



HAL
open science

Learning Visual Language Models for Video Understanding

Antoine Yang

► **To cite this version:**

Antoine Yang. Learning Visual Language Models for Video Understanding. Computer Vision and Pattern Recognition [cs.CV]. Ecole Normale Supérieure de Paris - ENS Paris, 2023. English. NNT : . tel-04307117v2

HAL Id: tel-04307117

<https://hal.science/tel-04307117v2>

Submitted on 28 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'École Normale Supérieure

**Learning visual language models
for video understanding**

Soutenu par

Antoine YANG

Le 23 Novembre 2023

École doctorale n°386

**Sciences Mathématiques
de Paris Centre**

Spécialité

Informatique

Composition du jury :

Matthieu CORD
Sorbonne Université

Président du jury

Dima DAMEN
University of Bristol

Rapporteur

Anna ROHRBACH
Technische Universität Darmstadt

Rapporteur

Josef SIVIC
Czech Technical University

Examineur

Ivan LAPTEV
Inria

Directeur de thèse

Cordelia SCHMID
Inria

Directrice de thèse

Résumé

Cette thèse est organisée en 8 chapitres, résumés ci-dessous.

Dans le chapitre 1, nous discutons de l'intérêt des modèles visuels de langage, des objectifs précis des modèles développés et des défis qu'ils doivent surmonter, comme résumé ci-dessous. Les humains sont des êtres multi-modaux, percevant le monde à travers leurs sens visuels complexes, tout en communiquant à travers le langage naturel. Cette intégration harmonieuse de la perception visuelle et du langage naturel forme la base de l'expérience humaine. Contrairement aux humains, les modèles d'apprentissage automatique sont généralement unimodaux, formés pour des tâches spécifiques en vision par ordinateur ou en traitement du langage naturel. Traditionnellement, ces modèles sont supervisés et dépendent d'annotations manuelles pour l'apprentissage. En revanche, les humains apprennent de manière auto-supervisée à partir de multiples modalités, ce qui inspire l'approche de cette thèse axée sur les modèles multi-modaux et l'apprentissage croisé de modalités. La thèse se concentre sur le développement de modèles multi-modaux, en particulier les modèles de langage visuel, capables de traiter conjointement des entrées visuelles et linguistiques et de générer des sorties linguistiques. Ces modèles pourraient combler le fossé entre les modèles d'apprentissage profond et les capacités humaines, en exploitant des données multi-modales à grande échelle via l'apprentissage croisé de modalités. Plus spécifiquement, la thèse explore le domaine de la compréhension vidéo, où les modèles visuels de langage peuvent extraire des informations sémantiques riches à partir de vidéos, y compris les métadonnées textuelles associées. Ces modèles offrent un potentiel considérable pour améliorer les interactions homme-machine, en comprenant le monde par la vision et en communiquant à travers le langage naturel. Les applications couvrent un large éventail de tâches vidéo, par exemple la recherche de vidéo à partir de requête en langage naturel, la génération de résumé textuel de vidéo, la génération de chapitres vidéo, les systèmes de recommandation vidéo, l'édition vidéo automatisée, la surveillance vidéo, la surveillance médicale, la réalité virtuelle, l'éducation, ou encore la modération de contenu.

L'objectif principal de cette thèse est de faire progresser le développement de modèles de langage visuel qui démontrent une compréhension approfondie du contenu vidéo en exploitant efficacement les indices visuels et linguistiques. Cela nécessite la conception d'architectures profondes et de techniques d'apprentissage appropriées, et leur application à des ensembles de données d'entraînement soigneusement sélectionnés. Pour évaluer l'efficacité des capacités détaillées de compréhension vidéo de nos modèles, nous nous concentrons sur plusieurs tâches importantes,

notamment la question-réponse vidéo, le repérage vidéo spatio-temporel et la description vidéo dense. La tâche de question-réponse vidéo est essentielle pour évaluer les capacités de compréhension vidéo détaillée des modèles de langage visuel. Elle consiste à générer une réponse en langage naturel à une question sur une vidéo, permettant d'évaluer la capacité des modèles à répondre de manière précise à des questions variées sur le contenu des vidéos. Nous explorons également la tâche de repérage vidéo spatio-temporel pour approfondir la compréhension fine de la vision et du langage. Cette tâche vise à localiser dans une vidéo non coupée un tube spatio-temporel correspondant à une requête linguistique, nécessitant une compréhension précise de l'espace et du temps. En outre, nous nous concentrons sur la tâche de description vidéo dense, qui implique de générer des descriptions en langage naturel localisées temporellement pour tous les événements se produisant dans une vidéo non coupée de plusieurs minutes. Cela nécessite une combinaison d'une compréhension précise des actions de bas niveau et d'une compréhension globale du récit de la vidéo. Enfin, nous abordons la tâche connexe de la génération de chapitres vidéo, qui consiste à segmenter temporellement la vidéo et à générer un chapitre en langage naturel pour chaque segment. Cette tâche a une application pratique en permettant aux utilisateurs de naviguer rapidement vers l'information qui les intéresse. En abordant ces tâches spécifiques, notre objectif est de faire progresser le développement de modèles de langage visuel capables d'une compréhension polyvalente du contenu vidéo, de traiter des tâches complexes de raisonnement et de générer des descriptions précises et contextuellement pertinentes.

Le développement de modèles de langage visuel capables de comprendre en détail le contenu des vidéos et de relever efficacement les défis présentés par les tâches précédemment exposées soulève plusieurs problématiques en termes de conception d'architecture neuronale et d'entraînement, comme décrit ci-dessous. Un premier défi consiste à connecter les modèles de vision et de langage. Les modalités visuelle et linguistique ont des formats différents : les images sont généralement représentées comme des signaux continus avec des pixels, tandis que le langage est symbolique et est représenté avec un vocabulaire fixe de jetons. De plus, la vision et le langage ont des structures et des sémantiques intrinsèquement différentes. Fusionner ces deux modalités dans un seul modèle nécessite la conception d'architectures appropriées et de techniques pour combler l'écart entre l'information visuelle et textuelle. Pour relever ce défi, nous développons des architectures de fusion médiane basées sur l'attention, permettant d'apprendre des représentations croisées détaillées en représentant à la fois les caractéristiques visuelles et le texte sous forme de jetons. Nous concevons également des techniques d'entraînement appropriées basées sur le gel des poids du modèle de langage pour préserver ses connaissances, ainsi que des approches de pré-entraînement multi-modal utilisant des jeux de données à grande échelle de vidéos Web. Un deuxième défi concerne la modélisation vidéo. Les vidéos contiennent beaucoup plus d'informations par rapport aux images, et une redondance significative d'informations entre les différentes images d'une même vidéo. Les tâches abordées dans cette thèse nécessitent de traiter des vidéos non coupées de plusieurs minutes, ce qui peut représenter des milliers d'images différentes. Nous devons donc concevoir des représentations vidéo efficaces qui capturent des informations sémantiques de haut niveau utiles sur le contenu de la vidéo. Nous visons également à développer des architectures qui peuvent représenter des vidéos non coupées et effectuer des

prédictions sur différentes périodes, afin de réaliser un repérage vidéo spatio-temporel et une description vidéo dense. Pour relever ces défis, nous proposons des architectures unifiées inspirées de travaux récents en vision par ordinateur, visant à être entraînaibles de bout en bout et à généraliser à diverses tâches vidéo. Un troisième défi concerne l'apprentissage à partir de vidéos Web. La collecte d'annotations pour les ensembles de données vidéo est coûteuse et chronophage. Pour surmonter cela, des travaux récents ont exploité des données vidéo disponibles sur le Web pour construire des ensembles de données vidéo-langage. Ces ensembles de données comprennent des vidéos narrées avec une transcription, ou des vidéos courtes accompagnées de descriptions. L'avantage de ces données est qu'elles peuvent être collectées automatiquement à grande échelle, capturant ainsi la diversité des vidéos du monde réel. Nous développons des techniques avancées pour tirer parti de ces vidéos Web dans l'entraînement de modèles de langage visuel pour des tâches complexes, telles que la réponse aux questions vidéo et la description vidéo dense. Ces techniques impliquent la génération de données avec des modèles de langage pré-entraînés, le gel des poids du modèle de langage, ou l'utilisation explicite des horodatages associés à la parole transcrite. Nous explorons également d'autres sources d'annotations à grande échelle sur le Web, telles que les chapitres annotés par les utilisateurs.

Dans le chapitre 2, nous présentons une revue de la littérature sur les travaux liés à cette thèse. Le chapitre est divisé en trois sections principales : (i) Modèles visuels, (ii) Modèles de langage et enfin (iii) Modèles de langage visuel.

Les chapitres 3 à 7 décrivent les 5 principales contributions de cette thèse. Dans les chapitres 3 et 4, nous présentons deux approches évolutives pour développer des modèles de réponses aux questions vidéo sans avoir recours à une annotation manuelle coûteuse. Dans le chapitre 5, nous expliquons TubeDETR, notre architecture basée sur un transformateur conçue pour la tâche de repérage vidéo spatio-temporel. Dans les chapitres 6 et 7, nous décrivons un nouveau modèle de langage visuel et un nouvel ensemble de données pour une description vidéo dense. Les détails sont donnés ensuite.

Dans le chapitre 3, nous proposons une approche pour générer automatiquement des données d'entraînement à grande échelle pour la question-réponse vidéo, évitant ainsi une annotation manuelle coûteuse. Nous utilisons la supervision croisée et appliquons des modèles de génération de questions basés uniquement sur le texte à la parole transcrite dans les vidéos narrées. À partir du jeu de données HowTo100M [Miech, 2019], nous générons le jeu de données HowToVQA69M avec 69 millions de triplets question-réponse-vidéo. Nous montrons qu'un transformateur de vidéo-question entraîné de manière contrastive avec un transformateur de réponse sur les ensembles de données générés est capable de répondre à des questions visuelles de manière zéro-shot (sans entraînement sur une seule image ou vidéo annotée manuellement) mieux que des références appropriées. De plus, notre méthode obtient des résultats compétitifs sur quatre jeux de données d'évaluation existants pour la question-réponse vidéo. Nous étendons également notre approche à des paires vidéo-description du Web pour générer le WebVidVQA3M avec 3 millions de triplets question-réponse vidéo à partir du jeu de données WebVid2M [Bain, 2021]. Pour une évaluation détaillée, nous introduisons également iVQA, un nouveau jeu de données de question-réponse

vidéo avec des biais linguistiques réduits et des annotations manuelles redondantes de haute qualité.

Dans le chapitre 4, nous proposons FrozenBiLM, une approche pour la question-réponse vidéo zéro-shot qui exploite directement des modèles de langage bidirectionnels gelés (BiLM) sans procédure de génération de données. En particulier, (i) nous combinons les entrées visuelles avec le BiLM gelé en utilisant des modules d’entraînement légers, (ii) nous entraînons de tels modules à l’aide de données multi-modales extraites du Web, et enfin (iii) nous effectuons l’inférence de question-réponse vidéo zéro-shot grâce à la modélisation de langage masqué, où le texte masqué est la réponse à une question donnée. Notre approche surpasse les méthodes autoregressives antérieures [Tsimpoukelli, 2021] tout en étant moins coûteuse, et améliore largement l’état de l’art antérieur en question-réponse vidéo zéro-shot sur huit ensembles de données variés.

Dans le chapitre 5, nous abordons le problème de la localisation d’un tube spatio-temporel dans une vidéo correspondant à une requête textuelle donnée. Nous proposons TubeDETR, une architecture basée sur un transformateur qui peut être entraîné de bout en bout pour le repérage vidéo spatio-temporel. Notre modèle comprend notamment : (i) un encodeur vidéo et texte efficace qui modélise les interactions multi-modales spatiales sur des images échantillonnées de manière clairsemée et (ii) un décodeur espace-temps qui effectue conjointement la localisation spatio-temporelle. Nous démontrons l’avantage de nos composants proposés grâce à une étude expérimentale approfondie. Avec un pré-entraînement image-texte [Kamath, 2021], TubeDETR améliore l’état de l’art antérieur sur les jeux de données d’évaluation exigeants VidSTG [Zhang, 2020d] et HC-STVG [Tang, 2021].

Dans le chapitre 6, nous abordons le problème de la génération de descriptions temporellement localisées pour tous les événements dans une vidéo non coupée. Nous proposons Vid2Seq, un modèle de langage visuel qui peut décrire de manière dense une vidéo en générant une seule séquence de jetons. L’architecture Vid2Seq augmente un modèle de langage avec des jetons de temps spéciaux, lui permettant de prédire de manière fluide les limites des événements et les descriptions textuelles dans la même séquence de sortie. Un tel modèle unifié nécessite des données d’entraînement à grande échelle, qui ne sont pas disponibles dans les ensembles de données annotés actuels. Nous montrons qu’il est possible de tirer parti de vidéos narrées non annotées pour la description dense de vidéos, en reformulant l’horodatage des phrases de la parole transcrite comme des limites de pseudo-événements et en utilisant les phrases de la parole transcrite comme des descriptions de pseudo-événements. Le modèle Vid2Seq résultant pré-entraîné sur l’ensemble de données YT-Temporal-1B [Zellers, 2022] améliore l’état de l’art antérieur sur une variété de jeux de données d’évaluation de description vidéo dense, tels que YouCook2 [Zhou, 2018a], ViTT [Huang, 2020b] et ActivityNet Captions [Krishna, 2017]. Vid2Seq généralise également bien aux tâches de description de paragraphes vidéo et de description de clips vidéo, ainsi qu’aux paramètres d’entraînement avec peu de données.

Dans le chapitre 7, nous proposons VidChapters-7M, un ensemble de données à grande échelle de vidéos annotées par des utilisateurs. VidChapters-7M est créé automatiquement à partir de vidéos en ligne de manière automatique en extrayant des chapitres annotés par les utilisateurs,

sans annotation manuelle supplémentaire. Nous introduisons les trois tâches suivantes basées sur ces données. Premièrement, la tâche de génération de chapitres vidéo consiste à segmenter temporellement la vidéo et à générer un titre de chapitre pour chaque segment. Pour disséquer davantage le problème, nous définissons également deux variantes de cette tâche: la génération de chapitres vidéo sachant les limites temporelles, qui nécessite de générer un titre de chapitre à partir d'un segment vidéo annoté, et le repérage de chapitres vidéo, qui nécessite de localiser temporellement un chapitre donné son titre annoté. Nous évaluons à la fois des approches simples et des modèles multi-modaux de l'état de l'art, y compris Vid2Seq, sur ces trois tâches. Nous montrons également que le pré-entraînement de Vid2Seq sur VidChapters-7M se transfère bien aux tâches de description vidéo dense, tant dans les paramètres de zéro-shot que de finetuning, améliorant largement l'état de l'art sur les jeux de données d'évaluation YouCook2 et ViTT.

Enfin, dans le chapitre 8, nous fournissons un résumé des contributions et discutons des pistes de recherche futures, telles que le développement de modèles capables de dialoguer précisément sur des entités spécifiques dans la vidéo, les modèles vidéos unifiés, le traitement de vidéos longues, l'annotation de jeux de données vidéos assistée par des modèles profonds, et les modèles de génération multi-modaux.

Abstract

As humans, we communicate with natural language and perceive the world through vision. Therefore, the goal of this thesis is to build and train machine learning models that combine the power of natural language processing with visual understanding, enabling a comprehensive and detailed comprehension of the content within videos. In particular, we develop visual language models capable of (i) answering natural language questions about videos (ii) localizing natural language queries spatially and temporally in untrimmed videos, and (iii) generating temporally-grounded natural language descriptions of all events in untrimmed videos.

First, we propose two scalable approaches to develop video question answering models without the need for costly manual annotation. This is unlike most current video question answering systems which are trained on large manually annotated datasets. We automatically generate video question answering data from narrated videos using text-only question-generation models. We then show that a multi-modal transformer trained contrastively on the generated data can answer visual questions in a zero-shot manner. In order to bypass the data generation procedure, we present an alternative approach, dubbed FrozenBiLM, that directly leverages bidirectional masked language models. This is done by adding light modules to incorporate vision into the language model, and training these modules on Web-scraped video-caption pairs while keeping the language model weights frozen to preserve its textual knowledge.

Second, we develop TubeDETR, a transformer model that can spatially and temporally localize a natural language query in an untrimmed video. Unlike prior spatio-temporal grounding approaches, TubeDETR can be effectively trained end-to-end on untrimmed videos, as it includes an efficient video and text encoder that models spatial multi-modal interactions over sparsely sampled frames and a space-time decoder that jointly performs spatio-temporal localization.

Third, we present a new model and a new dataset for multi-event understanding in untrimmed videos. We introduce the Vid2Seq model which generates dense natural language descriptions and corresponding temporal boundaries for all events in an untrimmed video by predicting a single sequence of tokens. Special time tokens interleave the text sentences to temporally ground them in the video. Moreover, Vid2Seq can be effectively pretrained on narrated videos at scale using transcribed speech as pseudo-supervision. Finally, we introduce VidChapters-7M, a large-scale dataset of user-chaptered videos. Based on this dataset, we evaluate state-of-the-art models on three tasks including video chapter generation. We also show that video chapter generation models transfer well to dense video captioning in both zero-shot and finetuning settings.

Acknowledgements

First, I warmly thank Antoine Miech, Josef Sivic, Ivan Laptev and Cordelia Schmid for advising me in the last three years. Your weekly feedback shaped the researcher I am today.

Special thanks to Dima Damen, Anna Rohrbach and Matthieu Cord for taking part in my defense, and to Rachel Bawden and Umut Simsekli for their valuable feedback as members of my doctoral committee.

I am grateful to Google for generously funding my research, to Inria Paris for providing such a pleasant research environment, and to IDRIS for maintaining the great Jean-Zay supercomputer.

To Mathieu, Schéhérazade, Julien, Marion, and Donia, I appreciated your assistance with administrative matters. Jean-Marc and Pierre-Guillaume, thank you for enabling the release of entertaining demos at Inria Paris.

To Lucas, Gabriel, and Zeeshan, it has been a privilege to co-advise you, and I am excited to continue our collaborations.

I am grateful to all members of the WILLOW and SIERRA teams for providing an exceptional lab environment. Special mention to Yana, Thomas E. and Pierre-Louis for being awesome mentors from my start. To Théophile, Ziad, Gaspard, and Bertille, it has been a pleasure sharing offices with you. Kudos to Yann, Ulysse and Francis for the bike rides, and to Charlotte, Guillaume, Fabian and Thomas C. for the runs. Thanks Elliot C.S., Adrien, Wilson, Elliot V., Céline, Oumayma, Bruno, Ricardo, Etienne, Minttu, Pauline, Tomás, Andrea, Kateryna, Viviane, Eugène, and everyone else with whom I shared nice moments.

Shoutouts to Lili M., Ahmet, Anurag, Mathilde, Paul H. S., Manjin, Lili G., Arsha, Xingyi, Chen, Xuehan and Jordi, for making my internship at Google such a delightful experience.

Lastly, my deepest gratitude goes to my loved ones for their unwavering support.

Contents

1	Introduction	1
1.1	Motivation	3
1.2	Goal	4
1.3	Challenges	6
1.4	Contributions	8
1.4.1	Publications	8
1.4.2	Software and dataset contributions	9
1.5	Outline	10
2	Related Work	12
2.1	Language models	12
2.1.1	Word embeddings	12
2.1.2	Recurrent neural networks	13
2.1.3	Transformers	14
2.2	Visual models	16
2.2.1	Image models	16
2.2.2	Video models	18
2.3	Visual language models	20
2.3.1	Image-language models	20
2.3.2	Video-language models	24
3	Learning to Answer Visual Questions from Web Videos	28
3.1	Introduction	29
3.2	Related Work	30
3.3	Large-scale generation of VideoQA data	33
3.3.1	Generating video-question-answer triplets	33
3.3.2	HowToVQA69M: a large-scale VideoQA dataset	34
3.4	VideoQA model and training procedure	37
3.4.1	VideoQA model	37
3.4.2	Training procedure	39
3.5	iVQA: a new VideoQA evaluation dataset	40
3.6	Experiments	42
3.6.1	Evaluation Protocol	43
3.6.2	Zero-shot VideoQA	44
3.6.3	VideoQA feature probe evaluation	46
3.6.4	Benefits of HowToVQA69M pretraining	46
3.6.5	Analysis of rare answers and question types	48
3.6.6	Comparison of VideoQA generation methods and VideoQA training datasets	50

3.6.7	Generalization to other video-text datasets	51
3.6.8	Importance of the visual modality in iVQA	52
3.6.9	Ablation studies	53
3.7	Conclusion	53
4	Zero-Shot Video Question Answering via Frozen Bidirectional Language Models	55
4.1	Introduction	56
4.2	Related Work	57
4.3	Method	58
4.3.1	Architecture	59
4.3.2	Cross-modal training	61
4.3.3	Adapting to downstream tasks	61
4.4	Experiments	63
4.4.1	Experimental setup	63
4.4.2	Ablation studies	65
4.4.3	Comparison with frozen autoregressive models	69
4.4.4	Comparison to the state of the art for zero-shot VideoQA	72
4.4.5	Freezing the BiLM is also beneficial in supervised settings	73
4.4.6	Qualitative analysis of the <i>frozen</i> self-attention patterns in FrozenBiLM	76
4.5	Conclusion	76
5	TubeDETR: Spatio-Temporal Video Grounding with Transformers	77
5.1	Introduction	78
5.2	Related Work	79
5.3	Method	80
5.3.1	Overview	80
5.3.2	Video-Text Encoder	81
5.3.3	Space-Time Decoder	83
5.3.4	Training loss	84
5.3.5	Weight initialization	85
5.4	Experiments	85
5.4.1	Experimental setup	85
5.4.2	Ablation studies	87
5.4.3	Comparison to the state of the art	90
5.4.4	Qualitative examples	90
5.4.5	Visualization of space, time and language attention patterns in the decoder	91
5.5	Conclusion	95
6	Vid2Seq: Large-Scale Pretraining of a Visual Language Model for Dense Video Captioning	96
6.1	Introduction	97
6.2	Related Work	98
6.3	Method	100
6.3.1	Model	100
6.3.1.1	Sequence construction.	101
6.3.1.2	Architecture.	102
6.3.2	Training	103
6.3.2.1	Pretraining on untrimmed narrated videos	103
6.3.2.2	Downstream task adaptation	104

6.4	Experiments	105
6.4.1	Experimental setup	105
6.4.2	Ablation studies	107
6.4.3	Comparison to the state of the art	111
6.4.4	Few-shot dense video captioning	112
6.4.5	Qualitative examples	113
6.5	Conclusion	115
7	VidChapters-7M: Video Chapters at Scale	117
7.1	Introduction	118
7.2	Related Work	119
7.3	VidChapters-7M: a large-scale dataset of user-chaptered videos	121
7.3.1	Data collection	121
7.3.2	Data processing	122
7.3.3	Data analysis	122
7.4	Experiments	127
7.4.1	Video chapter generation	128
7.4.2	Video chapter generation given ground-truth boundaries	131
7.4.3	Video chapter grounding	133
7.4.4	Transfer learning on dense video captioning	135
7.5	Conclusion	138
8	Conclusions	139
8.1	Contributions	139
8.2	Future work	140
8.2.1	Localized video dialog	140
8.2.2	Unified video model	141
8.2.3	Processing long videos	142
8.2.4	Model-assisted annotation of video datasets	142
8.2.5	Multi-modal generation	143
8.2.6	Ethical considerations	143
	Bibliography	145

Chapter 1

Introduction

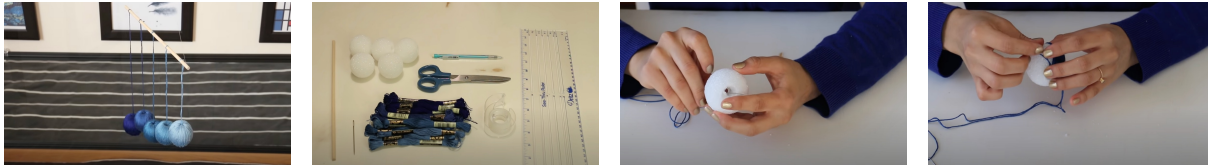
Humans perceive the world through their complex visual senses, encompassing the discernment of shapes, colors, depth, and motion, intricately weaving a vibrant tapestry of their surroundings. Simultaneously, humans communicate and convey their thoughts, emotions, and ideas through the versatile medium of natural language, employing a vast range of intricate linguistic structures, grammar, and vocabulary. This seamless integration of visual perception and natural language forms the foundation of human experience, enabling comprehension, interpretation, and effective communication.

However, machine learning models are traditionally trained to solve tasks within specific domains such as computer vision and natural language processing. State-of-the-art models in these fields typically apply modern deep learning techniques [LeCun, 2015] using large unimodal datasets like ImageNet [Russakovsky, 2015] in computer vision and BooksCorpus [Zhu, 2015] in natural language processing. These techniques learn the parameters of the model by backpropagating the error measured by an appropriate loss function with respect to the data.

Moreover, the traditional approach to develop effective deep learning models relies heavily on the laborious process of manually curating annotations linked to the raw data. This involves tasks such as categorizing images, as exemplified by the ImageNet dataset [Russakovsky, 2015], or writing question-answer pairs for textual corpora, as demonstrated by the SQuAD dataset [Rajpurkar, 2016]. These annotations are then integrated into the model's loss function, enhancing its ability to learn and generalize from the data.

In contrast, humans inherently learn visual and world knowledge from multiple modalities without need for an explicit teacher [Barlow, 1989]. In developmental psychology, this phenomenon is evidenced by the concept of re-entry [Edelman, 1987], where one modality triggers the memory of another.

Therefore, while building on the deep learning approaches mentioned above, this thesis aims at building and training multi-modal models that can use multiple modalities (for instance, vision and language). Such models could potentially bridge the gap between deep learning models and humans, by learning from Web-scale multi-modal data via cross-modal learning [Radford, 2021], where one modality serves as a supervision for the other.



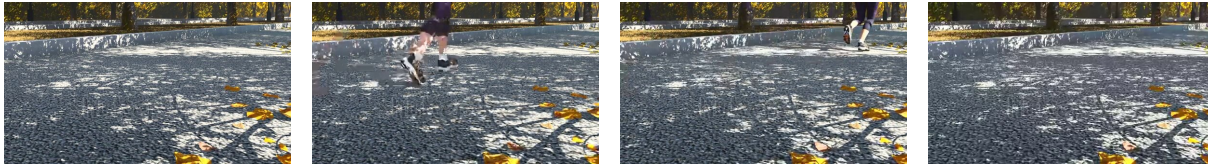
ASR: Hi friends! Today I am going to show you how to make the famous Montessori Gobbi Mobile.

ASR: You're also going to need a pencil, scissors, ruler, and tape.

ASR: Ok so we're going to take a needle and pull it through the hole in the center.

ASR: Now hold it in place with your finger.

(a) Example of a narrated video from the HowTo100M dataset [Miech, 2019]. For readability, we only show a few video frames with their associated segments of speech transcripts (ASR).



Web-sourced text description: Runners feet in a sneakers close up. realistic three dimensional animation.



Web-sourced text description: Snorkelers swimming in a calm blue sea with 3 windmills in the background.

(b) Examples of video-caption pairs from the WebVid10M dataset [Bain, 2021].

Figure 1.1: **Examples of web videos with their associated textual metadata.** In this thesis, we train visual language models for video understanding using such readily-available sources of data which can be automatically collected at scale (see Chapters 3, 4, 6 and 7).

In particular, in this thesis, we focus on developing visual language models that can jointly process visual and language inputs, and generate language output. Such models have the potential to enhance human-computer interactions by perceiving the world through vision and communicating through natural language. They are also general as various video tasks and applications can be formulated with visual and language inputs and language output.

Specifically, we develop visual language models in the context of video understanding. As illustrated in Figure 1.1, videos on the web are often associated with various sources of textual metadata, such as titles, descriptions, tags, chapters (contiguous, non-overlapping segments associated with a short text description and completely partitioning a video), texts displayed during the video or transcribed speech spoken during the video [Miech, 2019]. Therefore visual language models offer the potential to unlock rich semantics and contextual information from video content. Indeed, such models can not only understand the video content but also the various sources of metadata that are associated with the video. In the next section, we describe the wide range of applications that could be enabled by the development of successful visual language models for video understanding.

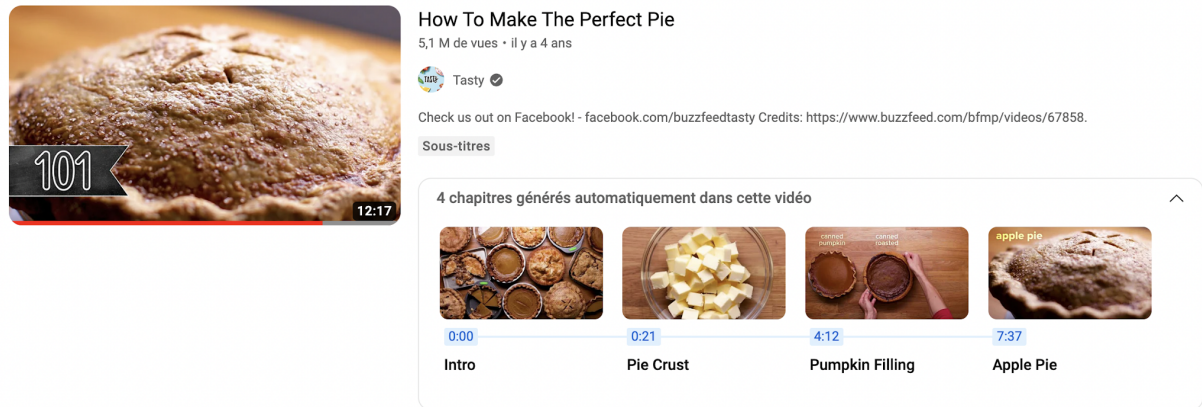


Figure 1.2: **Automatic video chapter generation** is an example of application of video understanding systems that enables users to quickly navigate to areas of interest and easily replay different parts of a video. We study this task in detail in Chapter 7.

1.1 Motivation

In recent times, digital video content has witnessed a remarkable upsurge, revolutionizing the way we consume and share information. Online platforms such as YouTube, Facebook, Instagram, and TikTok have become hubs for hosting vast amounts of video data, catering to billions of users worldwide. This explosion in video content has created a pressing need for efficient methods to analyze, interpret, and understand these videos at scale. Therefore video understanding systems have a wide range of applications across various domains and industries.

One crucial application of automatic video understanding systems is video search and retrieval. With the exponential growth of video data, the efficient localization of specific video clips most relevant to a user has become increasingly important to minimize the time and effort expended on manually navigating through vast video libraries.

Video summarization is another important application of automatic video understanding systems. Generating concise and informative video summaries in the form of textual or visual highlights that capture the essence of the video content can provide a quick overview of the video and facilitate content browsing, enabling users to determine whether a video is worth their time or to review important information without watching the entire video.

A related application is video chapter generation (see Figure 1.2), which consists in temporally segmenting the video into chapters and generating a chapter title for each segment. This application enables users to quickly navigate to areas of interest and easily replay different parts of a video.

Video recommendation systems can also benefit from automatic video understanding to deliver personalized and engaging content to users that best align with their interests. This improves user satisfaction, discovery of new content, and engagement with video platforms.

Moreover, automatic video understanding also has profound implications for video editing and production workflows. Video understanding systems can assist in automating various aspects of video editing, such as identifying potential flaws or errors in videos, suggesting im-

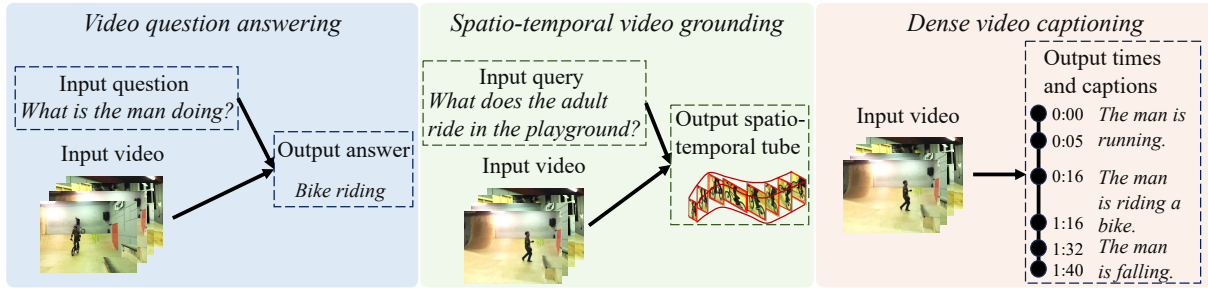


Figure 1.3: **Illustration of three representative tasks we tackle in this thesis:** video question answering (left, see Chapters 3 and 4), spatio-temporal video grounding (middle, see Chapter 5), dense video captioning (right, see Chapters 6 and 7).

provements, or automatically generating captions or subtitles for accessibility purposes. These systems can streamline the video production process, making it more efficient and cost-effective.

In addition, in video surveillance, video understanding systems can automatically analyze and interpret video streams, detecting and recognizing objects, activities, and anomalies in real time, thus enhancing public safety and security. They could also be used to assist in healthcare monitoring by analyzing video data to monitor patient activities, detect falls, assess rehabilitation progress, or track vital signs. Hence these systems have applications in elderly care, hospital settings, and remote patient monitoring, providing valuable insights for healthcare professionals and improving patient outcomes.

Furthermore, in domains such as virtual reality and education, video understanding systems are instrumental in unlocking new possibilities. In virtual reality applications, video understanding systems enable immersive experiences by analyzing and interpreting the user’s surroundings, enabling interaction and engagement within the virtual environment. In education, video understanding systems can facilitate multimedia learning experiences by automatically extracting relevant content, generating quizzes, or providing interactive annotations.

Finally, video understanding systems play a crucial role in content moderation on online platforms. They can automatically detect and flag inappropriate or harmful content, including explicit or violent scenes, hate speech, or copyright infringement. Therefore these systems help ensure user safety, maintain platform guidelines, and create a healthier online environment.

Note that these are a few examples of the diverse applications of video understanding systems. As technology advances and research progresses, the potential for video understanding to impact various fields and industries continues to expand, offering new opportunities for innovation and problem-solving.

1.2 Goal

Our primary objective is to advance the development of visual language models that exhibit a comprehensive understanding of video content by effectively leveraging both visual and language cues. This notably requires designing appropriate deep architectures and learning techniques, and applying them to well curated training datasets. To evaluate the efficacy of our models’

detailed video understanding capabilities, we focus on several important downstream tasks, including video question answering, spatio-temporal video grounding and dense video description, as described next in detail and illustrated in Figure 1.3. In particular, we strive to design models which could be used for various applications beyond the studied tasks. Therefore by addressing these specific tasks, our goal is to make advances in the development of visual language models that achieve a versatile understanding of video content, handle complex reasoning tasks, and generate accurate and contextually relevant descriptions.

An essential benchmark for evaluating the detailed video understanding capabilities of visual language models is the video question answering task [Jang, 2017; Lei, 2018a; Xu, 2017]. Given a natural language question about a video, this task requires to generate an appropriate natural language answer. Hence this task allows us to gauge their proficiency in accurately answering questions related to various aspects of the content of videos, going beyond video classification [Carreira, 2017]. In particular, questions involve recognizing objects, places, actions, colors, counting, understanding spatial relations, temporal dynamics and joint multi-modal reasoning over transcribed speech, visual and question inputs. By virtue of its open-ended nature, the video question answering task can be thought as a simple form of visual dialog [Das, 2017] about videos, hence can serve as an excellent testbed for visual language models. In comparison with more complex text generation tasks, this task also presents the advantage that it can be evaluated reliably with an interpretable accuracy metric, which evaluates the predicted answer and the ground-truth answer by string matching.

In addition, we explore the spatio-temporal video grounding task [Zhang, 2020d] to delve into fine-grained vision and language understanding. Given a natural language query and an untrimmed video that can span several minutes, this task aims at localizing in the video a spatio-temporal tube that refers to the language query. While video question answering predominantly involves learning representations at the video level, spatio-temporal video grounding requires precise spatial and temporal understanding. This task not only enables us to evaluate the fine-grained vision and language understanding but also may help develop visual language models that can better determine the relevant parts of a video that address the queried information in video question answering and other video-language tasks.

Furthermore, our aim is to develop visual language models capable of reasoning over multiple events in long videos. Such ability goes beyond image-level understanding which has been shown successful in many other video tasks [Buch, 2022]. To evaluate this ability, we focus on the dense video captioning task [Krishna, 2017], illustrated in Figure 1.4. This task involves generating temporally localized natural language descriptions for all events occurring within untrimmed videos lasting several minutes. Events have various temporal granularity, covering human activities or steps in instructional videos [Zhou, 2018a]. Hence dense video captioning requires a combination of accurate low-level action understanding and global story understanding. Finally, we consider the related task of video chapter generation, which requires temporally segmenting the video and generating a natural language chapter for each segment (see Figure 1.2). Therefore this task has a concrete practical application, as it can enable users to quickly navigate to the information of their interest. Compared with dense video captioning, the start of the chapter

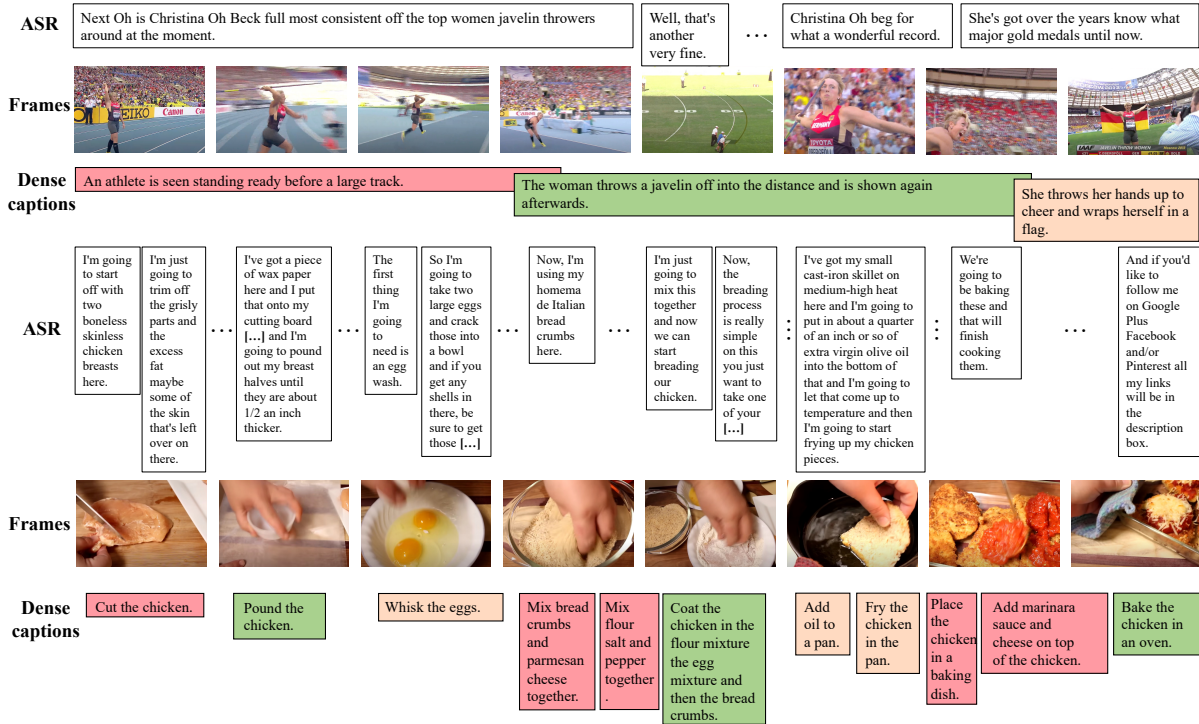


Figure 1.4: **Examples of untrimmed video with annotated dense captions from ActivityNet Captions [Krishna, 2017] (top) and YouCook2 [Zhou, 2018a] (bottom).** The horizontal span represents the temporal span of the events in the video. In this thesis, we build visual language models that have a detailed understanding of the different events happening at different times in the video (see Chapters 6 and 7).

of a given chapter is the end of previous one, the chapters cover the full video, and often have titles that are concise and substantially shorter than dense video captions.

1.3 Challenges

Developing visual language models that have a detailed understanding of the content of videos and that can effectively tackle the previously presented tasks presents several challenges in terms of neural architecture design and training, as described next.

Connecting vision models and language models. The visual and language modalities have different formats: images are typically represented as continuous signals with pixels, while language is symbolic and is represented with a fixed vocabulary of tokens. Moreover, vision and language have inherently different structures and semantics. While images are rich in spatial information, languages are structured with grammar and syntax. These differences in terms of data representations and semantics have historically resulted in different models being developed in computer vision and natural language processing, for instance, convolutional neural networks [Krizhevsky, 2012] for the former and recurrent neural networks [Sutskever, 2011] for the latter. Therefore fusing both modalities into a single model requires designing appropriate

architectures and techniques for bridging the gap between visual and textual information. In particular, the late fusion of visual and text features commonly used in joint embedding models [Radford, 2021] might not be suited to develop visual language models that have a detailed cross-modal understanding.

To address this challenge, we develop mid-fusion attention-based [Vaswani, 2017] architectures that enable learning detailed cross-modal representations by representing both visual features and text as tokens. To further bridge the modality gap, we design appropriate training techniques based on freezing the language model weights to preserve its knowledge (see Chapter 4) or multi-modal pretraining with contrastive [Radford, 2021] (see Chapter 3) and generative [Raffel, 2020] (see Chapter 6) approaches using large-scale datasets of web videos.

Video modeling. Compared to images, videos contain a lot more information and a significant redundancy of information between the different frames. For instance, in the last example of Figure 1.1, only a few pixels corresponding to the snorkelers swimming and the windmills movement change between the different frames. In particular, in this thesis, we consider multiple tasks that require processing untrimmed videos that can last over several minutes, potentially catering to thousands of video frames including various entities, scenes and actions, as illustrated in Figure 1.4. Hence we need to design efficient video representations that can capture useful high-level semantic information about the video content. Moreover, we wish to design architectures that can represent untrimmed videos and make predictions across different times, to perform spatio-temporal video grounding and dense video description. In contrast with prior video models that resort to two-stage approaches [Zhang, 2020d] or contain task-specific components [Wang, 2021d], we aim at developing end-to-end trainable unified models that can generalize to various video tasks.

To tackle these challenges, we propose unified architectures inspired by recent work in computer vision such as DETR [Carion, 2020] and Pix2Seq [Chen, 2022a]. The considered architectures represent videos with a few latent vectors (see Chapter 5), or tokenize time jointly with text (see Chapter 6). For compute efficiency, we either freeze the spatial branch of the visual backbone, or backpropagate gradients to this branch only on a few sampled frames (see Chapter 5). We also develop scalable techniques to train these architectures, leveraging image-text data (see Chapter 5) or web videos (see Chapters 3, 4, 6 and 7).

Learning from web videos. Deep learning models are typically trained on manually annotated datasets. Collecting natural language annotations for video datasets, however, is cumbersome, time consuming, expensive and therefore not scalable. This issue is compounded when collecting annotations for untrimmed videos that can last over several minutes. To address this issue, recent works have collected video-language datasets built on video data readily-available from the Web, as illustrated in Figure 1.1. Such data include narrated videos that contain transcribed speech [Miech, 2019], or short videos accompanied with their alt-text description [Bain, 2021]. The advantage of such data is that it can be collected automatically at scale, enabling to build video-text datasets that capture the diversity of real-world videos. These datasets enabled

to train joint video-text embedding spaces for text-video retrieval that achieve state-of-the-art results in zero-shot mode, where the pretrained model is directly applied to retrieval tasks, and fully-supervised settings, where the pretrained model is finetuned on manually annotated datasets.

In this thesis, we develop advanced techniques to leverage such web videos to train visual language models for complex tasks such as video question answering (see Chapters 3 and 4) or dense video description (see Chapters 6 and 7). These techniques involve generating data with pretrained language models (see Chapter 3), freezing the language model weights (see Chapter 4), or explicitly using the timestamps associated with the transcribed speech (see Chapter 6). We also explore other scalable sources of supervision from the Web, such as user-annotated chapters (see Chapter 7).

1.4 Contributions

In the following, we list the publications contributions, as well as the software and dataset releases that were performed during the course of this thesis. We will detail the contributions within five of the publications in Chapters 3 to 7.

1.4.1 Publications

The work done during this PhD led to the following publications:

- Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, Cordelia Schmid. Just Ask: Learning to Answer Questions from Millions of Narrated Videos. In ICCV 2021 (Oral). [Yang, 2021b] (Chapter 3).
- Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, Cordelia Schmid. Learning to Answer Visual Questions from Web Videos. In TPAMI Special Issue on the Best Papers of ICCV 2021 (journal extension of the ICCV 2021 paper). [Yang, 2022c] (Chapter 3).
- Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, Cordelia Schmid. Zero-Shot Video Question Answering via Frozen Bidirectional Language Models. In NeurIPS 2022. [Yang, 2022e] (Chapter 4).
- Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, Cordelia Schmid. TubeDETR: Spatio-Temporal Video Grounding with Transformers. In CVPR 2022 (Oral). [Yang, 2022d] (Chapter 5).
- Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, Cordelia Schmid. Vid2Seq: Large-Scale Pretraining of a Visual Language Model for Dense Video Captioning. In CVPR 2023. [Yang, 2023d] (Chapter 6).
- Antoine Yang, Arsha Nagrani, Ivan Laptev, Josef Sivic, Cordelia Schmid. VidChapters-7M: Video Chapters at Scale. In NeurIPS 2023 Track on Datasets and Benchmarks. [Yang, 2023c] (Chapter 7).

- Lucas Ventura, Antoine Yang, Cordelia Schmid, Gül Varol. CoVR: Learning Composed Video Retrieval from Web Video Captions. Work in progress. [Ventura, 2023] (<https://arxiv.org/abs/2308.14746>).

1.4.2 Software and dataset contributions

Software. The code for the five contribution chapters of this thesis is publicly released:

- Just Ask: The code for generating video question answering data, training and evaluating video question answering models, and pretrained models, are released as part of the project presented in [Yang, 2021b; Yang, 2022c] (Chapter 3). <https://github.com/antoyang/just-ask>. In addition, an online demo of zero-shot video question answering with the developed model is hosted at <http://videoqa.paris.inria.fr/>.
- FrozenBiLM: The code for training and evaluating video question answering models, and pretrained models, are released as part of the project presented in [Yang, 2022e] (Chapter 4). <https://github.com/antoyang/FrozenBiLM>.
- TubeDETR: The code for training and evaluating spatio-temporal video grounding models, and pretrained models, are released as part of the project presented in [Yang, 2022d] (Chapter 5). <https://github.com/antoyang/TubeDETR>. In addition, an online demo of spatio-temporal video grounding with the developed model is hosted at <http://stvg.paris.inria.fr/>.
- Vid2Seq: The code for training and evaluating dense video captioning models, and pretrained models, are released as part of the project presented in [Yang, 2023d] (Chapter 6), in collaboration with Google. <https://github.com/google-research/scenic/tree/main/scenic/projects/vid2seq>.
- VidChapters-7M: The code for training and evaluating video chapter generation models, and pretrained models, are released as part of the project presented in [Yang, 2023c] (Chapter 7). <https://github.com/antoyang/VidChapters>.

iVQA dataset. We have publicly released the iVQA dataset (<https://antoyang.github.io/just-ask.html#ivqa>) with the publication of [Yang, 2021b] (Chapter 3). The name stands for **I**nstructional **V**ideo **Q**uestion **A**nswering. The dataset contains 10K instructional videos and is designed for training and evaluating video question answering models. Each video is manually annotated with a question and *five* corresponding ground-truth answers to provide a well-defined evaluation. We also made efforts to avoid questions which can be answered without watching the video.

HowToVQA69M dataset. We have publicly released the HowToVQA69M dataset (<https://antoyang.github.io/just-ask.html#howtovqa>) with the publication of [Yang, 2021b] (Chapter 3). The dataset is automatically generated from the HowTo100M dataset [Miech, 2019] using question generation language models. HowToVQA69M contains 1M videos, 69M question-answers, and is designed to train video question answering models.

WebVidVQA3M dataset. We have publicly released the WebVidVQA3M dataset (<https://antoyang.github.io/just-ask.html#webvidvqa>) with the publication of [Yang, 2022c] (Chapter 3). The dataset is automatically generated from the WebVid2M dataset [Bain, 2021] using question generation language models. WebVidVQA3M contains 2M videos, 3M question-answers, and is designed to train video question answering models.

VidChapters-7M dataset. We have publicly released the VidChapters-7M dataset (<https://antoyang.github.io/vidchapters.html#data>) with the publication of [Yang, 2023c] (Chapter 7). VidChapters-7M includes 817K user-chaptered videos including 7M chapters in total. The dataset is automatically created from videos online in a scalable manner by scraping user-annotated chapters and hence without any additional manual annotation. It is designed to train and evaluate video chapter generation models, and to pretrain video-language models.

1.5 Outline

This thesis is organized into 8 chapters, including this introduction (Chapter 1).

Chapter 2 is a literature review of work related to this thesis. The chapter is divided into three main sections: (i) Visual models, (ii) Language models and finally (iii) Visual language models.

Chapters 3 to 7 describe the 5 main contributions of this thesis. In Chapters 3 and 4, we present two scalable approaches to develop video question answering models without the need for costly manual annotation. In Chapter 5, we explain TubeDETR, our transformer-based architecture designed for spatio-temporal video grounding. In Chapters 6 and 7, we describe a new visual language model and a new dataset for dense video description. Details are given next.

In Chapter 3, we propose an approach to automatically generate large-scale video question answering training data, avoiding expensive manual annotation. For this, we make use of cross-modal supervision and apply text-only question generation models to the transcribed speech in narrated videos. Starting from HowTo100M dataset [Miech, 2019], we generate the HowToVQA69M with 69M video question answering triplets. We show that a video-question transformer trained contrastively with an answer transformer on the generated datasets is capable of answering visual questions in a zero-shot manner (without training on a single manually annotated image or video) better than appropriate baselines. Furthermore, our method achieves competitive results on four existing video question answering benchmarks. Moreover, we extend our approach to web video-caption pairs and generate the WebVidVQA3M with 3M video question answering triplets starting from the WebVid2M dataset [Bain, 2021]. For a detailed evaluation, we also introduce iVQA, a new video question answering dataset with reduced language biases and high-quality redundant manual annotations.

In Chapter 4, we propose FrozenBiLM, an approach to zero-shot video question answering that directly leverages frozen bidirectional language models (BiLM) without data generation procedure. In particular, (i) we combine visual inputs with the frozen BiLM using light trainable modules, (ii) we train such modules using Web-scraped multi-modal data, and finally (iii) we

perform zero-shot video question answering inference through masked language modeling, where the masked text is the answer to a given question. Our approach outperforms prior autoregressive methods [Tsimpoukelli, 2021] while being lighter, and largely improves over the prior state of the art in zero-shot video question answering on a variety of eight video question answering datasets. It also demonstrates competitive performance in the few-shot and fully-supervised setting.

In Chapter 5, we consider the problem of localizing a spatio-temporal tube in a video corresponding to a given text query. We propose TubeDETR, an end-to-end transformer-based architecture for spatio-temporal video grounding. Our model notably includes: (i) an efficient video and text encoder that models spatial multi-modal interactions over sparsely sampled frames and (ii) a space-time decoder that jointly performs spatio-temporal localization. We demonstrate the advantage of our proposed components through an extensive ablation study. With image-text pretraining [Kamath, 2021], TubeDETR improves over the prior state of the art on the challenging VidSTG [Zhang, 2020d] and HC-STVG [Tang, 2021] benchmarks.

In Chapter 6, we consider the problem of generating temporally localized descriptions for all events in an untrimmed video. We propose Vid2Seq, a visual language model that can densely caption a video by generating a single sequence of tokens. The Vid2Seq architecture augments a language model with special time tokens, allowing it to seamlessly predict event boundaries and textual descriptions in the same output sequence. Such a unified model requires large-scale training data, which is not available in current annotated datasets. We show that it is possible to leverage unlabeled narrated videos for dense video captioning, by reformulating sentence boundaries of transcribed speech as pseudo event boundaries, and using the transcribed speech sentences as pseudo event captions. The resulting Vid2Seq model pretrained on the YT-Temporal-1B dataset [Zellers, 2022] improves over prior state of the art on a variety of dense video captioning benchmarks including YouCook2 [Zhou, 2018a], ViTT [Huang, 2020b] and ActivityNet Captions [Krishna, 2017]. Vid2Seq also generalizes well to the tasks of video paragraph captioning and video clip captioning, and to few-shot settings.

In Chapter 7, we propose VidChapters-7M, a large-scale dataset of user-chaptered videos. VidChapters-7M is automatically created from videos online in a scalable manner by scraping user-annotated chapters and hence without any additional manual annotation. We introduce the following three tasks based on this data. First, the video chapter generation task consists of temporally segmenting the video and generating a chapter title for each segment. To further dissect the problem, we also define two variants of this task: video chapter generation given ground-truth boundaries, which requires generating a chapter title given an annotated video segment, and video chapter grounding, which requires temporally localizing a chapter given its annotated title. We benchmark both simple baselines as well as state-of-the-art video-language models, including Vid2Seq, on these three tasks. We also show that pretraining Vid2Seq on VidChapters-7M transfers well to dense video captioning tasks both in the zero-shot and fine-tuning settings, largely improving the state of the art on the YouCook2 and ViTT benchmarks.

Finally, in Chapter 8, we provide a summary of contributions, discuss open problems, and point out promising future work directions.

Chapter 2

Related Work

In this chapter, we review the literature closely related to this thesis. The chapter is divided into three sections. We start first by presenting foundational work on language modeling in Section 2.1. Next, we study works related to learning visual representations in Section 2.2. Finally, we discuss works that learn joint vision and language models in Section 2.3.

2.1 Language models

We here describe related work that focuses on learning language models, on which we build to design visual language models in this thesis. We discuss word embeddings in Section 2.1.1, recurrent neural networks in Section 2.1.2 and transformers in Section 2.1.3.

2.1.1 Word embeddings

Unlike visual inputs, text data is discrete. To enable machines to understand and process natural language more effectively, text data is often represented with word embeddings. Word embeddings are distributed representations of words in a continuous vector space. They capture semantic and syntactic relationships between words. Several word embedding models have been proposed in the literature. Word2Vec [Mikolov, 2013] is a popular word embedding model that combines two approaches: Continuous Bag-of-Words (CBOW) and Skip-Gram. CBOW predicts a target word based on its context words, while Skip-Gram predicts context words given a target word. FastText [Bojanowski, 2016] is an extension of Word2Vec that introduces subword information into word embeddings. Instead of treating words as atomic units, FastText represents words as bags of character n-grams. This approach allows FastText to handle out-of-vocabulary words and capture morphological similarities. GloVe [Pennington, 2014] is another widely used word embedding model. GloVe embeddings are computed based on aggregated global word-word co-occurrence statistics from a corpus.

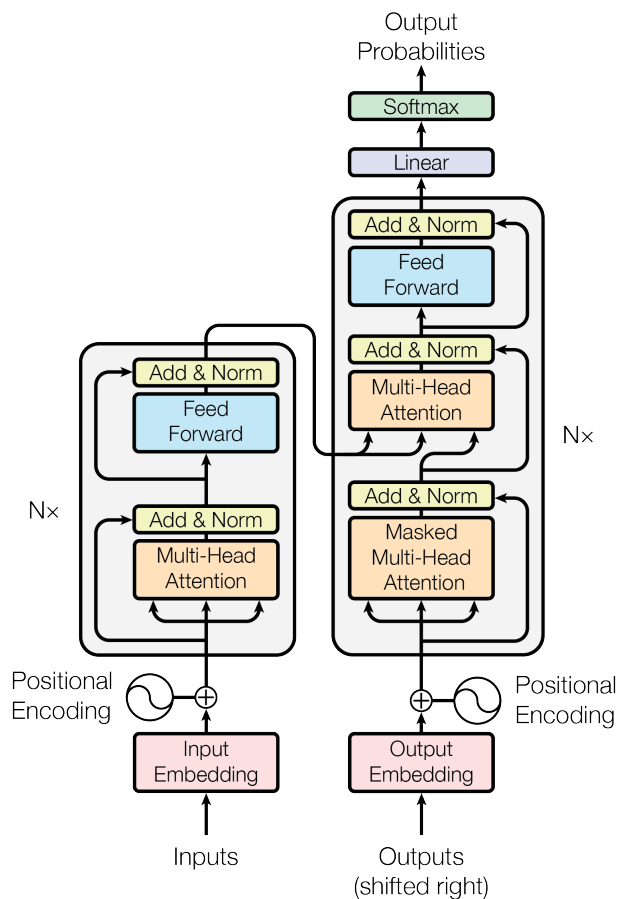


Figure 2.1: **The transformer architecture.** Illustration from [Vaswani, 2017].

2.1.2 Recurrent neural networks

Due to the sequential nature of text data, recurrent neural networks (RNNs) [Rumelhart, 1987] are a popular choice for modeling sequences of text. The vanilla RNN is characterized by recurrent connections that allow information to be processed across sequential data. However, Vanilla RNNs suffer from the vanishing/exploding gradient problem, where gradients either diminish or explode over time, making it difficult to capture long-term dependencies. Long-short term memory network (LSTM) [Hochreiter, 1997] is a type of RNN architecture that addresses the vanishing gradient problem by introducing a memory cell. LSTM has a more complex structure compared to Vanilla RNN, with three main components: input gate, forget gate, and output gate. These gates control the flow of information, allowing LSTMs to selectively update and output information from the memory cell. ELMo [Peters, 2018] showed the benefits of using contextualized word representations learnt with a bidirectional LSTM on a variety of natural language processing tasks including question answering, textual entailment and sentiment analysis. Gated Recurrent Unit (GRU) [Cho, 2014] is another variant of RNN that aims to simplify the LSTM architecture while maintaining its effectiveness. GRUs include a forget gate but no output gate, hence have fewer parameters than LSTMs.

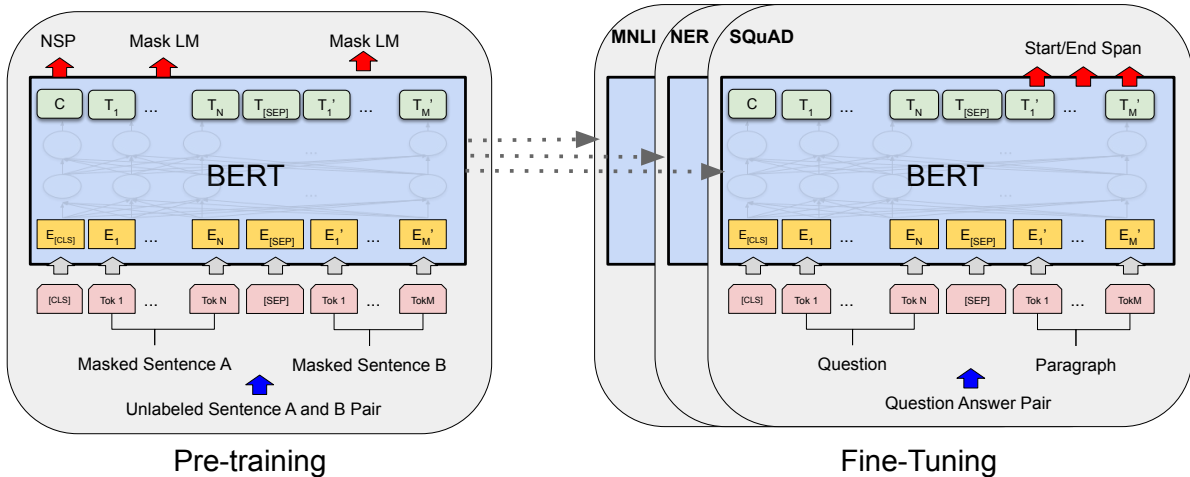


Figure 2.2: **BERT**: pretraining is done with masked language modeling and next sentence prediction on a large text corpus (left); the pretrained model can be simply finetuned on downstream tasks by adding a light output layer to the model (right). Illustration from [Devlin, 2019].

2.1.3 Transformers

The transformer architecture. The attention mechanism is a key concept used in modern deep learning models that process sequential data. [Bahdanau, 2015] showed the benefit of learning to attend to parts of a source sentence that are relevant to predict a target word in neural machine translation. [Vaswani, 2017] then showed that machine translation can be tackled without recurrent connection, exclusively relying on attention. This paper proposes the transformer architecture, which is an encoder-decoder model, as illustrated in Figure 2.1. With the transformer model, the text is transformed into a sequence of discrete tokens and mapped to an embedding space with a learnable token embedding layer. The token embeddings are then added with positional embeddings. The encoder embeds the source sequence and consists of blocks that interleave multi-head self-attention operations and feed-forward layers. The decoder predicts the output sequence and consists of blocks that interleave multi-head self-attention with appropriate masking to avoid future information to be used for previous predictions, cross-attention to the encoder outputs, and feed-forward layers. Layer normalizations [Ba, 2016] and residual connections [He, 2016] are used throughout the network. One advantage of the transformer architecture is that it models the context between every token and all other tokens in a unified way. The visual language models that we design in this thesis largely rely on the transformer architecture.

Pretraining. The transformer architecture motivated a variety of follow-up works, that notably demonstrated the benefit of pretraining transformers on web text corpora and finetuning them on the target tasks. Notably, BERT [Devlin, 2019] showed the benefits of pretraining transformer encoder on large-scale text data (namely, BookCorpus [Zhu, 2015] and Wikipedia), notably by using a masked language modeling objective that aims at predicting randomly masked tokens. For various natural language processing tasks such as question answering and language

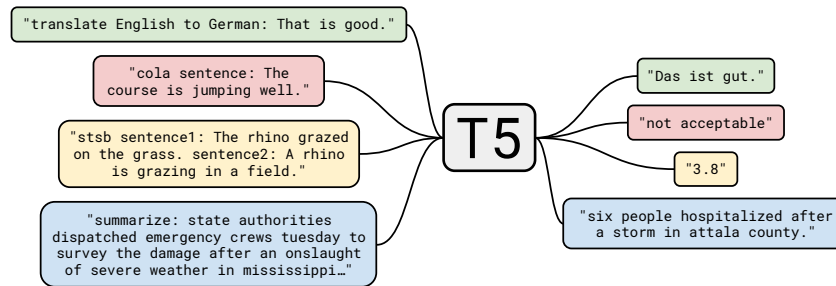


Figure 2.3: **The T5 text-to-text framework.** Illustration from [Raffel, 2020].

inference, the pretrained BERT model can be finetuned simply by adding a light output layer, see Figure 2.2. Encoder-only variants of the BERT architecture include DistilBERT [Sanh, 2019], SpanBERT [Joshi, 2020], RoBERTa [Liu, 2019b] and DeBERTa [He, 2021b]. Notably, RoBERTa improves the performance of BERT by training on more text data up to 160GB of uncompressed text. Furthermore, [Raffel, 2020] observe that all natural language processing tasks can be formulated as text-to-text (see Figure 2.3), and tackled with an encoder-decoder transformer, dubbed T5, via pretraining-finetuning. T5 is pretrained on the large C4 text corpus by encoding a corrupted sequence where some spans are randomly masked, and predicting the masked spans in the decoder. Moreover, FLAN [Wei, 2022a] exhibits the benefits of multi-task finetuning T5 on a variety of manually annotated natural language processing datasets to improve the zero-shot performance on unseen tasks.

Scaling. Another paradigm explored with transformer models pretrained on web text is zero-shot and few-shot learning. GPT-2 [Radford, 2019], which is a 1.5B parameter decoder-only transformer, can perform several natural language processing tasks like such as question answering, machine translation, reading comprehension, and summarization, without any explicit supervision when trained on large-scale text data with the language modeling objective. By scaling up to 175B parameters, GPT-3 [Brown, 2020] further exhibits emerging few-shot prompting learning abilities on many natural language processing tasks, where a few examples of the target task are provided to the language model as a prompt. Importantly, few-shot prompting does not require weight update or expensive manually annotated datasets. These results motivated several works to train large language models like the 280B-parameter Gopher [Rae, 2021] and the 540B-parameter PaLM [Chowdhery, 2022]. Chinchilla [Hoffmann, 2022] then showed that prior large language models are significantly undertrained, and that training the 70B-parameter Chinchilla model on 4 times more data than the 280B-parameter Gopher results in a stronger language model with the same training compute. [Hoffmann, 2022] also recommend to scale the training data and the language model size equally to obtain the best performance with a given amount of compute. However in many cases training longer a smaller language model is beneficial as it is faster at inference than larger ones. While the weights for GPT-3 and Chinchilla are not released, the LLaMa model [Touvron, 2023] was trained on publicly available datasets and publicly released, and its 65B-parameter variant achieves competitive performance compared to GPT-3 and Chinchilla by training longer compared to the Chinchilla scaling laws.

Moreover, InstructGPT [Ouyang, 2022] showed that finetuning GPT-3 with human feedback using reinforcement learning enables to improve its ability to follow instructions. For this, humans rank multiple outputs generated by the language model for various input prompts. This technique has notably been used to develop the popular ChatGPT application [OpenAI, 2023a]. The success of these language models also raised interest in prompt engineering, as evidenced by chain-of-thought prompting [Wei, 2022b] which encourages the language model to provide a series of intermediate reasoning steps.

2.2 Visual models

We here describe related work that focuses on learning visual backbones, on which we build to design visual language models in this thesis. We discuss image models in Section 2.2.1 and video models in Section 2.2.2.

2.2.1 Image models

Image classification models. A popular approach to learn visual representations consists in training deep convolutional networks for image classification on the large-scale ImageNet dataset [Russakovsky, 2015]. With their properties of translation equivariance and their weight sharing mechanism, these models are particularly suited to efficiently process images. A pioneering work in this domain is LeNet [Lecun, 1998], which was applied by several banks to recognise hand-written numbers on checks. With a deeper architecture, AlexNet [Krizhevsky, 2012] set new standards in ImageNet classification accuracy. GoogLeNet, also named Inception v1 [Szegedy, 2015] further improved over these results using small convolutions in parallel to make the network deeper and wider at a given compute budget and using batch normalization [Ioffe, 2015]. VGGNet [Simonyan, 2015] adopted an uniform architecture similar to AlexNet, based on 3x3 convolutions. By incorporating skip connections, ResNet [He, 2016] further scaled up to 152 layers while still having lower complexity than VGGNet, resulting in improved ImageNet classification accuracy, making it an appealing choice to extract visual features. Finally, following their success in natural language processing, recent works have shown that vision transformers (ViT) [Dosovitskiy, 2021] can also perform very well for image classification, especially when pretrained on large amount of data. These models apply the attention mechanism [Vaswani, 2017] to non-overlapping patches of the image.

Object detection models. Beyond image classification models, popular visual representations include object detection models typically trained on datasets like MS COCO [Chen, 2015] and Visual Genome [Krishna, 2016]. The development of modern object detection models also relies on deep convolutional neural networks. Notably, R-CNN [Girshick, 2014] extracts region proposals or regions of interest using an algorithm such as selective search and uses a convolutional neural network to classify the region proposals. Fast R-CNN [Girshick, 2015] further speeds up this process by feeding the image only once to the convolutional neural network and using ROI projection to project the region proposals into the feature space and classify

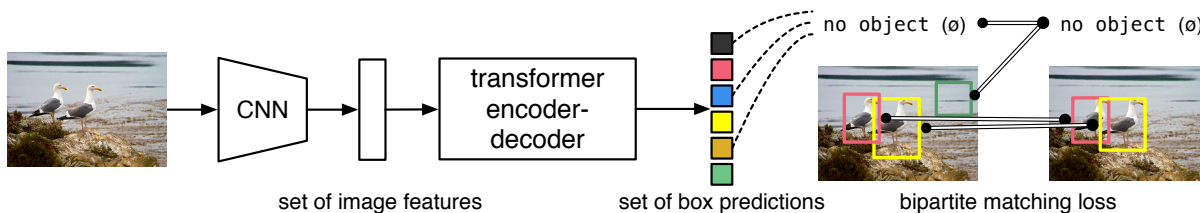


Figure 2.4: **DETR**: object detection is viewed as a set prediction task and tackled with a transformer encoder-decoder architecture. Illustration from [Carion, 2020].

them. Faster R-CNN [Ren, 2015b] bypasses the usage of region proposal extraction algorithm by proposing a Region Proposal Network (RPN), which shares full-image convolutional features with the detection network. Following a fundamentally different approach, the YOLO family of models [Redmon, 2016] achieve compelling speed and accuracy by dividing an image into a grid and detecting objects from anchors of the grid. While these methods originally make use of a convolutional backbone, vision transformers such as Swin Transformer [Liu, 2021b] have also shown to be competitive backbones for object detection. Finally, DETR [Carion, 2020] has showed the possibility of tackling object detection as a set prediction task with a transformer model that self-attends to learnable object queries and cross-attends to visual features, see Figure 2.4. Importantly, DETR achieves competitive performance without using task-specific tricks like non-maximum suppression procedure or anchor generation used in prior methods.

Self-supervised image models. In contrast with previously discussed supervised learning setups, various works have explored self-supervised settings which aim at learning visual representations directly from raw images without using manual annotations. Popular approaches can be divided into 4 categories: deep metric learning, self-distillation, canonical correlation analysis, and masked image modeling. Deep metric learning consists in encouraging semantically transformed versions of an input to have similar embeddings. For instance, SimCLR [Chen, 2020a] learns visual representations by encouraging similarity between two augmented views of an image. The transformed versions are referred to as positives, in contrast with negative instances that are samples we wish to make dissimilar to the positive ones. A popular objective for such contrastive learning setup is the InfoNCE loss [Oord, 2018]. Self-distillation methods such as BYOL [Grill, 2020], SimSIAM [Chen, 2021e], DINO [Caron, 2021] consist in feeding two different views to two encoders, and mapping one to the other with a predictor. These methods employ various techniques to avoid the encoders to collapse and predict a constant for every input, for instance, updating one of the two encoder weights with a running average of the other encoder’s weights. Canonical correlation analysis methods like SwAV [Caron, 2020], Barlow Twins [Zbontar, 2021] and VICReg [Bardes, 2022] infer the relationship between two variables by analyzing their cross-covariance matrix. Finally, masked image modeling approaches such as BEiT [Bao, 2021], MAE [He, 2022] and SimMIM [Xie, 2022] mask out portions of an image and teach a model to inpaint them.

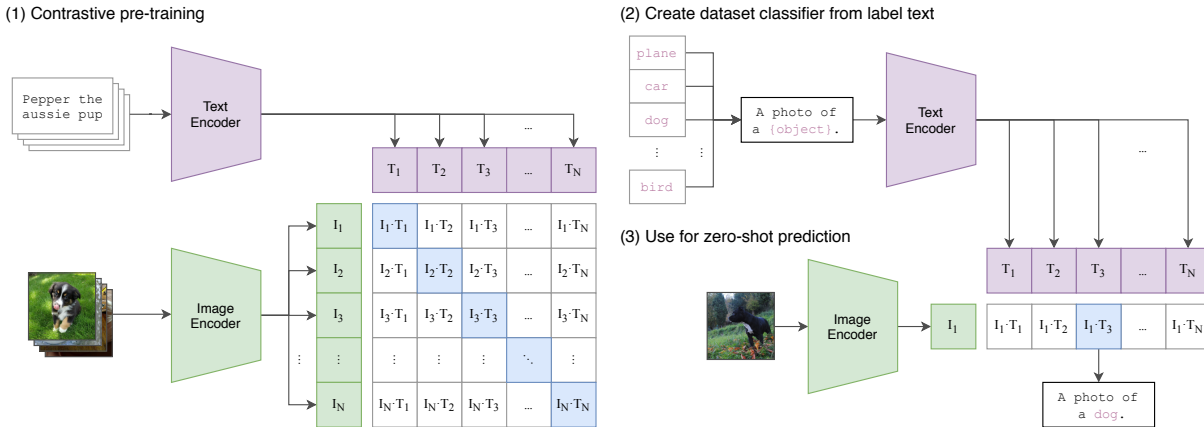


Figure 2.5: **CLIP**: during training, image embeddings are encouraged to be close to their corresponding text embeddings (left); the resulting joint-embedding model can be used to tackle various classification tasks by using textual labels (right). Illustration from [Radford, 2021].

Text-supervised image models. With the abundant number of image-text data available on the web, recent works have explored learning visual backbones from natural language supervision. Publicly available datasets of image-text pairs include SBU [Ordonez, 2011], Conceptual Captions [Sharma, 2018], Conceptual-12M [Changpinyo, 2021], RedCaps [Desai, 2021b] and LAION-5B [Schuhmann, 2022]. VirTex [Desai, 2021a] uses manually annotated captions from MS COCO [Chen, 2015] to learn a visual encoder, by attaching a text decoder head and training to generate the caption given the visual features. CLIP [Radford, 2021] trains an image encoder contrastively with a text encoder using the InfoNCE loss. The model can then transfer in a zero-shot manner to a variety of tasks like image classification, by encoding the image class as a text, as shown in Figure 2.5. After training on a large dataset of 400 million image-text pairs, CLIP demonstrates competitive results compared with fully-supervised approaches. In this thesis, several visual language models that we build rely on the CLIP visual encoder due to its strong ability to extract rich visual features. The CLIP approach has been followed by numerous works, like SLIP [Mu, 2022] which adds a self-supervised learning objective, or FLIP [Li, 2023d] which randomly masks out and removes a large portion of image patches to fasten CLIP training.

2.2.2 Video models

Action recognition models. Popular visual representations for video models include deep models pretrained for action recognition, for instance on the large Kinetics dataset [Carreira, 2017]. For this task, one way to represent videos is to see it as a time series of images and perform temporal fusion of image representations late in the network architecture, for instance using recurrent neural networks [Donahue, 2015]. Another way to represent videos for action recognition consists in modeling it as a 3D volume of pixels and using 3D convolutions, as done in C3D [Tran, 2015] and I3D [Carreira, 2017]. In particular, the I3D model inflates the filters and kernels of 2D convolutions from a pretrained ImageNet model and processes raw frames and optical flow in parallel, fusing their information late in the network. A major challenge with



Figure 2.6: **HowTo100M**: a dataset of 1.2M narrated videos. Illustration from [Miech, 2019].

such 3D models is their expensive computational cost. R(2+1D) [Tran, 2018] and S3D [Xie, 2018] tackle this issue by using a mix of 2D spatial convolutions and 1D temporal convolutions. SlowFast [Feichtenhofer, 2019] achieves compelling compute-performance trade-off by combining a slow pathway operating at low frame rate and a fast pathway operating at high frame rate. Finally, following their success in image classification, vision transformers have also been successfully applied to videos. In contrast with their convolutional counterparts, these models typically operate at a much lower frame rate, which make them suitable for end-to-end learning or fast feature extraction. For instance, ViViT [Arnab, 2021] shows that combining a spatial transformer with a temporal transformer achieves strong compute-accuracy trade-off compared to using a transformer on 3D tubelets. TimeSformer [Bertasius, 2021] consists of blocks that interleave spatial attention, temporal attention and feed-forward layers. Variants of these two architectures that incorporate more inductive bias include Video Swin Transformer [Liu, 2022b] and Multiscale Vision Transformer [Fan, 2021].

Self-supervised video models. Similar to the image domain, various works have explored self-supervised settings which aim at learning visual representations directly from raw videos without using manual annotations. To achieve this, popular objectives include temporal order verification [Misra, 2016; Xu, 2019], encouraging feature persistency over time [Feichtenhofer, 2021], contrastive learning [Dave, 2022; Han, 2020], predicting future representations [Han, 2019] or masked pixels [Tong, 2022].

Text-supervised video models. Similar to the image domain, recent works have explored learning video backbones from natural language supervision relying on web videos and their readily-available textual metadata. Publicly available datasets for this purpose are largely composed of short videos paired with captions, e.g. WebVid-10M [Bain, 2021] and VideoCC [Nagrani, 2022], or narrated videos with speech transcripts aligned over time (ASR), e.g. HowTo100M [Miech, 2019] (see Figure 2.6), YT-Temporal-1B [Zellers, 2021; Zellers, 2022] and HD-VILA-100M [Xue,

2022]. The speech transcripts are typically obtained using the YouTube API speech recognition service, although nowadays open-source models like Whisper [Radford, 2023] achieve state-of-the-art speech recognition performance. For instance, MIL-NCE [Miech, 2020] learns a visual backbone from the HowTo100M dataset via contrastive learning, using multiple positives per video to alleviate the weak temporal alignment between the narration and the visual content in narrated videos, and sampling multiple clips from the same video to obtain hard negative samples. TAN [Han, 2022] predicts the alignment between the narration and the video frames in a self-supervised fashion, and shows that this learnt alignment can help learning stronger backbones. LaViLa [Zhao, 2023] generates new narrations and paraphrases existing narrations to obtain additional training data that also enables learning stronger backbones.

Modeling long videos. While the previously described visual backbones typically process short videos that span several seconds, a wide range of works have explored video tasks that require modeling minutes-long videos, such as temporal action localization which requires temporally localizing and recognizing actions in untrimmed videos. Commonly used datasets for temporal action localization include ActivityNet-1.3 [Caba Heilbron, 2015] and THUMOS’14 [Idrees, 2017]. Due to finite GPU memory constraints, approaches for this task typically use pre-extracted video features [Bai, 2020; Chao, 2018; Lin, 2018; Lin, 2019; Long, 2019; Tan, 2021; Xu, 2020; Zhao, 2017a; Zhao, 2021], or operate at low spatial resolution [Lin, 2021a]. Moreover, most temporal action localization methods can be categorized into two groups: (i) single-stage detectors [Cheng, 2022a; Lin, 2017; Liu, 2020b; Zhang, 2018a; Zhang, 2022a], and (ii) two-stage detectors that require external action recognition classifiers [Bai, 2020; Lin, 2018; Lin, 2019; Lin, 2021a; Liu, 2019c; Qing, 2021; Shou, 2017; Xu, 2020; Zhao, 2017a; Zeng, 2019].

Another task that requires processing long videos is temporal action segmentation, which consists in predicting an action for each frame in an untrimmed video. Popular datasets for this task include Breakfast [Kuehne, 2014], 50Salads [Stein, 2013] and GTEA [Fathi, 2011]. Approaches for this task typically use pre-extracted features to capture local motion information. To capture long-range temporal patterns, the features are then refined by segmentation models like RNNs [Kuehne, 2018; Richard, 2016; Singh, 2016] or temporal convolutional neural networks [Ding, 2018; Farha, 2019; Gao, 2021; Lea, 2017; Lei, 2018b; Li, 2020c; Wang, 2020e].

2.3 Visual language models

We here describe related work that focuses on learning visual language models, building on the two previous sections. We discuss image-language models in Section 2.3.1 and video-language models in Section 2.3.2.

2.3.1 Image-language models

Image-text tasks. There is a long research history of connecting visual perception and linguistic comprehension. Various tasks have been proposed, including text-image retrieval [Chen,

2015] which requires retrieving an image most relevant to a text query in a database of images, visual question answering [Antol, 2015; Goyal, 2017] which requires answering natural language questions about an image, image captioning [Vinyals, 2015] which requires describing an image in natural language and visual reasoning [Suhr, 2019; Zellers, 2019] which requires determining whether a sentence is true about an image. To solve these tasks, early works developed task-specific models. For instance, for the visual question answering task, early successful models [Yang, 2016; Anderson, 2018] typically include a text encoder, an image feature extractor, a multi-modal fusion module with attention and an answer classifier over all possible answers. For the image-text retrieval task, a standard approach [Chowdhury, 2018; Wu, 2017] consists in learning a joint embedding space where image and textual inputs are close in that space if and only if they are semantically similar. For the image captioning task, a common approach [Karpathy, 2015] encodes an image and decodes this representation in textual form autoregressively.

Encoder-only models. The development of pretrained transformer encoders like BERT [Devlin, 2019] that can be simply finetuned for different tasks in natural language processing inspired research in developing similar models for vision and language. Processing vision and language inputs with a transformer requires decomposing the image representation into tokens. A popular way to achieve this consists in pre-extracting object embeddings, as done in VisualBERT [Li, 2019a], ViLBERT [Lu, 2019], VL-BERT [Su, 2019], Oscar [Li, 2020d], UNITER [Chen, 2020b], VILLA [Gan, 2020] and VinVL [Zhang, 2021a]. Another way is to use grid or patch features, as done in PixelBERT [Huang, 2020c], SOHO [Huang, 2021b], ALBEF [Li, 2021a], METER [Dou, 2022b] and FLAVA [Singh, 2022]. Patch features notably enable end-to-end training of the model with the visual backbone without relying on datasets manually annotated with object regions, as done in ALBEF for instance. We can further divide these architectures into two types: self-attentional ones like VisualBERT use self-attention over visual and text embeddings, and cross-attentional ones like ViLBERT make use of cross-attention between a text tower and an image tower to model multi-modal interactions. These models are typically pretrained on large datasets of image-text pairs scrapped from the web like Conceptual Captions [Sharma, 2018], with objectives like masked language modeling and image-text matching. Finetuning for downstream tasks can then be done by adding light layers on top of the transformer encoder, for instance a linear classifier for visual question answering.

Encoder-decoder and decoder-only models. Following the success of transformer encoder-decoder or decoder-only text generation models like T5 [Raffel, 2020] or GPT-3 [Brown, 2020], a plethora of works explored extending these models to vision and language inputs. This includes VL-T5 [Cho, 2021], which unifies visual question answering, visual grounding and image-text matching as text generation. For pretraining, VL-T5 uses a T5-style span unmasking objective, an image-text matching objective and a visual question answering objective, on the MS COCO [Chen, 2015] and Visual Genome [Krishna, 2016] datasets. While VL-T5 relies on manual annotations for pretraining, SimVLM [Wang, 2022f] shows that a single prefix language



Figure 2.7: **Examples of multi-modal few-shot learning with frozen language models.** Illustration from [Tsimpoukelli, 2021].

modeling objective can be effectively used to pretrain an encoder-decoder vision-language transformer on large-scale image-text data. Unlike most of these architectures that include a text encoder, GIT [Wang, 2022a] shows that simply using an image encoder and a text decoder can be sufficient. CoCa [Yu, 2022a] further unifies generative architectures like SimVLM and contrastive architectures like CLIP by combining an image encoder, a text-only decoder and a multi-modal decoder. BLIP [Li, 2022c] shows that model-generated captions can be used as data augmentation to improve the pretraining of the visual language model on datasets of image-text pairs.

Scaling. The benefits of scaling the size of language models have also been transferred to vision and language tasks. A popular technique to achieve this is to freeze a large language model to preserve its knowledge and incorporate vision inputs as a prefix [Tsimpoukelli, 2021]. In particular, Frozen [Tsimpoukelli, 2021] demonstrates that this enables the emergence of few-shot learning ability in vision-language tasks, as shown in Figure 2.7. To increase the learning ability of the model, light adapter layers can be inserted inside the transformers [Eichenberg, 2021]. The FrozenBiLM model we present in Chapter 4 does fall in the family of visual language models built on top of frozen language models using light adapter layers. Also relying on a frozen language model, Flamingo [Alayrac, 2022] demonstrates impressive few-shot prompting ability on multi-modal tasks by training on webpages that contain interleaved images and texts. The Flamingo architecture is built on the Chinchilla language model and an image backbone trained similarly as CLIP, both components being connected via cross-attention. While Flamingo is only trained with the language modeling objective, BLIP-2 [Li, 2023a] uses a two-stage training strategy with a multi-objective representation learning stage to improve the relevance of the visual prefix. Furthermore, PaLI [Chen, 2023b] exhibits the benefits of scaling the size of the visual backbone for the performance on vision and language tasks.

Spatially-grounded tasks. Spatially-grounded tasks that require localizing entities mentioned in text are a key challenge for image-text transformers that do not rely on object features,

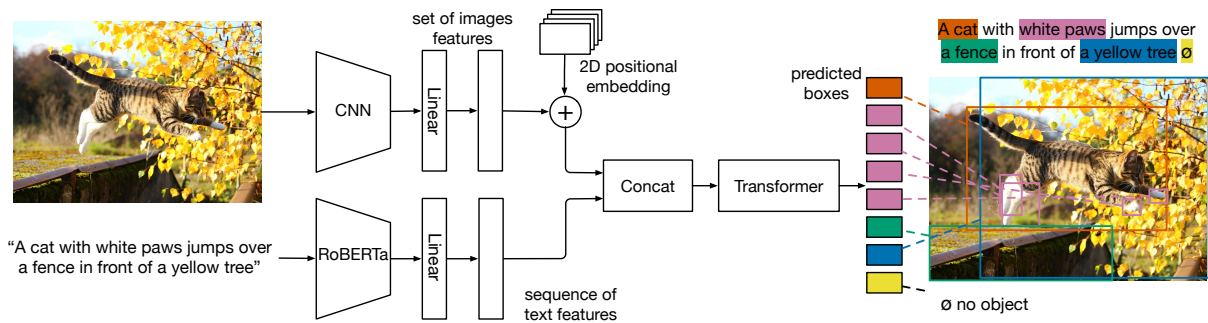


Figure 2.8: **MDETR**: given an input image and an input text, the model outputs boxes together with their alignment in the text. Illustration from [Kamath, 2021].

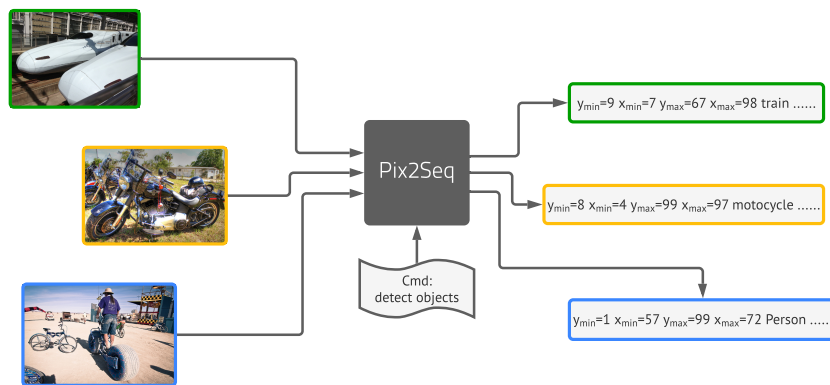


Figure 2.9: **Pix2seq**: object detection is viewed as a language modeling problem, where tokens represent spatial coordinates or class labels. Illustration from [Chen, 2022a].

as they cannot be simply formulated as text generation or classifying a global multi-modal representation. This includes phrase grounding which consists in grounding each entity mentioned by a noun phrase in the caption to a region in the image, and referring expression comprehension which consists in localizing a target object in an image described by a referring expression phrased in natural language. Following the success of DETR [Carion, 2020] in object detection, [Kamath, 2021] develop MDETR, an end-to-end transformer that detects objects in an image conditioned on a raw text query (see Figure 2.8). MDETR is pretrained with box prediction losses, a soft-token prediction loss, and a contrastive alignment loss, on 1.3M image-text pairs manually annotated with region-text alignment, comprising MS COCO [Chen, 2015], Visual Genome [Krishna, 2016] and Flickr30k [Plummer, 2015]. FIBER [Dou, 2022a] then shows that a single model can tackle image-level tasks like visual question answering and region-level tasks like phrase grounding, by first training on large-scale image-text data then on datasets of image-text pairs annotated with region-text alignment. Moreover, GLIP [Li, 2022f] leverages large-scale image-text data for pretraining by generating bounding boxes in a self-training fashion, and unifies object detection and phrase grounding. GLIPv2 [Zhang, 2022b] further extends GLIP to support image-level understanding tasks as well. However, these models require specific heads for localization. A promising approach to unify text outputs and box outputs is Pix2seq [Chen, 2022a], which tackles object detection by generating a single sequence of tokens

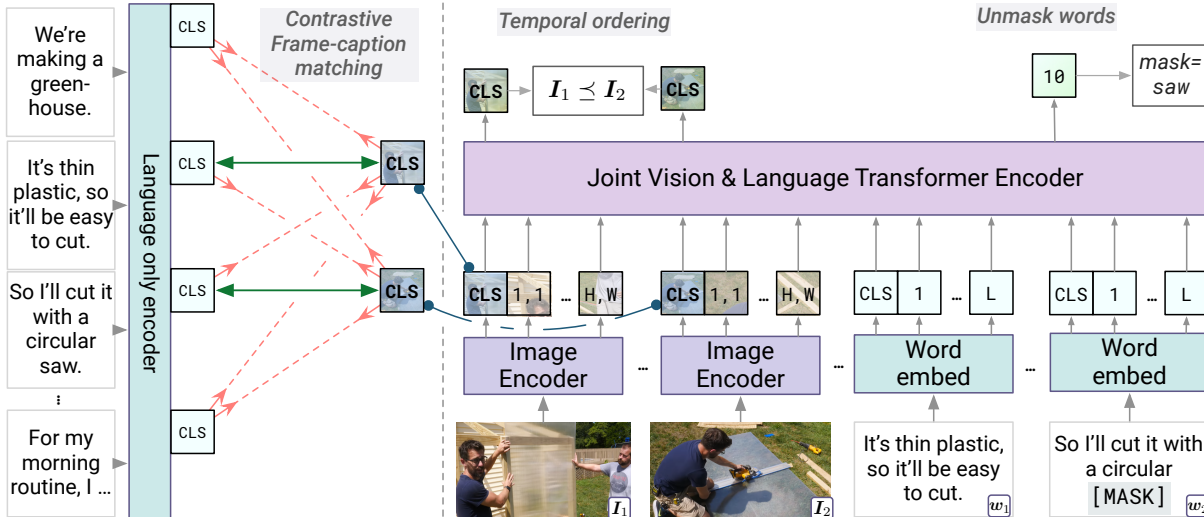


Figure 2.10: **MERLOT**: a vision encoder and a multi-modal transformer encoder are trained from scratch on narrated videos. Illustration from [Zellers, 2021].

comprising tokens that represent spatial coordinates and tokens that represent labels. Following Pix2seq, UniTAB [Yang, 2021e] and OFA [Wang, 2022c] both consist of image-text models that can output text and spatial coordinates. This is done by combining text tokens and tokens representing spatial coordinates. These models can leverage both large datasets of image-text pairs and smaller datasets of image-text pairs manually annotated with region-text alignment.

2.3.2 Video-language models

Global video-text tasks. In analogy with the image domain, various video and language tasks such as text-to-video retrieval [Xu, 2015], video question answering [Tapaswi, 2016; Xu, 2017] and video captioning [Venugopalan, 2015] have been proposed. Models developed for these tasks are similar to their image counterparts, although there are a few differences in the visual encoding and the fusion of the visual representation with language. For instance, early successful video question answering works [Xu, 2017; Jang, 2017] combine a visual encoder which includes appearance and motion features, a text encoder, a multi-modal fusion module reasoning over spatially pooled features and an answer classifier. Popular datasets for this task include MSRVT-QA [Xu, 2017], MSVD-QA [Xu, 2017], ActivityNet-QA [Yu, 2019], TGIF-QA [Jang, 2017], How2QA [Li, 2020b], TVQA [Lei, 2018a] and LSMDC-FiB [Maharaj, 2017]. Additionally, early video captioning efforts [Pan, 2016; Yu, 2016] use a convolutional video encoder and a recurrent text decoder reasoning over spatially pooled features.

Video-language models. Recent works have focused on developing unified models for video-text tasks. For instance, VideoBERT [Sun, 2019b] learns a video-language transformer encoder where the video is tokenized via vector quantization. VideoBERT is pretrained on narrated videos with a masked token modeling objective and a video-text matching loss, and can be finetuned for various tasks like video captioning by adding a simple head on top of the trans-

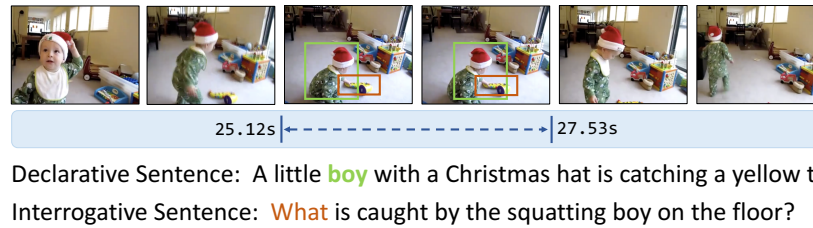


Figure 2.11: **Spatio-temporal video grounding:** the goal is to localize a spatio-temporal tube referred by a text query in an untrimmed video. Illustration from [Zhang, 2020d].

former. ActBERT [Zhu, 2020] also relies on a transformer encoder but represents the video with action features and object features. In contrast with these flat transformer encoder architectures, HERO [Li, 2020b] proposes a hierarchical model with a local visual-linguistic transformer and a global temporal transformer. With the success of vision transformers and end-to-end image-text transformers, a few works have explored end-to-end learning a video-language transformer encoder, notably using narrated videos. For instance, MERLOT [Zellers, 2021] pretrains a joint video-language transformer encoder only from scratch only on narrated videos, as seen in Figure 2.10. MERLOT shows the benefits of diversifying and scaling up the dataset of narrated videos used for pretraining, up to 5M videos, which is 5 times bigger than HowTo100M. Other efforts to pretrain video-language transformers from scratch include Frozen [Bain, 2021] and BridgeFormer [Ge, 2022], which focus on the retrieval task. While these architectures are discriminative, video models capable of generating text have also been developed, such as VX2TEXT [Lin, 2021b] which represents a video by textual labels derived from off-the-shelf video models, MV-GPT [Seo, 2022] which is pretrained to predict narration given previously spoken narration, and Flamingo [Alayrac, 2022] which can be also be applied to videos.

Multi-modality. A specificity of videos is their multi-modal nature – videos may contain audio, and transcribed speech in the audio channel often give cues to solve video and language tasks. This is specifically the case in video datasets of instructional videos such as How2QA [Li, 2020b], YouCook2 [Zhou, 2018b], ViTT [Huang, 2020b], the iVQA dataset we present in Chapter 3, or video datasets of TV shows like TVQA [Lei, 2018a], VIOLIN [Liu, 2020a] or VLEP [Lei, 2020d], or datasets designed for multi-modal understanding such as MUSIC-AVQA [Li, 2022b]. Raw audio cues beyond speech transcripts may also be useful to solve these tasks. Similar to vision and language, there is a long history of research in fusing vision and audio modalities [Chen, 1998; Kazakos, 2019; Nagrani, 2021; Ramachandram, 2017; Xiao, 2020]. In addition, multiple works have pretrained video-language-audio encoders, for instance VATT [Akbari, 2021] and MERLOT Reserve [Zellers, 2022]. Notably, MERLOT Reserve pretraining consists in jointly predicting text and audio given visual inputs, and uses the YT-Temporal-1B dataset that includes 18M narrated videos.

Temporally-grounded tasks. Another specificity of videos compared with images is their temporal aspect. Multiple tasks have been studied to further understand this aspect. The tempo-

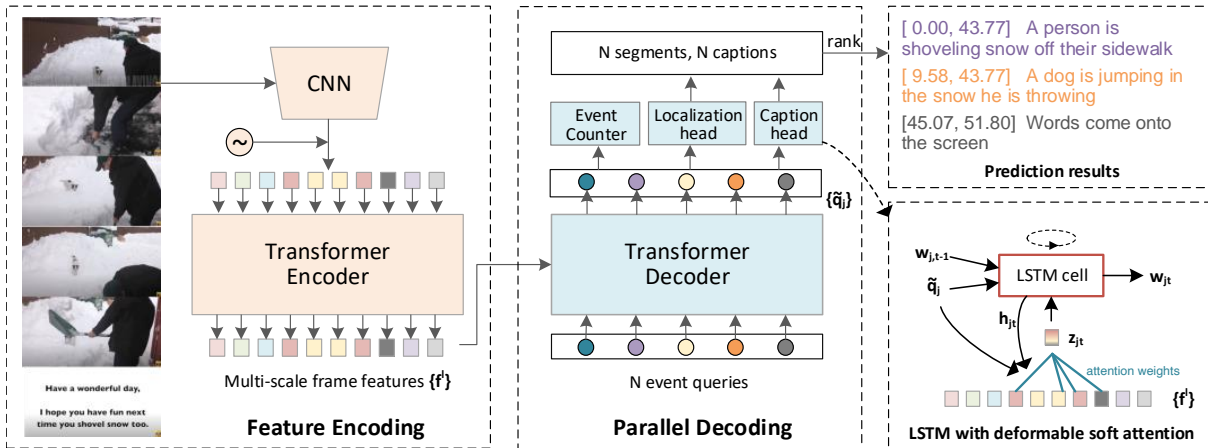


Figure 2.12: **PDVC**: the model jointly predicts captions and their temporal locations for dense video captioning. Illustration from [Wang, 2021d].

ral language grounding task consists in temporally localizing a language query in an untrimmed video. Popular methods for this task include anchor-based methods, regression-based methods and span-based methods. An example of anchor-based method is 2D-TAN [Zhang, 2020c], which models the temporal relations between video moments by a two-dimensional map, where one dimension indicates the starting time of a moment and the other indicates the end time. An example of regression-based method are DRN [Zeng, 2020], which uses the distances between the frame within the ground truth and the starting/ending frame as dense supervisions. An example of span-based method is VSLNet [Zhang, 2020a], which predicts the probability of each video frame being the start and end positions of the target moment. In contrast with the standard Charades-STA [Gao, 2017a], ActivityNet Captions [Krishna, 2017] and TACoS [Rohrbach, 2014] datasets, the QVHighlights dataset [Lei, 2021a] includes multiple relevant moments per video. [Lei, 2021a] also propose Moment-DETR, a model that can detect multiple moments in a video by tackling this task as a set prediction problem similar to DETR. A variant of the temporal language grounding task consists in retrieving a moment in a small set of moments, as proposed in DiDeMo [Hendricks, 2018].

Another variant consists in localizing both spatially and temporally the language query, as done in spatio-temporal video grounding [Zhang, 2020d], see Figure 2.11. This task was originally tackled with two-stage approaches building on pre-extracted object or tube proposals [Zhang, 2020d; Tang, 2021]. Recent methods, including STVGBert [Su, 2021] and the TubeDETR model presented in Chapter 5, consist of a single stage as they perform spatio-temporal video grounding without relying on pre-extracted proposals.

Another interesting task that requires processing untrimmed videos is dense video captioning [Krishna, 2017], which involves temporally localizing and captioning all events in an untrimmed video as illustrated in Figure 1.4. The majority of existing methods for dense video captioning [Krishna, 2017; Iashin, 2020a; Iashin, 2020b; Wang, 2018b; Wang, 2020c] consist of a temporal localization stage followed by an event captioning stage. To enrich inter-task interactions, recent works jointly train the captioning and localization modules. This includes

PDVC [Wang, 2021d] which follows a DETR-style design (see Figure 2.12), the Vid2Seq architecture presented in Chapter 6, as well as various other methods [Chadha, 2021; Chen, 2021b; Deng, 2021a; Li, 2018a; Mun, 2019; Rahman, 2019; Shen, 2017; Shi, 2019a; Wang, 2018b; Zhou, 2018c].

Large-scale video datasets. We previously discussed datasets of short video-caption pairs [Bain, 2021] or narrated videos [Miech, 2019]. In Chapter 3, we also present an approach to leverage these datasets to automatically generate video question answering training data. A handful of works have explored learning video models from other scalable sources of annotations for videos. This includes movie scripts [Laptev, 2008] which describe how the scenes should look like before shooting them, audio narration [Rohrbach, 2015] describing the visual cues and actions in a movie for the visually impaired, and TV subtitles [Lei, 2020c] which describe the plot of the TV show hence do not necessarily describe visual elements but are more widely available than movie scripts and audio narration. Moreover, the YFCC100M dataset [Thomee, 2016] includes web metadata like titles, descriptions and tags. In addition, [Hanu, 2022] show that text-video retrieval can be improved using user-comments. Moreover, in the egocentric video understanding literature, due to the unavailability of large-scale video data available online, manual annotation has been largely scaled up with the Epic-Kitchens dataset [Damen, 2018] and more recently the Ego4D dataset [Grauman, 2022]. In Chapter 7, we explore another scalable source of annotations for videos which consists of user-annotated chapters.

Chapter 3

Learning to Answer Visual Questions from Web Videos

Recent methods for visual question answering rely on large-scale annotated datasets. Manual annotation of questions and answers for videos, however, is tedious, expensive and prevents scalability. In this chapter, we propose to avoid manual annotation and generate a large-scale training dataset for video question answering making use of automatic cross-modal supervision. We leverage a question generation transformer trained on text data and use it to generate question-answer pairs from transcribed video narrations (see Figure 3.1). Given narrated videos, we then automatically generate the HowToVQA69M dataset with 69M video-question-answer triplets. To handle the open vocabulary of diverse answers in this dataset, we propose a training procedure based on a contrastive loss between a video-question multi-modal transformer and an answer transformer. We introduce the zero-shot VideoQA task and the VideoQA feature probe evaluation setting and show excellent results, in particular for rare answers. Furthermore, our method achieves competitive results on MSRVTT-QA [Xu, 2017], ActivityNet-QA [Yu, 2019], MSVD-QA [Xu, 2017] and How2QA [Li, 2020b]. We also use our method to generate the WebVidVQA3M dataset from the WebVid dataset, i.e. videos with alt-text annotations, and show its benefits for training VideoQA models. Finally, for a detailed evaluation we introduce iVQA, a new VideoQA dataset with reduced language bias and high-quality manual annotations. Code, datasets and trained models are available on our project webpage [Yang, 2021a].



Figure 3.1: Given videos with transcribed speech (left) or “alt-text” annotations (right), we leverage language models and cross-modal supervision to obtain large-scale VideoQA data.

3.1 Introduction

Answering questions about videos requires a detailed understanding of the visual content and its association with the natural language. Indeed, given the large diversity of questions, methods for Video Question Answering (VideoQA) should reason about scenes, objects and human actions as well as their complex temporal interactions.

Current approaches to VideoQA rely on deep fully-supervised models trained on manually annotated datasets with question and answer pairs [Fan, 2019b; Huang, 2020a; Jiang, 2020a; Jiang, 2020b; Le, 2020a; Lei, 2021b; Li, 2019b]. Collecting and annotating VideoQA datasets, however, is cumbersome, time consuming, expensive and therefore not scalable. As a result, current VideoQA datasets are relatively small (see Figure 3.2). This limitation hinders the progress in the field as state-of-the-art VideoQA models often require a large amount of training data.

In this work, we address the scale issue with a new approach for automatically generating VideoQA datasets as illustrated in Figure 3.1. The idea is to leverage cross-modal supervision together with text-only tools for question generation and to automatically annotate VideoQA data from a *large amount of videos with readily-available text annotations* in the form of transcribed narrations or “alt-text” annotations available with the video on the Internet. Inspired by the recent progress in language generation using transformer-based language models [Brown, 2020], we leverage transformers trained on a question-answering text corpus to generate a diverse set of non-scripted questions and corresponding open-vocabulary answers from text. By applying these transformers to speech transcripts of narrated videos from the large-scale HowTo100M dataset [Miech, 2019] we create HowToVQA69M, an open-ended VideoQA dataset with 69 million video-question-answer triplets and a diverse set of more than 16M unique answers (see Figure 3.3). We also extend our approach to web videos with readily-available alt-text descriptions and generate the WebVidVQA3M dataset from the WebVid2M dataset [Bain, 2021]. As shown in Figure 3.2, our HowToVQA69M and WebVidVQA3M datasets are orders of magnitude larger compared to prior VideoQA datasets.

Given the limited diversity of existing datasets, current methods typically reduce VideoQA to a classification problem, where frequent answers are assigned to unique classes. Typically, up to 5K unique possible answers are considered. Such an approach, however, does not scale to the open vocabulary of 16M different answers in HowToVQA69M. To address this problem and to enable video question answering with highly diverse questions and answers, we introduce a training procedure based on contrastive learning between a video-question multi-modal transformer and an answer transformer that can handle free-form answers. This bypasses the need to define a discrete set of answer classes.

The goal of our work is to advance truly open-ended and generic solutions to VideoQA. To evaluate generalization, we propose a new zero-shot VideoQA task where we prohibit any manual supervision of visual data during training, and a new VideoQA feature probe evaluation setting where only the final projection layers of the network are finetuned on the target dataset. Our VideoQA model, trained on HowToVQA69M, demonstrates excellent zero-shot results on

multiple existing datasets, especially for rare answers. Additionally, we find that our VideoQA model exhibits strong performance in the VideoQA feature probe evaluation setting. Moreover, when finetuned on target datasets, our model achieves competitive results on MSRVTT-QA [Xu, 2017], ActivityNet-QA [Yu, 2019], MSVD-QA [Xu, 2017] and How2QA [Li, 2020b]. We further show the generalizability of our approach by showing the benefits of WebVidVQA3M for training VideoQA models.

Initial experiments have shown that existing benchmarks for open-ended VideoQA [Xu, 2017; Yu, 2019] contain a language bias [Goyal, 2017], i.e. their questions can often be answered without looking at the video. To better evaluate the impact of visual information in VideoQA, we introduce a new open-ended VideoQA dataset (iVQA) with manually collected questions and answers, where we exclude questions that could be answered without watching the video. Moreover, to account for multiple possible answers, iVQA contains five independently collected answers for each question.

In summary, our work makes the following three contributions:

- (i) We introduce an approach to automatically generate a large-scale VideoQA dataset, HowToVQA69M. Relying on cross-modal supervision, we use transformers trained on an existing text-only question-answering corpus and generate video-question-answer triplets from videos and transcribed narrations. We also apply our method to video alt-text pairs and generate the WebVidVQA3M dataset.
- (ii) We train a VideoQA model on the automatically generated data via contrastive learning between a multi-modal video-question transformer and an answer transformer. We show the efficiency of our model for the new zero-shot VideoQA task and the new VideoQA feature probe task. Our model achieves competitive results in four existing VideoQA benchmarks.
- (iii) Finally, we introduce a new manually annotated open-ended VideoQA benchmark iVQA that excludes non-visual questions and contains multiple possible answers for each question.

3.2 Related Work

Visual Question Answering (VQA). VQA is typically tackled by classifying the image-question (or video-question) representation into a fixed vocabulary of answers. Various approaches to combine spatial image representations and sequential question representations have been proposed [Anderson, 2018; Ben-Younes, 2017; Fukui, 2016; Lu, 2016; Xiong, 2016; Xu, 2016a; Yang, 2016]. More specifically to the video domain (VideoQA), spatio-temporal video representations in terms of motion and appearance have been used in [Dang, 2021; Fan, 2019b; Gao, 2018; Huang, 2020a; Jang, 2017; Jiang, 2020a; Jiang, 2020b; Le, 2020a; Le, 2020b; Lei, 2021b; Li, 2019b; Park, 2021; Seo, 2021a; Xu, 2017; Xue, 2018; Yu, 2021; Zha, 2019; Zhuang, 2020].

Methods above are limited to pre-defined vocabularies of answers and are difficult to apply outside of specific datasets. To address this problem, [Hu, 2018] propose a joint embedding where

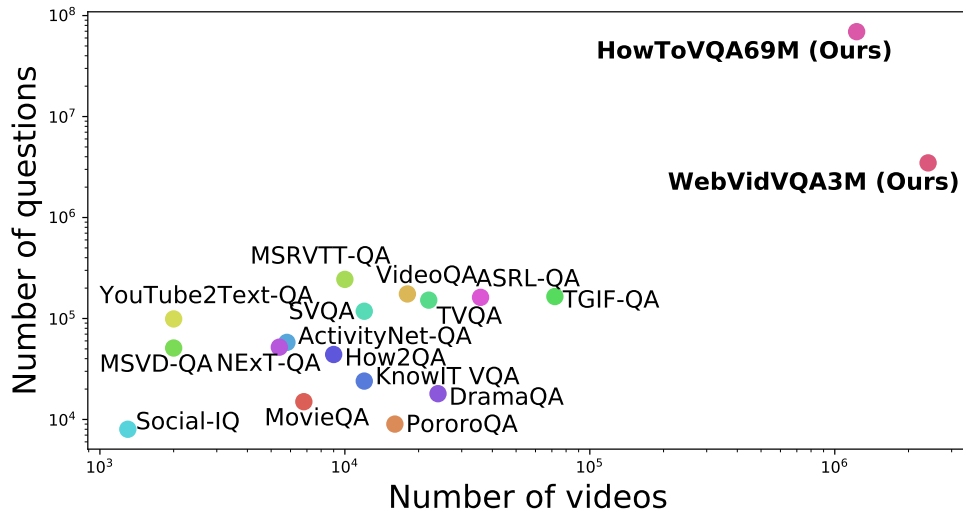


Figure 3.2: Comparison of our large-scale HowToVQA69M and WebVidVQA3M datasets with existing VideoQA datasets.

image-question representations can be matched with free-form answers. Our VideoQA model follows this idea, but instead of relying on manually annotated datasets of limited scale, we train it on a large-scale VideoQA dataset that we automatically generate. In contrast to some previous works using additional video features such as subtitles [Chadha, 2021; Kim, 2020a; Kim, 2020b; Kim, 2021a; Lei, 2018a; Lei, 2020b; Li, 2020b; Lin, 2021b; Tapaswi, 2016; Winterbottom, 2020; Yang, 2020a], our video representation is exclusively based on visual information, as we focus on the detailed visual understanding of videos.

To evaluate the generalization of VQA models, Teney and Hengel [Teney, 2016] define zero-shot VQA by answering previously unseen questions, which is a related but less challenging task compared to the zero-shot VQA task we propose in Section 3.6.2. Vatashsky and Ullman [Vatashsky, 2020] address VQA using MS COCO image annotations [Lin, 2014], while our zero-shot model is trained with no manual annotations. Our proposed zero-shot VQA task is analogous to zero-shot video retrieval [Miech, 2020] or zero-shot action recognition [Radford, 2021]. We further propose a VQA feature probe evaluation setting where only the final heads of the network are finetuned on the downstream dataset while all other pretrained weights are kept frozen. This setting is analogous to the linear probe evaluation setting commonly used in self-supervised image recognition [Caron, 2020; Caron, 2021; Chen, 2021f] or self-supervised action recognition [Radford, 2021] but with multiple layers in the head rather than just a single (linear) layer. Visual question generation (VQG) has been introduced in [Mostafazadeh, 2016]. The methods in [Li, 2018b] and [Shah, 2019] propose to jointly learn VQG and VQA to improve the image VQA task. However, these works do not generate questions to obtain additional training data, but use visual data annotation for VQG as an additional loss.

VideoQA datasets. Manually collecting and annotating video-question-answer triplets is cumbersome, costly and difficult to scale. As a result, current VideoQA datasets [Castro, 2020; Choi, 2021; Colas, 2020; Fan, 2019a; Garcia, 2020; Jang, 2017; Kim, 2017; Lei, 2018a; Li,

2020b; Mun, 2017; Sadhu, 2021; Song, 2018; Tapaswi, 2016; Xiao, 2021; Xu, 2017; Ye, 2017; Yu, 2019; Zadeh, 2019; Zeng, 2017] are limited in size, as the largest, TGIF-QA [Jang, 2017], contains only 72K annotated clips (see Figure 3.2 for more details). To address this issue, several works have explored leveraging manually annotated video descriptions [Jang, 2017; Wang, 2020d; Xu, 2017; Zeng, 2017; Zhao, 2020; Zhao, 2017b; Zhao, 2018] for automatic generation of VideoQA datasets, using rule-based [Heilman, 2010; Ren, 2015a] approaches. Similarly, in the image domain, [Banerjee, 2021] has recently proposed to use annotated image captions from COCO [Chen, 2015] to generate question-answer pairs using a template-based approach [Ren, 2015a].

Instead, we propose to use video annotations in the form of transcribed narrations or alt-text descriptions that are available at large-scale with no manual supervision. Moreover, rule-based generation requires the manual creation of rules by experts which is expensive, and has also been recently outperformed by neural question generation [Du, 2017; Yao, 2018; Zhou, 2017] as used in our approach.

Large-scale pretraining for vision and language. Several recent methods [Alberti, 2019b; Chen, 2020b; Desai, 2021a; Huang, 2020c; Huang, 2021b; Li, 2020a; Li, 2019a; Li, 2020d; Lu, 2019; Lu, 2020; Su, 2019; Tan, 2019; Zhou, 2020] pretrain multi-modal vision-language representations, such as transformers, using datasets with image captions, e.g. COCO [Chen, 2015], Conceptual Captions [Sharma, 2018] and Visual Genome [Krishna, 2016]. These methods are often optimized using generic objectives such as masked language losses and losses for text-image matching and image caption generation. In our work, we pretrain models using large amounts of narrated videos. In contrast to task-agnostic pretraining in the previous work, we show the benefits of task-specific pretraining for our target VideoQA task.

Learning from web videos. In this work, we exploit noisy correlations between videos and readily-available text annotations in unlabeled web videos from the recent HowTo100M [Miech, 2019] and WebVid2M [Bain, 2021] datasets. Methods using such readily-available data have shown significant improvements on several tasks including video retrieval [Bain, 2021; Gabeur, 2020], action localization [Miech, 2019], action recognition [Miech, 2020] and video captioning [Luo, 2020b; Sun, 2019a; Sun, 2019b; Zhu, 2020], sometimes outperforming fully-supervised baselines. Others have used videos with readily available text annotations for the VideoQA task. In detail, [Amrani, 2021] propose a text-video pretraining approach and finetune their model for VideoQA. [Li, 2020b] propose HERO, a pretraining approach restricted to multiple-choice VideoQA, for which questions and answers are treated as a single text stream. [Seo, 2021b] propose a pretraining approach based on next utterance prediction and finetune their model for VideoQA. [Seo, 2021b] propose a pretraining approach based on a mix of frame-level and video-level objectives and finetune for VideoQA. Differently to these methods with task-agnostic pretraining, we propose a pretraining approach specifically dedicated for VideoQA using automatically generated question and answer pairs from readily available text annotations, and show in Section 3.6 the benefits of our approach.

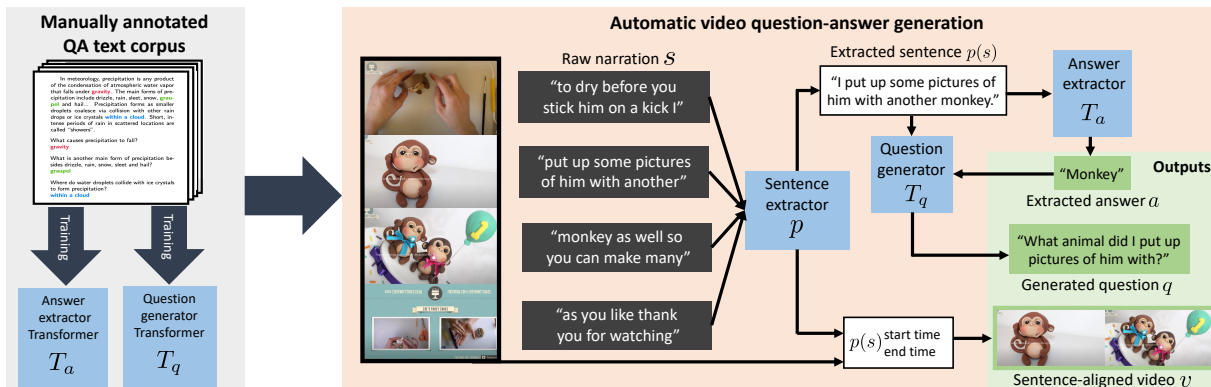


Figure 3.3: **Our automatic approach for large-scale generation of video-question-answer triplets from narrated (subtitled) videos.** First, at the language-only training phase (left), the transformer-based answer extractor T_a and question generator T_q are trained [Raffel, 2020] on a manually annotated text-only question-answer corpus. Then video-question-answer triplets are automatically generated from narrated videos (right). Individual sentences are extracted from the ASR-transcribed narration using a punctuator p . Each extracted sentence is analyzed with an answer extractor T_a and a question generator T_q to produce answer a and question q . The timestamps of the narration are used to obtain a video clip v temporarily aligned to the extracted sentence to form the output video-question-answer triplet (v, q, a) .

3.3 Large-scale generation of VideoQA data

This section presents our approach to generate large-scale VideoQA datasets from videos with readily available text annotations. We illustrate the proposed approach on instructional videos with text annotations in the form of transcribed narrations, which in many cases describe the content of the videos. Section 3.3.1 presents our proposed generation procedures. Section 3.3.2, then, describes the resulting HowToVQA69M dataset. Our approach can be easily adapted to other type of content, for example, shorter web-videos with with readily text annotations in the form of alt-text, as will be shown in the result section (Section 3.6.4).

3.3.1 Generating video-question-answer triplets

We tackle the task of generating video-question-answer triplets from a large-scale instructional video dataset with transcribed spoken narration [Miech, 2019]. This is a challenging task because of transcription errors and lack of punctuation. We also wish to obtain highly diverse data. To address these issues, we propose to leverage powerful language models trained on text data. Our approach is illustrated in Figure 3.3 and details are given next.

We first present details about the generation procedure. Let s be the transcribed speech data obtained with automatic speech recognition (ASR). First, we use a recurrent neural network p , to infer punctuation in the transcribed speech data. We denote the punctuated transcript as $p(s)$. We extract video clips v temporally aligned with the inferred sentences $p(s)$ using the ASR timestamps. We found that the generation works significantly better when applied to sentences rather than the original sentence fragments from the HowTo100M dataset, see Table 3.1. Second,

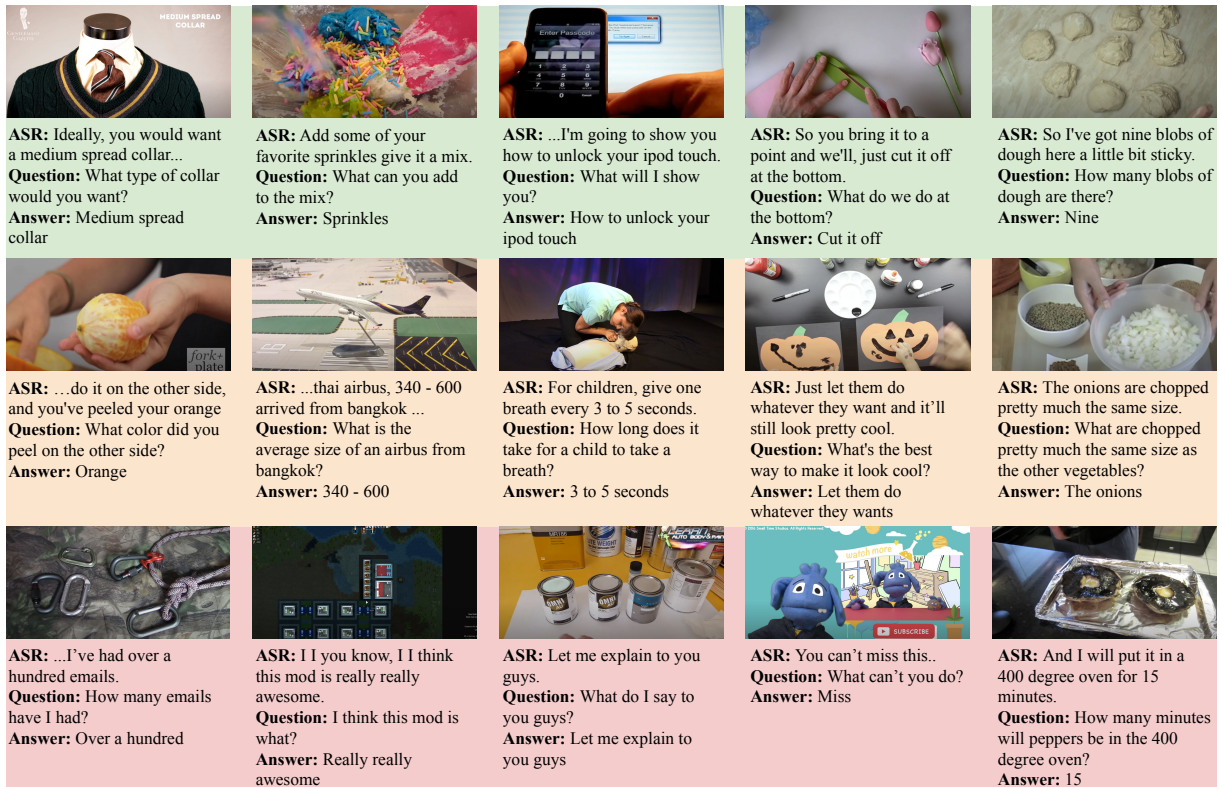


Figure 3.4: Examples of video-question-answer triplets generated from narrated videos in our HowToVQA69M dataset. The green color (first row) indicates relevant examples, the orange color (second row) indicates a failure of the question-answer generation, and the red color (third row) indicates that the generated question-answer is unrelated to the visual content.

for each sentence, we apply a transformer T_a , to extract a set of potential answers: $a = T_a(p(s))$. Third, we use another transformer T_q to generate a question given each transcript sentence and each extracted answer such that: $q = T_q(a, p(s))$. The output is a set of video-question-answer triplets (v, q, a) .

We now explain details of the language models and their training procedure. For ASR, we follow [Miech, 2019] and use the readily-available ASR data provided by YouTube. For punctuation p , we use the BRNN model from [Tilk, 2016] and the weights available at [Tilk, 2017] trained on IWSLT2011 [Federico, 2012]. For T_a and T_q , we use the transformer-based T5-small and T5-base models [Raffel, 2020], respectively. We follow [Alberti, 2019a; Chan, 2019; Lopez, 2020] and use the weights available at [Patil, 2020] trained for answer span extraction and answer-aware question generation, respectively, on SQuADv1 [Rajpurkar, 2016]. SQuADv1 is a text-only question-answering dataset consisting of questions for which the answer is a segment of text extracted from a paragraph.

3.3.2 HowToVQA69M: a large-scale VideoQA dataset

We have applied the previously described procedure to all 1.2M original videos from the HowTo100M dataset [Miech, 2019]. The result is HowToVQA69M, a dataset of 69,270,581 video clip, question

Punctuation	Generation method	Correct Samples	QA Generation Failure	QA unrelated to video
✓	[Heilman, 2010]	17	54	29
✗	Ours	23	49	28
✓	Ours	30	31	39

Table 3.1: Manual evaluation of our generation method (with and without punctuation) on a random sample of 100 examples compared with a rule-based question-answer generation of [Heilman, 2010]. Numbers are obtained with majority voting between 5 annotators.

Question Type	Total	Correct Samples (%)	QA Generation Failure (%)	QA unrelated to video (%)
Attribute	25	28	32	40
Object	17	41	24	35
Action	16	69	19	13
Counting	13	23	15	62
Place	7	0	86	14
People	7	0	43	57
Other	15	13	27	60

Table 3.2: Manual evaluation of our generation method on 100 randomly chosen generated examples split by question type. Results are obtained by majority voting among 5 annotators.

MSVD-QA or ActivityNet-QA, for which answers are on average shorter than 2 words. Each clip lasts 12.1 seconds on average. The distribution of clip duration has a peak at around seven seconds with a long tail of longer clips. These statistics demonstrate the diversity of our HowToVQA69M dataset, in terms of videos, questions and answers.

Word clouds¹ for questions and answers in HowToVQA69M are shown in Figure 3.6 and illustrate the diverse vocabulary in HowToVQA69M as well as the presence of speech-related words such as *okay, right, oh*.

Manual evaluation of HowToVQA69M. As shown in Figure 3.4, HowToVQA69M annotations are noisy, which can be attributed to: (i) errors in speech transcription, (ii) speech not describing the video content, or (iii) errors in question-answer generation. We manually evaluate the quality of 100 randomly sampled (v, q, a) triplets in HowToVQA69M by collecting 5 different annotations for each triplet to reduce variance and report results in Table 3.1. Among 100 triplets generated by our method we find 30 to be correctly generated and matching well to the video content, 31 are incorrectly generated and 39 are correctly generated but unrelated to the video content. To demonstrate the influence of the different components of our automatic question-answer generation procedure, we compare our results with (i) a variant of our approach that does not split transcribed narrations into sentences using a punctuator, and (ii) a rule-based approach [Heilman, 2010] for question-answer generation. Table 3.1 confirms the importance of punctuation and demonstrates the superior performance of our generation method compared

¹Word clouds were generated using https://github.com/amueller/word_cloud.

to [Heilman, 2010]. Further comparison with [Heilman, 2010] is given in Section 3.6.6. In terms of inter-rater agreement, for the 300 generated video-question-answer triplets (100 for each generation method), 94 were in an agreement of all 5 annotators, 198 in an agreement of at least 4 annotators, and 299 in an agreement of at least 3 annotators. This high agreement of annotators demonstrates the reliability of the results in Table 3.1.

We further manually classify the 100 video-question-answer triplets obtained with our method by the question type (“Attribute”, “Object”, “Action”, “Counting”, “Place”, “People”, or “Other”), evaluate the quality of generated triplets for different question types and report results in Table 3.2. Out of the 6 most common categories, we observe that questions related to “Action” lead to the best annotations, “Counting” questions lead to the highest number of QAs unrelated to the video content, and questions related to “Place” lead to the highest number of QA generation errors. Qualitatively, we found that actions are often depicted in the video, while counted quantities (e.g. time, weight, length) mentioned in the speech are hard to guess from the video only. We describe next how we use HowToVQA69M to train our VideoQA model.

3.4 VideoQA model and training procedure

This section presents our VideoQA model (Section 3.4.1) and describes the training procedure (Section 3.4.2). Figure 3.7 gives an overview of the model.

3.4.1 VideoQA model

As illustrated in Figure 3.7, our VideoQA model is composed of two branches: (i) a video-question module f based on a transformer [Vaswani, 2017] and a mapping from the CLS token with a linear function. It takes a pair of video v and question q as input, models the multi-modal temporal interactions between v and q and then outputs an embedding vector $f(v, q) \in \mathbb{R}^d$. (ii) The second branch is a text encoder g that embeds an answer a as $g(a) \in \mathbb{R}^d$. We will denote our model as $VQA-T$, standing for VideoQA-Transformer. Note that using the joint (video, question) and answer embeddings allows us to deal with a large open vocabulary of answers present in our new HowToVQA69M dataset as the model can measure similarity between the input video-question embedding and the embedding of any answer. This is in contrast to using a classification answer module [Huang, 2020a; Jiang, 2020a; Jiang, 2020b; Le, 2020a; Zhuang, 2020] that can choose only from a fixed predefined vocabulary of answers. Our embedding can be also easily finetuned on the different downstream VideoQA datasets, which may contain new answers that have not been seen at training. In contrast, the classification answer module has to be retrained when the vocabulary of answers changes. Next, we give details of the language and video representations, and of the video-question multi-modal transformer and the answer transformer.

Word representation. The question and answer are separately tokenized with the Word-Pieces embedding [Wu, 2016] and fed to DistilBERT [Sanh, 2019]. DistilBERT is a light version

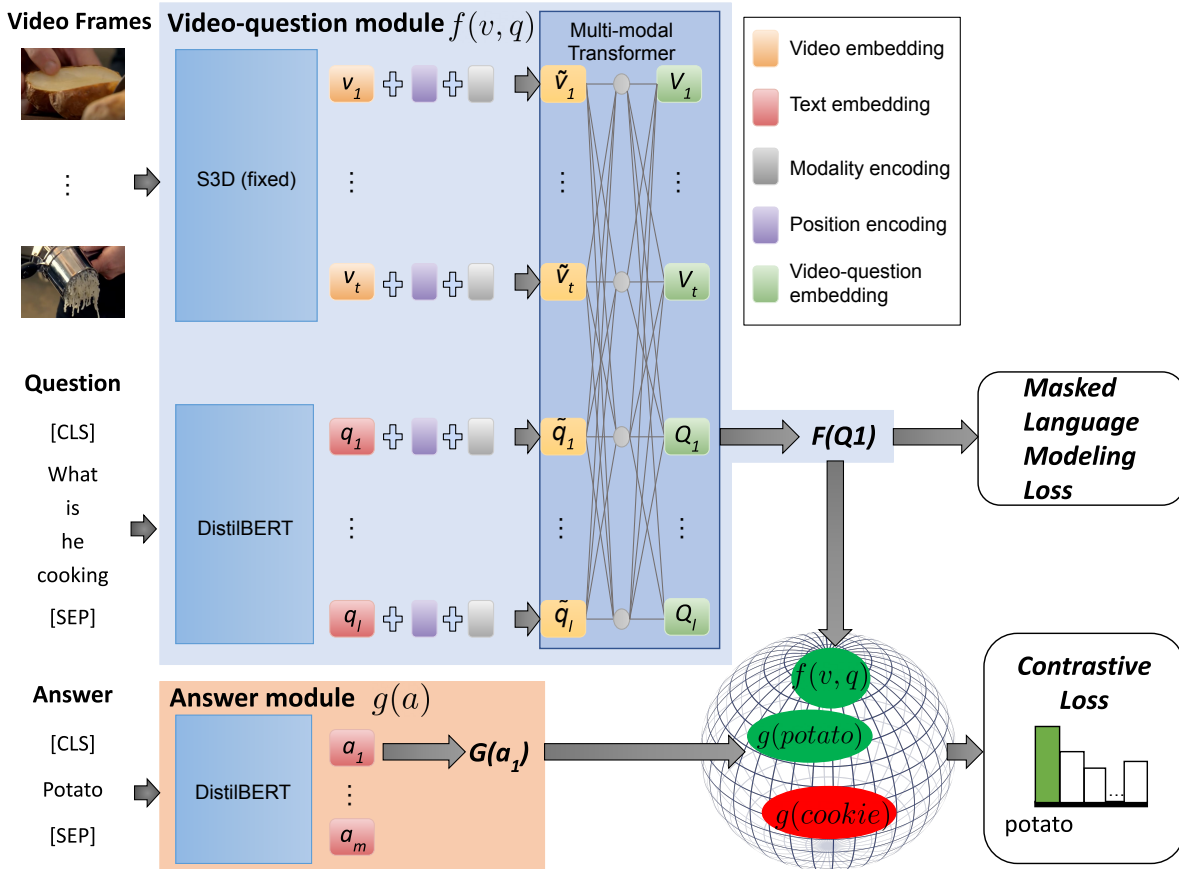


Figure 3.7: **Overview of our VideoQA training architecture.** Our model is composed of a video-question module f based on a multi-modal transformer (top) and an answer module g based on DistilBERT [Sanh, 2019] encoder (bottom). For pretraining, we use a contrastive loss and a masked language modeling loss (right).

of BERT [Devlin, 2019] pretrained in a self-supervised fashion on English Wikipedia and the Toronto Book Corpus [Zhu, 2015].

Video representation. We use a frozen S3D [Xie, 2018] pretrained on HowTo100M [Miech, 2019] using MIL-NCE [Miech, 2020]. This model is pretrained from scratch on HowTo100M only.

Video-question multi-modal transformer. The input video representation, obtained from a fixed S3D model [Xie, 2018], is composed of t features denoted $v = [v_1, \dots, v_t] \in \mathbb{R}^{d_v \times t}$ where d_v is the dimension of the video features, and t is the number of extracted features, one per second. The contextualized representation of the question, provided by the DistilBERT model [Sanh, 2019], is composed of l token embeddings denoted as $q = [q_1, \dots, q_l] \in \mathbb{R}^{d_q \times l}$ where d_q is the dimension of the DistilBERT embedding and l is the number of tokens in the question. The inputs to our video-question multi-modal transformer are then defined as a concatenation of

question token embeddings and video features

$$u(v, q) = [\tilde{q}_1, \dots, \tilde{q}_l, \tilde{v}_1, \dots, \tilde{v}_t] \in \mathbb{R}^{d \times (l+t)}, \quad (3.1)$$

with

$$\tilde{q}_s = dp(\sigma(W_q q_s + b_q) + pos_s + mod_q), \quad (3.2)$$

and

$$\tilde{v}_s = dp(\sigma(W_v v_s + b_v) + pos_s + mod_v), \quad (3.3)$$

where $W_q \in \mathbb{R}^{d_q \times d}$, $b_q \in \mathbb{R}^d$, $W_v \in \mathbb{R}^{d_v \times d}$, $b_v \in \mathbb{R}^d$ and learnable parameters, $mod_q \in \mathbb{R}^d$ and $mod_v \in \mathbb{R}^d$ are learnt modality encodings for video and question, respectively, and $[pos_1, \dots, pos_{l+t}] \in \mathbb{R}^{d \times (l+t)}$ are fixed sinusoidal positional encodings. σ is a Gaussian Error Linear Unit [Hendrycks, 2016] followed by a Layer Normalization [Ba, 2016] and dp refers to Dropout [Srivastava, 2014].

The multi-modal transformer is a transformer with N layers, h heads, dropout probability p_d , and hidden dimension d_h . The outputs of the multi-modal transformer $[Q_1, \dots, Q_l, V_1 \dots V_t] \in \mathbb{R}^{d \times (l+t)}$ are contextualized representations over tokens in the question and temporal video representations. Finally, the fused video-question embedding $f(v, q)$ is obtained as

$$F(Q_1) = W_{vq} dp(Q_1) + b_{vq}, \quad (3.4)$$

where $W_{vq} \in \mathbb{R}^{d \times d}$, $b_{vq} \in \mathbb{R}^d$ are learnable parameters and Q_1 is the multi-modal contextualized embedding of the [CLS] token in the question, as shown in Figure 3.7.

Answer transformer. The contextualized representation of the answer, provided by the DistilBERT model [Sanh, 2019], is composed of m token embeddings denoted as $a = [a_1, \dots, a_m] \in \mathbb{R}^{d_a \times m}$ where d_a is the dimension of the DistilBERT embedding and m is the number of tokens in the answer. Our answer embedding $g(a)$ is then obtained as

$$G(a_1) = W_a a_1 + b_a, \quad (3.5)$$

where $W_a \in \mathbb{R}^{d_a \times d}$, $b_a \in \mathbb{R}^d$ are learnable parameters and a_1 is the contextualized embedding of the [CLS] token in the answer, as shown in Figure 3.7.

3.4.2 Training procedure

This section describes the training of our VideoQA model on the HowToVQA69M dataset and its finetuning on downstream VideoQA datasets.

Training on HowToVQA69M. We wish to make a pair of video and question (v, q) close to its correct answer a measured by the dot product of their embeddings, $f(v, q)^\top g(a)$. In contrast, the incorrect answers should be far, i.e. the dot product with their embeddings should be small.

This can be done by maximizing the following contrastive objective:

$$\max_{f,g} \sum_{i=1}^n \log \left(\frac{e^{f(v_i, q_i)^\top g(a_i)}}{e^{f(v_i, q_i)^\top g(a_i)} + \sum_{(v', q', a') \sim \mathcal{N}_i} e^{f(v', q')^\top g(a')}} \right), \quad (3.6)$$

where (v_i, q_i, a_i) represents a generated triplet (video clip, question, answer) from HowToVQA69M. Given a specific positive triplet (v_i, q_i, a_i) , we construct the set \mathcal{N}_i of negative triplets by concatenating incorrect answers a_j within the training batch to the video-question pair (v_i, q_i) as: (v_i, q_i, a_j) with $a_j \neq a_i$. In particular, if the same negative answer a_j is present multiple times in a batch, we only count it once. We found that sampling the same negative answer multiple times leads to worse results (see Section 3.6.9), which we believe is due to different distributions of answers in the pretraining and downstream datasets. Removing duplicate negatives helps to mitigate this difference.

Finetuning on downstream VideoQA datasets. We leverage the model pretrained on HowToVQA69M and finetune it on a downstream VideoQA dataset that typically has a smaller vocabulary of answers V (e.g. $|V| \sim 4000$). To this end, we adapt the training objective in (3.6) by constructing the negative set \mathcal{N}_i from *all* incorrect answers in V . Note that in such setting (3.6) becomes equivalent to optimizing the standard cross-entropy objective. In the specific case of multiple-choice VideoQA, the set of negatives \mathcal{N}_i is the set of incorrect answers for each sample.

Masked Language Modeling (MLM). In addition to the contrastive loss (3.6) we apply the masking loss [Devlin, 2019] to question tokens during both pretraining and finetuning. We found this to have a positive regularization effect when finetuning the DistilBERT weights (see Section 3.6.9).

3.5 iVQA: a new VideoQA evaluation dataset

In this section we present our **Instructional VQA** dataset (iVQA). We start from a subset of HowTo100M videos and manually annotate video clips with questions and answers. We aim (i) to provide a well-defined evaluation by including five correct answer annotations per question and (ii) to avoid questions which can be answered without watching the video. The dataset is described below.

iVQA Data Collection. iVQA videos are obtained by randomly sampling 7-30 sec. video clips from the HowTo100M dataset [Miech, 2019]. We avoid overlap between datasets and make sure iVQA and HowToVQA69M have no videos in common. Each clip is manually annotated with one question and 5 answers on Amazon Mechanical Turk. We ask workers to annotate questions about objects and scenes in the video and remove videos that could not be annotated. The correctness of annotations is manually verified by the authors. Moreover, we manually

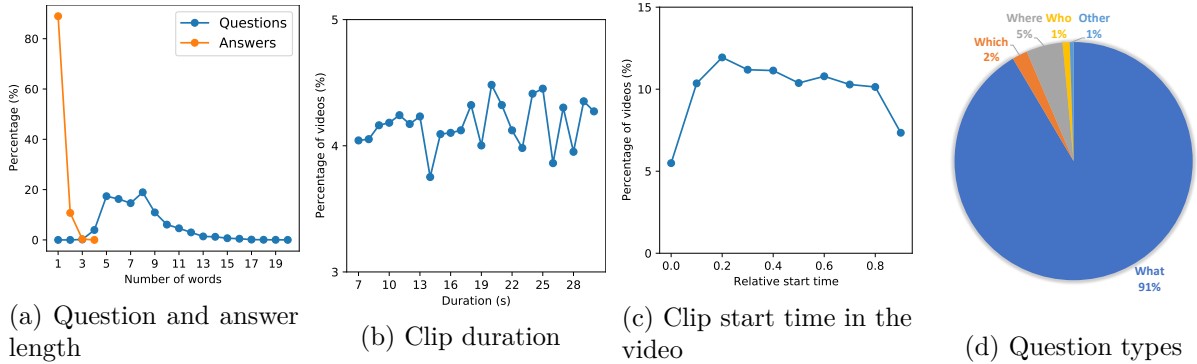


Figure 3.8: **Statistics of the iVQA dataset.** (a) Distribution of length of questions and answers. (b) Distribution of video clip duration in seconds. (c) Distribution of video clip relative start time in the original video. (d) Distribution of question types.

reduce the language bias by excluding questions that could be answered without watching the video. To increase diversity, each question is answered by 5 different workers. The answers are restricted to 4 words and are complemented by a confidence level. Questions that receive multiple answers with low confidence are removed. We further describe our data collection interfaces in [Yang, 2021b] (Appendix C.1.).

Statistical analysis of iVQA. iVQA contains 10,000 video clips with one question and five corresponding answers per clip. We split the dataset into 60%/20%/20% train/validation/test subsets. Figure 3.8 shows the distributions of question length, answer length, clip duration, clip relative start time in the original video and question types. The average duration of video clips is 18.6 seconds. Clip duration and start time distributions are almost uniform because we randomly sampled both the duration and the start time to obtain the clips, which results in a high video content diversity. Most questions are about objects (*What* questions make up 91% of the data), while some are about places (*Where* questions make up 5% of the data) and people (*Who* questions make up 1% of the data). On average, questions and answers contain 7.6 and 1.1 words, respectively. Answers are in great majority one or two words, which is a result of our collection procedure.

The majority of questions have a consensus between at least 2 annotators, i.e. at least 2 annotators providing the same answer. In detail, we observe that 27.0% of questions lead to a perfect consensus among the five answer annotators, 48.4% of questions lead to a consensus among at least four annotators, and 77.3% lead to a consensus among at least three annotators. All but six questions lead to a consensus between at least two annotators. Additionally, 27.5% of questions have two different answers that had a consensus between at least two annotators. Similarly to [Antol, 2015], this motivates us to define the following accuracy measure for a given answer a : $acc(a) = \min(\frac{\#\text{ground truth answers} = a}{2}, 1)$. This metric assigns 100% accuracy to answers confirmed by at least 2 annotators, 50% accuracy to answers confirmed by only 1 annotator and 0% otherwise. Note that this definition is specific to our set-up where we have *multiple* ground truth answers per question.



Figure 3.9: Word clouds for questions and answers in our iVQA dataset. The frequent occurrence of location and time-specific words (*behind*, *front*, *right*, *left*, *first*, *end*, *beginning*) indicates the presence of the spatial and temporal context within iVQA questions. We can also observe the task-specific vocabulary in iVQA answers related to the domains of cooking, hand crafting and gardening.

Word clouds for questions and answers in the iVQA dataset in Figure 3.9 demonstrate the relation of iVQA to the domains of cooking, hand crafting and gardening. These word clouds also indicate that questions in iVQA often require spatial reasoning (*behind*, *front*, *right*, *left*) and temporal understanding (*first*, *end*, *left*, *beginning*) of the video. The most frequent answer (*spoon*) in iVQA corresponds to 2% of all answers in the dataset. In contrast, the most frequent answers in other existing VideoQA datasets account for more than 9% of all answers in these datasets (we have verified this for MSRVTT-QA, MSVD-QA and ActivityNet-QA). As a consequence, the *most frequent answer baseline* is significantly lower for our iVQA dataset compared to other VideoQA datasets. We further evaluate the language bias in iVQA in Section 3.6.8.

3.6 Experiments

This section demonstrates the benefits of training using our generated HowToVQA69M dataset and compares our method to the state of the art. We first outline the used datasets, baseline methods and implementation details in Section 3.6.1. We then present results for the novel zero-shot VideoQA task in Section 3.6.2. Next we present results for the novel VideoQA feature probe evaluation setting in Section 3.6.3. The comparison to the state of the art in VideoQA and alternative training strategies is given in Section 3.6.4. Section 3.6.5 presents results for rare answers and split per question type. Then we compare our VideoQA generation approach to previous methods in Section 3.6.6. We also apply our approach to another video-text datasets in Section 3.6.7. Finally, we show the importance of the visual modality in iVQA in Section 3.6.8

and present ablation studies in Section 3.6.9.

3.6.1 Evaluation Protocol

Datasets. We use three datasets for training and five datasets for evaluation as described below. We follow previous evaluation protocols for open-ended settings [Le, 2020a] and use a fixed vocabulary of training answers. Unless stated otherwise, we report top-1 test accuracy and use original splits for training, validation and test.

For training we use our new **HowToVQA69M** dataset introduced in Section 3.3.2 with 90% and 10% videos in training and validation subsets. For comparison, we also train our model using a large-scale text-video dataset, **HowTo100M** [Miech, 2019], that contains videos with transcribed narrations but *no video-question-answer* triplets. Test and validation videos of downstream datasets are excluded from HowTo100M and HowToVQA69M. To evaluate the general applicability of our approach, we generate another automatic VQA dataset based on **WebVid2M** [Bain, 2021], which consists of 2.5M video-text pairs scraped from the web where video captions are obtained from readily-available alt-text descriptions, see Section 3.6.7.

We evaluate results on four open-ended VideoQA downstream datasets: **MSRVTT-QA** [Xu, 2017], **MSVD-QA** [Xu, 2017], **ActivityNet-QA** [Yu, 2019] and our new **iVQA** dataset (see Section 3.5). We also evaluate on a multiple-choice VideoQA dataset **How2QA** [Li, 2020b] where each question is associated with one correct and three incorrect answers. For MSRVTT-QA and MSVD-QA, we follow [Le, 2020a] and use a vocabulary of the top 4000 training answers for MSRVTT-QA, and all 1852 training answers for MSVD-QA. For our iVQA dataset and ActivityNet-QA, we consider all answers that appear at least twice in the training set, resulting in 2348 answers for iVQA and 1654 answers for ActivityNet-QA.

Baselines. To evaluate the contribution of the visual modality, we compare our $VQA-T$ model with its language-only variant $QA-T$. $QA-T$ does not use video input, *i.e.* we set the input v of the video-question transformer to zero during both training and testing (see Figure 3.7). To evaluate our generated dataset, we also compare $VQA-T$ trained on HowToVQA69M and on HowTo100M. Since HowTo100M has no (v, q, a) triplets, we only train the f branch of $VQA-T$ on HowTo100M using the standard masking and cross-modal matching losses [Chen, 2020b; Li, 2020b; Lu, 2019; Sun, 2019b; Zhu, 2020]. In the zero-shot setting we evaluate $VQA-T$ trained on HowTo100M by computing $f(v, [q, a])$ for concatenated pairs of questions and answers $[q, a]$. During finetuning we also initialize the g branch of $VQA-T$ with parameters of the text encoding obtained from f .

Implementation details. For the VideoQA generation, the input sequence to the answer extractor and question generation transformers are truncated and padded up to a maximum of 32 tokens. The question decoding is done with beam search keeping track of the 4 most probable states at each level of the search tree. We have used the original captions (including stop words) from the HowTo100M dataset [Miech, 2019] and removed word repetitions from adjacent clips.

Method	Pretraining Data	iVQA		MSRVTT-QA		MSVD-QA		ActivityNet-QA		How2QA
		Top-1	Top-10	Top-1	Top-10	Top-1	Top-10	Top-1	Top-10	Top-1
Random	\emptyset	0.09	0.9	0.02	0.2	0.05	0.5	0.05	0.5	25.0
QA-T	HowToVQA69M	4.4	23.2	2.5	6.5	4.8	15.0	11.6	45.8	38.4
VQA-T	HowTo100M	1.9	11.9	0.3	3.4	1.4	10.4	0.3	1.9	46.2
VQA-T (Ours)	HowToVQA69M	12.2	43.3	2.9	8.8	7.5	22.4	12.2	46.5	51.1

Table 3.3: Comparison with baselines for zero-shot VideoQA. Top-1 and top-10 (for open-ended datasets) accuracy are reported.

For the VideoQA model, we use the following hyperparameters: $l = 20$, $t = 20$, $m = 10$, $d = 512$, $d_h = 2048$, $N = 2$, $H = 8$, $p_d = 0.1$, $d_q = d_a = 768$, $d_v = 1024$. The video features are sampled at equally spaced timestamps, and padded to length t . Sequences of question and answer tokens are truncated and padded to length l and m , respectively. Attention is computed only on non-padded sequential video and question features.

For the training on HowToVQA69M, we use the Adam optimizer [Kingma, 2015] and mini-batches with 4096 video clips sampled from 128 random videos. We use a cosine annealing learning rate schedule with initial value of 5×10^{-5} . The optimization over 10 epochs lasts 2 days on 8 Tesla V100 GPUs. For finetuning, we use a cosine annealing learning rate schedule with initial value of 1×10^{-5} , a batch size of 256 and training runs for 20 epochs. The final model is selected by the best performance on the validation set.

For the masked language modeling objective, a token is corrupted with a probability 15%, and replaced 80% of the time with [MASK], 10% of the time with the same token and 10% of the time with a randomly sampled token. To guess which token is masked, each sequential question output Q_i of the multi-modal transformer is classified in a vocabulary of 30,522 tokens, and we use a cross-entropy loss.

For the variant *VQA-T* trained directly on HowTo100M, in the video-text cross-modal matching objective, we sample one video negative and one text negative per (positive) video-text pair, and use a binary cross-entropy loss. The cross-modal matching module is used to perform zero-shot VideoQA for this variant, by computing scores for $f(v, [q, a])$ for all possible answers a , for each video-question pair (v, q) .

3.6.2 Zero-shot VideoQA

In this section, we address the *zero-shot VideoQA* task where we prohibit any manual supervision of visual data during training. We explore this setup to evaluate the generalization of *VQA-T* trained on HowToVQA69M to unseen downstream datasets. For consistency, we use the vocabulary of answers from downstream datasets during testing (see Section 3.6.1).

Zero-shot results are presented in Table 3.3. We first observe that the use of visual cues by *VQA-T* outperforms *QA-T* when both models are trained on HowToVQA69M. This demonstrates the importance of the cross-modality in HowToVQA69M despite the VideoQA annotation being exclusively generated from text-only methods. Since HowToVQA69M has been generated using no manual annotation of visual data, our approach is scalable and can lead to further

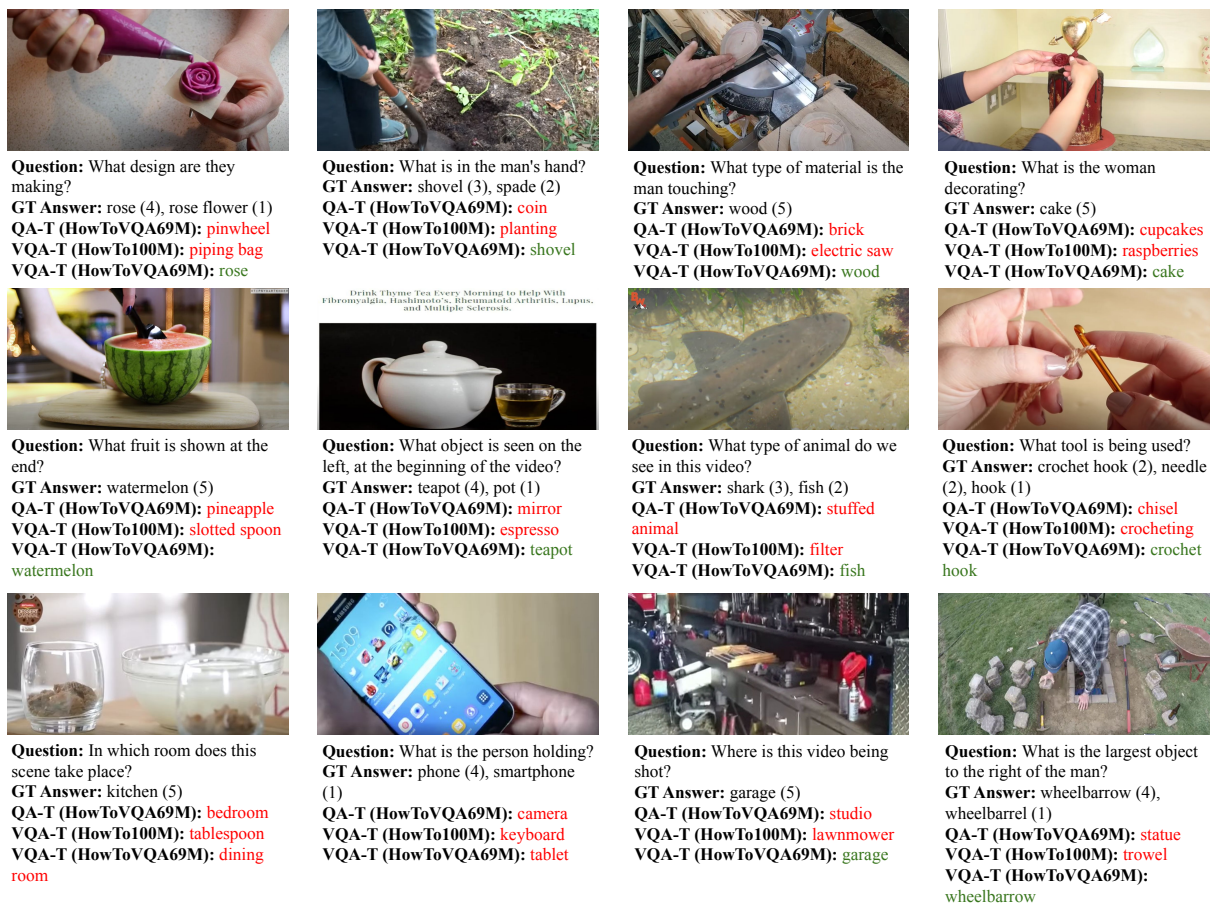


Figure 3.10: **Zero-shot VideoQA on iVQA**. The top 4 rows illustrate successful predictions of our model, while the bottom-most row displays failure cases. The values next to the ground truth (GT) answers indicate the number of annotators that gave the answer. We show more examples on our webpage [Yang, 2021a].

improvements by increasing the dataset size, as we discuss in Section 3.6.9.

Training on HowToVQA69M significantly outperforms the training on HowTo100M and the random baseline. This confirms the advantage of our HowToVQA69M dataset for the VideoQA task over other generic text-video datasets that do not contain video-question-answer triplets. We emphasize that our training does not use any information about target VideoQA datasets. Qualitative results for zero-shot VideoQA are presented for our approach and compared with baselines in Figure 3.10. We observe that *QA-T* (trained on HowToVQA69M) provides plausible but video-unrelated answers to the questions. Moreover, *VQA-T* (trained on HowTo100M) is able to reply with answers related to the visual content, but doesn't take into account the question. Our *VQA-T* model trained on HowToVQA69M, on the other hand, correctly understands questions and uses information in the video to provide correct answers, confirming results in Table 3.3. We also illustrate some failure cases in Figure 3.10, showing that our zero-shot *VQA-T* model can fail to understand fine variations in the video or the language, confusing a *kitchen* with a *dining room* or a *phone* with a *tablet*.

Method	Pretraining data	iVQA	MSRVTT	MSVD	ActivityNet	How2QA
			QA	QA	QA	
VQA-T	\emptyset	3.8	23.2	21.8	22.9	55.3
QA-T	HowToVQA69M	11.4	27.0	29.5	27.6	64.7
VQA-T	HowTo100M	13.8	27.0	32.9	24.7	63.9
VQA-T	HowToVQA69M	24.5	32.9	39.0	30.6	72.9

Table 3.4: Probe evaluation of different pretraining strategies. In each case, only the last projection layers in the model were finetuned on the downstream VideoQA datasets. Top-1 accuracy is reported.

Pretraining data	iVQA	MSRVTT	MSVD	ActivityNet	How2QA
		QA	QA	QA	
\emptyset	23.0	39.6	41.2	36.8	80.8
HowTo100M	28.1	40.4	43.5	38.1	81.9
HowToVQA69M	35.4	41.5	46.3	38.9	84.4

Table 3.5: Benefits of pretraining our *VQA-T* model on our new HowToVQA69M dataset (last row) compared to no pretraining (first row) or pretraining on HowTo100M (second row). In each case our *VQA-T* model was then finetuned on the downstream VideoQA datasets. Top-1 accuracy is reported.

3.6.3 VideoQA feature probe evaluation

In this section we further evaluate the generalization capabilities of the multi-modal representation learnt by our pretrained model. To this end, we analyze the effect of *VQA-T* pretraining followed by the *finetuning of the final projection layers* on the target datasets. More precisely, only the final MLP in the video-question module and the final linear layer in the answer module are finetuned. All other weights in the model (notably the video, question, answer representations and the multi-modal transformer) are kept frozen after the large-scale pre-training.

Results for VideoQA feature probe evaluation are reported in Table 3.4. Similarly as for the zero-shot setting, we observe that the use of visual cues in *VQA-T* outperforms *QA-T* when both models are pretrained on HowToVQA69M. Additionally, pretraining on HowToVQA69M significantly outperforms the pretraining on HowTo100M and the probe baseline trained from scratch. Note that the probe baseline trained from scratch, despite having notably randomly initialized frozen multi-modal transformer weights, achieves reasonable absolute results as it can exploit dataset biases, which are further ablated in Section 3.6.8. Interestingly, we find that on the iVQA dataset, the probe evaluation of our model pretrained on HowToVQA69M (24.5% accuracy, first line in Table 3.4) outperforms the fully supervised model trained from scratch (23.0% accuracy, first line in Table 3.5). These results further confirms the quality of our multi-modal representation learnt from HowToVQA69M.

3.6.4 Benefits of HowToVQA69M pretraining

This section evaluates the effect of *VQA-T* pretraining in combination with finetuning on target datasets. As shown in Table 3.5, pretraining on HowToVQA69M provides consistent and

Method	Pretraining data	MSRVTT	MSVD
		QA	QA
E-SA [Xu, 2017]		29.3	27.6
ST-TP [Jang, 2017]		30.9	31.3
AMU [Xu, 2017]		32.5	32.0
Co-mem [Gao, 2018]		32.0	31.7
HME [Fan, 2019b]		33.0	33.7
LAGCN [Huang, 2020a]		—	34.3
HGA [Jiang, 2020b]		35.5	34.7
QueST [Jiang, 2020a]		34.6	36.1
HCRN [Le, 2020a]		35.6	36.1
MASN [Seo, 2021a]		35.2	38.0
Bridge to Answer [Park, 2021]		36.9	37.2
OCRL+LOGNet [Dang, 2021]		36.0	38.2
ClipBERT [Lei, 2021b]	COCO + Visual Genome	37.4	—
[Jin, 2021]	Conceptual Captions	37.6	38.2
SSML [Amrani, 2021]	HowTo100M	35.1	35.1
CoMVT [Seo, 2021b]	HowTo100M	39.5	42.6
SiaSamRea [Yu, 2021]	COCO + Visual Genome	41.6	45.5
MERLOT [Zellers, 2021]	YT-Temporal-180M	43.1	—
VQA-T	\emptyset	39.6	41.2
VQA-T	HowToVQA69M	41.5	46.3
VQA-T	HowToVQA69M+ WebVidVQA3M	41.8	47.5

Table 3.6: Comparison with state of the art on MSRVTT-QA and MSVD-QA (top-1 accuracy).

	Pretraining data	ActivityNet	How2QA
		QA	
E-SA [Yu, 2019]		31.8	—
MAR-VQA [Zhuang, 2020]		34.6	—
HERO [Li, 2020b]	HowTo100M + TV Dataset	—	74.1
CoMVT [Seo, 2021b]	HowTo100M	38.8	82.3
SiaSamRea [Yu, 2021]	COCO + Visual Genome	39.8	84.1
MERLOT [Zellers, 2021]	YT-Temporal-180M	41.4	—
VQA-T	\emptyset	36.8	80.8
VQA-T	HowToVQA69M	38.9	84.4
VQA-T	HowToVQA69M+ WebVidVQA3M	39.0	85.3

Table 3.7: Comparison with state of the art on ActivityNet-QA and the public val set of How2QA (top-1 accuracy).

significant improvements for all datasets when compared to pretraining on HowTo100M and no pretraining. In particular, we observe the largest improvement for our new iVQA dataset which comes from the same domain as HowToVQA69M. Hence, the automatic generation of training data for other domains using our method can lead to further improvements on other datasets.

We compare our pretrained model to the state-of-the-art in VideoQA in Tables 3.6-3.7. No-

Pretraining Data	Finetuning	iVQA				MSRVTT-QA				MSVD-QA				ActivityNet-QA			
		Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
\emptyset	✓	38.4	16.7	5.9	2.6	68.4	44.1	32.9	8.1	71.2	53.7	28.9	8.8	65.6	49.0	25.7	3.9
HowTo100M	✓	46.7	22.0	8.6	3.6	65.2	46.4	34.9	10.6	74.8	58.8	30.6	10.5	67.5	53.3	25.9	4.1
HowToVQA69M	✗	9.0	8.0	9.5	7.7	0.2	6.4	2.4	3.0	9.3	9.0	6.9	4.8	36.3	5.7	3.7	1.5
HowToVQA69M	✓	47.9	28.1	15.6	8.5	66.9	46.9	36.0	11.5	74.7	59.0	35.0	14.1	66.3	53.0	28.0	5.0

Table 3.8: Results of our $VQA-T$ model with different training strategies, on subsets of iVQA, MSRVTT-QA, MSVD-QA and ActivityNet-QA, corresponding to four quartiles with Q1 and Q4 corresponding to samples with the most frequent and the least frequent answers, respectively.

Pretraining Data	Finetuning	MSRVTT-QA						MSVD-QA					
		What	Who	Number	Color	When	Where	What	Who	Number	Color	When	Where
\emptyset	✓	33.4	49.8	83.1	50.5	78.5	40.2	31.5	54.9	82.7	50.0	74.1	46.4
HowTo100M	✓	34.3	50.2	82.7	51.8	80.0	41.5	34.3	58.6	82.4	62.5	77.6	50.0
HowToVQA69M	✗	1.8	0.7	66.3	0.6	0.6	4.5	7.8	1.7	74.3	18.8	3.5	0.0
HowToVQA69M	✓	35.5	51.1	83.3	49.2	81.0	43.5	37.9	58.0	80.8	62.5	77.6	46.4

Table 3.9: Effect of our pretraining per question type on MSRVTT-QA and MSVD-QA.

Pretraining Data	Finetuning	Motion	Spatial	Temporal	Yes-No	Color	Object	Location	Number	Other
\emptyset	✓	23.4	16.1	3.8	65.6	31.3	26.4	33.7	48.0	33.6
HowTo100M	✓	26.6	17.7	3.5	67.5	32.8	25.3	34.0	50.5	35.8
HowToVQA69M	✗	2.3	1.1	0.3	36.3	11.3	4.1	6.5	0.2	4.7
HowToVQA69M	✓	28.0	17.5	4.9	66.3	34.3	26.7	35.8	50.2	36.8

Table 3.10: Effect of our pretraining per question type on ActivityNet-QA.

tably, $VQA-T$ pretrained on HowToVQA69M outperforms previous methods using comparable pretraining data on all tested datasets. In particular, our method improves over CoMVT [Seo, 2021b] that has been pretrained on HowTo100M. We note that the recent SiaSamRea approach [Yu, 2021] improves over our method on MSRVTT-QA (+0.1%) and ActivityNet-QA (+0.9%), but achieves lower results on MSVD-QA (-0.8%) and How2QA (-0.3%). However, SiaSamRea leverages manually annotated visual data for pretraining (COCO [Chen, 2015] and Visual Genome [Krishna, 2016]). We also note that MERLOT [Zellers, 2021] improves over our method on MSRVTT-QA and ActivityNet-QA, but uses the YT-Temporal-180M dataset for pretraining. This dataset includes HowTo100M but is significantly larger and more diverse (6 millions YouTube videos instead of 1 million).

3.6.5 Analysis of rare answers and question types

Results for rare answers. Training on downstream VideoQA datasets typically leads to particularly large improvements for questions with most frequent answers. As shown in Table 3.8, our approach brings significant improvements both for common and rare answers compared to models trained from scratch or pretrained on HowTo100M. We also find that our pretrained model, in the zero-shot setting, performs similarly across the different quartiles, with the exception of ActivityNet-QA, which includes in its most common answers *yes*, *no*. Interestingly, for the most rare answers in iVQA (Q3 and Q4) our model without finetuning (zero-shot mode) out-

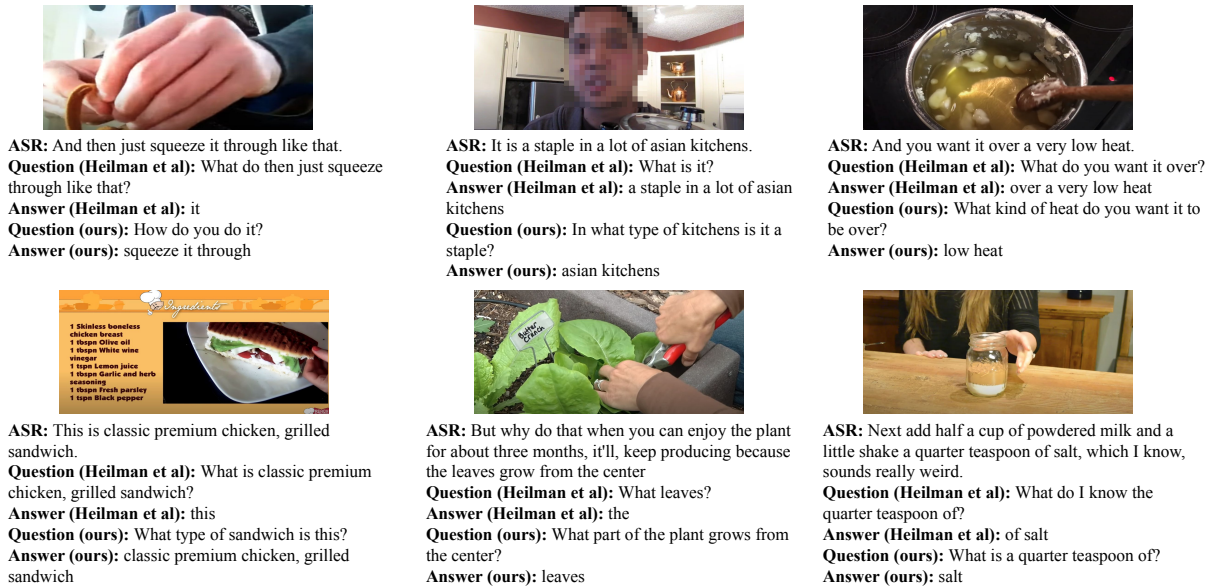


Figure 3.11: Qualitative examples of video-question-answer triplets generated with our trained language models compared to [Heilman, 2010], illustrating the higher quality and diversity of triplets obtained with our generation method.

Generation Method	Zero-shot			Finetune		
	iVQA	ActivityNet QA	How2QA	iVQA	ActivityNet QA	How2QA
[Heilman, 2010]	7.4	1.1	41.7	31.4	38.5	83.0
Ours	12.2	12.2	51.1	35.4	38.9	84.4

Table 3.11: Comparison of our question-answer generation approach with [Heilman, 2010], evaluated by downstream performance of the model trained on the generated VideoQA data.

performs finetuned models that have not been pretrained on HowToVQA69M. We conclude that VideoQA specific pretraining on additional large-scale, diverse data helps improve generalization of VideoQA models.

Note that in order to have a consistent evaluation with other experiments, we keep the same train vocabulary at test time. This implies that a significant part of answers in the test set is considered wrong because the answer is not in the vocabulary. This represents 16% of answers in iVQA, 3% of answers in MSRVTT-QA, 6% for MSVD-QA and 19% for ActivityNet-QA. Note, however, that our joint embedding framework could allow for different vocabularies to be used at the training and test time.

Results split per question type. We also present results per question type for MSRVTT-QA, MSVD-QA and ActivityNet-QA in Tables 3.9 and 3.10. Compared to the model trained from scratch or the model pretrained on HowTo100M, we observe consistent improvements by our model for most categories.

Pretraining Data	Zero-shot				Finetune			
	iVQA	MSRVTT QA	ActivityNet QA	How2QA	iVQA	MSRVTT QA	ActivityNet QA	How2QA
\emptyset	—	—	—	—	23.0	39.6	36.8	80.8
MSRVTT-QA	8.6	—	1.7	42.5	25.2	—	37.5	80.0
ActivityNet-QA	5.5	2.7	—	40.8	24.0	39.9	—	80.7
HowToVQA69M	12.2	2.9	12.2	51.1	35.4	41.5	38.9	84.4

Table 3.12: Comparison of our training on HowToVQA69M with cross-dataset transfer using the previously largest open-ended VideoQA dataset (MSRVTT-QA) and the largest manually annotated open-ended VideoQA dataset (ActivityNet-QA).

3.6.6 Comparison of VideoQA generation methods and VideoQA training datasets

Comparison of VideoQA generation methods. We compare our question-answer generation approach to [Heilman, 2010], that was notably used in [Xu, 2017; Zeng, 2017; Zhao, 2020; Zhao, 2017b; Zhao, 2018] to generate VideoQA data from video descriptions. We run the method of [Heilman, 2010] on sentences extracted from HowTo100M, apply our pretraining method on the generated data and show results in Table 3.11. Note that we do not choose MSRVTT-QA and MSVD-QA as downstream datasets for this comparison because their evaluation sets were automatically generated using [Heilman, 2010]. We find that our generation method leads to significantly better performance both in zero-shot and finetuning settings. We supplement this quantitative comparison with a qualitative comparison shown in Figure 3.11. We found that compared to [Heilman, 2010] our generation method provides higher quality as well as higher diversity of question-answer pairs when applied to the uncurated sentences extracted from speech in narrated videos. This further demonstrates the benefit of our transformer-based question-answer generation approach compared to previous rule-based methods.

Comparison of VideoQA training datasets. We also evaluate the importance of our generated HowToVQA69M dataset by comparing our results to cross-dataset transfer using existing VideoQA datasets. We define cross-dataset transfer as a procedure where we pretrain our VideoQA model on a VideoQA dataset and then finetune and test it on another VideoQA dataset. The training follows the procedure described for finetuning in Section 3.4.2. We report results for cross-dataset transfer in Table 3.12. Note that we do not use MSVD-QA as downstream dataset as its test set has been automatically generated with the same method [Heilman, 2010] as MSRVTT-QA. As can be observed, our approach with pretraining on HowToVQA69M significantly outperforms cross-dataset transfer models using the previously largest VideoQA dataset (MSRVTT-QA), or the largest manually annotated VideoQA dataset (ActivityNet-QA), both for the zero-shot and finetuning settings, on all four downstream datasets. We emphasize that our dataset is generated relying on text-only annotations, while MSRVTT-QA was generated using manually annotated video descriptions and ActivityNet-QA was manually collected. These results further demonstrate the benefit of our HowToVQA69M dataset for



Figure 3.12: Examples of questions-answers generated from video alt-text pairs from the WebVid2M dataset [Bain, 2021]. **The green color** (first row) indicates relevant examples, **the orange color** (second row) indicates a failure of the question-answer generation, and **the red color** (third row) indicates that the generated question-answer is unrelated to the visual content.

Pretraining Data	Zero-shot					Finetune				
	iVQA	MSRVTT QA	MSVD QA	ActivityNet QA	How2QA	iVQA	MSRVTT QA	MSVD QA	ActivityNet QA	How2QA
\emptyset	—	—	—	—	—	23.0	39.6	41.2	36.8	80.8
WebVidVQA3M	7.3	5.3	12.3	6.2	49.8	28.1	41.2	45.4	38.1	82.4
HowToVQA69M	12.2	2.9	7.5	12.2	51.1	35.4	41.5	46.3	38.9	84.4
HowToVQA69M + WebVidVQA3M	13.3	5.6	13.5	12.3	53.1	35.2	41.8	47.5	39.0	85.3

Table 3.13: Comparison of our VideoQA training datasets generated with different video-text data source, evaluated by downstream performance of the model pretrained on the generated data in zero-shot mode and after finetuning.

training VideoQA models.

3.6.7 Generalization to other video-text datasets

In this section, we show that our VideoQA generation approach can be generalized to other sources of non-manually annotated video-text paired data. For this, we extend and apply our generation pipeline presented in Section 3.3.1 to videos with alt-text description, i.e. alt-text HTML attribute associated with videos, from the WebVid2M dataset [Bain, 2021].

Method	iVQA	MSRVTT QA	MSVD QA	ActivityNet QA	How2QA
QA-T	14.1	32.8	32.6	30.4	76.6
VQA-T	23.0	39.6	41.2	36.8	80.8

Table 3.14: Comparison of *QA-T* and *VQA-T* models trained from scratch (without pretraining) on downstream datasets.

WebVidVQA3M dataset. We first explain how we adapt our generation pipeline detailed in Section 3.3.1 to video alt-text pairs. As captions in WebVid2M are relatively short, we do not apply the punctuation model but directly apply the question-answer generation models on the captions. Captions in WebVid2M are also not temporally localized, so the generated question-answers are not temporally localized either. They instead refer to the whole videos, which are typically short (4 seconds on average). Applying our generation pipeline to WebVid2M [Bain, 2021], we generate WebVidVQA3M, a dataset of 3,476,610 question-answers associated with 2,404,871 videos. Examples of generated samples are illustrated in Figure 3.12. These examples show that despite a substantial visual-linguistic domain difference compared to HowTo100M, our approach is able to generate relevant VideoQA data. We believe that qualitatively, the generated QA data from WebVidVQA3M are of better quality than the generated QA data from HowToVQA69M (see Section 3.3.2). We argue that WebVid2M [Bain, 2021] has a better visual-linguistic correlation and a higher quality of text data compared to HowTo100M [Miech, 2019], which facilitates the VideoQA generation.

Benefits of training on WebVidVQA3M. We next apply our pretraining method on the generated data and show results in Table 3.13. We also explore combining both datasets with a simple curriculum learning strategy, where our model initially pretrained on HowToVQA69M is further trained on WebVidVQA3M. We find that training only on WebVidVQA3M gives competitive performance both in the zero-shot setting and the finetuning setting. Notably, it significantly improves over the variant trained from scratch in the finetuning setting. This shows that our approach can be generalized to other sources of video and text data. Additionally, we find that combining the two datasets for pretraining results in additional improvements both for zero-shot and finetuning. Therefore the difference with previous methods is also increased (see Tables 3.6-3.7). Note that as WebVidVQA3M is significantly smaller than HowToVQA69M, our training runs faster on this dataset (20 GPUH instead of 350 GPUH), which gives a practical advantage to WebVidVQA3M. We have open-sourced WebVidVQA3M annotations to facilitate future research.

3.6.8 Importance of the visual modality in iVQA

We show in Table 3.14 that *QA-T* is a strong baseline compared to *VQA-T* on existing VideoQA datasets, when both are trained from scratch. However, on iVQA, *VQA-T* improves even more over *QA-T* than with other datasets, as measured by absolute improvement in top-1 accuracy.

MLM	Sampling without answer repetition	Zero-shot		Finetune	
		iVQA	MSVD-QA	iVQA	MSVD-QA
\times	\times	11.1	6.1	34.7	45.6
\times	\checkmark	12.1	7.0	34.3	45.0
\checkmark	\times	10.9	6.4	34.3	45.1
\checkmark	\checkmark	12.2	7.5	35.4	46.3

Table 3.15: Effect of MLM loss and our negative sampling strategy on HowToVQA69M training.

Pretraining data size	Zero-shot		Finetune	
	iVQA	MSVD-QA	iVQA	MSVD-QA
0%	—	—	23.0	41.2
1%	4.5	3.6	24.2	42.8
10%	9.1	6.2	29.2	44.4
20%	9.5	6.8	31.3	44.8
50%	11.3	7.3	32.8	45.5
100%	12.2	7.5	35.4	46.3

Table 3.16: Effect of the training size of HowToVQA69M.

This suggests that the visual modality is more important in iVQA than in other VideoQA datasets.

3.6.9 Ablation studies

Pretraining losses. As shown in Table 3.15, removing duplicate negative answers in our contrastive loss, as discussed in Section 3.4.2, is beneficial notably in the zero-shot setting. Moreover, adding the MLM loss during pretraining improves the downstream results for both zero-shot and finetuning when used in combination with our contrastive learning strategy. These results motivate our proposed pretraining approach.

Importance of scale. Results of our method after pretraining on different fractions of HowToVQA69M are shown in Table 3.16. We construct these subsets such that larger subsets include the smaller ones. These results suggest that the scale is an important factor and that we can expect further improvements with additional pretraining data, both in the zero-shot and finetuning settings.

3.7 Conclusion

We propose a novel and scalable approach for training VideoQA models without manually annotated visual data. We automatically generate HowToVQA69M – a large-scale VideoQA training dataset generated from narrated videos with readily-available speech transcripts, significantly exceeding existing datasets by size and diversity. We demonstrate several benefits of pretraining on HowToVQA69M. We are the first to demonstrate zero-shot VideoQA results while using

no manually annotated images or videos for training. We also introduce the VideoQA feature probe evaluation setting and show strong generalization capabilities of the multi-modal representation learnt by our pretrained model. Furthermore, finetuning our HowToVQA69M pretrained model on downstream tasks achieves competitive performance on MSRVTT-QA, ActivityNet-QA, MSVD-QA and How2QA. Moreover, we show that our approach generalizes to other sources of web videos by generating the WebVidVQA3M from video alt-text pairs and showing its benefits for VideoQA training. We further validate our approach on our new manually-collected iVQA benchmark.

Limitations Generating question answering data at scale is computationally expensive (the cost is about 10K GPUH for the HowToVQA69M dataset). Moreover, our generation method relies on text-only question-answering manual annotations from the SQuADv1 dataset [Rajpurkar, 2016] to train the question generation models. Furthermore, our VideoQA model cannot make use of the speech modality: if we would train it with transcribed speech input, it would simply learn shortcuts between the transcribed speech, the question and the answer as the question-answer pair is generated from the transcribed speech. We propose an alternative solution to the zero-shot VideoQA problem that alleviates the aforementioned limitations in the following chapter.

Chapter 4

Zero-Shot Video Question Answering via Frozen Bidirectional Language Models

In the previous chapter, we address the video question answering task (VideoQA) in the zero-shot setting, with no manual annotation of visual question-answer. However, the previous approach requires an expensive data generation procedure. An alternative approach adapts *frozen autoregressive* language models pretrained on web-scale text-only data to multi-modal inputs. In contrast, in this chapter, we build on *frozen bidirectional* language models (BiLM) and show that such an approach provides a stronger and cheaper alternative for zero-shot VideoQA. In particular, (i) we combine visual inputs with the frozen BiLM using light trainable modules, (ii) we train such modules using Web-scraped multi-modal data, and finally (iii) we perform zero-shot VideoQA inference through masked language modeling, where the masked text is the answer to a given question (see Figure 4.1). Our proposed approach, FrozenBiLM, outperforms the prior state of the art in zero-shot VideoQA by a significant margin on a variety of datasets, including LSMDC-FiB [Maharaj, 2017], iVQA [Yang, 2021b], MSRVT-QA [Xu, 2017], MSVD-QA [Xu, 2017], ActivityNet-QA [Yu, 2019], TGIF-FrameQA [Jang, 2017], How2QA [Li, 2020b] and TVQA [Lei, 2018a]. It also demonstrates competitive performance in the few-shot and fully-supervised setting. Our code and models are publicly available at [Yang, 2022a].

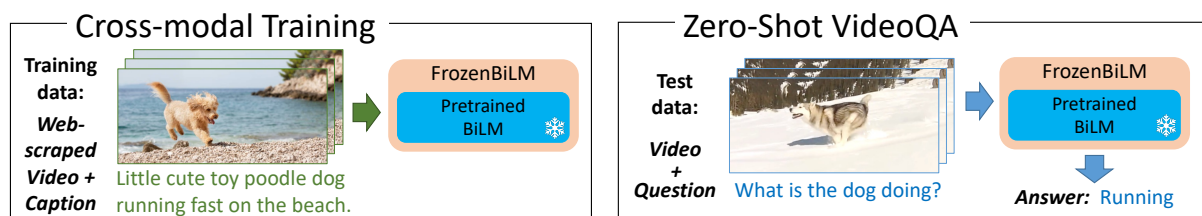


Figure 4.1: Our model FrozenBiLM builds on a pretrained and *frozen* bidirectional language model (BiLM), and is trained from web-scraped video-caption pairs. FrozenBiLM excels in the zero-shot video question answering task without using any explicit visual question-answer supervision.

4.1 Introduction

Video question answering (VideoQA) is a challenging task that requires fine-grained multi-modal understanding. State-of-the-art approaches to VideoQA [Le, 2020a; Yu, 2021; Zellers, 2021] rely on large video datasets manually annotated with question-answer pairs. Yet, collecting such annotations is time consuming, expensive and therefore not scalable. This has motivated the development of *zero-shot* VideoQA approaches [Yang, 2021b; Yang, 2022c; Zellers, 2022], that use no visual question-answer annotation for training, see Figure 4.1.

Recently, a promising line of work builds on *frozen* large autoregressive language models [Eichenberg, 2021; Mokady, 2021; Tsimpoukelli, 2021; Wang, 2022e; Yang, 2021f; Zeng, 2023] for zero-shot visual question answering. This has been motivated by the findings from GPT-3 [Brown, 2020] which exhibits strong zero-shot text-only question answering abilities from large autoregressive language models. Such models [Brown, 2020; Raffel, 2020; So, 2021; Vaswani, 2017] can predict an arbitrarily long sequence of text, one token at each step from left to right. However, they usually require billion parameters to work well, making them computationally expensive to train, and challenging to deploy in practice.

In contrast, recent work in natural language [Mahabadi, 2022; Schick, 2021a; Schick, 2021b; Tam, 2021] demonstrates strong zero-shot performance for lighter bidirectional language models (BiLM). Such models [Devlin, 2019; He, 2021b; Joshi, 2020; Lan, 2020; Liu, 2019b; Sanh, 2019] can predict a few masked tokens in an input sequence given left and right context in a single forward pass. These works cast downstream tasks in *cloze* form¹ [Taylor, 1953], similar to the masked language modeling task (MLM) [Devlin, 2019] solved by these models at pretraining. This motivates us to tackle diverse zero-shot multi-modal tasks (open-ended VideoQA [Xu, 2017], multiple-choice VideoQA [Lei, 2018a] and fill-in-the-blank [Maharaj, 2017]) by formulating them in *cloze* form and leveraging the text-only knowledge of pretrained BiLM.

To adapt a pretrained BiLM to multi-modal inputs, we combine it with a frozen pretrained visual backbone and a set of lightweight additional modules including adapters [Houlsby, 2019]. We train these modules on Web-scraped video-text data using a simple visually-conditioned MLM loss. We preserve the uni-modal knowledge of a BiLM by *freezing* its weights. To our knowledge, our approach is the first to explore the zero-shot visual-linguistic capabilities of *frozen non-autoregressive* language models.

We show that our approach largely improves over the prior state of the art on various zero-shot VideoQA benchmarks. Furthermore, we demonstrate that *frozen bidirectional* language models perform better while being cheaper to train than *frozen autoregressive* language models [Tsimpoukelli, 2021]. Moreover, our ablation studies show (i) the ability of our model to effectively perform zero-shot multi-modal reasoning using both visual cues and speech transcripts, (ii) the importance of adapters combined with *frozen* pretrained language models, (iii) the impact of multi-modal data scale, (iv) the impact of the language model size and of bidirectional modeling. Our approach also performs competitively in the fully-supervised setting. Indeed, we show the benefits of *freezing* the weights of a BiLM when using VideoQA training

¹“Cloze test” is an exercise test where certain portions of text are occluded or masked and need to be filled-in.

data, while updating considerably less parameters compared to alternative methods. Finally, we introduce a new few-shot VideoQA task in which we finetune our pretrained model on a small fraction of the downstream training dataset, and show promising results in this setting.

In summary, our contributions are three-fold:

- (i) We present FrozenBiLM, a framework that handles multi-modal inputs using *frozen* bidirectional language models and enables zero-shot VideoQA through masked language modeling.
- (ii) We provide an extensive ablation study and demonstrate the superior performance of our framework in the zero-shot setting when compared to previous autoregressive models.
- (iii) Our approach improves over the prior state of the art in zero-shot VideoQA by a significant margin. FrozenBiLM also demonstrates competitive performance in the fully-supervised setting and shows strong results in the few-shot VideoQA setting which we introduce.

4.2 Related Work

Zero-shot VideoQA. A vast majority of VideoQA approaches rely on relatively small, manually annotated VideoQA datasets [Amrani, 2021; Castro, 2020; Chadha, 2021; Choi, 2021; Colas, 2020; Dang, 2021; Fan, 2019b; Gao, 2018; Garcia, 2020; Huang, 2020a; Jiang, 2020a; Jiang, 2020b; Kim, 2020a; Kim, 2020b; Kim, 2017; Kim, 2021a; Le, 2020a; Le, 2020b; Lei, 2020b; Li, 2019b; Lin, 2021b; Mun, 2017; Park, 2021; Sadhu, 2021; Seo, 2021b; Seo, 2022; Song, 2018; Tapaswi, 2016; Xiao, 2021; Xue, 2018; Yang, 2020a; Ye, 2017; Zha, 2019; Zhuang, 2020]. Recently, a few work [Yang, 2021b; Zellers, 2022] have explored zero-shot approaches for VideoQA, where models are *only* trained on automatically mined video clips with short text descriptions. In contrast to VideoQA annotations, such video-text pairs are readily-available at scale on the Web [Bain, 2021; Miech, 2019; Zellers, 2021]. In particular, Yang et al. [Yang, 2021b] automatically generate VideoQA training data using language models [Raffel, 2020] pretrained on a manually annotated text-only question-answer corpus [Rajpurkar, 2016]. Reserve [Zellers, 2022] uses GPT-3 [Brown, 2020] to rephrase questions into sentences completed by a multi-modal model. In contrast to these prior works [Yang, 2021b; Zellers, 2022], our method does not require any kind of explicitly annotated language dataset or the use of data generation pipelines for zero-shot VideoQA. Note that BLIP [Li, 2022c] studies a related setting where a model trained on manually annotated image-question-answer triplets is transferred to VideoQA, which is a less challenging task. Also note that VideoCLIP [Xu, 2021] considers a related zero-shot multiple-choice video-to-text retrieval task as VideoQA, but in this setting the model is not provided with natural language questions.

Visual language models. As language models require large amounts of training data to perform well [Hoffmann, 2022], recent works have studied transferring pretrained language models [Brown, 2020; Wang, 2021a] to image-text tasks. VisualGPT [Chen, 2021a] and VC-GPT [Luo, 2022] showed the benefit of initializing the weights of an image captioning model with

a pretrained autoregressive language-only model. Recent work pushed this idea further by *freezing* the weights of a pretrained autoregressive language model for tackling vision and language tasks [Alayrac, 2022; Eichenberg, 2021; Mokady, 2021; Tsimpoukelli, 2021; Wang, 2022e; Yang, 2021f; Zeng, 2023]. Our approach also leverages a *frozen* pretrained language model. Similar to MAGMA [Eichenberg, 2021], we also use adapter layers [Houlsby, 2019; Hu, 2022a]. However, we differ from these approaches as we propose to instead use lighter *bidirectional masked language models*, instead of autoregressive ones, and rely on a masked language modeling objective (MLM) instead of an autoregressive one. Moreover, our model is specifically designed for videos, for which high-quality visual question answering annotation is even more scarce compared to still images [Eichenberg, 2021; Mokady, 2021; Tsimpoukelli, 2021; Yang, 2021f]. We also explore the use of the speech modality, and tackle tasks which are challenging for autoregressive language models such as video-conditioned fill-in-the-blank [Maharaj, 2017]. Finally we show in Section 4.4.3 the superior performance of frozen bidirectional language models in comparison with autoregressive ones [Tsimpoukelli, 2021].

Masked Language Modeling in vision and language. The MLM objective was initially introduced in natural language [Devlin, 2019; Lan, 2020; Liu, 2019b] to pretrain bidirectional transformers and learn generic representations. This approach achieved state-of-the-art results in many language tasks after finetuning on downstream datasets. Its success inspired numerous works to adapt it to train multi-modal transformer models on paired visual-linguistic data [Chen, 2020b; Fu, 2021; Gan, 2020; Hendricks, 2021; Huang, 2020c; Kim, 2021b; Lei, 2021b; Li, 2020a; Li, 2019a; Li, 2020b; Li, 2020d; Li, 2021a; Li, 2022a; Lu, 2019; Lu, 2020; Shen, 2021; Singh, 2022; Su, 2019; Sun, 2019b; Tan, 2019; Wang, 2023; Wang, 2021b; Yu, 2020; Zellers, 2021; Zhou, 2020; Zhu, 2020]. However, these works typically use it to learn generic visual-linguistic representations by updating the transformer weights, and then use expensive manual supervision to train randomly initialized task-specific answer classifiers for VQA [Chen, 2020b; Gan, 2020; Li, 2020a; Li, 2021a; Li, 2020d; Lu, 2019; Shen, 2021; Singh, 2022; Su, 2019; Tan, 2019; Wang, 2021b; Yu, 2020] or VideoQA [Fu, 2021; Lei, 2021b; Li, 2022a; Wang, 2023; Zellers, 2021]. In contrast, we tackle *zero-shot* VideoQA, i.e. without using *any* manual annotation. Moreover, we do not update the transformer weights during cross-modal training, but instead exhibit the benefits of *freezing* these weights after text-only pretraining, for both zero-shot and fully-supervised VideoQA (see Sections 4.4.2 and 4.4.5).

4.3 Method

This section presents our approach to tackle *zero-shot* video question answering. Here, zero-shot means that we do not use *any* visual question answering annotation and only rely on scalable data from the Web. Our approach starts with two strong pretrained components: (i) a text-only bidirectional masked language model (BiLM) pretrained on data from the Internet, which has the capability of zero-shot question answering but is not capable of visual reasoning, and (ii) a vision encoder pretrained to map images to text descriptions, but which does not have the ability

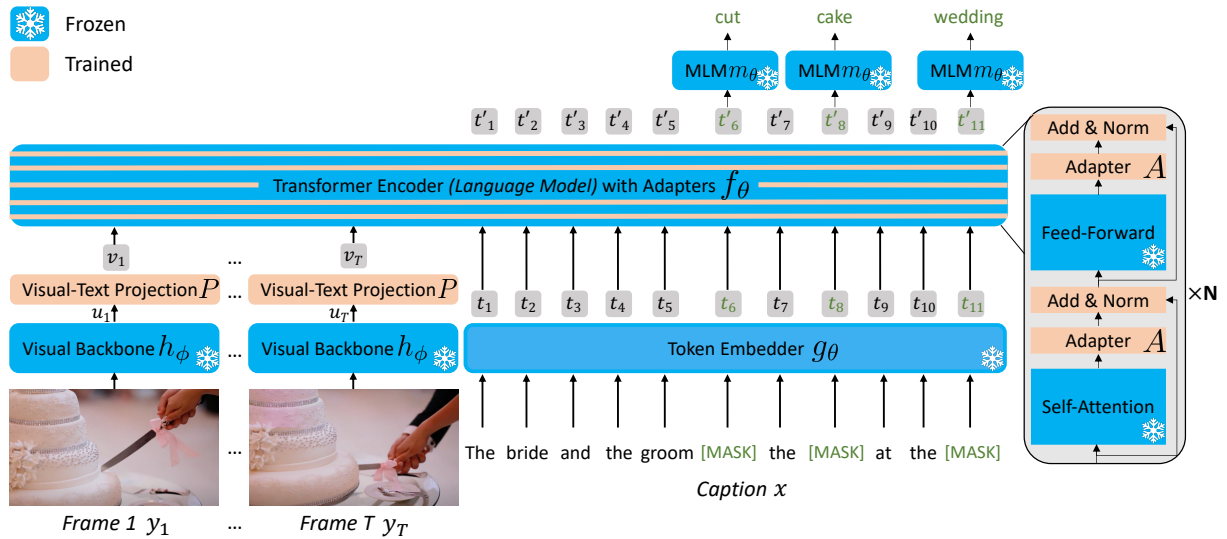


Figure 4.2: **Our training architecture** consists of a large *frozen* bidirectional language model (BiLM) and a *frozen* pretrained visual encoder (in blue), complemented with additional lightweight trainable modules (in orange): (1) a visual-to-text projection module P (on the left), which maps the *frozen* visual features to the joint visual-text embedding space and (2) a set of small adapter modules A (on the right) in between the *frozen* transformer blocks. The pretrained normalization layers in the BiLM (on the right) are also finetuned.

to perform visual question answering. We aim at connecting these two components while keeping the language component *frozen* to avoid catastrophic forgetting [De Lange, 2021], where the large language model would specialize to a new task while forgetting its initial capabilities. The end-goal is to design a unified model having the best of both worlds: visual understanding capabilities of a powerful visual encoder and question answering capabilities of a powerful language model. This requires several technical innovations, which are described in the rest of this section. First, we explain in Section 4.3.1 how we augment a *frozen* pretrained bidirectional masked language model with new layers to enable joint video and language reasoning, see Figure 4.2. Second, we present in Section 4.3.2 how we train these layers on video-text data scraped from the Web [Bain, 2021]. Finally, we describe in Section 4.3.3 how we enable zero-shot predictions for several video-language downstream tasks, including open-ended VideoQA, by casting them in a *cloze* form, similar to the masked language modeling task solved during training.

4.3.1 Architecture

The proposed architecture, illustrated in Figure 4.2, brings together a powerful *frozen* pretrained bidirectional language model with a strong visual encoder. The difficulty lies in enabling multi-modal reasoning while keeping the large language model *frozen*. To address this challenge, we unify these two models via a visual-to-text projection module together with small adapter modules inserted within the frozen language model. Next, we describe in more detail the three main components of the architecture: (i) the *frozen* pretrained bidirectional language model, (ii) the pretrained video encoder and (iii) the lightweight modules that seamlessly connect the

two components.

Frozen Bidirectional Masked Language Model. Our method starts from a pretrained bidirectional language model based on a Transformer encoder [Vaswani, 2017]. The input text is decomposed into a sequence of tokens $x = \{x_i\}_1^L \in [1, V]^L$ by a tokenizer of a vocabulary size V . The language model, parameterized by θ , makes use of an embedding function g_θ which independently transforms each token into a D -dimensional continuous embedding $t = \{t_i\}_1^L := \{g_\theta(x_i)\}_1^L \in \mathbb{R}^{L \times D}$, a Transformer encoder f_θ which computes interactions between all input tokens and outputs contextualized representations $t' = \{t'_i\}_1^L$, and a masked language modeling (MLM) classifier head m_θ which independently maps the D -dimensional continuous embedding for each token t'_i to a vector of logits parameterizing a categorical distribution over the vocabulary V . This distribution is referred to by $\log p_\theta(x) := \{m_\theta(t'_i)\}_1^L \in \mathbb{R}^{L \times V}$. We assume that the language model is pretrained, i.e. θ has been optimised with a standard MLM objective [Devlin, 2019] on a large dataset of text from the Web. We show in Section 4.4.2 that this text-only pretraining has a crucial importance for zero-shot VideoQA.

Pretrained Video Encoder. The video is represented by a sequence of frames $y = \{y_i\}_1^T$. Each frame is forwarded separately through a visual backbone h_ϕ , which outputs one feature vector per frame $u = \{u_i\}_1^T := \{h_\phi(y_i)\}_1^T \in \mathbb{R}^{T \times D_u}$. In detail, the visual backbone is CLIP ViT-L/14 [Dosovitskiy, 2021; Radford, 2021] at resolution 224×224 pixels, pretrained to map images to text descriptions with a contrastive loss on 400M Web-scraped image-text pairs. The backbone is kept frozen throughout our experiments. Note that a CLIP-baseline for zero-shot VideoQA results in poor performance, see Section 4.4.4.

Connecting the Frozen Language and Frozen Vision components. The video features are incorporated into the language model as a prompt [Lester, 2021; Li, 2021c; Zhou, 2022] v of length T (Figure 4.2, left). This prompt is obtained by linearly mapping the visual features u to the text token embedding space via a visual-to-text projection $P \in \mathbb{R}^{D_u \times D}$, i.e. $v = \{v_i\}_1^T := \{P(u_i)\}_1^T$. The prompt is then concatenated with the text embeddings before being forwarded to the transformer encoder that models joint visual-linguistic interactions. We show in Section 4.4.2 that incorporating the input video considerably improves zero-shot VideoQA results. In addition, to learn powerful multi-modal interactions while keeping the transformer encoder weights *frozen*, we equip the transformer encoder with lightweight adapter modules A [Houlsby, 2019] (Figure 4.2, right). We use an adapter which transforms the hidden state z with a multi-layer perceptron transformation and a residual connection, i.e. $A(z) = z + W^{up}\psi(W^{down}z)$ with $W^{down} \in \mathbb{R}^{D \times D_h}$, $W^{up} \in \mathbb{R}^{D_h \times D}$, D the hidden dimension of the transformer, D_h the bottleneck dimension, and ψ a ReLU activation function. D_h is typically set to be smaller than D such that the adapters are lightweight. In detail, we add an adapter module before the layer normalization, after each self-attention layer and each feed-forward layer of the transformer encoder.

4.3.2 Cross-modal training

We wish to train the newly added modules introduced in the previous section (shown in orange in Figure 4.2) for the VideoQA task. This is hard because we assume that no explicit manual annotation for the VideoQA task is available, such annotations being expensive and therefore hard to obtain at scale. Instead we train our architecture using *only* readily-available video-caption pairs scraped from the Web. Such data is easy to obtain [Bain, 2021; Miech, 2019; Zellers, 2021], ensuring the scalability of our approach.

During training, we keep the weights of the pretrained BiLM and pretrained visual backbone *frozen* as previously explained. We train from scratch the parameters of (i) the visual-to-text projection module P and (ii) the adapter modules A . We show in Section 4.4.2 the importance of *freezing* the BiLM weights combined with training the adapter modules. Note that all normalization layers [Ba, 2016] of the pretrained BiLM are also updated to adjust to the new distribution of the training data. We denote all the trainable parameters of our model by the subscript μ . In practice, they sum up to about 5% of the BiLM parameters, hence the training of our model is computationally efficient.

We use a visually-conditioned masked language modeling objective (MLM), in which some text tokens $\{x_m\}$ are randomly masked and the model has to predict these tokens based on the surrounding text tokens and the video input. Formally, we minimize the following loss:

$$\mathcal{L}_\mu(x, y) = -\frac{1}{M} \sum_m \log p_\mu(\tilde{x}, y)_m^{x_m}, \quad (4.1)$$

where \tilde{x} is the corrupted text sequence, y is the sequence of video frames, $p_\mu(\tilde{x}, y)_m^{x_m}$ is the probability for the (masked) m -th token in \tilde{x} to be x_m , and M is the number of masks in the sequence \tilde{x} . In detail, we follow [Devlin, 2019] and corrupt 15% of text tokens, replacing them 80% of the time with a mask token, 10% of the time with the same token and 10% of the time with a randomly sampled token.

4.3.3 Adapting to downstream tasks

After training, our model is able to fill gaps in the input text given an input video together with left and right textual context as part of the input text. We wish to apply our model *out-of-the-box* to predict an answer given a question about a video. The video can optionally come with textual subtitles obtained using automatic speech recognition. To avoid using manual supervision, we formulate the downstream tasks in *cloze* form [Schick, 2021a; Taylor, 1953], i.e. such that the model only has to fill-in a mask token in the input prompt similarly to the MLM objective optimized during training. The adaptation to the downstream tasks brings several challenges, as described next. First, we describe how we formulate the input text prompts for several downstream tasks. Then, we explain how we map the mask token from the input text prompt to an answer via a *frozen* answer embedding module. Finally, we present how we finetune our architecture in a supervised setting.

Input prompt engineering. We describe how we design the input text prompts for several downstream video-language tasks. Each downstream task is formulated as a masked language modeling problem. This allows us to apply FrozenBiLM out-of-the-box. A [CLS] token and a [SEP] token are respectively inserted at the start and the end of each sequence following [Devlin, 2019].

Open-ended VideoQA. Given a question and a video, the task is to find the correct answer in a large vocabulary \mathcal{A} of about 1K answers. Answers are concise, i.e. the great majority of answers consist of one word [Jang, 2017; Xu, 2017; Yang, 2021b; Yu, 2019]. We design the following prompt:

```
"[CLS] Question: <Question>? Answer: [MASK]. Subtitles: <Subtitles> [SEP]"
```

Multiple-choice VideoQA. Given a question and a video, the task is to find the correct answer in a small number of candidates C , typically up to 5 choices [Lei, 2018a; Li, 2020b]. We set the vocabulary to $\mathcal{A} = [\text{Yes}, \text{No}]$ and compute a confidence score for each candidate by using the following prompt:

```
"[CLS] Question: <Question>? Is it '<Answer Candidate>'? [MASK]. Subtitles: <Subtitles> [SEP]"
```

We choose the best option by selecting the candidate with the highest *Yes* logit value.

Video-conditioned fill-in-the-blank task. Given a video and a sentence with a blank space, the task is to fill in the blank with the correct word from a vocabulary \mathcal{A} of about 1K answers. We replace the blank in the sentence with a mask token, and design the following prompt:

```
"[CLS] <Sentence with a [MASK] token>. Subtitles: <Subtitles> [SEP]"
```

Note that all prompts are prepended with the video prompt (see Section 4.3.1) before being forwarded to the transformer encoder.

Answer embedding module. For each downstream task, we wish to map the mask token in the input text prompt to an actual answer prediction in the set of possible answers \mathcal{A} , as described above. For this we use the *frozen* MLM classifier head m_θ . However, $m_\theta \in \mathbb{R}^{V \times D}$ covers V different tokens where $V \gg N$ and $N \approx 1,000$ is the size of \mathcal{A} . Therefore, we introduce a task-specific answer classification head l which linearly maps a contextualized mask representation t'_i to a vector of logits parameterizing a categorical distribution over the vocabulary \mathcal{A} , i.e. $l \in \mathbb{R}^{N \times D}$. We set the weights of this answer module l with the corresponding weights of the pretrained MLM classifier m_θ for one-token answers. In the case of multi-token answers, we average the weights of their different tokens. We, hence, enable zero-shot inference at test time. We also discuss other alternative strategies to handle multi-token answers in Appendix Section 4.4.2.

Fully-supervised training. To evaluate our approach on fully-supervised benchmarks, we also explore finetuning of our model on datasets that provide manual annotations for the target task. To this end, we train the same parameters as explained in Section 4.3.2, while keeping the transformer weights and the answer embedding module *frozen*. For open-ended VideoQA and video-conditioned fill-in-the-blank, we use a cross-entropy loss on the task-specific vocabulary

A. For multiple-choice VideoQA, we use a binary cross-entropy loss applied to each answer candidate. We show in Section 4.4.5 the benefit of *freezing* the language model weights during fully-supervised training.

4.4 Experiments

This section demonstrates the benefits of our FrozenBiLM framework and compares our method to the state of the art. We first outline our experimental setup in Section 4.4.1. We then present ablation studies in Section 4.4.2. Next we compare our bidirectional framework to its autoregressive variant in Section 4.4.3. The comparison to the state of the art in zero-shot VideoQA and qualitative results are presented in Section 4.4.4. We then finetune our model on the VideoQA task in Section 4.4.5, where we show few-shot and fully-supervised results. Finally, we provide an analysis of the *frozen* self-attention patterns in FrozenBiLM in Section 4.4.6.

4.4.1 Experimental setup

Frozen bidirectional language model. We use a tokenizer based on SentencePiece [Kudo, 2018] with $V = 128,000$, and a bidirectional language model with 900M parameters, DeBERTa-V2-XLarge [He, 2021b], trained with the MLM objective on a corpus of 160G text data. We also show how our approach generalizes to other MLM-pretrained bidirectional language models such as BERT [Devlin, 2019] in Section 4.4.2.

Datasets. For training we use the publicly available **WebVid10M** dataset [Bain, 2021], which consists of 10 million of video-text pairs scraped from the Shutterstock website where video captions are obtained from readily-available alt-text descriptions. We evaluate results on eight downstream datasets covering a wide range of textual and video domains (e.g. GIFs, YouTube videos, TV shows, movies), and multiple VideoQA paradigms: open-ended VideoQA (**iVQA** [Yang, 2021b], **MSRVTT-QA** [Xu, 2017], **MSVD-QA** [Xu, 2017], **ActivityNet-QA** [Yu, 2019] and **TGIF-QA** FrameQA [Jang, 2017]), multiple-choice VideoQA (**How2QA** [Li, 2020b] and **TVQA** [Lei, 2018a]) and video-conditioned fill-in-the-blank (**LSMDC-Fill-in-the-blank** [Maharaj, 2017]). Unless stated otherwise, we report top-1 test accuracy using the original splits for training, validation and test. Below we describe the downstream datasets in more detail.

LSMDC-FiB [Maharaj, 2017] is an open-ended video-conditioned fill-in-the-blank task which consists in predicting masked words in sentences that describe short movie clips [Rohrbach, 2015; Rohrbach, 2017]. It contains 119K video clips and 349K sentences, split into 297K/22K/30K for training/validation/testing.

iVQA [Yang, 2021b] is a recently introduced open-ended VideoQA dataset, focused on objects, scenes and people in instructional videos [Miech, 2019]. It excludes non-visual questions, and contains 5 possible correct answers for each question for a detailed evaluation. It contains 10K video clips and 10K questions, split into 6K/2K/2K for training/validation/testing.

MSRVTT-QA [Xu, 2017], **MSVD-QA** [Xu, 2017] and **TGIF-FrameQA** [Jang, 2017] are popular open-ended VideoQA benchmarks automatically generated from video descriptions [Chen,

2011; Li, 2016; Xu, 2016b]. Questions are of five types for MSRVTT-QA and MSVD-QA: what, who, how, when and where; and four types for TGIF-QA: object, number, color and location. MSRVTT-QA contains 10K video clips and 243K question-answer pairs, split into 158K/12K/73K for training/validation/testing. MSVD-QA contains 1.8K video clips and 51K question-answer pairs, split into 32K/6K/13K for training/validation/testing. TGIF-QA contains 46K GIFs and 53K question-answer pairs, split into 39K/13K for training/testing.

ActivityNet-QA [Yu, 2019] is an open-ended VideoQA dataset consisting of long videos [Caba Heilbron, 2015] (3 minutes long on average), and covering 9 question types (motion, spatial, temporal, yes-no, color, object, location, number and other). It contains 5.8K videos and 58K question-answer pairs, split into 32K/18K/8K for training/validation/testing.

How2QA [Li, 2020b] is a multiple-choice VideoQA dataset focused on instructional videos [Miech, 2019]. Each question is associated with one correct and three incorrect answers. It contains 28K video clips and 38K questions, split into 35K/3K for training/validation. We report results on the public validation set for comparison with prior work [Seo, 2021b; Yang, 2021b; Yu, 2021].

TVQA [Lei, 2018a] is a multiple-choice VideoQA dataset focused on popular TV shows. Each question is associated with one correct and four incorrect answers. It contains 22K video clips and 153K questions, split into 122K/15K/15K for training/validation/testing. The test set is hidden and only accessible a limited number of times via an online leaderboard. We report results on the validation set for the ablation studies and on the hidden test set for the comparison to the state of the art.

Implementation Details. As for the architecture hyperparameters, we truncate text sequences up to $L = 256$ tokens. Video features are extracted by sampling $T = 10$ frames, each resized at 224×224 pixels, from the video. These frames are sampled at temporally equal distance, with a minimum distance of 1 second. For videos shorter than T seconds, we pad the video prompt up to T tokens. The dimension of the visual features from ViT-L/14 [Dosovitskiy, 2021] is $D_f = 768$. The transformer encoder from DeBERTa-V2-XLarge [He, 2021b] has 24 layers, 24 attention heads, a hidden dimension of $D = 1536$ and an intermediate dimension in the feed-forward layers of 6144. For the adapters [Houlsby, 2019], we use a bottleneck dimension of $D_h = \frac{D}{8} = 192$.

For training, we use the Adam optimizer [Kingma, 2015] with $\beta = (0.9, 0.95)$ and no weight decay. We use Dropout [Srivastava, 2014] with probability 0.1 in the adapters and in the transformer encoder. When finetuning the language model weights, we divide the batch size by a factor 2 so to accommodate with the GPU memory constraints.

For cross-modal training on WebVid10M, we use a total batch size of 128 video-caption pairs split in 8 NVIDIA Tesla V100 GPUs. The training for 2 epochs on WebVid10M lasts 20 hours on 8 Tesla V100 GPUs. We use a fixed learning rate of $3e^{-5}$ for the variant with adapters. We find that the variant without adapters that freezes the language model weights prefers a higher learning rate of $3e^{-4}$, and that the variant *UnfrozenBiLM* that finetunes the language model weights prefers a lower one of $1e^{-5}$.

To finetune our model on downstream datasets, we use a total batch size of 32 video-question-

	LM	<i>Frozen</i>	Adapters	Fill-in-the-blank		Open-ended				Multiple-choice	
	Pretraining	LM		LSMDC	iVQA	MSRVTT QA	MSVD QA	ActivityNet QA	TGIF QA	How2QA	TVQA
1.	✗	✗	✗	0.5	0.3	0.1	0.0	0.5	0.0	32.4	20.7
2.	✓	✗	✗	37.1	21.0	17.6	31.9	20.7	30.7	45.7	45.6
3.	✓	✓	✗	50.7	27.3	16.8	32.2	24.7	41.0	53.5	53.4
4.	✓	✓	✓	51.5	26.8	16.7	33.8	25.9	41.9	58.4	59.2

Table 4.1: The effect of initializing and training various parts of our model evaluated on zero-shot VideoQA. All models are trained on WebVid10M and use multi-modal inputs (video, speech and question) at inference.

answer triplets (respectively 32 video-sentence pairs) split in 4 NVIDIA Tesla V100 GPUs for open-ended VideoQA datasets (respectively video-conditioned fill-in-the-blank datasets) and 16 video-question pairs split in 8 NVIDIA Tesla V100 GPUs for multiple-choice VideoQA datasets. We train for 20 epochs for all downstream datasets except LSMDC-FiB for which we find that training for 5 epochs leads to similar validation results. We warm up the learning rate linearly for the first 10% of iterations, followed by a linear decay of the learning rate (down to 0) for the remaining 90%. On each dataset, we run a random search and select the learning rate based on the best validation results. We search over 10 learning rates in the range $[1e^{-5}, 1e^{-4}]$ for variants that freeze the language model weights, and $[5e^{-6}, 5e^{-5}]$ for the variant *UnfrozenBiLM* that finetunes the language model weights.

In the zero-shot open-ended VideoQA setting, we use an answer vocabulary composed of the top 1,000 answers in the corresponding training dataset, following [Zellers, 2021]. In the fully-supervised setting, we experiment both with the vocabulary composed of the top 1,000 answers and the vocabulary composed of all answers appearing at least twice in the corresponding training dataset and choose the one leading to best validation results. Following [Zellers, 2021], questions with out-of-vocabulary answer are not used for finetuning, and are automatically considered as incorrect during evaluation.

4.4.2 Ablation studies

In this section, we evaluate the zero-shot performance of different variants of our method. By default, we use the *frozen* pretrained DeBERTa-V2-XLarge language model and train the visual-to-text-projection layer together with adapters for 2 epochs on WebVid10M. We refer to this default model as *FrozenBiLM*. This model uses three input modalities in terms of video, question, and speech, the prompts and the inference strategy described in Section 4.3.3, $T = 10$ video frames, $D_h = 192$ hidden dimension in the adapters, and the ViT-L/14 CLIP visual backbone.

Ablation of the model training. We ablate the effect of initializing parameters of the language model, freezing its weights and training adapters in Table 4.1. We observe that the language model pretraining is crucial. Indeed, a model with randomly initialized language weights (row 1) performs poorly compared to models initialized with language pretrained weights

	Visual	Speech	Fill-in-the-blank	Open-ended					Multiple-choice	
			LSMDC	iVQA	MSRVTT QA	MSVD QA	ActivityNet QA	TGIF QA	How2QA	TVQA
1.	✗	✗	47.9	11.0	6.4	11.3	22.6	32.3	29.6	23.2
2.	✗	✓	49.8	13.2	6.5	11.7	23.1	32.3	45.9	44.1
3.	✓	✗	50.9	26.2	16.9	33.7	25.9	41.9	41.9	29.7
4.	✓	✓	51.5	26.8	16.7	33.8	25.9	41.9	58.4	59.2

Table 4.2: Impact of the visual and speech modalities on zero-shot VideoQA. Rows 1 and 2 report results for a pretrained language model without any visual input. Rows 3 and 4 give results for a FrozenBiLM model pretrained on WebVid10M.

(rows 2 to 4). Moreover, the model which updates the language model weights (row 2) during cross-modal training performs considerably worse compared to variants that *freeze* them (rows 3 and 4). This shows the benefit of *freezing* the language model for zero-shot VideoQA. We also notice the benefit of the adapter layers by comparing rows 3 and 4, especially for multiple-choice datasets. Finally, we note that training variants with the *frozen* language model is twice faster compared to updating all parameters, as there is a significantly lower number of parameters to be trained.

Impact of modalities. Table 4.2 shows the impact of the visual and speech modalities on the zero-shot performance of our model. First, we evaluate the text-only performance of our model using neither visual input nor speech input in row 1. We can observe that adding speech (row 2) marginally improves the results and that the importance of speech highly depends on the dataset. When adding vision (rows 3 and 4), the performance increases significantly, e.g. +13.6% accuracy on iVQA and +22.1% on MSVD-QA between rows 4 and 2. Finally, the model with vision also benefits from the speech, e.g. +16.5% accuracy on How2QA and +29.5% accuracy on TVQA (compare rows 3 and 4).

Note that in practice, speech is missing for many videos, as we obtain the speech directly from the YouTube API and many videos are no longer available. Exceptions are How2QA and TVQA for which the authors [Lei, 2018a; Li, 2021b] provide speech for all videos. Consequently, we have speech data for only 44.3%, 14.2%, 8.2%, 7.1% and 25.3% of test samples in LSMDC-FiB, iVQA, MSRVTT-QA, MSVD-QA and ActivityNet-QA respectively. GIFs in TGIF-QA do not contain speech.

Size of the cross-modal training dataset.

Zero-shot results of *FrozenBiLM* after training for a fixed number of iterations on different fractions of WebVid10M are shown in Table 4.3. We construct these subsets such that larger subsets include the smaller ones. We find that performance increases monotonically with more multi-modal training data.

	Training Data	MSVD-QA	How2QA
1.	WebVid1K	13.6	53.0
2.	WebVid10K	22.7	54.9
3.	WebVid200K	27.8	56.0
4.	WebVid2M	30.1	57.4
5.	WebVid10M	33.8	58.4

Table 4.3: Dependency on the size of the training set. Zero-shot results are presented for different fractions of the WebVid10M dataset used for training.

Template	iVQA	MSRVTT QA	MSVD QA	ActivityNet QA	TGIF QA
1. "[CLS] Question: <Question>? Answer: [MASK]. Subtitles: <Subtitles> [SEP]"	26.8	16.7	33.8	25.9	41.9
2. "[CLS] Q: <Question>? A: [MASK]. S: <Subtitles> [SEP]"	27.4	16.2	32.5	25.5	41.9
3. "[CLS] <Question>? [MASK]. <Subtitles> [SEP]"	23.1	13.6	28.0	21.6	25.2

Table 4.4: Impact of the prompt on the zero-shot open-ended VideoQA performance.

Template	How2QA	TVQA
1. "[CLS] Question: <Question>? Is it "<Answer Candidate>"? [MASK]. Subtitles: <Subtitles> [SEP]"	58.4	59.7
2. "[CLS] Q: <Question>? Is it "<Answer Candidate>"? [MASK]. S: <Subtitles> [SEP]"	57.7	58.2
3. "[CLS] <Question>? <Answer Candidate>? [MASK]. <Subtitles> [SEP]"	47.6	55.0

Table 4.5: Impact of the prompt on the zero-shot multiple-choice VideoQA performance.

Size of the language model. In Table 4.9, we ablate the importance of the language model size for the zero-shot performance. Note that when comparing different language models, we use no adapters to avoid biases related to the choice of the bottleneck dimension hyperparameter [Houlsby, 2019]. We find that using the 900M-parameter DeBERTA-V2-XLarge (row 6) outperforms the 300M-parameter BERT-Large (row 5) which also improves over the 100M-parameter BERT-Base (row 4).

Prompt design. Our text input prompts include a suffix just to the right of the mask token which consists in a point and an end-of-sentence token for the variant without speech (or a point followed by the speech subtitles for the variant with speech). We found that removing this suffix leads to a considerable drop of performance (e.g. the test accuracy on MSVD-QA in the row 3 of Table 4.2 drops from 33.7% to 2.8%). Note that we do not observe such a large drop in performance when removing the [CLS] token, e.g. the accuracy on MSVD-QA drops only from 33.8% to 33.2%. This shows that the bidirectional nature of our framework is a key factor for the performance. Intuitively, this suffix forces the model to provide a concise answer. Such a hard constraint cannot be given to unidirectional autoregressive models compared next in Section 4.4.3.

We also ablate the importance of the prompt design on the zero-shot VideoQA performance. We report results with alternative prompts in Tables 4.4 and 4.5. We find that replacing the words “Question”, “Answer” and “Subtitles” by “Q”, “A” and “S”, respectively, in the templates described in Section 4.3.3 does not impact the zero-shot VideoQA accuracy (compare rows 2 and 1 in Tables 4.4 and 4.5). However, completely removing “Question”, “Answer”, “Subtitles” and “is it” in the templates results in a significant drop of performance (compare rows 3 and 1 in Tables 4.4 and 4.5). We conclude that it is important to have tokens that link the different textual inputs.

Method	Fill-in-the-blank	Open-ended					Multiple-choice	
	LSMDC	iVQA	MSRVTT QA	MSVD QA	ActivityNet QA	TGIF QA	How2QA	TVQA
<i>FrozenBiLM</i> (Ours)	51.5±0.1	28.3±0.9	14.4±1.4	30.0±2.2	25.4±0.7	39.7±2.1	57.9±0.6	57.9±1.2

Table 4.6: Impact of the random seed for zero-shot VideoQA, reporting mean and standard deviation over 5 cross-modal training runs with different random seeds.

Inference Strategy	Fill-in-the-blank	Open-ended				
	LSMDC	iVQA	MSRVTT QA	MSVD QA	ActivityNet QA	TGIF QA
1. Average token embeddings	51.5	26.8	16.7	33.8	25.9	41.9
2. Multiple mask tokens	51.0	27.0	17.1	34.4	26.1	42.0

Table 4.7: Impact of the inference strategy on the zero-shot open-ended VideoQA performance.

Impact of the random seed. To verify the robustness of our approach with respect to the random seed, we run cross-modal training for FrozenBiLM with 5 different random seeds. We report the mean and standard deviation of zero-shot accuracy in Table 4.6. We observe that the random seed does not affect the comparison to prior work done in Section 4.4.4, which is reported with a single run for fair comparison with prior work.

Multi-token inference strategy. For multi-token answers in the open-ended VideoQA setting, our FrozenBiLM simply averages the weights of different answer tokens. However, such simple scheme does not preserve the semantic structure of the answer. Hence we here investigate and compare another possible inference strategy in the zero-shot setting and discuss potential sources of improvement. We take inspiration from [Jiang, 2020c] and performs zero-shot VideoQA inference by using multiple mask tokens decoded in parallel. Then, for each video-question pair, we do one forward pass through the model per possible number of mask tokens (typically, 1 to 5) in order to score all possible answers in vocabulary \mathcal{A} . The score of a given answer is then obtained by multiplying the probability of its individual tokens, possibly normalized by its number of tokens. As shown in Table 4.7, we observe that such a decoding strategy (row 2) does not significantly improve the accuracy of our model over the one used in FrozenBiLM (row 1). We hypothesize that this is due to the fact that the current open-ended VideoQA datasets [Jang, 2017; Xu, 2017; Yang, 2021b; Yu, 2019] contain a great majority of short answers, e.g. 99% of the answers in the MSRVTT-QA test set are one-token long with our tokenizer [Kudo, 2018]. Additionally, a possible solution to further improve the decoding in this alternative scheme is to increase the length of the masked spans at pretraining, as in [Joshi, 2020]. [Salazar, 2020] provides another potential solution to score multi-token answers in our framework, by masking tokens one by one and computing pseudo-likelihood scores.

Number of frames, adapters hidden dimension, and size and pretraining of the visual backbone. In Table 4.8, we analyze the impact of the number of frames T used by the model,

T	D_h	Visual Backbone	Fill-in-the-blank	Open-ended					Multiple-choice		
			LSMDC	iVQA	MSRVTT QA	MSVD QA	ActivityNet QA	TGIF QA	How2QA	TVQA	
1.	1	192	ViT-L/14 (CLIP)	50.4	24.8	12.4	28.3	24.9	41.5	54.3	54.6
2.	10	96	ViT-L/14 (CLIP)	52.4	28.6	13.7	29.0	25.1	42.3	59.3	58.0
3.	10	384	ViT-L/14 (CLIP)	51.4	27.5	15.6	31.2	23.9	41.8	58.0	57.8
4.	10	192	ViT-B/16 (ImageNet)	49.4	23.8	13.3	25.7	25.1	36.8	56.5	57.2
5.	10	192	ViT-B/16 (CLIP)	50.8	25.5	14.6	30.3	25.6	41.0	57.6	58.2
6.	10	192	ViT-L/14 (CLIP)	51.5	26.8	16.7	33.8	25.9	41.9	58.4	59.2

Table 4.8: Impact of the number of frames T used by the model, the hidden dimension D_h in the adapters and the visual backbone on the zero-shot VideoQA results. All models are trained on WebVid10M and use multi-modal inputs (video, speech and question) at inference.

Method	Language Model	# LM params	Train time (GPUH)	MSRVTT	MSVD	ActivityNet	TGIF	iVQA	
				QA	QA	QA	QA	QA	QA
Autoregressive	1. GPT-Neo-1.3B	1.3B	200	6.6	4.2	10.1	17.8	14.4	14.4
	2. GPT-Neo-2.7B	2.7B	360	9.1	7.7	17.8	17.4	20.1	20.1
	3. GPT-J-6B	6B	820	21.4	9.6	26.7	24.5	37.3	37.3
Bidirectional	4. BERT-Base	110M	24	12.4	6.4	11.7	16.7	23.1	23.1
	5. BERT-Large	340M	60	12.9	7.1	13.0	19.0	21.5	21.5
	6. DeBERTa-V2-XLarge	890M	160	27.3	16.8	32.2	24.7	41.0	41.0

Table 4.9: Comparison of autoregressive language models (top) and bidirectional language models (bottom) for zero-shot VideoQA. All variants are trained on WebVid10M for the same number of epochs.

the hidden dimension in the adapters D_h and the size and pretraining of the visual backbone. We first observe that using 10 frames significantly improves over using a single frame (compare rows 1 and 5). Next we note that using a hidden dimension of 96 or 384 in the adapters instead of 192 does not change the results significantly (see rows 2, 3 and 6). Moreover, we find that scaling up the size of the visual backbone is beneficial, as using ViT-L/14 instead of ViT-B/16, both being trained on CLIP [Radford, 2021], slightly improves the results (compare rows 4 and 6). Furthermore, we observe that the pretraining of the visual backbone is crucial, as using ViT-B/16 pretrained on 400M image-text pairs from CLIP significantly improves over using ViT-B/16 pretrained on ImageNet-21K, i.e. 22M image-label pairs (compare rows 4 and 5).

4.4.3 Comparison with frozen autoregressive models

In this section, we compare our bidirectional framework using language models of various sizes to the larger, autoregressive GPT-based counterparts recently used for zero-shot image question answering [Tsimpoukelli, 2021; Yang, 2021f]. For fair comparison, we adapt autoregressive models to video and language inputs similarly as our bidirectional models. In detail, autoregressive variants train a similar visual-to-text projection by using a left-to-right language modeling loss [Tsimpoukelli, 2021]. All models in our comparison are trained on WebVid10M for the same number of epochs. At inference, autoregressive variants use the same template as [Tsimpoukelli,

Method	Training Data	T	Fill-in-the-blank	Open-ended					Multiple-choice	
			LSMDC	iVQA	MSRVTT QA	MSVD QA	ActivityNet QA	TGIF QA	How2QA	TVQA
Random	—	—	0.1	0.1	0.1	0.1	0.1	0.1	25	20
CLIP ViT-L/14 [Radford, 2021]	400M image-texts	✗	1.2	9.2	2.1	7.2	1.2	<u>3.6</u>	47.7	26.1
Just Ask [Yang, 2022c]	HowToVQA69M + WebVidVQA3M	✗	—	13.3	5.6	13.5	<u>12.3</u>	—	<u>53.1</u>	—
Reserve [Zellers, 2022]	YT-Temporal-1B	✗	31.0	—	5.8	—	—	—	—	—
<i>FrozenBiLM</i> (Ours)	WebVid10M	✗	<u>50.9</u>	<u>26.2</u>	16.9	<u>33.7</u>	25.9	41.9	41.9	<u>29.7</u>
<i>FrozenBiLM</i> (Ours)	WebVid10M	✓	51.5	26.8	<u>16.7</u>	33.8	25.9	41.9	58.4	59.7

Table 4.10: Comparison with the state of the art for zero-shot VideoQA. T denotes Transcribed Speech.

Method	Motion	Spatial	Temporal	Yes-No	Color	Object	Location	Number	Other
Just Ask [Yang, 2021b]	2.3	1.1	0.3	36.3	11.3	4.1	6.5	0.2	4.7
FrozenBiLM (Ours)	12.7	6.8	1.6	53.2	16.5	17.9	18.1	26.2	25.8

Table 4.11: Zero-shot VideoQA results segmented per question type on the ActivityNet-QA dataset, compared with Just Ask [Yang, 2021b].












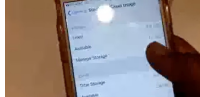



Method	MSRVTT-QA						MSVD-QA					
	What	Who	Number	Color	When	Where	What	Who	Number	Color	When	Where
Just Ask [Yang, 2021b]	1.8	0.7	66.3	0.6	0.6	4.5	7.8	1.7	74.3	18.8	3.5	0.0
FrozenBiLM (Ours)	10.7	28.7	55.0	11.4	9.2	9.3	26.0	45.0	69.9	56.3	5.2	17.9

Table 4.12: Zero-shot VideoQA results segmented per question type on the MSRVTT-QA dataset (left) and the MSVD-QA dataset (right), compared with Just Ask [Yang, 2021b].






2021] to which we prepend speech subtitles, greedily decode sequences as [Tsimpoukelli, 2021], and use the same answer vocabulary as bidirectional models. Autoregressive variants select the top answer that maximizes the log-likelihood when appended to the question prompt. Here also, we use no adapters for all models, such that the architecture of autoregressive models closely follows [Tsimpoukelli, 2021]. This is to avoid biases related to the tuning of the bottleneck reduction hyperparameter in the adapters [Houlsby, 2019].

We compare autoregressive and bidirectional language models in terms of accuracy and efficiency in Table 4.9. We observe that our bidirectional framework (rows 4-6) achieves significantly better zero-shot performance-efficiency trade-off compared to its autoregressive counterpart (rows 1-3). For instance, our framework with BERT-Base [Devlin, 2019] (row 4) outperforms the autoregressive variant based on GPT-Neo-1.3B [Black, 2021] (row 1) which uses 12 times more parameters and 8 times more training time. Likewise, our framework with DeBERTa-V2-XLarge [He, 2021b] (row 6) improves over the autoregressive variant based on GPT-J-6B [Wang, 2021a] (row 3) that has 7 times more parameters and requires 5 times more training time, showing the efficiency of our *bidirectional* framework for zero-shot VideoQA.

(a) Zero-Shot open-ended VideoQA on the iVQA and ActivityNet-QA datasets.

 Question: What is the man holding at the start of the video? GT Answer: guitar, electric guitar Just Ask: typewriter UnFrozenBiLM: beer FrozenBiLM (text-only): scissors FrozenBiLM (ours): guitar	 Question: What item hanging on the wall features a tree? GT Answer: quilt Just Ask: christmas sock UnFrozenBiLM: fabric FrozenBiLM (text-only): tree FrozenBiLM (ours): quilt	 Question: What is the sitting man doing? GT Answer: knit sweater Just Ask: tie cow UnFrozenBiLM: swimming FrozenBiLM (text-only): eating FrozenBiLM (ours): knit sweater	 Question: Where is the woman sitting on? GT Answer: camel Just Ask: horse yard UnFrozenBiLM: desert FrozenBiLM (text-only): chair FrozenBiLM (ours): camel	 Question: What is the color of the cabinet door in the video? GT Answer: red Just Ask: dresser UnFrozenBiLM: blue FrozenBiLM (text-only): black FrozenBiLM (ours): red
 Question: What are the men standing in front of? GT Answer: fireplace Just Ask: cabinets UnFrozenBiLM: kitchen FrozenBiLM (text-only): building FrozenBiLM (ours): fireplace	 Question: Which category of sports does this sport belong to? GT Answer: surfing Just Ask: second UnFrozenBiLM: swimming FrozenBiLM (text-only): 1 FrozenBiLM (ours): surfing	 Question: Is there green grass on the roof? GT Answer: yes Just Ask: no UnFrozenBiLM: no FrozenBiLM (text-only): no FrozenBiLM (ours): yes	 Question: How many people are there in the video? GT Answer: 1 Just Ask: 2 UnFrozenBiLM: 4 FrozenBiLM (text-only): 2 FrozenBiLM (ours): 1	
 Question: What did the man with the backpack walk into? GT Answer: bakery, bake shop Just Ask: stores UnFrozenBiLM: wall FrozenBiLM (text-only): water FrozenBiLM (ours): restaurant	 Question: What is the person changing on the phone? GT Answer: settings Just Ask: colors UnFrozenBiLM: camera FrozenBiLM (text-only): phone FrozenBiLM (ours): wallpaper	 Question: What is the silver object behind the woman on counter? GT Answer: toaster Just Ask: mirror UnFrozenBiLM: salt FrozenBiLM (text-only): coin FrozenBiLM (ours): spoon	 Question: What organism is shown at the end of the video? GT Answer: bird Just Ask: worms UnFrozenBiLM: beef FrozenBiLM (text-only): octopus FrozenBiLM (ours): chicken	

(b) Zero-shot video-conditioned fill-in-the-blank on the LSMDC FiB dataset.

 Sentence: Someone ____ him to the truck and across the street. GT Answer: chases UnFrozenBiLM: follow FrozenBiLM (text-only): drags FrozenBiLM (ours): chases	 Sentence: Each singer in the front row ____ a huge toad. GT Answer: holds UnFrozenBiLM: plays FrozenBiLM (text-only): wears FrozenBiLM (ours): holds	 Sentence: He hurries up the ____ walkway to his house and enters. GT Answer: front UnFrozenBiLM: screen FrozenBiLM (text-only): wooden FrozenBiLM (ours): front	 Sentence: A woman wraps food in newspapers and brings it over to their ____. GT Answer: table UnFrozenBiLM: man FrozenBiLM (text-only): home FrozenBiLM (ours): table	
---	--	---	--	---

(c) Zero-shot multiple-choice VideoQA on the How2QA and TVQA datasets.






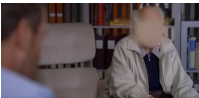



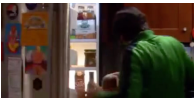
 Question: Why did the speaker opened a folder on his computer? A0: to show pictures of digital numbers A1: to show photographs he has taken A2: to show desktop wallpapers A3: to show programs he downloaded GT Answer: A0 UnFrozenBiLM: A2 FrozenBiLM (text-only): A1 FrozenBiLM (ours): A0	 Question: Where is the person in the clip most likely located? A0: home A1: corporate office A2: sports stadium A3: emergency room GT Answer: A0 UnFrozenBiLM: A3 FrozenBiLM (text-only): A2 FrozenBiLM (ours): A0	 Question: What is the man doing to the branches? A0: He is burning them. A1: He is burying them. A2: He is throwing them in water. A3: He's painting them. GT Answer: A0 UnFrozenBiLM: A3 FrozenBiLM (text-only): A2 FrozenBiLM (ours): A0	 Question: When did the chef flipped over the layer of rice and seaweed? A0: after she sprinkled sesame A1: after she added cucumber A2: after she added fish A3: after she cut the cucumbers GT Answer: A0 UnFrozenBiLM: A3 FrozenBiLM (text-only): A1 FrozenBiLM (ours): A0	
 Question: Where is the man with glasses after Dr Lisa Cuddy leaves the room? A0: Leaning against the bookcase A1: Sitting on a white chair A2: Standing behind Dr House A3: Laying on the floor next to the desk A4: Sitting in a wheel chair GT Answer: A1 UnFrozenBiLM: A0 FrozenBiLM (text-only): A3 FrozenBiLM (ours): A1	 Question: What adjustment does Beckett do before going to talk with Mr caraway? A0: She puts on lipstick A1: She puts on glasses A2: She ties back her hair A3: She changes into a skirt A4: She zips up her jacket GT Answer: A4 UnFrozenBiLM: A2 FrozenBiLM (text-only): A2 FrozenBiLM (ours): A4	 Question: What color was the bowl beside the stove when Robin was making crepes? A0: Orange A1: Red A2: White A3: Blue A4: Green GT Answer: A4 UnFrozenBiLM: A0 FrozenBiLM (text-only): A3 FrozenBiLM (ours): A4	 Question: What did Raj do after he discovered the wine bottle was empty? A0: Raj laughed out loud A1: Raj called Howard on the phone A2: Raj put the bottle down and got cake to eat from the refrigerator A3: Raj ran in a circle A4: Raj went to the bathroom GT Answer: A2 UnFrozenBiLM: A1 FrozenBiLM (text-only): A3 FrozenBiLM (ours): A2	

Figure 4.3: Zero-Shot VideoQA results. We show more examples at [Yang, 2022a].

Method	Pretraining Data	Finetuning Data	iVQA	MSRVTT	MSVD	ActivityNet	TGIF
				QA	QA	QA	QA
BLIP [Li, 2022c]	129M image-text pairs	VQA	—	19.2	35.2	—	—
FrozenBiLM (no image-VQA training)	WebVid10M	\emptyset	26.8	16.7	33.8	25.9	41.9
FrozenBiLM (no cross-modal training)	\emptyset	VQA	14.6	6.9	12.6	22.6	33.3
FrozenBiLM (Ours)	WebVid10M	VQA	34.6	22.2	39.0	33.1	43.4

Table 4.13: Results of our model after cross-modal training, finetuning on the open-ended image-VQA dataset [Antol, 2015] and directly evaluating on open-ended VideoQA without using any VideoQA supervision, as in BLIP [Li, 2022c].

4.4.4 Comparison to the state of the art for zero-shot VideoQA

Zero-shot VideoQA. Table 4.10 presents results of our method in comparison to the state of the art in *zero-shot* VideoQA settings [Yang, 2021b], i.e. when using no manually annotated visual data for training. Our approach outperforms previous methods by a significant margin on all 8 datasets. In particular, FrozenBiLM outperforms Reserve [Zellers, 2022], which is trained on one billion YouTube video clips jointly with vision, language and sound, Just Ask [Yang, 2022c], which uses large-scale automatically generated VideoQA data, and a CLIP baseline [Radford, 2021] matching the text concatenating question and answer to the middle frame of the video. Note that FrozenBiLM performs competitively even when using no speech input. We also provide results segmented per question type for ActivityNet-QA in Table 4.11, and for MSRVTT-QA and MSVD-QA in Table 4.12. Compared to Just Ask [Yang, 2021b], we observe large and consistent improvements over all question categories, except for the *number* category on MSRVTT-QA and MSVD-QA. These results show that our approach is efficient in the diverse question categories of zero-shot VideoQA. In summary, our evaluation shows the excellent performance of our model in the challenging zero-shot setup.

Comparison with BLIP. In addition to the previously described zero-shot results, we here investigate a different but related *zero-shot* setting defined in BLIP [Li, 2022c], where a network trained on manually annotated image-VQA annotations is evaluated directly on open-ended VideoQA datasets. In detail, BLIP uses the open-ended image-VQA dataset [Antol, 2015] for finetuning after pretraining on 129M image-text pairs, including MS COCO [Chen, 2015] and Visual Genome [Krishna, 2016] which are manually annotated. To adapt our model to this setting, we finetune our model FrozenBiLM pretrained on WebVid10M on the image-VQA dataset using the same procedure as for finetuning on VideoQA datasets (see Section 4.3.3), i.e. notably with a *frozen* language model. In particular, we finetune on VQA for 10 epochs with an initial learning rate of $1e^{-5}$ which is warmed up for the first 10% iterations, and linearly decayed to 0 for the remaining 90% iterations. Table 4.13 shows that the resulting model not only improves over our model without image-VQA finetuning (i.e. in zero-shot mode as defined in Section 4.1) or our model trained on VQA only (i.e. without cross-modal training), but also substantially outperforms BLIP on both MSRVTT-QA and MSVD-QA. These results further demonstrate the strong capabilities of FrozenBiLM in settings where no VideoQA annotation is

available.

Zero-shot image-VQA. We next evaluate our pretrained model on the VQAv2 [Antol, 2015] validation set in the zero-shot setting, i.e. without any supervision of visual questions and answers. Frozen [Tsimpoukelli, 2021] achieves 29.5% accuracy in this setting using an autoregressive language model. In comparison, our FrozenBiLM model is 7 times smaller than Frozen and achieves 45.0% accuracy. We conclude that our model can perform competitively on the image-VQA tasks despite being tailored for videos.

Qualitative zero-shot VideoQA results. Figure 4.3 illustrates qualitative results of zero-shot VideoQA for our FrozenBiLM model and compares them to Just Ask [Yang, 2022c], as well as to variants of our approach that do not *freeze* the language model (UnFrozenBiLM) and use no visual modality (text-only), as evaluated in Section 4.4.2. We observe that the *unfrozen* variant can predict answers that lack text-only commonsense reasoning, e.g. in the third example of Figure 4.3a, it is unlikely that a sitting man is swimming; in the first example of Figure 4.3b, the word *follow* is grammatically incorrect; in the second example of Figure 4.3b, it is unlikely that a singer *plays* a toad. The text-only variant does have strong language understanding, but makes visually-unrelated predictions. In contrast, consistently with our quantitative results, our model FrozenBiLM is able to correctly answer various questions in the diverse VideoQA paradigms (open-ended VideoQA, video-conditioned fill-in-the-blank, multiple-choice VideoQA), showing both a strong textual commonsense reasoning and a complex multi-modal understanding.

However, our zero-shot model still underperforms compared to VideoQA-supervised models (see Table 4.15) and we analyze its failure cases in the last row of Figure 4.3a. Qualitatively, we find that the zero-shot model can fail on examples requiring complex temporal or spatial understanding e.g. in the third example, the model does not detect a toaster behind the woman; in the second example, it gets confused as the person browses through many different tabs from their phone. It can also be semantically inaccurate, as in the first example, the model confuses a restaurant with a bakery; in the fourth example, it confuses a chicken with another kind of bird.

4.4.5 Freezing the BiLM is also beneficial in supervised settings

Fully-supervised VideoQA. We next present an evaluation in a supervised setup where we finetune FrozenBiLM on a downstream VideoQA task. We emphasize that we also keep our pretrained language model weights *frozen* all throughout finetuning. As shown in Table 4.14, our approach improves the state of the art on LSMDC-FiB, iVQA, MSRVTT-QA, MSVD-QA, ActivityNet-QA and How2QA. In particular, FrozenBiLM outperforms strong recent baselines such as All-in-one [Wang, 2023] on 2/3 datasets, VIOLET [Fu, 2021] on 3/4 datasets and MERLOT [Zellers, 2021] on 4/5 datasets. Our approach has significantly less trainable parameters compared to the state of the art [Fu, 2021; Wang, 2023; Zellers, 2021] as we *freeze* the weights of the pretrained language model. We ablate this major difference in Table 4.14, and find that our *FrozenBiLM* with the *frozen* language model performs better and trains twice faster compared

Method	# Trained Params	Fill-in-the-blank LSMDC	Open-ended					Multiple-choice	
			iVQA	MSRVTT QA	MSVD QA	ActivityNet QA	TGIF QA	How2QA	TVQA
HCRN [Le, 2021]	44M	—	—	35.4	36.8	—	57.9	—	71.4*
HERO [Li, 2020b]	119M	—	—	—	—	—	—	74.1*	73.6*
ClipBERT [Lei, 2021b]	114M	—	—	37.4	—	—	60.3	—	—
Just Ask [Yang, 2022c]	157M	—	35.4	41.8	47.5	39.0	—	85.3	—
SiaSamRea [Yu, 2021]	—	—	—	41.6	45.5	39.8	60.2	84.1	—
MERLOT [Zellers, 2021]	223M	52.9	—	43.1	—	<u>41.4</u>	69.5	—	78.7*
Reserve [Zellers, 2022]	644M	—	—	—	—	—	—	—	86.1*
VIOLET [Fu, 2021]	198M	53.7	—	43.9	47.9	—	<u>68.9</u>	—	—
All-in-one [Wang, 2023]	110M	—	—	<u>46.8</u>	48.3	—	66.3	—	—
UnFrozenBiLM (Ours)	890M	<u>58.9*</u>	37.7*	45.0*	53.9*	43.2*	66.9	87.5*	79.6*
<i>FrozenBiLM</i> w/o adapters	1M	60.4*	38.2*	43.2*	51.7*	38.3*	66.5	79.3*	—
<i>FrozenBiLM</i> w/o CM training	<u>30M</u>	57.1*	34.3*	46.2*	51.9*	41.8*	67.4	75.8*	—
<i>FrozenBiLM</i> w/o speech (Ours)	<u>30M</u>	58.6	39.7	47.0	<u>54.4</u>	43.2	68.6	81.5	57.5
<i>FrozenBiLM</i> (Ours)	<u>30M</u>	63.5*	<u>39.6*</u>	47.0*	54.8*	43.2*	68.6	<u>86.7*</u>	<u>82.0*</u>

Table 4.14: Comparison with the state of the art, and the variant UnFrozenBiLM which does not freeze the language model weight, on fully-supervised benchmarks. * denotes results obtained with speech input. CM training denotes Cross-Modal training. Results of *FrozenBiLM* w/o adapters and *FrozenBiLM* w/o CM training on TVQA are not reported given that the hidden test set used here can only be accessed a limited number of times.

Variant	Supervision	Fill-in-the-blank LSMDC	iVQA	Open-ended				Multiple-choice			
				MSRVTT QA	MSVD QA	ActivityNet QA	TGIF QA	How2QA	TVQA		
1. UnFrozenBiLM	0% (zero-shot)		37.1	21.0	17.6	31.9	20.7	30.7	45.7	29.7	
2. FrozenBiLM	0% (zero-shot)		51.5	26.8	16.7	33.8	25.9	41.9	58.4	59.7	
3. UnFrozenBiLM	1% (few-shot)		46.2	23.5	33.4	43.7	31.6	51.7	68.0	—	
4. FrozenBiLM	1% (few-shot)		56.9	31.1	36.0	46.5	33.2	55.1	71.7	72.5	
5. UnFrozenBiLM	10% (few-shot)		52.6	29.5	38.9	49.8	36.5	57.8	73.2	—	
6. FrozenBiLM	10% (few-shot)		59.9	35.3	41.7	51.0	37.4	61.2	75.8	77.6	
7. UnFrozenBiLM	100% (fully-supervised)		58.9	37.7	45.0	53.9	43.2	66.9	87.5	—	
8. FrozenBiLM	100% (fully-supervised)		63.5	39.6	47.0	79.6	54.8	43.2	68.6	86.7	82.0

Table 4.15: Few-shot results, by finetuning *FrozenBiLM* using a small fraction of the downstream training dataset, compared with the variant UnFrozenBiLM which does not freeze the language model weights. Few-shot results of UnFrozenBiLM on TVQA are not reported given that the hidden test set used here can only be accessed a limited number of times.

to UnFrozenBiLM where we update the language model during training. This shows that *freezing* the language model is not only beneficial for zero-shot but also in fully-supervised settings, therefore suggesting that our FrozenBiLM framework also provides a parameter-efficient solution for VideoQA training. We also note that FrozenBiLM performs competitively even without speech input, although speech helps significantly for the performance on LSMDC, How2QA and TVQA. Additionally, we find that cross-modal training is significantly beneficial in this setting. Finally, we note that training adapters has a considerable importance on the performance in this setting.

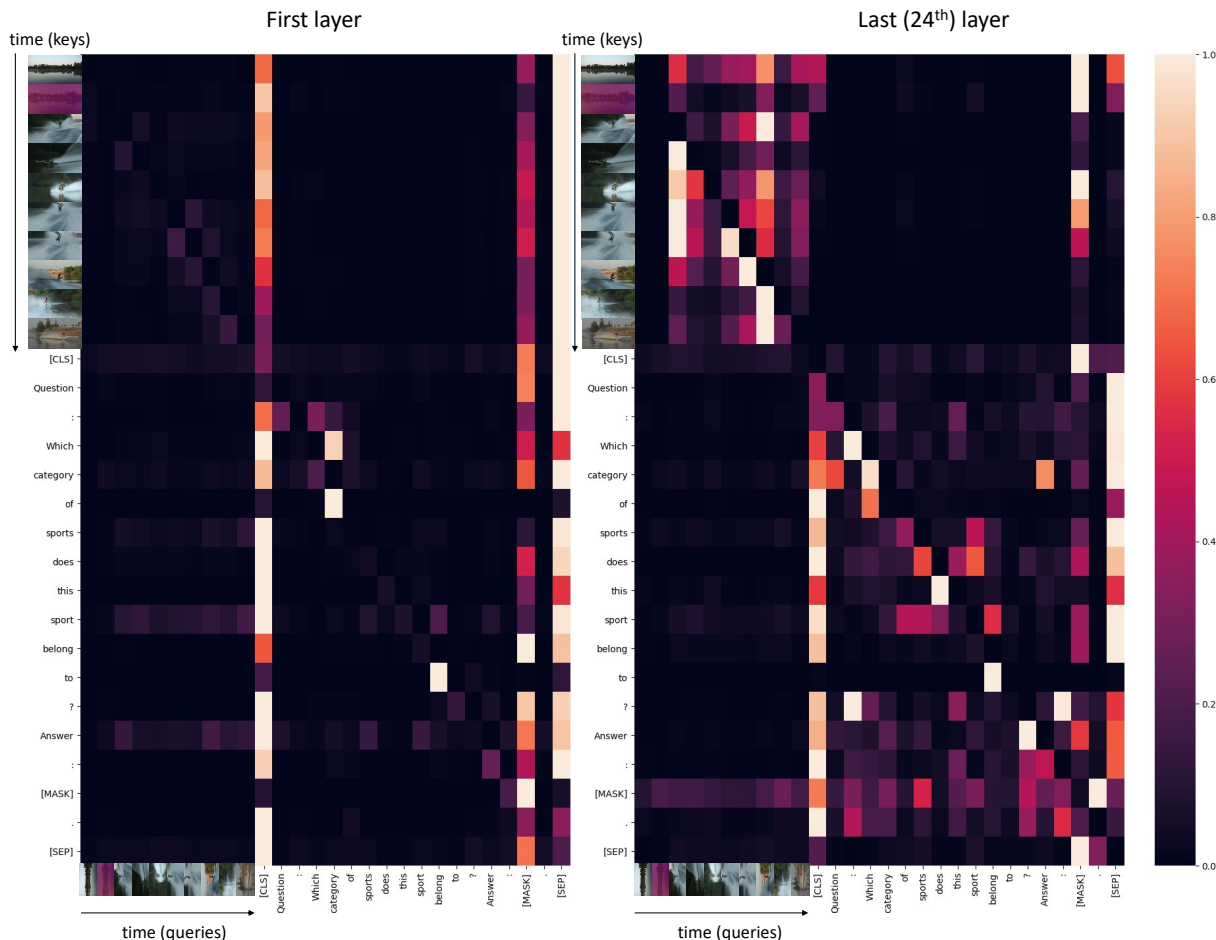


Figure 4.4: **FrozenBiLM self-attention visualization for zero-shot VideoQA.** Visualization of the attention weights between the different visual tokens from the video prompt and the textual tokens from the text embedder, for the second example of the second row in Figure 4.3a. A column corresponds to the weights of the different visual and text tokens for the given token. These attention weights are averaged across all 24 heads, and renormalized by the maximum weight for each token (i.e. each column) for the purpose of visualization. Lighter colors correspond to higher attention weights (see the colorbar on the right). Note that the self-attention weights are *frozen* after text-only pretraining.

Few-shot VideoQA. The low number of trainable parameters when training *FrozenBiLM* makes it particularly well-suited in the low data regime. To verify this, we explore a few-shot VideoQA setting where we finetune our pretrained model using varying fractions of VideoQA training data. From Table 4.15 we observe significant improvements over zero-shot when using only 1% of training data. Moreover, consistently with results in the zero-shot and fully-supervised settings, we find that freezing the language model combined with training adapters outperforms finetuning the language model (compare rows 3 and 4, or rows 5 and 6). Interestingly, the difference is larger when using 1% of the downstream training dataset (rows 3 and 4) compared to using 10% (rows 5 and 6) or 100% (rows 7 and 8). These results demonstrate that our approach is particularly efficient in settings where VideoQA annotations are scarce.

4.4.6 Qualitative analysis of the *frozen* self-attention patterns in Frozen-BiLM

We show in Section 4.4.2 that the visual modality is crucial for the zero-shot VideoQA performance. Here we further analyze qualitatively *how*, for zero-shot VideoQA, our model makes use of the visual modality through self-attention layers which are *frozen* after text-only pretraining. Figure 4.4 illustrates the self-attention patterns in FrozenBiLM for the second example in the first row of Figure 4.3a. Despite the freezing, we observe that these layers actually enable visual-linguistic interactions. Indeed, in the first layer (Figure 4.4, left), the [CLS], [MASK] and [SEP] tokens significantly attend to the visual tokens. Moreover, we observe substantially different patterns in the last layer (Figure 4.4, right): while the [MASK] token still attends to visual tokens, the different visual tokens at different timesteps attend between each other and the [CLS] and [SEP] tokens mainly attend to other text tokens. Consistently with results presented in Section 4.4.2, this qualitative analysis suggests that the *frozen* self-attention layers in FrozenBiLM do enable visual-linguistic interactions.

4.5 Conclusion

We present FrozenBiLM, a framework that extends *frozen* bidirectional language models to multi-modal inputs by training additional modules on web-scraped data, and that tackles zero-shot VideoQA through masked language modeling. We provide extensive ablation studies and show the efficiency of our framework compared to its autoregressive variant. Frozen-BiLM improves over the prior state-of-the-art zero-shot VideoQA on the LSMDC-FiB, iVQA, MSRVTT-QA, MSVD-QA, ActivityNet-QA, TGIF-FrameQA, How2QA and TVQA datasets, performs competitively in fully-supervised settings and exhibits strong performance in the few-shot VideoQA setting we newly introduce.

Limitations. Promising directions not explored in this work include scaling the size of a bidirectional language model to several billion parameters, and additional training on large datasets of YouTube videos with accompanying speech transcripts and/or audio [Zellers, 2022]. Also, our model cannot be applied out-of-the-box to complex multi-modal text generation tasks such as video captioning [Alayrac, 2022].

Chapter 5

TubeDETR: Spatio-Temporal Video Grounding with Transformers

We consider the problem of localizing a spatio-temporal tube in a video corresponding to a given text query (see Figure 5.1). This is a challenging task that requires the joint and efficient modeling of temporal, spatial and multi-modal interactions. To address this task, we propose TubeDETR, a transformer-based architecture inspired by the recent success of such models for text-conditioned object detection. Our model notably includes: (i) an efficient video and text encoder that models spatial multi-modal interactions over sparsely sampled frames and (ii) a space-time decoder that jointly performs spatio-temporal localization. We demonstrate the advantage of our proposed components through an extensive ablation study. We also evaluate our full approach on the spatio-temporal video grounding task and demonstrate improvements over prior state of the art on the challenging VidSTG [Zhang, 2020d] and HC-STVG [Tang, 2021] benchmarks. Our code and models are publicly available at [Yang, 2022b].

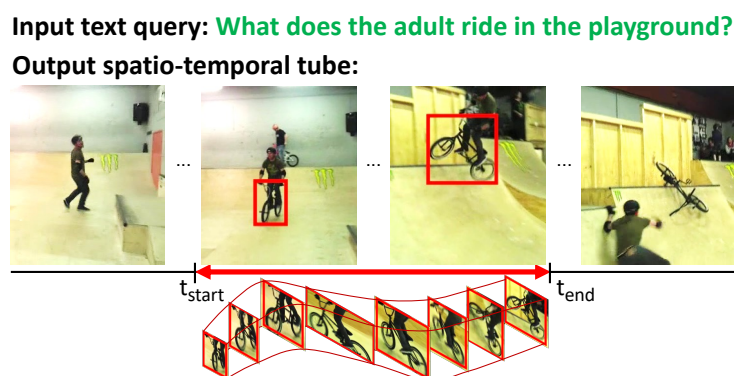


Figure 5.1: Spatio-temporal video grounding requires reasoning about space, time, and language.

5.1 Introduction

Grounding natural language in visual content is a fundamental skill to build powerful and explainable vision and language models. In particular, understanding the association of language with spatial regions and temporal boundaries in videos is particularly important to analyze and improve multi-modal video models. This goes beyond associating a global visual representation with a textual representation [Radford, 2021; Miech, 2020], as it requires to reason about detailed spatio-temporal visual representations and their association with natural language, as illustrated in Figure 5.1.

Spatio-temporal video grounding, recently introduced in [Zhang, 2020d], is an interesting and challenging task that lies at the intersection of visual grounding [Hu, 2016; Nagaraja, 2016; Vasudevan, 2018] and temporal localization [Hendricks, 2017; Gao, 2017a; Chen, 2018]. Given an untrimmed video and a textual description of an object, spatio-temporal video grounding aims at localizing a spatio-temporal tube (i.e. a sequence of bounding boxes) for the target object described by the input text. This task is particularly challenging as videos are highly diverse and often present challenging scenarios where different entities have similar appearance or perform similar actions within one scene.

The success of attention-based models in natural language processing [Vaswani, 2017; Devlin, 2019] has recently inspired approaches to integrate transformers into computer vision tasks, such as image classification [Dosovitskiy, 2021], object detection [Carion, 2020], semantic segmentation [Liu, 2021b] or action recognition [Arnab, 2021; Bertasius, 2021; Zhang, 2021b; Patrick, 2021]. Notably, with DETR [Carion, 2020], transformers have shown competitive performance on object detection while removing the need of multiple hand-designed components encoding a prior knowledge about this task. More recently, MDETR [Kamath, 2021] has extended this framework for various text-conditioned object detection tasks in the image domain, such as phrase grounding, referring expression comprehension and segmentation.

Inspired by these works, and the fact that attention-based architectures are an intuitive choice for modelling multi-modal and spatio-temporal contextual relationships in videos, we develop a transformer encoder-decoder model for spatio-temporal video grounding, as illustrated in Figure 5.2. While existing approaches for this task rely on pre-extracted object proposals [Zhang, 2020d], tube proposals [Tang, 2021] or upsampling layers [Su, 2021], our architecture simply reasons about abstractions called *time queries* to jointly perform temporal localization and visual grounding. Our framework enables to use the same representations for both subtasks in order to learn powerful contextualized representations.

More specifically, our architecture includes key components to jointly model temporal, spatial and multi-modal interactions. Our video-text encoder efficiently encodes spatial and multi-modal interactions by computing these interactions over sparsely sampled frames, and separately recovers temporally local information with a lightweight fast branch. Our space-time decoder models temporal interactions with temporal self-attention layers, and spatial and multi-modal interactions with time-aligned cross-attention layers. Spatio-temporal video grounding is then tackled with multiple heads on top of the decoder outputs, which predict the object boxes and

temporal start and end probabilities. We conduct various ablation studies, where we notably show the benefit of our video-text encoder in terms of performance-memory trade-off, and the efficiency of our space-time decoder in terms of spatio-temporal grounding results. Finally, we show that our method significantly improves over prior state-of-the-art methods on two benchmarks, VidSTG [Zhang, 2020d] and HC-STVG [Tang, 2021].

In summary, our contributions are three-fold:

- (i) We propose a novel architecture for spatio-temporal video grounding that performs this task with a space-time transformer decoder.
- (ii) We propose a dual-stream encoder that efficiently encodes spatial and multi-modal interactions, based on a slow multi-modal stream and a lightweight fast visual stream.
- (iii) We conduct comprehensive experiments on two benchmarks, VidSTG and HC-STVG, showing the effectiveness of our framework for the spatio-temporal video grounding task. Our approach, referred to as TubeDETR, outperforms all prior state-of-the-art methods by a large margin.

5.2 Related Work

Spatio-temporal video grounding. Visual grounding consists in spatially localizing an object given a referring expression, and has been an active area of research both in the image domain [Deng, 2018; Hu, 2016; Hu, 2017; Liu, 2021a; Nagaraja, 2016; Wang, 2021c; Xiao, 2017; Yu, 2017; Zhang, 2018b; Zhuang, 2018] and the video domain [Huang, 2018; Shi, 2019b; Vasudevan, 2018]. A standard paradigm consists in using pre-extracted object proposals [Liu, 2017; Liu, 2019a; Wang, 2019a; Yamaguchi, 2017; Yang, 2019a; Yang, 2019b; Yu, 2018], while some recent works [Deng, 2021b; Huang, 2021a; Kamath, 2021; Liao, 2020; Luo, 2020a; Yang, 2019c; Yang, 2020b] have proposed one-stage approaches which do not rely on such proposals. Our work follows the one-stage framework of MDETR [Kamath, 2021], but extends it to spatio-temporal video grounding with temporal localization losses (see Equation 5.1), slow-fast encoding (see Figure 5.3), and space-time decoding (see Figure 5.4).

A separate line of work focuses on temporally localizing moments in a video given a natural language query [Chen, 2018; Chen, 2019a; Chen, 2019b; Gao, 2017a; He, 2019; Hendricks, 2017; Hendricks, 2018; Lin, 2020b; Mithun, 2019; Rodriguez, 2020; Wang, 2019b; Wang, 2020a; Yuan, 2019; Zhang, 2019b; Zhang, 2019a; Zhang, 2020a; Zhang, 2020c; Zeng, 2020]. These works build architectures that reason about time but do not preserve spatial information. Spatio-temporal video grounding lies at the intersection of temporal localization and visual grounding. While some approaches [Chen, 2019c; Tang, 2021; Yamaguchi, 2017] rely on pre-extracted tube proposals, or object proposals [Zhang, 2020d], our method does not require any pre-extracted proposals. A recent work [Su, 2021] proposes STVGBert, a one-stage approach that extends the ViBERT model [Lu, 2019] pretrained on Conceptual Captions [Sharma, 2018] to this task.

STVGBert uses deconvolutions to perform visual grounding, and symmetrically models temporal and spatial interactions. In contrast, our architecture performs visual grounding with a transformer decoder, and separately reasons about the temporal and spatial dimensions.

Temporal modeling for video understanding. The rise of powerful models for image understanding such as ViT [Dosovitskiy, 2021] or DETR [Carion, 2020] has fostered research extending these models to the video domain [Arnab, 2021; Bertasius, 2021; He, 2021a; Lei, 2021a; Patrick, 2021; Zhang, 2021b]. In particular, [Lei, 2021a] propose an architecture that views moment retrieval as a direct set prediction problem, but is unsuitable to visual grounding as it does not preserve spatial information. [He, 2021a] extend the DETR framework to videos, and propose an architecture built with sequentially added modules on top of Deformable DETR [Zhu, 2021], while ours is built on inner modifications of a pretrained encoder and decoder and also reasons about language. Our dual-branch encoder is also related to SlowFast networks [Feichtenhofer, 2019; Xiao, 2020] which combine fast and slow video streams. In contrast, in our case, both streams operate on features extracted from the same backbone, and our dual-stream architecture is motivated by the computational complexity related to multi-modal modeling.

Vision and language. Transformer-based architectures have become ubiquitous in various vision and language tasks [Chen, 2020b; Chen, 2021d; Cornia, 2020; Desai, 2021a; Huang, 2020c; Kim, 2021b; Li, 2020a; Li, 2020d; Lu, 2019; Lu, 2020; Su, 2019; Tan, 2019; Zhou, 2020]. Most video-text transformers rely either on pre-extracted object features [Zhu, 2020], or spatially pooled features [Gabeur, 2020; Ging, 2020; Li, 2020b; Sun, 2019b; Yang, 2021b; Zhou, 2018c], which do not preserve detailed spatial information. In contrast, our architecture is designed to preserve spatial information to perform visual grounding. Some recent works propose transformer-based architectures reasoning on videos and text that do preserve spatial information [Akbari, 2021; Bain, 2021; Lei, 2021b; Zellers, 2021]. However, these works typically aim to learn global video representations to tackle video-level prediction tasks, while we focus on learning detailed frame-level representations to address a dense prediction task requiring spatial and temporal localization.

5.3 Method

We first give an overview of our model in Section 5.3.1. Next, we describe in detail the two main components of our model, the video-text encoder (Section 5.3.2) and the space-time decoder (Section 5.3.3). Then in Section 5.3.4 we explain the loss used to train our model. Finally in Section 5.3.5 we present how we initialize our model weights.

5.3.1 Overview

Our objective is, given a video and a language query, to output a spatio-temporal tube, i.e. a sequence of bounding boxes with temporal boundaries, grounding the language query in the video.

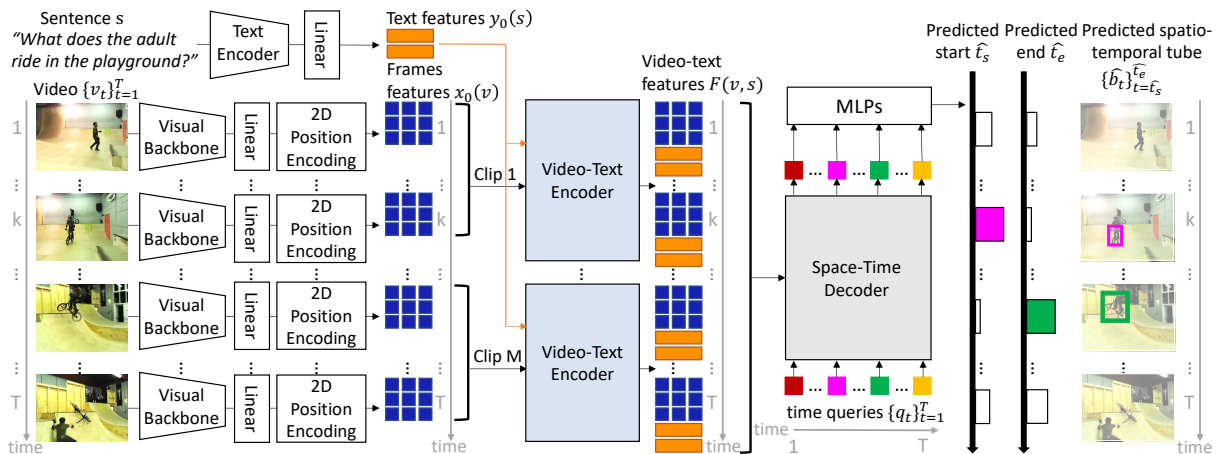


Figure 5.2: **TubeDETR model overview.** All input video frames v_t and the sentence s are first processed with a Visual Backbone and a Text Encoder. The resulting text and video features $y_0(s)$ and $x_0(v)$ are then jointly encoded with a Video-Text Encoder that computes spatial and multi-modal interactions for M short clips of k frames (about 1 second). The resulting video-text features $F(v, s)$ are then decoded into the output spatio-temporal tube \hat{b} using a Space-Time Decoder that jointly reasons about time, space and text over the entire video.

This is challenging as it requires modelling long-range *spatial* and *temporal* interactions between the language query and the video where the video may have hundreds of frames represented by tens of thousands spatio-temporal video features. Hence efficiency is a major challenge. To address this issue we design an encoder-decoder architecture, illustrated in Figure 5.2, that enables accurate yet efficient modelling of video-language spatial and temporal interactions across the entire video. In particular, our two-stream video-text encoder (Section 5.3.2) models video-language interactions only over short clips of about one second but allows for detailed spatial localization. Our space-time decoder (Section 5.3.3) then models long-range temporal interaction over the entire video to produce a temporally consistent output and accurate predictions of the start and end times of the output spatio-temporal tube.

5.3.2 Video-Text Encoder

Our encoder is illustrated in Figure 5.3 and described next. Its objective is to model spatial and multi-modal interactions between the language query and the video to accurately spatially ground the query in each frame. To achieve this, we leverage the ability of the self-attention layers to jointly model spatial and visual-linguistic interactions [Kamath, 2021; Lei, 2021b; Huang, 2020c]. However, computing self-attention between visual features and textual features for every frame is computationally expensive. For this reason, we propose to compute spatial and multi-modal interactions only for every k -th frame. We denote the resulting stream as *slow multi-modal* branch. We use a separate lightweight *fast visual-only* branch that preserves the original frame rate and allows us to recover some of the high frequency spatio-temporal details lost by the sparse sampling in the slow branch.

Formally, our encoder takes as input a set of 2D flattened image features $x_0(v) \in \mathbb{R}^{T \times HW \times d}$

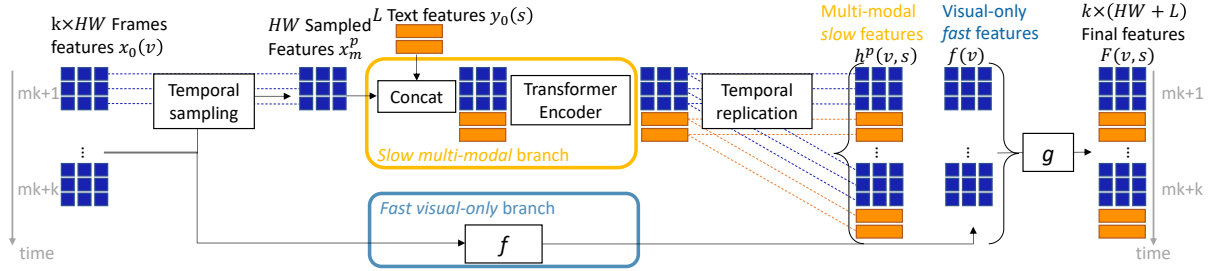


Figure 5.3: **Video-Text Encoder** takes as input a set of 2D flattened image features $x_0(v)$ together with a set of text features $y_0(s)$ from the query sentence, and outputs a set of video-text features $F(v, s)$, one for each frame. Top: the *Slow multi-modal* branch first samples video features x_m^p , one from every k frames. Then it computes multi-modal interactions between the sampled features x_m^p and text features y_0 using a transformer encoder. The temporal sampling reduces the number of video features in order to efficiently compute the attention-based interactions. Bottom: lightweight “Fast visual-only” branch f processes features from *all* frames but without any attention layers for increased efficiency. Features from both branches are then combined in module g into the final set of per-frame features $F(v, s)$.

from the visual backbone for all T frames of the input video together with a set of L text features $y_0(s) \in \mathbb{R}^{L \times d}$ extracted by the text encoder from the query sentence, and outputs a set of video-text features $F(v, s) \in \mathbb{R}^{T \times (HW+L) \times d}$, one for each frame. Next we give the details of the Slow and Fast branches, and the final feature aggregation module.

Slow multi-modal branch. The goal of this branch (see top of Figure 5.3) is to model interactions between visual and textual representations. This branch first samples features from *one* frame for a short clip of k consecutive frames. A typical clip length is one second, i.e. $k = 5$ with a standard frame rate of 5 frames per second [Zhang, 2020d]. Formally, the resulting feature map is written as $x^p \in \mathbb{R}^{M \times HW \times d}$ where $M = \lceil \frac{T}{k} \rceil$ is the number of clips, k is the length of the clip and T is the length of the entire video. We then concatenate, for each clip m , its visual features x_m^p with text features $y_0(s)$ and forward it to a N-layer transformer encoder. The outputs are contextualized visual-text representations $h^p(v, s) \in \mathbb{R}^{M \times (HW+L) \times d}$, which effectively combine information from the input video v and the query sentence s .

Fast visual-only branch. The previously explained temporal sparse sampling scheme reduces significantly the memory requirements of the video-text encoder but results in a loss of spatio-temporal details which are important for spatio-temporal video grounding. To alleviate this issue, we introduce module f (see bottom of Figure 5.3) which operates on *2D flattened image features for all frames*. Formally, given feature map $x_0(v)$, this module outputs visual features $f(v) \in \mathbb{R}^{T \times HW \times d}$. This *fast* branch preserves the spatial and temporal resolution of the features but is computationally light as it does not compute any multi-modal or spatial interactions. For additional efficiency, at training time, this branch does not back-propagate gradients to the visual backbone. Furthermore, we show in Section 5.4.2 that it is able, when combined with the temporally sparse features obtained from the slow branch, to recover some of the temporal information lost during the temporal sampling.

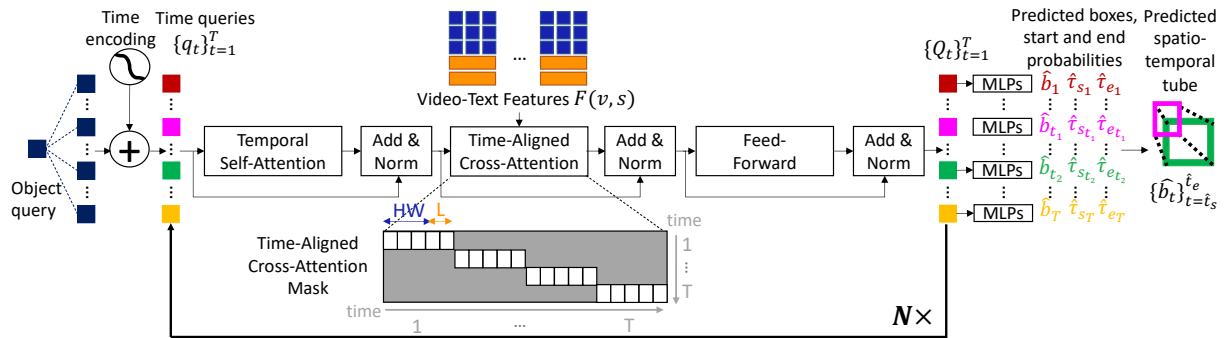


Figure 5.4: **Space-Time Decoder.** The decoder is composed of N repeated blocks. In each block, time queries q_t successively attend to each other via *temporal self-attention* and to their respective time-aligned video-text features $F(v, s)$ via *time-aligned cross-attention*. The cross-attention mask (bottom) indicates the non-zero weights (white) between the input $HW + L$ video-language features for each of the T input frames (x-axis) and T time queries (y-axis). The cross-attention mask ensures that each time query q_t only cross-attends to video-text features $F(v, s)$ at the corresponding frame t , which significantly increases efficiency of the decoder and enables decoding entire videos of T frames. The temporal modelling over the entire length of the video is ensured by the temporal self-attention layers.

Slow-Fast feature aggregation. We now describe the *slow* and *fast* branches aggregation module (see Figure 5.3, right), which fuses information from both branches and outputs final video-text features. To match the temporal dimension of the output from the *fast* branch $f(v)$, the output of the *slow* multi-modal branch $h^p(v, s)$ is temporally replicated k times for each clip resulting in video-text encodings $h(v, s) \in \mathbb{R}^{T \times (HW+L) \times d}$. These encodings are a concatenation of text-contextualized visual encodings $h_v(v, s) \in \mathbb{R}^{T \times HW \times d}$ and visually-contextualized textual encodings $h_s(v, s) \in \mathbb{R}^{T \times L \times d}$. The text-contextualized visual encodings $h_v(v, s)$ are combined with the outputs of the *fast* branch with an additional aggregation module g and a residual connection, resulting in aggregated visual encodings $F_v(v, s) = g(h_v(v, s), f(v)) + h_v(v, s)$. The final output of our video-text encoder is obtained by concatenating these aggregated visual encodings with the visually-contextualized textual encodings, i.e. $F(v, s) = [F_v(v, s), h_s(v, s)] \in \mathbb{R}^{T \times (HW+L) \times d}$. In detail, the module g is implemented as a sum followed by a linear layer, i.e. $g(h_v(v, s), f(v)) = \text{Linear}(h_v(v, s) + f(v))$.

5.3.3 Space-Time Decoder

Our decoder is illustrated in Figure 5.4 and detailed next. Its objective is to model the temporal interactions within the entire video of T frames and decode the multi-modal features from the encoder into a temporally coherent output tube with accurate start and end times. This is achieved by an efficient decoder architecture that alternates (i) *temporal self-attention* layers, which model *temporal* interactions across the entire video, with (ii) *time-aligned cross attention* layers, which efficiently incorporate the video-text features for individual frames obtained from the encoder. In detail, the decoder operates on T positional encodings $\{q_t\}_{t=1}^T$, one per frame, referred to as time queries. The initial encoding of each time query is obtained by summing

a learnt object encoding common to all frames, and a frozen sinusoidal time encoding. The decoder also takes as input $T \times (HW + L)$ video-language embeddings $F(v, s)$ output from the video-text encoder. The decoder is a succession of N decoder blocks. Each block is composed of temporal self-attention, time-aligned cross-attention, and feed-forward layers, interleaved with normalization [Ba, 2016], as shown in Figure 5.4. The decoder outputs refined time queries $\{Q_t\}_{t=1}^T$, which are contextualized across all frames in the video together with video-text features produced by the encoder. The refined time queries are then jointly used for outputting the spatio-temporal video tube that grounds the input sentence in the video. The individual layers are described in detail next.

Temporal self-attention. The T input time queries q_t attend to each other using the temporal self-attention layer. This layer is in each of the N blocks of the decoder and is responsible for modelling the long-range temporal interactions in the entire video. This is possible because of the relatively low complexity of this layer, which does not depend on the spatial resolution of the input video.

Time-aligned cross-attention. Allowing each time query to cross-attend to all $T \times (HW + L)$ video-text features can be highly computationally expensive due to the large number of video frames T and a large spatial resolution HW of the video features. Instead, in our cross-attention module, each time query q_t only cross-attends to its temporally corresponding multi-modal features $F(v, s)[t]$ at frame t . Note that with our time-aligned cross-attention formulation, the time encoding and the temporal self-attention layers are all the more important, as they are responsible for the temporal modelling across the entire video. Without them, our decoder would be decoding each frame independently. Their importance is ablated in Section 5.4.2.

Prediction heads. The output of the decoder is a set of refined time queries $\{Q_t\}_{t=1}^T$. They are jointly used for visual grounding and temporal localization to simultaneously obtain predictions for *all frames of the video*. In detail, normalized coordinates of all bounding boxes (2D center and size) $\hat{b} \in [0, 1]^{T \times 4}$ are predicted with a 3-layer MLP. Probabilities of the start and the end of the output video tube, $\hat{\tau}_s \in [0, 1]^T$ and $\hat{\tau}_e \in [0, 1]^T$, respectively, are predicted with 2-layer MLPs. At inference time, the start and end times of the output tube, \hat{t}_s and \hat{t}_e , are computed by choosing the maximum of the joint start and end probability distribution $(\hat{\tau}_s, \hat{\tau}_e) \in [0, 1]^{T \times T}$ with invalid combinations where $\hat{t}_e \leq \hat{t}_s$ masked out. The predicted spatio-temporal tube $\{\hat{b}_t\}_{t=\hat{t}_s}^{\hat{t}_e}$ is composed from bounding boxes \hat{b}_t predicted within the chosen start and end times \hat{t}_s and \hat{t}_e .

5.3.4 Training loss

The input training data is in the form of a set of videos, where each video is annotated with a query sentence s and the corresponding video tube b composed of a set of bounding boxes and corresponding start and end times, t_s and t_e . Inspired by [Rodriguez, 2020], we construct a target start (respectively end) distribution $\tau_s \in [0, 1]^T$ (respectively τ_e) which follows a quantized Gaussian centered at $t_s \in [0, T - 1]$ (respectively t_e) with standard deviation 1. We train our

architecture with a linear combination of four losses

$$\begin{aligned} \mathcal{L} = & \lambda_{\mathcal{L}_1} \mathcal{L}_{\mathcal{L}_1}(\hat{b}, b) + \lambda_{gIoU} \mathcal{L}_{gIoU}(\hat{b}, b) \\ & + \lambda_{KL} \mathcal{L}_{KL}(\hat{\tau}_s, \hat{\tau}_e, \tau_s, \tau_e) + \lambda_{att} \mathcal{L}_{att}(A) \end{aligned} \quad (5.1)$$

where $b \in [0, 1]^{4(t_e - t_s + 1)}$ denotes the normalized ground truth box coordinates and \hat{b} the predicted bounding boxes and $A \in [0, 1]^{T \times T}$ denotes the temporal self-attention matrix. Finally, different λ_{\bullet} are scalar weights of the individual losses. $\mathcal{L}_{\mathcal{L}_1}$ is a \mathcal{L}_1 loss on bounding box coordinates. \mathcal{L}_{gIoU} is a generalized ‘‘intersection over union’’ (IoU) loss [Rezatofighi, 2019] on the bounding boxes. Both \mathcal{L}_1 and \mathcal{L}_{gIoU} are used for spatial grounding. $\mathcal{L}_{KL}(\hat{\tau}_s, \hat{\tau}_e, \tau_s, \tau_e)$ is the Kullback-Leibler divergence loss measuring the distance between the predicted and the target start distribution as well as the distance between the predicted and the target end distribution [Rodriguez, 2020]. $\mathcal{L}_{att}(A)$ is a guided attention loss [Rodriguez, 2020] that encourages weights corresponding to time queries outside of the temporal boundaries to be lower than the weights inside these boundaries. \mathcal{L}_{KL} and $\mathcal{L}_{att}(A)$ are both used for temporal grounding. Losses are computed at each layer of the decoder following [Carion, 2020].

5.3.5 Weight initialization

We initialize our architecture with weights from MDETR [Kamath, 2021] pretrained on Flickr30k [Plummer, 2015], MS COCO [Chen, 2015] and Visual Genome [Krishna, 2016]. In detail, weights of our video-text encoder are initialized from the MDETR multi-modal encoder, except for the fast and aggregation modules. We also use the weights from the MDETR single-image multi-object decoder to initialize our multi-frame single-object space-time decoder, except for the temporal localization head. We show the benefit of this initialization notably by comparing it to an ImageNet initialization, i.e. using a visual backbone pretrained on ImageNet with a randomly initialized transformer, in Section 5.4.2. We also evaluate a MDETR-equivalent baseline in Section 5.4.2.

5.4 Experiments

This section demonstrates the effectiveness of our architecture and compares our method to the state of the art. We first introduce the datasets, evaluation metrics and implementation details in Section 5.4.1. We then present ablation studies in Section 5.4.2. The comparison to the state of the art in spatio-temporal video grounding is given in Section 5.4.3. Next we show qualitative results in Section 5.4.4. Finally we present a visualization of space, time and language attention patterns in the decoder in Section 5.4.5.

5.4.1 Experimental setup

Datasets. We evaluate our approach on the VidSTG [Zhang, 2020d] and HC-STVG [Tang, 2021] datasets. Both are annotated with spatio-temporal tubes corresponding to text queries.

VidSTG consists of 99,943 sentence descriptions with 44,808 declarative sentences and 55,135 interrogative sentences describing 79 types of objects appearing in 10,303 different videos. The dataset is divided into training, validation and test subsets with 80,684, 8,956 and 10,303 distinct sentences respectively, and 5,436, 602 and 732 distinct videos respectively. **HC-STVG** consists of videos in multi-person scenes, each annotated with one sentence referring to a person. For ablation, we use the second improved version of the dataset **HC-STVG2.0** which is divided into training and validation subsets with 10,131 and 2,000 video-sentence pairs, respectively. The test set is not publicly available at the time of writing. To compare with prior work, we use the first version of the dataset **HC-STVG1** which is divided into training and test subsets with 4,500 and 1,160 video-sentence pairs, respectively.

Evaluation metrics. We follow [Zhang, 2020d] and define $vIoU$ as $vIoU = \frac{1}{|S_u|} \sum_{t \in S_i} IoU(\hat{b}_t, b_t)$ where S_u (respectively S_i) is the set of frames in the union (respectively intersection) between the ground truth (GT) and the predicted timestamps. \hat{b}_t (respectively b_t) are the predicted (respectively GT) boxes at time t . To evaluate spatio-temporal video grounding, we use m_vIoU , which is the average of $vIoU$. We also use $vIoU@R$, the proportion of samples for which $vIoU > R$. To isolate the evaluation of temporal localization, we use m_tIoU which is the average of temporal IoU between the GT start and end and the predicted start and end. Likewise, to evaluate spatial grounding only, we use m_sIoU , which is computed by using the GT start and end times. We also report peak GPU memory usage during training (Mem.) to measure the memory footprint of alternative models.

Implementation details. The visual backbone is ResNet-101 [He, 2016], the text encoder is RoBERTa [Liu, 2019b] and the fast module f is a linear layer. Following [Zhang, 2020d], we sample 5 frames per second for videos, and for videos with more than 200 sampled frames we uniformly sample 200 frames. We use hyper-parameters $T = 200$, $N = 6$, $d = 256$, $\lambda_{L_1} = 5$, $\lambda_{giou} = 2$, $\lambda_{KL} = 10$ and $\lambda_{att} = 1$. We train our networks for 10, 20 and 40 epochs on VidSTG, HC-STVG2.0 and HC-STVG1, respectively. The final model is selected based on the best spatio-temporal video grounding performance on the validation set. For the largest dataset VidSTG, the optimization takes 2 days on 16 Tesla V100 GPUs.

In our transformer, the number of heads is 8 and the hidden dimension of the feed-forward layers is 2048. We set the initial learning rates to $1e^{-5}$ for the visual backbone, and $5e^{-5}$ for the rest of the network. The learning rate follows a linear schedule with warm-up for the text encoder and the learning rate is constant for the rest of the network. We use the AdamW optimizer [Loshchilov, 2019] and weight-decay $1e^{-4}$. Video data augmentation includes spatial random resizing, spatial random cropping preserving box annotations, and temporal random cropping preserving the annotated time interval. Dropout [Srivastava, 2014] with probability 0.1 is applied in our transformer layers, and dropout with probability 0.5 is applied in the temporal localization head. We use exponential moving average with a decay rate of 0.9998, and an effective batch size of 16 videos. For temporal stride $k = 1$ the fast and aggregation modules in the encoder are not active, as their goal is to recover local spatial and temporal

	Time Encoding	Self Attention	Declarative Sentences					Interrogative Sentences				
			m_tIoU	m_vIoU	vIoU @0.3	vIoU @0.5	m_sIoU	m_tIoU	m_vIoU	vIoU @0.3	vIoU @0.5	m_sIoU
1.	\times	-	24.4	13.6	17.8	7.3	51.9	23.5	11.1	13.3	5.2	43.1
2.	\times	Temporal	25.3	14.1	18.6	7.3	52.3	25.0	12.1	15.4	5.9	43.3
3.	\checkmark	-	42.1	23.2	31.8	19.5	51.3	41.5	19.7	26.2	15.8	42.5
4.	\checkmark	Temporal	46.4	26.6	36.1	24.7	52.8	45.6	22.5	30.8	19.8	43.6

Table 5.1: Effect of the time encoding and the temporal self-attention in our space-time decoder on the VidSTG validation set.

	Pre-Training	Decoder Self-Attention Transfer	Declarative Sentences					Interrogative Sentences				
			m_tIoU	m_vIoU	vIoU @0.3	vIoU @0.5	m_sIoU	m_tIoU	m_vIoU	vIoU @0.3	vIoU @0.5	m_sIoU
1.	\times	\times	42.9	19.8	26.7	16.8	41.1	42.8	18.0	23.9	14.6	36.5
2.	\checkmark	\times	44.0	24.5	32.9	21.5	51.5	43.6	20.8	27.5	17.2	42.6
3.	\checkmark	\checkmark	46.4	26.6	36.1	24.7	52.8	45.6	22.5	30.8	19.8	43.6

Table 5.2: Effect of the weight initialization for our model on the VidSTG validation set.

information when $k > 1$.

5.4.2 Ablation studies

In this section, we ablate the hyper-parameters of our model and evaluate alternative design choices of the encoder and decoder. Unless stated otherwise, we use spatial frame resolution of 224 pixels and temporal stride $k = 5$.

Space-time decoder. We first ablate the design choices of the proposed space-time decoder. We compare our full decoder model with variants without time encoding, without temporal self-attention and without both. The variant without both corresponds to a space-only decoder, similar to MDETR [Kamath, 2021] applied independently to every frame. Table 5.1 shows that there is a substantial improvement over the space-only decoder when using both time encoding and temporal self-attention (+18.3% on $vIoU@0.3$ for declarative sentences and +17.5% on $vIoU@0.3$ for interrogative sentences between rows 1 and 4). The gain comes mostly from the temporal localization (+22.0% on m_tIoU for declarative sentences and +22.1% on m_tIoU for interrogative sentences), while the spatial grounding moderately increases (+0.9% in m_sIoU for declarative sentences and +0.5% in m_sIoU for interrogative sentences). Furthermore, we can observe that the time encoding brings most of the gain (+14.0% on $vIoU@0.3$ for declarative sentences and +12.9% on $vIoU@0.3$ for interrogatives sentences between rows 1 and 3). Finally, the temporal self-attention results in an additional improvement (+4.3% on $vIoU@0.3$ for declarative sentences and +4.6% on $vIoU@0.3$ for interrogative sentences between rows 3 and 4) over using time encoding only.

Fast	Res.	Temp. Stride	Declarative Sentences					Interrogative Sentences					Mem. (GB)	
			m_tIoU	m_vIoU	vIoU@0.3	vIoU@0.5	m_sIoU	m_tIoU	m_vIoU	vIoU@0.3	vIoU@0.5	m_sIoU		
1.	—	224	1	46.9	27.6	37.7	25.7	54.2	46.1	23.3	31.3	20.8	44.9	23.9
2.	✓	224	2	46.6	27.4	38.0	25.7	54.3	45.5	23.0	31.3	20.7	44.7	16.2
3.	✓	224	5	46.4	26.6	36.1	24.7	52.8	45.6	22.5	30.8	19.8	43.6	11.8
4.	✓	288	2	47.0	28.2	38.3	26.3	55.7	46.0	24.1	32.4	22.0	46.3	23.7
5.	✓	320	3	46.9	28.3	39.2	26.4	56.0	45.9	24.0	32.8	21.5	46.4	23.6
6.	✓	352	4	47.2	28.7	39.6	27.1	56.4	46.6	24.2	33.2	21.7	46.2	24.4
7.	✗	352	4	47.1	27.1	37.4	24.1	53.7	46.2	22.9	31.3	19.6	44.0	18.1
8.	✓	384	5	47.4	28.4	38.9	27.0	55.3	46.4	24.0	32.8	21.7	45.6	26.1

Table 5.3: Comparison of performance-memory trade-off with various temporal strides k , frame spatial resolutions (Res.), with or without the fast branch in our video-text encoder, on the VidSTG validation set.

	Fast	Res.	Temp. Stride	m_tIoU	m_vIoU	vIoU@0.3	vIoU@0.5	m_sIoU	Mem. (GB)
1.	—	224	1	52.8	35.0	55.3	28.3	63.9	14.3
2.	✓	224	2	53.7	35.8	56.7	29.6	64.3	10.2
3.	✓	224	5	53.2	35.0	54.5	29.0	63.2	8.0
4.	✓	288	2	53.9	36.4	58.1	30.7	65.4	13.9
5.	✓	320	3	53.6	36.2	57.5	30.4	65.2	13.8
6.	✓	352	4	53.9	36.4	58.8	30.6	64.9	14.3
7.	✗	352	4	53.1	34.7	55.9	27.4	63.0	11.3
8.	✓	384	5	53.6	36.3	57.5	30.4	65.3	15.2

Table 5.4: Comparison of performance-memory trade-off with various temporal strides k , spatial resolutions (Res.), with or without the fast branch in our video-text encoder, on the HC-STVG2.0 validation set.

Initialization. We now ablate the importance of initializing our model with pretrained MDETR [Kamath, 2021] weights. In Table 5.2, we compare this initialization to ImageNet initialization, and a variant that does not transfer the spatial self-attention weights from MDETR decoder to the temporal self-attention in our space-time decoder. At pretraining time, this self-attention was used to model spatial relationships between different objects in the same image, while the temporal self-attention in our decoder models temporal relationships between the same object in different frames of a video. We find that pretraining is highly beneficial (+9.4% on $vIoU@0.3$ for declarative sentences and +6.9% on $vIoU@0.3$ for interrogative sentences between rows 1 and 3), especially for the spatial grounding performance (+11.7% on m_sIoU for declarative sentences and +7.1% on m_sIoU for interrogative sentences). Additionally, we observe the benefit of using the spatial self-attention weights from the MDETR decoder to initialize the temporal self-attention in our decoder (+3.2% on $vIoU@0.3$ for declarative sentences and +3.3% on $vIoU@0.3$ for interrogative sentences between rows 2 and 3).

Impact of spatial resolution and temporal stride k . In this section, we analyze the impact of the frame resolution and the temporal stride k . In Tables 5.3 and 5.4, we show that increasing the resolution is an important factor of performance for spatio-temporal video

Slow	Spatial Pool.	f	g	Declarative Sentences					Interrogative Sentences				
				m_tIoU	m_vIoU	$vIoU@0.3$	$vIoU@0.5$	m_sIoU	m_tIoU	m_vIoU	$vIoU@0.3$	$vIoU@0.5$	m_sIoU
1. ✗	✗	Linear	Sum + Linear	42.7	18.6	25.0	14.8	39.6	42.5	16.9	22.0	12.9	35.1
2. ✓	-	0	0	46.2	24.9	34.4	21.8	49.7	45.1	20.9	28.3	17.9	40.5
3. ✓	✓	Linear	Sum + Linear	45.8	25.0	34.7	22.1	50.2	44.9	21.1	29.2	17.8	40.9
4. ✓	✗	Linear	Product + σ	46.2	26.2	36.0	23.9	52.0	45.4	22.1	30.1	18.8	43.0
5. ✓	✗	Transformer	Sum + Linear	46.4	26.4	36.4	23.8	52.8	45.3	22.2	30.2	19.6	43.3
6. ✓	✗	Linear	Sum + Linear	46.4	26.6	36.1	24.7	52.8	45.6	22.5	30.8	19.8	43.6

Table 5.5: Comparison of designs for the video-text encoder, with or without the slow branch, with or without spatial pooling in the fast branch, with variants of the fast module f and aggregation module g , on the VidSTG validation set.

grounding, on both the VidSTG and HC-STVG2.0 datasets (see rows 2 and 4). However, it also results in significantly higher memory usage (16.2GB vs 23.7GB). As a consequence, the variant using temporal stride $k = 1$ is challenging to train on VidSTG with a resolution higher than 224 on a Tesla V100 32GB GPU. At a fixed 224 resolution, increasing the temporal stride k to 2 or 5 reduces the peak memory usage by 7.7GB or 12.1GB, respectively (see row 1 vs 2 or 3, respectively). Our proposed video-text encoder enables us to train on higher resolutions at a given memory usage. This leads to a better performance-memory trade-off (rows 4, 5, 6, 8) than the baseline variant with temporal stride $k = 1$ (row 1). In particular, the best spatio-temporal video grounding results (m_vIoU and $vIoU@R$) over the two datasets are obtained with temporal stride $k = 4$ and resolution 352 (row 6).

We note that as the resolution increases, performance gains obtained by its further increase are expected to be lower as they are limited by the original video resolution. For instance, the average video pixel height in VidSTG and HCSTVG2.0 is 440 and 490 pixels, respectively.

Impact of the fast branch. Finally, we validate the importance of our fast branch by comparing, for the best variant, temporal stride $k = 4$ and resolution 352, our slow-fast video-text encoder to a slow-only variant that corresponds to $f = 0$ and $g = 0$. In this case the video-text features are the slow video-text features. By comparing rows 6 and 7 in Tables 5.3 and 5.4, our fast branch significantly improves the spatio-temporal video grounding performance (+2.2% $vIoU@0.3$ for declarative sentences on VidSTG, and +1.9% $vIoU@0.3$ for interrogative sentences on VidSTG, and +2.9% $vIoU@0.3$ on HC-STVG2.0) with low computational memory overhead. This shows that the fast branch recovers useful spatio-temporal details lost by the temporal sampling operation in the slow branch.

Design of the fast and aggregation modules. Here we further ablate the fast and aggregation modules f and g used in our dual-branch encoder. We report results in Table 5.5. The comparison between our slow-fast design (row 6) and the slow-only variant (row 2) is discussed in the previous paragraph. Likewise, we compare our slow-fast design to a fast-only variant (row 1). The fast-only variant does not use the slow multi-modal branch, in which case the video-text features are the fast visual-only features concatenated with the text features. As shown in Table 5.5, our slow-fast design outperforms the fast-only variant, showing the importance of the slow

Method	Pretraining Data	VidSTG								HC-STVG1		
		Declarative Sentences				Interrogative Sentences				m_vIoU	$vIoU@0.3$	$vIoU@0.5$
		m_tIoU	m_vIoU	$vIoU@0.3$	$vIoU@0.5$	m_tIoU	m_vIoU	$vIoU@0.3$	$vIoU@0.5$			
1. STGRN [Zhang, 2020d]	VG	48.5	19.8	25.8	14.6	47.0	18.3	21.1	12.8	—	—	—
2. STGVT [Tang, 2021]	VG + CC	—	21.6	29.8	18.9	—	—	—	—	18.2	26.8	9.5
3. STVGBert [Su, 2021]	IN + VG + CC	—	24.0	30.9	18.4	—	22.5	26.0	16.0	20.4	29.4	11.3
4. TubeDETR (Ours)	IN	43.1	22.0	29.7	18.1	42.3	19.6	26.1	14.9	21.2	31.6	12.2
5. TubeDETR (Ours)	IN + VG + F + C	48.1	30.4	42.5	28.2	46.9	25.7	35.7	23.2	32.4	49.8	23.5

Table 5.6: Comparison to the state of the art on the VidSTG test set and the HC-STVG1 test set. IN: ImageNet, VG: Visual Genome, CC: Conceptual Captions, F: Flickr, C: MS COCO.

multi-modal branch. We further compare the design of our fast and aggregation modules f and g (row 6) to other alternatives: row 3, a variant with the same primitives f and g but with f operating on features pooled over the spatial dimension; row 4, a variant which uses the same fast module f but a gating aggregation module $g(h_v(v, t), f(v)) = \sigma(h_v(v, t)) * f(v)$ where σ is the sigmoid function; row 5, a variant that uses the same aggregation module g but a fast temporal transformer module f , which models temporal interactions between spatially-detailed features. As shown in Table 5.5, our design outperforms row 3, showing that preserving spatial information for each frame is crucial for the effectiveness of the fast branch. Additionally, our design slightly improves over row 4, indicating that further forcing the network to use the slow branch is not helpful. Finally, our design slightly improves over row 5, suggesting that additional modeling of temporal interactions in our encoder is not necessarily helpful.

5.4.3 Comparison to the state of the art

In this section, we compare our approach to state-of-the-art methods in spatio-temporal video grounding. We report results for the model achieving the best validation results in the previous ablation studies, i.e. our space-time decoder with time encoding and temporal self-attention, temporal stride $k = 4$ and resolution 352. The focus of our work is on the spatio-temporal video grounding metrics (m_vIoU and $vIoU@R$). As shown in Table 5.6, only using ImageNet to initialize the visual backbone (row 4), our TubeDETR performs competitively despite using less annotations. Furthermore, if we use MDETR initialization (row 5), our TubeDETR outperforms by a large margin all previous methods (rows 1, 2 and 3) on both datasets. STGRN [Zhang, 2020d] achieves similar m_tIoU (measuring only temporal localization), but it defines a hand-crafted set of possible window widths to tackle temporal localization, while we consider all possible windows, i.e. any starting frame i and ending frame j with $i < j$. These results demonstrate the excellent performance of our architecture for spatio-temporal video grounding.

5.4.4 Qualitative examples

We show qualitative examples of our predictions on the VidSTG test set in Figure 5.5. These examples show that our model is able to predict meaningful and accurate spatio-temporal tubes associated with the input text queries. In particular, in the first example, our model correctly



Figure 5.5: Qualitative examples of spatio-temporal tubes predicted by our model (light yellow), compared with ground truth (light green), on the VidSTG test set. The first three examples illustrate successful predictions of our method. In the last example the method confuses the small sports ball in the background with a balloon. We show more examples on our webpage [Yang, 2022b].

detects the temporal moment corresponding to the cat biting the adult. In the second example, our model localizes the spatio-temporal tube corresponding to a man quickly grabbing a very small sports ball and in the third example it is able to localize the skis under the adult while skiing. However, as shown in the last example, it may fail to understand fine details in the query and the video. Note that the balloon and the ball are visually and semantically similar. A careful analysis is required to understand the difference.

5.4.5 Visualization of space, time and language attention patterns in the decoder

This section illustrates attention mechanisms of our space-time decoder over space, language and time for the spatio-temporal video grounding example presented in Figure 5.7. For this example the time-aligned cross-attention for the visual modality is also shown in Figure 5.7. We note that spatially, attention at each timestep is particularly focused on humans that are receiving the sports ball and gesturing. Additionally, the time-aligned cross-attention for the textual modality is illustrated in Figure 5.6. We observe that the words *adult* and *grabs* are

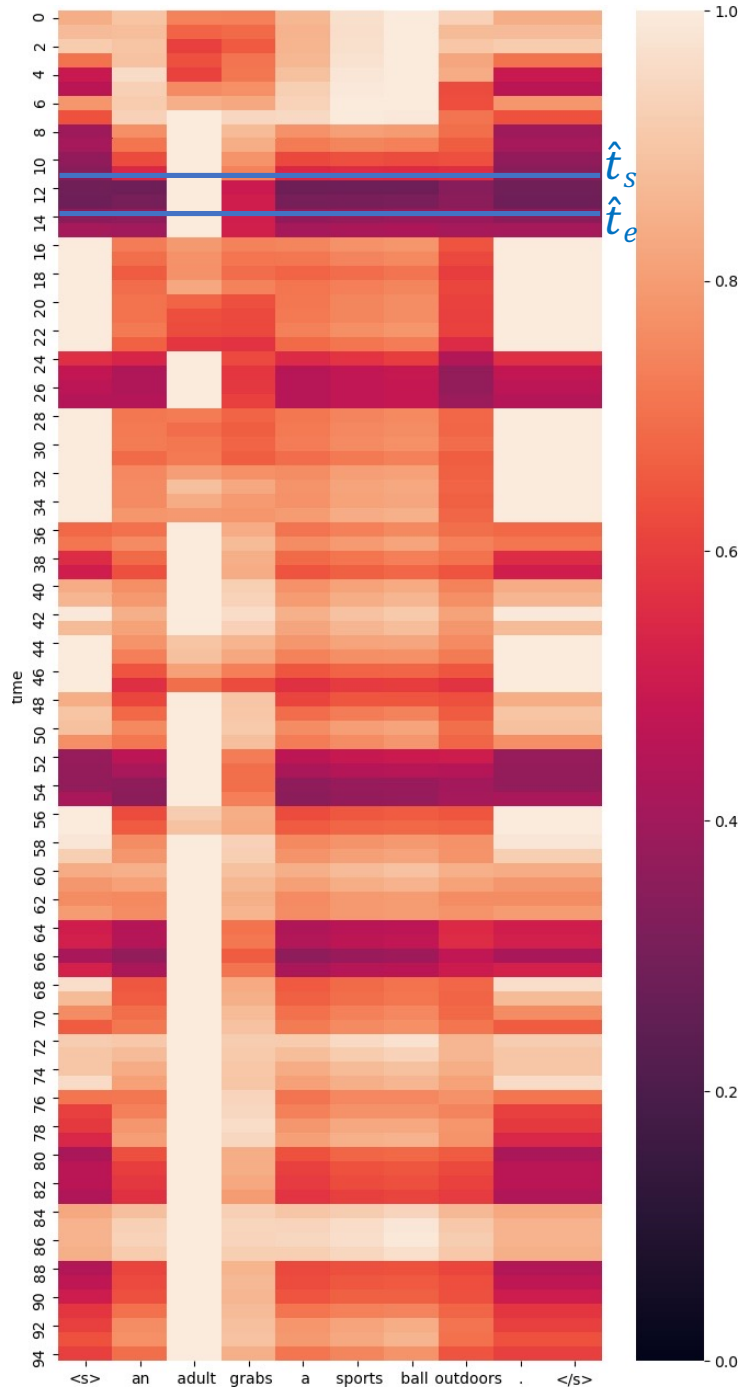


Figure 5.6: **Time-aligned cross-attention visualization (textual modality)**. Visualization of the attention weights between the time query (y -axis) and its time-aligned visually-contextualized text features (x -axis) at different times in our space-time decoder. These attention weights are averaged across all 8 heads and all 6 layers, and renormalized by the maximum weight at each timestep (i.e. each row) for the purpose of visualization. Lighter colors correspond to higher attention weights (see the colorbar on the right).

the most attended overall, and that attention weights on the different words (e.g. *sports* and *ball*) vary over time. \hat{t}_s and \hat{t}_e in Figure 5.6 denote the predicted start and end times of the

Query: An adult grabs a sports ball outdoors.

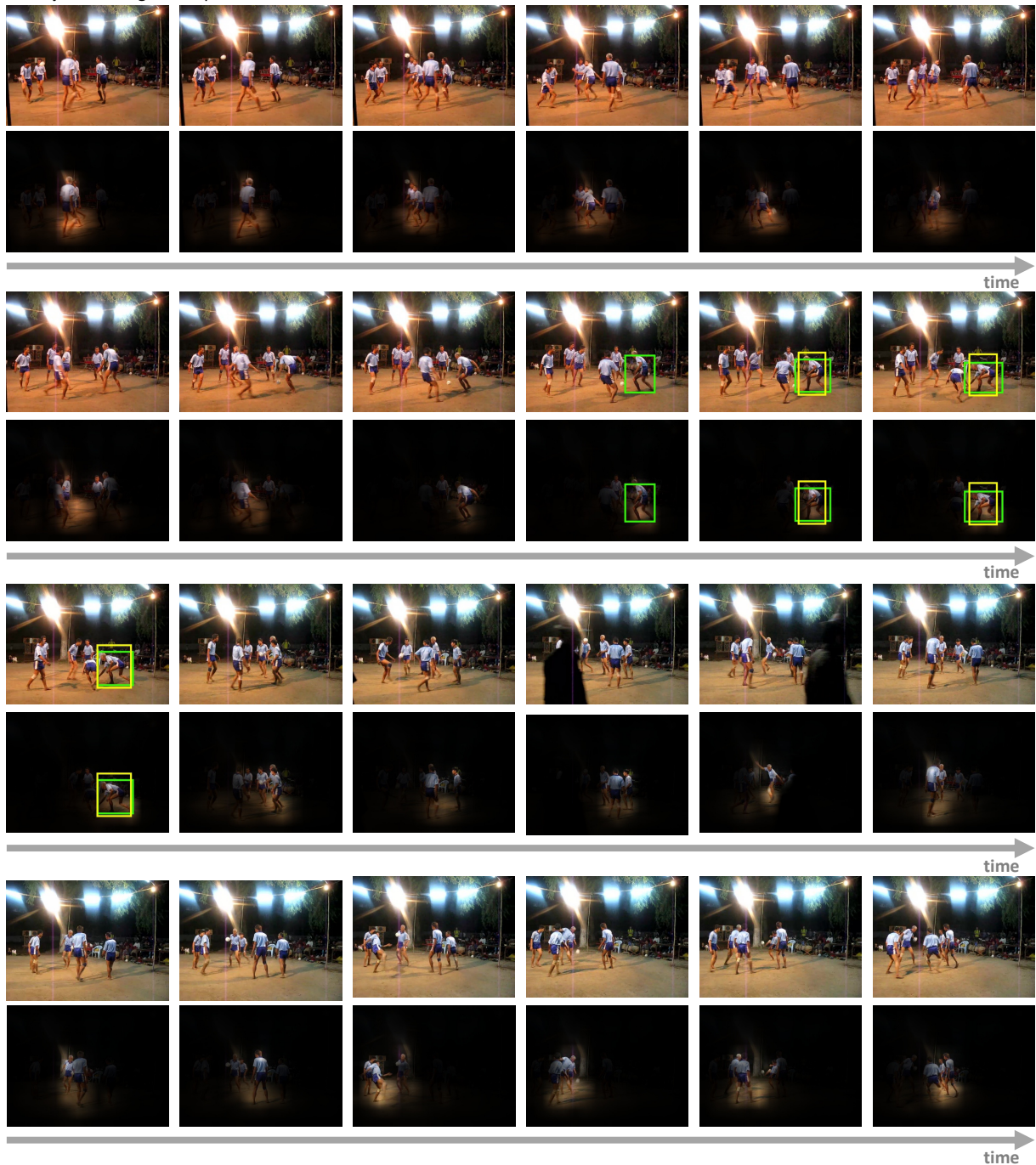


Figure 5.7: **Time-aligned cross-attention visualization (visual modality)**. Top rows: Input frames with the predicted (yellow) and ground truth (green) spatio-temporal tubes overlaid. Bottom rows: Visualization of the attention weights between the time query and its time-aligned text-contextualized visual features at different times in our space-time decoder. These attention weights are averaged across all 8 heads and all 6 layers, and renormalized by the maximum weight at each timestep for the purpose of visualization. Attention at each timestep is particularly focused on humans that are receiving the sports ball and gesturing.

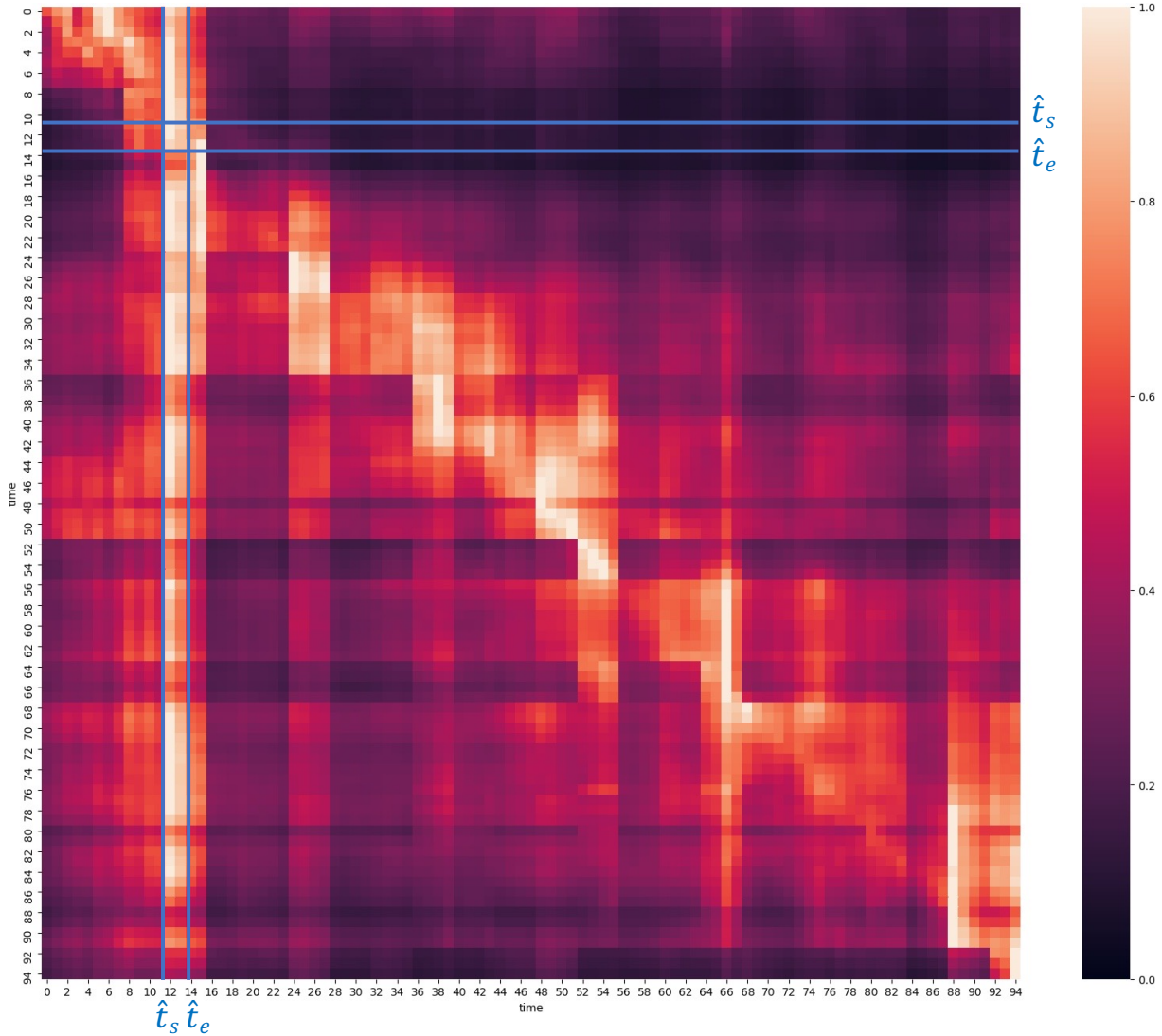


Figure 5.8: **Temporal self-attention visualization.** Visualization of the attention weights between the different time queries in our space-time decoder. The column t corresponds to the weights of the different time queries for the time query at time t . These attention weights are averaged across all 8 heads and all 6 layers, and renormalized by the maximum weight at each timestep (i.e. each column) for the purpose of visualization. \hat{t}_s and \hat{t}_e denote the predicted start and end times of the output tube. Lighter colors correspond to higher attention weights (see the colorbar on the right).

output tube. Next, the temporal self-attention is illustrated in Figure 5.8. We notice long-range temporal interactions: a certain number of time queries attend to various temporally distant time queries, e.g. time queries located around the start of the video between the eighth and sixteenth frames.

5.5 Conclusion

We propose TubeDETR, a novel transformer-based architecture for spatio-temporal video grounding. TubeDETR tackles this task with a space-time transformer decoder combined with a video-text encoder that efficiently encodes spatial and multi-modal interactions. We demonstrate the effectiveness of our space-time decoder, and the benefits of our video-text encoder in terms of performance-memory trade-off. Finally, our approach outperforms prior state-of-the-art methods on two benchmarks, VidSTG and HC-STVG.

Limitations. Our architecture is limited to detecting a single spatio-temporal tube for a given natural language query. Therefore future work could extend our space-time decoder to detect multiple objects per frame or multiple events per video. Moreover, TubeDETR is built using vanilla self-attention which has quadratic complexity with respect to the number of input tokens. Hence investigating more efficient alternatives to self-attention, such as the ones studied for natural language [Beltagy, 2020; Choromanski, 2021; Kitaev, 2020; Tay, 2021; Wang, 2020b; Wu, 2020; Zaheer, 2020], is another promising direction for future research in developing efficient end-to-end video-language models. Finally, this work focuses on the spatio-temporal video grounding task itself. It would be interesting to study if such systems can be integrated into video question answering systems to improve them.

Chapter 6

Vid2Seq: Large-Scale Pretraining of a Visual Language Model for Dense Video Captioning

In this chapter, we introduce Vid2Seq, a multi-modal single-stage dense event captioning model pretrained on narrated videos which are readily-available at scale. The Vid2Seq architecture augments a language model with special time tokens, allowing it to seamlessly predict event boundaries and textual descriptions in the same output sequence (see Figure 6.1). Such a unified model requires large-scale training data, which is not available in current annotated datasets. We show that it is possible to leverage unlabeled narrated videos for dense video captioning, by reformulating sentence boundaries of transcribed speech as pseudo event boundaries, and using the transcribed speech sentences as pseudo event captions. The resulting Vid2Seq model pretrained on the YT-Temporal-1B dataset [Zellers, 2022] improves over the prior state of the art on a variety of dense video captioning benchmarks including YouCook2 [Zhou, 2018a], ViTT [Huang, 2020b] and ActivityNet Captions [Krishna, 2017]. Vid2Seq also generalizes well to the tasks of video paragraph captioning and video clip captioning, and to few-shot settings. Our code and models are publicly available at [Yang, 2023a].

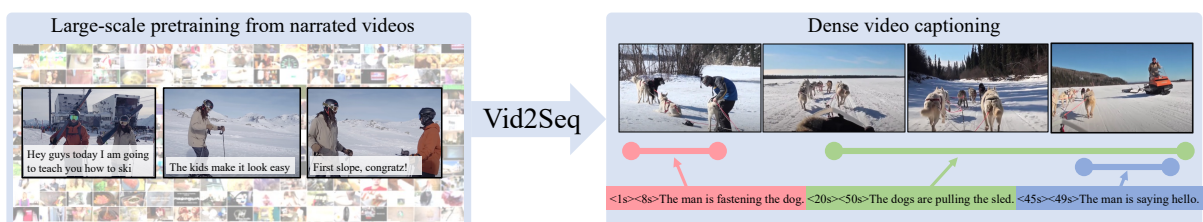


Figure 6.1: **Vid2Seq** is a visual language model that predicts dense event captions together with their temporal grounding in the video by generating a *single* sequence of tokens (right). This ability is enabled by large-scale pretraining on unlabeled narrated videos (left).

6.1 Introduction

Dense video captioning requires the temporal localization and captioning of all events in an untrimmed video [Krishna, 2017; Wang, 2021d; Zhou, 2018c]. This differs from standard video captioning [Lin, 2022b; Luo, 2020b; Seo, 2022], where the goal is to produce a single caption for a given short video clip. Dense captioning is significantly more difficult, as it raises the additional complexity of localizing the events in minutes-long videos. However, it also benefits from long-range video information. This task is potentially highly useful in applications such as large-scale video search and indexing, where the video content is not segmented into clips.

Existing methods mostly resort to two-stage approaches [Krishna, 2017; Wang, 2018b; Iashin, 2020a], where events are first localized and then captioned. To further enhance the inter-task interaction between event localization and captioning, some approaches have introduced models that jointly solve the two tasks [Deng, 2021a; Wang, 2021d; Zhou, 2018c]. However, often these approaches still require task-specific components such as event counters [Wang, 2021d]. Furthermore, they exclusively train on manually annotated datasets of limited size [Huang, 2020b; Krishna, 2017; Zhou, 2018b], which makes it difficult to effectively solve the task. To address these issues, we take inspiration from recent sequence-to-sequence models pretrained on Web data which have been successful on a wide range of vision and language tasks [Chen, 2022a; Yang, 2021d; Chen, 2023b; Alayrac, 2022; Wang, 2022f].

First, we propose a video language model, called Vid2Seq. We start from a language model trained on Web text [Raffel, 2020] and augment it with special *time tokens* that represent timestamps in the video. Given video frames and transcribed speech inputs, the resulting model jointly predicts all event captions and their corresponding temporal boundaries by generating a *single* sequence of discrete tokens, as illustrated in Figure 6.1 (right). Such a model therefore has the potential to learn multi-modal dependencies between the different events in the video via attention [Vaswani, 2017]. However this requires large-scale training data, which is not available in current dense video captioning datasets [Huang, 2020b; Krishna, 2017; Zhou, 2018b]. Moreover, collecting manual annotations of dense captions for videos is expensive and prohibitive at scale.

Hence we propose to pretrain Vid2Seq by leveraging unlabeled narrated videos which are readily-available at scale. To do this, we reformulate sentence boundaries of transcribed speech as pseudo event boundaries, and use the transcribed speech sentences as pseudo event captions. We then pretrain Vid2Seq with a generative objective, that requires predicting the transcribed speech given visual inputs, and a denoising objective, which masks spans of transcribed speech. Note that transcribed speech may not describe the video content faithfully, and is often temporally misaligned with the visual stream [Han, 2022; Ko, 2022; Miech, 2020]. For instance, from the example in Figure 6.1 (left), one can understand that the grey skier has descended a slope from the last speech sentence which is said *after* he actually descended the slope. Intuitively, Vid2Seq is particularly suited for learning from such noisy supervision as it jointly models *all* narrations and the corresponding timestamps in the video.

We demonstrate the effectiveness of our pretrained model through extensive experiments. We

show the importance of pretraining on untrimmed narrated videos, the ability of Vid2Seq to use both the visual and speech modalities, the importance of the pretraining objectives, the benefit of joint caption generation and localization, as well as the importance of the language model size and the scale of the pretraining dataset. The pretrained Vid2Seq model achieves state-of-the-art performance on various dense video captioning benchmarks [Huang, 2020b; Krishna, 2017; Zhou, 2018b]. Our model also excels at generating paragraphs of text describing the video: without using ground-truth event proposals at inference time, our model outperforms all prior approaches including those that rely on such proposals [Lei, 2020a; Zhou, 2019; Park, 2019]. Moreover, Vid2Seq generalizes well to the standard task of video clip captioning [Chen, 2011; Xu, 2016b]. Finally, we introduce a new few-shot dense video captioning setting in which we finetune our pretrained model on a small fraction of the downstream training dataset and show benefits of Vid2Seq in this setting.

In summary, we make the following contributions:

- (i) We introduce Vid2Seq for dense video captioning. Given multi-modal inputs (transcribed speech and video), Vid2Seq predicts a single sequence of discrete tokens that includes caption tokens interleaved with special *time tokens* that represent event timestamps.
- (ii) We show that transcribed speech and corresponding timestamps in unlabeled narrated videos can be effectively used as a source of weak supervision for dense video captioning.
- (iii) Finally, our pretrained Vid2Seq model improves over the prior state of the art on three dense video captioning datasets (YouCook2, ViTT, ActivityNet Captions), two video paragraph captioning benchmarks (YouCook2, ActivityNet Captions) and two video clip captioning datasets (MSR-VTT, MSVD), and also generalizes well to few-shot settings.

6.2 Related Work

Dense video captioning. Dense video captioning lies at the intersection of event localization [Heilbron, 2016; Escorcia, 2016; Gao, 2017b; Lin, 2018; Lin, 2019; Lin, 2020a; Shou, 2016; Zhao, 2017a] and event captioning [Gao, 2017c; Lin, 2022b; Pan, 2017; Wang, 2018c; Wang, 2018a]. The majority of existing methods for dense video captioning [Krishna, 2017; Iashin, 2020a; Iashin, 2020b; Wang, 2018b; Wang, 2020c] consist of a temporal localization stage followed by an event captioning stage. To enrich inter-task interactions, recent works [Chadha, 2021; Chen, 2021b; Deng, 2021a; Li, 2018a; Mun, 2019; Rahman, 2019; Shen, 2017; Shi, 2019a; Wang, 2018b; Wang, 2021d; Zhou, 2018c] jointly train the captioning and localization modules. In particular, [Wang, 2021d] propose to view dense video captioning as a set prediction task, and jointly perform event localization and captioning for each event in parallel. In contrast, our model generates event boundaries and captions conditioned on the previously generated events. [Deng, 2021a] propose to first generate a paragraph and then ground each sentence in the video. We also generate all captions as a single output sequence, however our output already includes event timestamps. [Zhang, 2022c] propose to generate event boundaries sequentially, but separately perform event localization and single event captioning, and only use visual input. Most

related to our work, [Zhu, 2022b] also perform dense video captioning by generating a single output sequence. Their method, however, infers event locations directly from the timestamps of transcribed speech and, hence, can only detect events that closely follow the speech. In contrast, our model generates event timestamps as special tokens and can produce dense captions for videos with limited speech, as we demonstrate on the ActivityNet Captions dataset.

Video and language pretraining. Following the success of image-text pretraining [Singh, 2022; Yu, 2022a; Hu, 2022b; Chen, 2020b; Dou, 2022b; Dou, 2022a; Gan, 2020; Huang, 2021b; Jia, 2021; Kamath, 2021; Kim, 2021b; Li, 2020a; Li, 2020d; Li, 2021a; Li, 2022c; Li, 2022d; Lu, 2019; Lu, 2020; Su, 2019; Tan, 2019; Tsimpoukelli, 2021; Desai, 2021b; Wang, 2021b; Yu, 2020; Yuan, 2021; Zhang, 2022b; Zhou, 2020], recent works have explored video-text pretraining [Wang, 2022b; Akbari, 2021; Alayrac, 2022; Bain, 2021; Fu, 2021; Ge, 2022; Han, 2022; Ko, 2022; Lei, 2021b; Li, 2020b; Li, 2022a; Miech, 2019; Miech, 2020; Nagrani, 2022; Seo, 2021b; Seo, 2022; Sun, 2019b; Xue, 2022; Xu, 2021; Wang, 2023; Yang, 2021b; Yang, 2022c; Yang, 2022e; Yang, 2022d; Zellers, 2021; Zellers, 2022; Xue, 2022]. These methods show strong improvements on various tasks such as text-video retrieval [Bain, 2021; Miech, 2020], video question answering [Yang, 2021b; Zellers, 2021] and video clip captioning [Alayrac, 2022; Seo, 2022]. While these works mostly learn global video representations to tackle video-level prediction tasks, we here focus on learning detailed representations to address a dense prediction task requiring reasoning over multiple events in untrimmed videos. Several works have explored long-form video-text pretraining [Sun, 2022] and video-text pretraining for temporal localization tasks [Cao, 2022a; Lei, 2021a; Lin, 2022a; Wang, 2022g; Xu, 2022; Yang, 2021c]. However these works focus on video understanding tasks while our pretraining approach is tailored for a generative task that not only requires the model to reason over multiple events in the video, but also to describe them by natural language.

A few works explore pretraining for dense video captioning. [Zhang, 2022c] pretrain on ActivityNet Captions to improve the downstream performance on the same dataset. In contrast, we propose a pretraining method that does not rely on *any* manual annotation, and show its benefits on multiple downstream datasets. [Huang, 2020b] explore pretraining on narrated instructional videos, but only consider event captioning using ground truth proposals as their model does not handle localization. Finally, [Huang, 2020b; Zhu, 2022b] explore pretraining on a domain specific text-only dataset [Koupaei, 2018]. In contrast, we propose to pretrain on a generic video corpus [Zellers, 2022] and show benefits on various domains.

Unifying tasks as language modeling. Recent works [Chen, 2022a; Chen, 2022b; Chen, 2022c; Chen, 2023b; Cho, 2021; Kolesnikov, 2022; Li, 2022e; Wang, 2022c; Yang, 2021d; Zhu, 2022a] have shown that it is possible to cast various computer vision problems as a language modeling task, addressing object detection [Chen, 2022a], grounded image captioning [Yang, 2021d] or visual grounding [Zhu, 2022a]. In this work we also cast visual localization as a language modeling task. However, unlike prior work focused on image-level spatial localization, we address the different problem of event localization *in time*, in untrimmed videos.

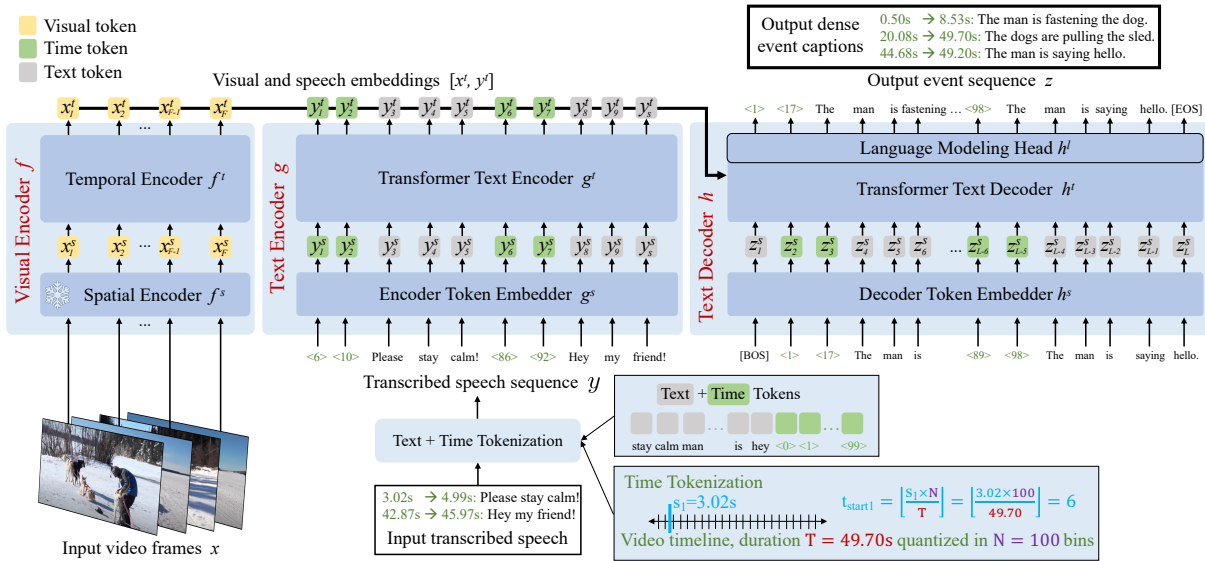


Figure 6.2: **Vid2Seq model overview.** We formulate dense event captioning as a sequence-to-sequence problem, using special *time tokens* to allow the model to seamlessly understand and generate sequences of tokens containing both textual semantic information and temporal localization information grounding each text sentence in the video. In detail, all input video frames x and the transcribed speech sequence y are first processed with a Visual Encoder f (a frozen Spatial Encoder f^s followed by a Temporal Encoder f^t) and a Text Encoder g (a Token Embedder g^s followed by a Transformer Encoder g^t), respectively. Then the Text Decoder h (composed of a Token Embedder h^s , a Transformer Encoder h^t and a Language Modeling Head h^l) autoregressively generates the output event sequence z by cross-attending to the visual and speech embeddings x^t and y^t .

6.3 Method

The goal of dense video captioning is to temporally localize and describe with natural language *all* events in an untrimmed input video. Therefore a key challenge is to effectively model the relationships between the different events in the video, as for example, it is easier to predict that the dogs are pulling the sled if we know that the man has just fastened a dog (see Figure 6.1 (right)). Furthermore, due to the dense nature of the task, there can be many events in a long video and the requirement is to output a natural language caption for each event. Hence, another key challenge is that the manual collection of annotations for this task is particularly expensive. To tackle these challenges, we first develop a unified multi-modal model that jointly predicts event boundaries and captions as a single sequence of tokens, as explained in Section 6.3.1 and Figure 6.2. Second, we design a pretraining strategy that effectively leverages cross-modal supervision in the form of transcribed speech from unlabeled narrated videos by reformulating sentence boundaries as pseudo event boundaries, as presented in Section 6.3.2 and Figure 6.3.

6.3.1 Model

We wish to design a model for dense video captioning that can capture relationships between events using visual and (transcribed) speech cues in order to effectively localize and describe

these events in untrimmed minutes-long videos. To tackle this challenge, we cast dense video captioning as a sequence-to-sequence problem where the input and output sequences contain both the semantic information about the event in the form of natural language descriptions and the temporal localization of the events in the form of temporal timestamps. In addition, to best leverage both the visual and the language signal, we develop an appropriate multi-modal encoder-decoder architecture. As illustrated in Figure 6.2, our architecture takes as input video frames $x = \{x_i\}_{i=1}^F$ together with the transcribed speech sequence $y = \{y_j\}_{j=1}^S$. The output of our model is an event sequence $z = \{z_k\}_{k=1}^L$, where each event contains both its textual description and timestamps corresponding to the temporal event locations in the video. Below we explain the structure of the transcribed speech and event sequences constructed for our model as well as details of our model architecture.

6.3.1.1 Sequence construction.

To model inter-event relationships in dense event captioning annotations (or the readily-available transcribed narration, see Section 6.3.2), we cast dense video captioning as predicting a single output sequence of tokens z . This output event sequence is constructed by leveraging a text tokenizer augmented with special *time tokens*. Furthermore, we enable our architecture to jointly reason about the semantic and temporal information provided in the transcript of the input narration by constructing the input transcript sequence y in a similar manner as the event sequence z . Details are given next.

Time tokenization. We start from a text tokenizer with a vocabulary size V , and augment it with N additional time tokens, resulting in a tokenizer with $V + N$ tokens. The time tokens represent relative timestamps in a video, as we quantize a video of duration T into N equally-spaced timestamps. In detail, we use the SentencePiece tokenizer [Kudo, 2018] with vocabulary size $V = 32,128$ and $N = 100$.

Event sequence. Our introduced tokenizer enables us to construct sequences that contain both video timestamps and text video descriptions. We next explain how we construct the output event sequence z . Note that videos have a variable number of events in standard dense video captioning datasets [Huang, 2020b; Krishna, 2017; Zhou, 2018b]. Each event k is characterized by a text segment, a start time and an end time. We first construct for each event k a sequence by concatenating its start time token t_{start_k} , its end time token t_{end_k} and its text tokens $[z_{k_1}, \dots, z_{k_{l_k}}]$. Then we order all these sequences in increasing order of their start times and concatenate them. In practice, each text segment ends with a dot symbol indicating the separation between different events. Finally, the event sequence is obtained by prepending and appending a BOS and an EOS tokens to indicate the start and the end of sequence, respectively, *i.e.*, $z = [BOS, t_{start_1}, t_{end_1}, z_{1_1}, \dots, z_{1_{l_1}}, t_{start_2}, \dots, EOS]$.

¹<https://cloud.google.com/speech-to-text/docs/automatic-punctuation>.

Transcribed speech sequence. To enable the model to use both the transcribed speech and its corresponding timestamps, we convert the speech transcript into a speech sequence y similarly as the input training dense event captions z . This is done by segmenting the raw speech transcript into sentences with the Google Cloud API¹, and using each transcribed speech sentence with its corresponding timestamps analogously as an event in the previously explained process.

6.3.1.2 Architecture.

We wish to design an architecture that can effectively model relationships between different events in untrimmed minutes-long videos. To tackle this challenge, we propose a multi-modal encoder-decoder architecture, illustrated in Figure 6.2, that gradually refines and outputs the event sequence described above. In detail, given an untrimmed minutes-long video, the visual encoder f embeds its frames while the text encoder g embeds transcribed speech and the corresponding timestamps. Then a text decoder h predicts event boundaries and text captions using the visual and transcribed speech embeddings. The individual modules are described next.

Visual encoder. The visual encoder operates on a sequence of F frames $x \in \mathbb{R}^{F \times H \times W \times C}$ where H , W and C are the height, width and the number of channels of each frame. A visual backbone f^s first encodes each frame separately and outputs frame embeddings $x^s = f^s(x) \in \mathbb{R}^{F \times d}$, where d is the embedding dimension. Then a transformer encoder [Vaswani, 2017] f^t models temporal interactions between the different frames, and outputs F contextualized visual embeddings $x^t = f^t(x^s + x^p) \in \mathbb{R}^{F \times d}$, where $x^p \in \mathbb{R}^{F \times d}$ are learnt temporal positional embeddings, which communicate time information from visual inputs to the model. In detail, the visual backbone is CLIP ViT-L/14 [Dosovitskiy, 2021; Radford, 2021] at resolution 224×224 pixels, pretrained to map images to text descriptions with a contrastive loss on Web-scraped image-text pairs. We keep the backbone frozen for efficiency.

Text encoder. The text encoder operates on a transcribed speech sequence of S tokens $y \in \{1, \dots, V+N\}^S$, where V is the text vocabulary size, N is the size of the vocabulary of time tokens and S is the number of tokens in the transcribed speech sequence. Note that the transcribed speech sequence includes time tokens to input the temporal information from the transcribed speech into the model. An embedding layer $g^s \in \mathbb{R}^{(V+N) \times d}$ embeds each token independently and outputs semantic embeddings $y^s = g^s(y) \in \mathbb{R}^{S \times d}$. Then a transformer encoder g^t computes interactions in the transcribed speech sequence and outputs S contextualized speech embeddings $y^t = g^t(y^s) \in \mathbb{R}^{S \times d}$.

Text decoder. The text decoder generates the event sequence z by using the encoder embeddings, which are obtained by concatenating the visual and speech embeddings x^t and y^t . The text decoder is based on a causal transformer decoder h^t that cross-attends to the encoder outputs, and at each autoregressive step k , self-attends to the previously generated tokens $\hat{z}_{<k}^t$ to output a contextualized representation $z_k^t = h^t(h^s(\hat{z}_{<k}^t), x^t, y^t) \in \mathbb{R}^d$ where $h^s \in \mathbb{R}^{(V+N) \times d}$ is

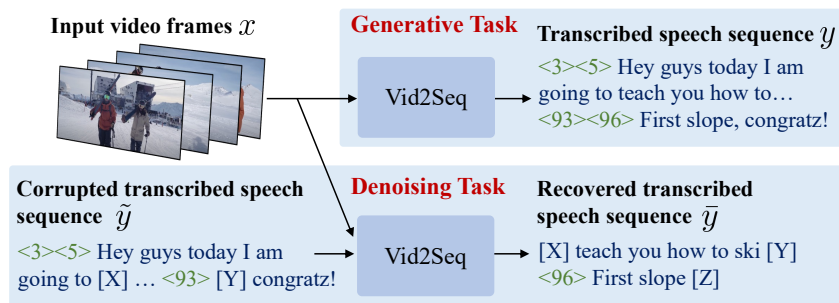


Figure 6.3: **Pretraining tasks.** To train Vid2Seq on unlabeled narrated videos, we design two pretraining objectives. **Top:** generative objective, given visual inputs x only, the task is to generate the transcribed speech sequence y . **Bottom:** denoising objective, given visual inputs x and the corrupted speech sequence \tilde{y} , the task is to generate the sequence of recovered speech segments \bar{y} .

the decoder token embedding layer. Then a language modeling head $h^l \in \mathbb{R}^{d \times (V+N)}$ predicts a probability distribution over the joint vocabulary of text and time tokens in order to predict the next token in the event sequence, *i.e.*, $z_k^l = h^l(z_k^t) \in \mathbb{R}^{V+N}$.

Text initialization. We initialize the text encoder and the text decoder with T5-Base [Raffel, 2020] which has been pretrained on Web text corpora with a denoising loss. Therefore their implementation and parameters also closely follow T5-Base, *e.g.* they use relative positional embeddings and share their token embedding layer $g^s = h^s \in \mathbb{R}^{(V+N) \times d}$.

6.3.2 Training

In this Section, we describe how we leverage a large amount of unlabeled narrated videos to train the previously described dense event captioning model. We first present the pretraining method used to effectively train Vid2Seq using cross-modal supervision in readily-available narrated videos in Section 6.3.2.1 and Figure 6.3. Then we explain how we finetune our architecture for various downstream tasks including dense event captioning in Section 6.3.2.2.

6.3.2.1 Pretraining on untrimmed narrated videos

We wish to leverage narrated videos for pretraining as they are easily available at scale [Miech, 2019; Zellers, 2022]. However these videos do not contain dense event captioning annotations. Therefore we use as supervisory signal the transcribed speech sentences and their corresponding timestamps. As speech transcripts are not always visually grounded and often temporally misaligned [Han, 2022; Ko, 2022; Miech, 2020], we note that they only provide *weak* supervision. Furthermore, speech transcripts drastically differ from dense event captioning annotations. For instance, in the YT-Temporal-1B dataset [Zellers, 2022], a video contains 120 speech sentences on average which is an order of magnitude more than the number of events in standard dense video captioning datasets [Zhou, 2018b; Huang, 2020b; Krishna, 2017]. Our Vid2Seq model is particularly suitable for using such weak supervision as it constructs the speech sequence

similarly as a manually annotated event sequence, and jointly contextualizes the speech boundaries and semantic information on the level of potentially minutes-long videos (see Section 6.3.1) rather than at a shorter clip-level, enabling our model to learn long-term relationships between the different speech segments: in experiments we show that pretraining on entire minutes-long videos is highly beneficial.

We next describe the two proposed training objectives, which are both based on a maximum likelihood objective. Formally, given visual inputs x , encoder text sequence y and a decoder target text sequence z , both objectives are based on minimizing the following loss:

$$\mathcal{L}_\theta(x, y, z) = -\frac{1}{\sum_{k=1}^{L-1} w_k} \sum_{k=1}^{L-1} w_k \log p_\theta(z_{k+1}|x, y, z_{1:k}), \quad (6.1)$$

where L is the length of the decoder target sequence, w_k is the weight for k -th token in the sequence, which we set to $w_k = 1 \forall k$ in practice, θ denotes the trainable parameters in the model and p_θ is the output probability distribution over the vocabulary of text and time tokens.

Generative objective. This objective uses the transcribed speech as a (pseudo-)supervisory signal to teach the decoder to predict a sequence of events given visual inputs. Given video frames x , which are fed to the encoder, the decoder has to predict the transcribed speech sequence y (see Figure 6.3), which serves as a proxy dense event captioning annotation. Note that no text input is given to the encoder for this task as using transcribed speech both as input and target would lead the model to learn text-only shortcuts.

Denoising objective. As no text input is given to the encoder for the generative proxy task, the generative objective only trains the visual encoder and the text decoder, but not the text encoder. However when our model is used for dense video captioning, the text encoder has a significant importance as it encodes speech transcripts. Hence we introduce a denoising objective that aims at jointly aligning the visual encoder, the text encoder and the text decoder. Inspired by T5 [Raffel, 2020] in the text domain, we randomly mask spans of (text and time) tokens in the transcribed speech sequence with a probability P and an average span length M . The encoder input is composed of the video frames x together with the corrupted speech sequence \tilde{y} , which contains sentinel tokens that uniquely identify the masked spans. The decoder then has to predict a sequence \bar{y} constructed with the corresponding masked spans for each sentinel token, based on visual inputs x and speech context \tilde{y} (see Figure 6.3).

6.3.2.2 Downstream task adaptation

Our architecture and task formulation enables us to tackle dense video captioning with a generic language modeling training objective and inference procedure. Note that as a by-product of our generic architecture, our model can also be used to generate paragraphs about entire videos by simply removing the time tokens from the output sequence, and can also be easily adapted to video clip captioning with the same finetuning and inference recipe.

Finetuning. To finetune our model for dense video captioning, we use a maximum likelihood objective based on the event sequence (see Equation 6.1). Given video frames x and speech transcripts y , the decoder has to predict the event sequence z .

Inference. The text decoder autoregressively generates the event sequence by sampling from the model likelihood. In practice, we use beam search as we find that it improves the captioning quality compared with argmax sampling or nucleus sampling. Finally, the event sequence is converted into a set of event predictions by simply reversing the sequence construction process.

6.4 Experiments

This section demonstrates the effectiveness of our pretrained Vid2Seq model and compares our method to the state of the art. We first outline our experimental setup in Section 6.4.1. We then present ablation studies in Section 6.4.2. The comparison to the state of the art in dense video captioning, video paragraph captioning and video clip captioning is presented in Section 6.4.3. Next, we present results in a new few-shot dense video captioning setting in Section 6.4.4. Finally, we show qualitative results in Section 6.4.5.

6.4.1 Experimental setup

Datasets. For pretraining, following prior work showing the benefits of pretraining on a diverse and large dataset [Zellers, 2021], we use the **YT-Temporal-1B** dataset [Zellers, 2022], which includes 18 million narrated videos collected from YouTube. We evaluate Vid2Seq on three downstream dense video captioning datasets: YouCook2 [Zhou, 2018b], ViTT [Huang, 2020b] and ActivityNet Captions [Krishna, 2017]. For video clip captioning, we use two standard benchmarks, **MSR-VTT** [Xu, 2016b] and **MSVD** [Chen, 2011]. For all datasets, we follow the standard splits for training, validation and testing. Note that we only use videos available on YouTube at the time of the work, resulting in 10 to 20% less videos than in the original datasets. We describe below the downstream datasets in more detail.

YouCook2 has 2K untrimmed videos of cooking procedures. On average, each video lasts 320s and is annotated with 7.7 temporally-localized sentences. The dataset is split into 1,333 videos for training and 457 videos for validation.

ViTT consists of 8K untrimmed instructional videos. On average, each video lasts 250s and is annotated with 7.1 temporally-localized short tags. The dataset is split into 5,476, 1,102 and 1,094 videos for training, validation and testing, respectively. Videos in the validation and test sets are provided with multiple sets of dense event captioning annotations. Following [Huang, 2020b], we treat each set of annotations as a single example during evaluation and discard videos with more than 3 sets of annotations.

ActivityNet Captions contains 14,934 untrimmed videos of various human activities. On average, each video lasts 120s and is annotated with 3.7 temporally-localized sentences. Different from YouCook2 and ViTT where most videos contain transcribed speech content, we find that 68% of videos in ActivityNet Captions do not have transcribed narration. On average, each

video lasts 120s and is annotated with 3.7 temporally-localized sentences. The dataset is split into 10,009 and 4,925 videos for training and validation, respectively. Videos in the validation set are provided with two sets of dense video captioning annotations. Following prior work [Wang, 2021d], we use both sets of annotations for evaluation, by computing the average of the scores over each set for SODA_c and by using the standard evaluation tool [Krishna, 2017] for all other dense event captioning metrics. For video paragraph captioning, we follow [Wang, 2021d] and report results on the 'val-ae' split that includes 2,460 videos [Zhou, 2019; Lei, 2020a].

MSR-VTT [Xu, 2016b] consists of 10,000 open domain video clips. The duration of each video clip is between 10 and 30 seconds. 20 natural language descriptions are manually annotated for each clip. The dataset is split into 6,513, 497 and 2,990 videos for training, validation and testing, respectively.

MSVD [Chen, 2011] consists of 1,970 open domain video clips. The duration of each video clip is between 10 and 30 seconds. Each video clip has roughly 40 manually annotated captions. The dataset is split into 1,200, 100 and 670 videos for training, validation and testing, respectively.

Implementation details. Our code is implemented in Jax and based on the Scenic library [Dehghani, 2022]. We extract video frames at 1FPS, and subsample or pad the sequence of frames to F frames where we set $F = 100$. The text encoder and decoder sequence are truncated or padded to $L = S = 1000$ tokens. The visual temporal transformer encoder f^t , the text encoder g^t and the text decoder h^t all have 12 layers, 12 heads, embedding dimension 768, and MLP hidden dimension of 2048. The text encoder and decoder sequences are truncated or padded to $L = S = 1000$ tokens during pretraining, and $S = 1000$ and $L = 256$ tokens during finetuning. At inference, we use beam search decoding where we track the top 4 sequences and apply a length normalization of 0.6. Our model has 314M trainable parameters.

For training, we use the Adam optimizer [Kingma, 2015] with $\beta = (0.9, 0.999)$ and no weight decay. We pretrain our model for 200,000 iterations with a batch size of 512 videos split on 64 TPU v4 chips, which lasts a day. We sum both pretraining objectives with equal weighting to get our final pretraining loss. During pretraining, we use a learning rate of $1e^{-4}$, warming it up linearly (from 0) for the first 1000 iterations, and keeping it constant for the remaining iterations. During finetuning, we use a learning rate of $3e^{-4}$, warming it up linearly (from 0) for the first 10% of iterations, followed by a cosine decay (down to 0) for the remaining 90%. During finetuning, we use a batch size of 32 videos split on 16 TPU v4 chips. We finetune for 40 epochs on YouCook2, 20 epochs on ActivityNet Captions and ViTT, 5 epochs on MSR-VTT and 10 epochs on MSVD. We clip the maximum norm of the gradient to 0.1 during pretraining, and 1 during finetuning. For data augmentation, we use random temporal cropping. For regularization, we use label smoothing [Szegedy, 2016] with value 0.1 and dropout [Srivastava, 2014] with probability 0.1.

Evaluation metrics. For captioning, we use CIDEr [Vedantam, 2015] (C) and METEOR [Banerjee, 2005] (M). For dense video captioning, we follow the commonly used evaluation tool [Krishna, 2017] which calculates matched pairs between generated events and the ground truth

Pretraining input		YouCook2			ActivityNet			
Untrimmed	Time tokens	S	C	F1	S	C	F1	
1.	<i>No pretraining</i>	4.0	18.0	18.1	5.4	18.8	49.2	
2.	X	X	5.5	27.8	20.5	5.5	26.5	52.1
3.	✓	X	6.7	35.0	23.3	5.6	27.4	52.2
4.	✓	✓	7.9	47.1	27.3	5.8	30.1	52.4

Table 6.1: **Ablation showing the impact of using untrimmed videos and adding time tokens during pretraining.** When we use untrimmed video-speech inputs, time information from transcribed speech sentence boundaries is integrated via time tokens.

Max number of narrations		YouCook2			ActivityNet		
		S	C	F1	S	C	F1
1.	<i>No pretraining</i>	4.0	18.0	18.1	5.4	18.8	49.2
2.	1	6.0	32.1	22.1	5.1	22.9	48.1
3.	10	6.5	34.6	23.6	5.4	27.1	50.3
4.	∞	7.9	47.1	27.3	5.8	30.1	52.4

Table 6.2: **Ablation showing the importance of pretraining on long narrated videos,** by varying the maximum number of narration sentences that a randomly cropped video can cover. ∞ means the cropping is unrestricted and can sample arbitrarily long videos.

across IoU thresholds of $\{0.3, 0.5, 0.7, 0.9\}$, and compute captioning metrics over the matched pairs. However, these metrics do not take into account the story of the video. Therefore we also use SODA_c [Fujita, 2020] (S) for an overall dense video captioning evaluation. To further isolate the evaluation of event localization, we report the average precision and average recall across IoU thresholds of $\{0.3, 0.5, 0.7, 0.9\}$ and their harmonic mean, the F1 Score.

6.4.2 Ablation studies

The default Vid2Seq model predicts both text and time tokens, uses both visual frames and transcribed speech as input, builds on the T5-Base language model and the CLIP ViT-L/14 visual backbone, and is pretrained on untrimmed videos from YT-Temporal-1B with both the generative and denoising losses. Below we ablate the importance of each of these factors on the downstream dense video captioning performance by reporting results on YouCook2 and ActivityNet Captions validation sets.

Pretraining on untrimmed narrated videos by exploiting transcribed speech sentence boundaries. In Table 6.1, we evaluate the effectiveness of our pretraining task formulation that uses untrimmed videos and integrates sentence boundaries of transcribed speech via time tokens. In contrast, most video clip captioning pretraining methods [Huang, 2020b; Luo, 2020b; Seo, 2022] use short, trimmed, video-speech segments for pretraining. We adapt this strategy in our model and find that it indeed yields significant performance improvements over the baseline that uses no video-text pretraining (row 2 vs row 1). However, larger improvements are obtained by using untrimmed video-speech inputs (row 3 vs row 2). Moreover, using time

	Finetuning Input		Pretraining loss		YouCook2			ActivityNet		
	Visual	Speech	Generative	Denoising	S	C	F1	S	C	F1
1.	✓	✗	<i>No pretraining</i>		3.0	15.6	15.4	5.4	14.2	46.5
2.	✓	✓	<i>No pretraining</i>		4.0	18.0	18.1	5.4	18.8	49.2
3.	✓	✗	✓	✗	5.7	25.3	23.5	5.9	30.2	51.8
4.	✓	✓	✓	✗	2.5	10.3	15.9	4.8	17.0	48.8
5.	✓	✓	✓	✓	7.9	47.1	27.3	5.8	30.1	52.4

Table 6.3: **Effect of input modalities and pretraining losses.**

	Captioning	Pretraining	YouCook2			ActivityNet		
			Recall	Precision	F1	Recall	Precision	F1
1.	✗	✗	17.8	19.4	17.7	47.3	57.9	52.0
2.	✓	✗	17.2	20.6	18.1	42.5	64.1	49.2
3.	✗	✓	25.7	21.4	22.8	52.5	53.0	51.1
4.	✓	✓	27.9	27.8	27.3	52.7	53.9	52.4

Table 6.4: **Effect of joint captioning and localization on the localization performance.** The variant that does not caption corresponds to a localization-only variant that only predicts time tokens.

tokens to integrate time information from transcribed speech drastically improves performance (row 4 vs row 3). This shows the benefits of exploiting sentence boundaries of transcribed speech via time tokens and of using untrimmed videos during pretraining.

Pretraining on long narrated videos. In Table 6.2, we further evaluate the importance of sampling long narrated videos during pretraining. By default, at each training iteration, we randomly temporally crop each narrated video without constraints, resulting in a video that can span over hundreds of transcribed speech sentences. We here evaluate a baseline that constrains this cropping process such that the cropped video only spans over a given maximum number of narration sentences. Even with a maximum of 10 narration sentences, this baseline significantly underperforms our model trained in default settings where we sample longer untrimmed narrated videos (rows 1, 2 and 3). This demonstrates that our model benefits from pretraining on long narrated videos.

Input modalities and pretraining objectives. In Table 6.3, we analyze the importance of input modalities and pretraining tasks on the downstream dense video captioning performance. The model with visual inputs only (no transcribed speech as input) benefits significantly from pretraining with the generative objective (row 3 vs row 1). This shows the effectiveness of using the transcribed speech as a proxy annotation for dense video captioning pretraining. However, this model is pretrained with visual inputs only and its performance largely drops when it is finetuned with both visual and transcribed speech inputs (row 4 vs row 3). With both modalities, adding the denoising loss strongly benefits our model (row 5 vs rows 4 and 2). We conclude that the denoising objective benefits multi-modal reasoning.

	Language Model	Pretraining		YouCook2			ActivityNet		
		# Videos	Dataset	S	C	F1	S	C	F1
1.	T5-Small	15M	YTT	6.1	31.1	24.3	5.5	26.5	52.2
2.	T5-Base	\emptyset	\emptyset	4.0	18.0	18.1	5.4	18.8	49.2
3.	T5-Base	15K	YTT	6.3	35.0	24.4	5.1	24.4	49.9
4.	T5-Base	150K	YTT	7.3	40.1	26.7	5.4	27.2	51.3
5.	T5-Base	1M5	YTT	7.8	45.5	26.8	5.6	28.7	52.2
6.	T5-Base	1M	HTM	8.3	48.3	26.6	5.8	28.8	53.1
7.	T5-Base	15M	YTT	7.9	47.1	27.3	5.8	30.1	52.4

Table 6.5: **Effect of language model size and pretraining data.** HTM: HowTo100M [Miech, 2019], YTT: YT-Temporal-1B [Zellers, 2022].

	Pretraining Data	Model	YouCook2			ActivityNet		
			S	C	F1	S	C	F1
1.	ImageNet	ViT-B/16	6.6	40.2	24.3	4.5	17.2	49.3
2.	CLIP	ViT-B/16	7.7	46.3	26.5	5.6	28.4	51.7
3.	CLIP	ViT-L/14	7.9	47.1	27.3	5.8	30.1	52.4

Table 6.6: **Ablation on the pretraining data and model size of the visual backbone f^s .**

Effect of captioning on localization. In Table 6.4, we compare the event localization performance of our model with a localization-only variant that only predicts event boundaries. We find that the model that jointly predicts event boundaries and captions localizes better and benefits more from pretraining than the localization-only baseline (row 4 vs row 3), which demonstrates the importance of contextualizing the noisy timestamps of the transcribed speech with the speech semantic content during pretraining.

Model size and pretraining data. In Table 6.5, we show that the language model size has a great importance on the performance, as the model with T5-Base outperforms its variant with T5-Small (row 7 vs row 1). We also evaluate the importance of the size of the pretraining dataset of narrated videos by constructing subsets such that larger subsets include the smaller ones. We find that scaling up the size of the pretraining dataset is beneficial, and that our pretraining method yields important benefits when only using 150K narrated videos for pretraining (row 4). We further show that our pretraining method generalizes well to the HowTo100M dataset [Miech, 2019]. The model pretrained on HowTo100M (row 6) actually achieves best results on YouCook2, as these datasets are from a similar domain.

Visual features. In Table 6.6, we further analyze the importance of the pretraining dataset and size of the visual backbone f^s . We find that CLIP pretraining [Radford, 2021] considerably improves over ImageNet pretraining [Steiner, 2022] with the same ViT-B/16 visual backbone model (row 2 vs 1). Furthermore, scaling up the visual backbone size from ViT-B/16 to ViT-L/14 brings additional improvements (row 3 vs 2).

	Language Model	Video-text	YouCook2			ActivityNet		
	Initialization	Pretraining	S	C	F1	S	C	F1
1.	X	X	0.9	4.2	7.6	4.3	23.7	41.2
2.	✓	X	4.0	18.0	18.1	5.4	18.8	49.2
3.	X	✓	8.8	51.3	28.4	5.7	28.7	51.2
4.	✓	✓	7.9	47.1	27.3	5.8	30.1	52.4

Table 6.7: Ablation on language model initialization and pretraining.

	Tokenization	N	YouCook2			ActivityNet		
			S	C	F1	S	C	F1
1.	Absolute	20	0.3	0.2	0.9	3.2	23.0	23.1
2.	Absolute	100	3.5	25.7	12.0	4.8	25.5	41.5
3.	Absolute	500	7.9	39.8	24.3	5.4	28.1	48.6
4.	Relative	20	7.2	39.6	23.7	5.6	29.0	49.4
5.	Relative	100	7.9	47.1	27.3	5.8	30.1	52.4
6.	Relative	500	7.2	40.0	25.0	5.7	28.6	52.5

Table 6.8: Ablation on time tokenization (relative or absolute) and the number of time tokens N .

Language model initialization and pretraining. In Table 6.7, we further investigate the importance of initializing the language model from weights pretrained on Web text. Without pretraining on narrated videos, we find that text-only initialization is helpful (rows 1 and 2). Interestingly, after pretraining on narrated videos, we find that text-only initialization has little importance (rows 3 and 4), as it slightly improves the performance on ActivityNet Captions while resulting in a slight drop of performance on YouCook2. We believe that this may be because of the domain gap between Web text and the imperative-style dense captions in YouCook2, which are more similar to transcribed speech in YT-Temporal-1B.

Time tokenization and number of time tokens. In Table 6.8, we ablate the time tokenization process presented in Section 6.3.1. Our default time tokens represent relative timestamps in a video, as we quantize a video of duration T into N equally-spaced timestamps. Another possibility is to use time tokens that represent absolute timestamps in the video, *i.e.*, the k -th token represents the k -th second in the video. For both these variants, we vary the number of time tokens N . For the relative time tokens, increasing N makes the quantization more fine-grained but also spreads the data into more time tokens. On the other hand, for the absolute time tokens, increasing N increases the video duration that the time tokens can cover. We find that the best dense video captioning results are obtained with the relative time tokens and $N = 100$ time tokens (row 5).

Sequence construction. In Table 6.9, we further ablate the sequence construction process presented in Section 6.3.1. Our default sequence inserts the start and end time tokens of each segment before its corresponding text sentence. Another possibility is to insert time tokens after each corresponding text sentence. We find that both variants achieve similar results (rows 2 and

	Dot symbol between segments	Time tokens Position	YouCook2			ActivityNet		
			S	C	F1	S	C	F1
1.	✗	<i>After text</i>	7.9	48.3	26.7	5.6	29.8	51.1
2.	✓	<i>After text</i>	8.3	50.9	26.2	5.7	30.4	51.8
3.	✗	<i>Before text</i>	8.0	50.0	27.3	5.6	28.2	50.7
4.	✓	<i>Before text</i>	7.9	47.1	27.3	5.8	30.1	52.4

Table 6.9: **Ablation on the sequence construction process.**

	Temporal embeddings	YouCook2			ActivityNet		
		S	C	F1	S	C	F1
1.	✗	6.8	42.0	24.9	5.3	27.0	50.6
2.	✓	7.9	47.1	27.3	5.8	30.1	52.4

Table 6.10: **Ablation on the temporal positional embeddings.**

4), with the default sequence (row 4) resulting in slightly higher event localization performance (F1 Score) but slightly lower dense captioning results overall. Furthermore, we observe that the dot symbols indicating the separation between different events have low importance (rows 1 and 2, rows 3 and 4).

Temporal positional embeddings. In Table 6.1, we show that time tokens in the speech sequence provide temporal information about the speech transcript to our model. In Table 6.10, we also evaluate the importance of the temporal positional embeddings which communicate temporal information from the visual stream to our model. We find that these temporal embeddings are beneficial (row 2 vs 1).

6.4.3 Comparison to the state of the art

Dense video captioning. In Table 6.11, we compare our approach to state-of-the-art dense video captioning methods using cross-entropy training ¹ on the YouCook2, ViTT and ActivityNet Captions datasets. Vid2Seq sets new state of the art on all three datasets. In particular, our method improves the CIDEr metric by 18.2 and 0.8 points on YouCook2 and ActivityNet Captions over PDVC. Our method also outperforms E2ESG [Zhu, 2022b] which uses in-domain text-only pretraining on Wikihow. These results demonstrate the strong dense event captioning ability of our pretrained Vid2Seq model.

Event localization. In Table 6.12, we evaluate the event localization performance of our dense video captioning model in comparison with prior work. On both YouCook2 and ViTT, Vid2Seq outperforms prior work [Zhu, 2022b] tackling dense video captioning as a single sequence generation task. However, our model underperforms compared to PDVC [Wang, 2021d] and UEDVC [Wang, 2021d] on ActivityNet Captions. We emphasize that our approach integrates less prior knowledge about temporal localization than both these approaches, which include task

¹We do not include methods directly optimizing the test metric [Deng, 2021a; Mun, 2019].

Method	Backbone	YouCook2 (val)			ViTT (test)			ActivityNet (val)		
		S	C	M	S	C	M	S	C	M
MT [Zhou, 2018c]	TSN	—	6.1	3.2	—	—	—	—	9.3	5.0
ECHR [Wang, 2020c]	C3D	—	—	3.8	—	—	—	3.2	14.7	7.2
PDVC [Wang, 2021d]	TSN	4.4	22.7	4.7	—	—	—	5.4	29.0	8.0
PDVC [Wang, 2021d] [†]	CLIP	4.9	28.9	5.7	—	—	—	6.0	29.3	7.6
UEDVC [Zhang, 2022c]	TSN	—	—	—	—	—	—	5.5	—	—
E2ESG [Zhu, 2022b]	C3D	—	25.0*	3.5	—	25.0	8.1	—	—	—
Vid2Seq (Ours)	CLIP	7.9	47.1	9.3	13.5	43.5	8.5	5.8	30.1	8.5

Table 6.11: Comparison to the state of the art for dense video captioning. * Results provided by the authors. † Results of our experiments using the official codebase.

Method	Backbone	YouCook2 (val)		ViTT (test)		ActivityNet (val)	
		Recall	Precision	Recall	Precision	Recall	Precision
PDVC [Wang, 2021d]	TSN	—	—	—	—	55.4	58.1
PDVC [Wang, 2021d] [†]	CLIP	—	—	—	—	53.2	54.7
UEDVC [Zhang, 2022c]	TSN	—	—	—	—	59.0	60.3
E2ESG [Zhu, 2022b]	C3D	20.7*	20.6*	32.2*	32.1*	—	—
Vid2Seq (Ours)	CLIP	27.9	27.8	42.6	46.2	52.7	53.9

Table 6.12: Comparison to the state of the art for event localization. * Results provided by the authors. † Results of our experiments using the official codebase.

specific components such as event counters [Wang, 2021d] or separately train a model for the localization subtask [Zhang, 2022c].

Video paragraph captioning. In Table 6.13, we compare our approach to state-of-the-art video paragraph captioning methods on the YouCook2 and ActivityNet Captions datasets. Vid2Seq outperforms all prior methods on both datasets, including the ones using ground-truth event boundary proposals at inference time [Dai, 2019; Lei, 2020a; Zhou, 2018c; Zhou, 2019; Wang, 2021d; Park, 2019], showing strong video paragraph captioning ability.

Video clip captioning. In Table 6.14, we compare our approach to state-of-the-art video clip captioning methods on the MSR-VTT and MSVD datasets. Vid2Seq improves over prior methods in their respective pretraining data setting while using a comparable number of trained parameters. We conclude that our pretrained Vid2Seq model generalizes well to the standard video clip captioning setting.

6.4.4 Few-shot dense video captioning

To further evaluate the generalization capabilities of our pretrained Vid2Seq model, we propose a new few-shot dense video captioning setting where we finetune Vid2Seq using only a fraction of the downstream training dataset. From Table 6.15, we observe important improvements when using 10% compared to 1% of training data (row 3 vs 1). We further find that pretraining is essential in the few-shot setting (see row 2 vs 1 for instance).

Method	Backbone	YouCook2 (val)		ActivityNet (val-ae)	
		C	M	C	M
<i>With GT Proposals</i>					
VTransformer [Zhou, 2018c]	V (ResNet-200) + F	32.3	15.7	22.2	15.6
Transformer-XL [Dai, 2019]	V (ResNet-200) + F	26.4	14.8	21.7	15.1
MART [Lei, 2020a]	V (ResNet-200) + F	35.7	15.9	23.4	15.7
GVDSup [Zhou, 2019]	V (ResNet-101) + F + O	—	—	22.9	16.4
AdvInf [Park, 2019]	V (ResNet-101) + F + O	—	—	21.0	16.6
PDVC [Wang, 2021d]	V + F (TSN)	—	—	27.3	15.9
<i>With Learnt Proposals</i>					
MFT [Xiong, 2018]	V + F (TSN)	—	—	19.1	14.7
PDVC [Wang, 2021d]	V + F (TSN)	—	—	20.5	15.8
PDVC [Wang, 2021d] [†]	V (CLIP)	—	—	23.6	15.9
Vid2Seq (Ours)	V (CLIP)	50.1	24.0	28.0	17.0

Table 6.13: Comparison to the SoTA for video paragraph captioning. [†] Results of our experiments using the official codebase. V/F/O refers to visual/flow/object features.

Method	Trained Parameters	Pretraining Data	MSR-VTT (test)		MSVD (test)	
			C	M	C	M
ORG-TRL [Zhang, 2020e]	—	∅	50.9	28.8	95.2	36.4
SwinBERT [Lin, 2022b]	229M	∅	53.8	29.9	120.6	41.3
Vid2Seq (Ours)	314M	∅	57.2	30.0	120.3	41.4
MV-GPT [Seo, 2022]	354M	HowTo100M	60.0	29.9*	—	—
Vid2Seq (Ours)	314M	HowTo100M	61.5	30.4	140.6	44.5
Vid2Seq (Ours)	314M	YT-Temporal-1B	64.6	30.8	146.2	45.3

Table 6.14: Comparison to the SoTA for video clip captioning. * indicates results re-evaluated by the same evaluation toolkit.

	Data	Pretrain	YouCook2			ViTT			ActivityNet		
			S	C	M	S	C	M	S	C	M
1.	1%	✗	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1
2.	1%	✓	2.4	10.1	3.3	2.0	7.4	1.9	2.2	6.2	3.2
3.	10%	✗	0.1	0.0	0.2	3.3	0.4	3.3	3.4	11.9	4.6
4.	10%	✓	3.8	18.4	5.2	10.7	28.6	6.0	4.3	20.0	6.1
5.	50%	✗	1.8	8.5	2.4	6.5	18.7	3.9	4.6	13.1	6.3
6.	50%	✓	6.2	32.1	7.6	12.5	38.8	7.8	5.4	27.5	7.8
7.	100%	✗	4.0	18.0	4.6	7.9	21.2	6.2	5.4	18.8	7.1
8.	100%	✓	7.9	47.1	9.3	13.5	43.5	8.5	5.8	30.1	8.5

Table 6.15: **Few-shot dense video captioning**, by finetuning FrozenBiLM using a small fraction of the downstream training dataset, compared with a non-pretrained variant.

6.4.5 Qualitative examples

In Figures 6.4 and 6.5, we show examples of dense event captioning predictions from Vid2Seq on ActivityNet Captions and YouCook2, respectively. These results show that Vid2Seq can predict meaningful dense captions and event boundaries in diverse scenarios, with or without transcribed speech input, *e.g.*, series of instructions in cooking recipes (Figure 6.5) or actions

Chapter 6. Vid2Seq: Large-Scale Pretraining of a Visual Language Model for Dense Video Captioning

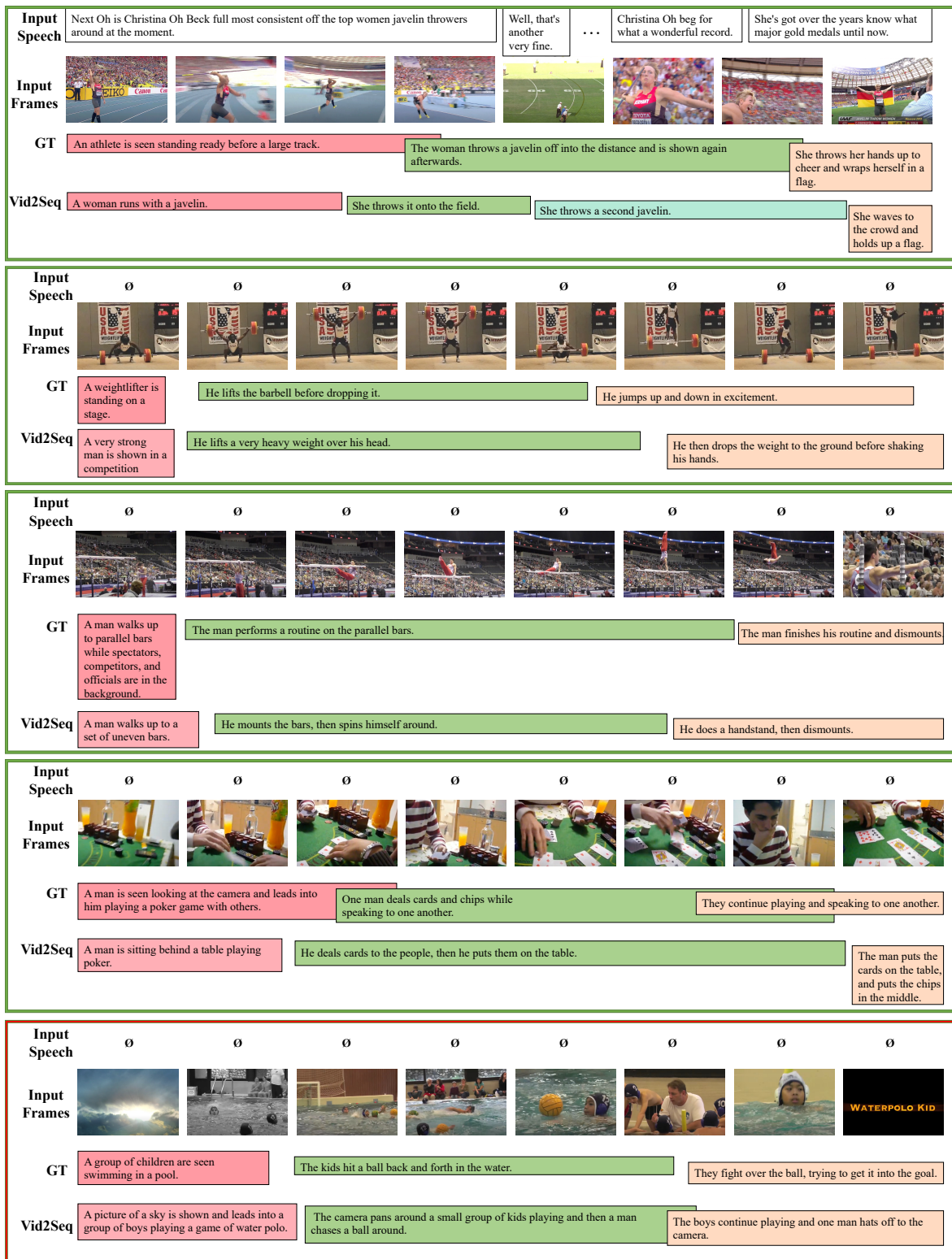


Figure 6.4: Examples of dense event captioning predictions of Vid2Seq on ActivityNet Captions validation set, compared with ground-truth. The first four examples show successful predictions, while the last example illustrates a failure case where the model hallucinates events that are not visually grounded ('one man hats off to the camera').

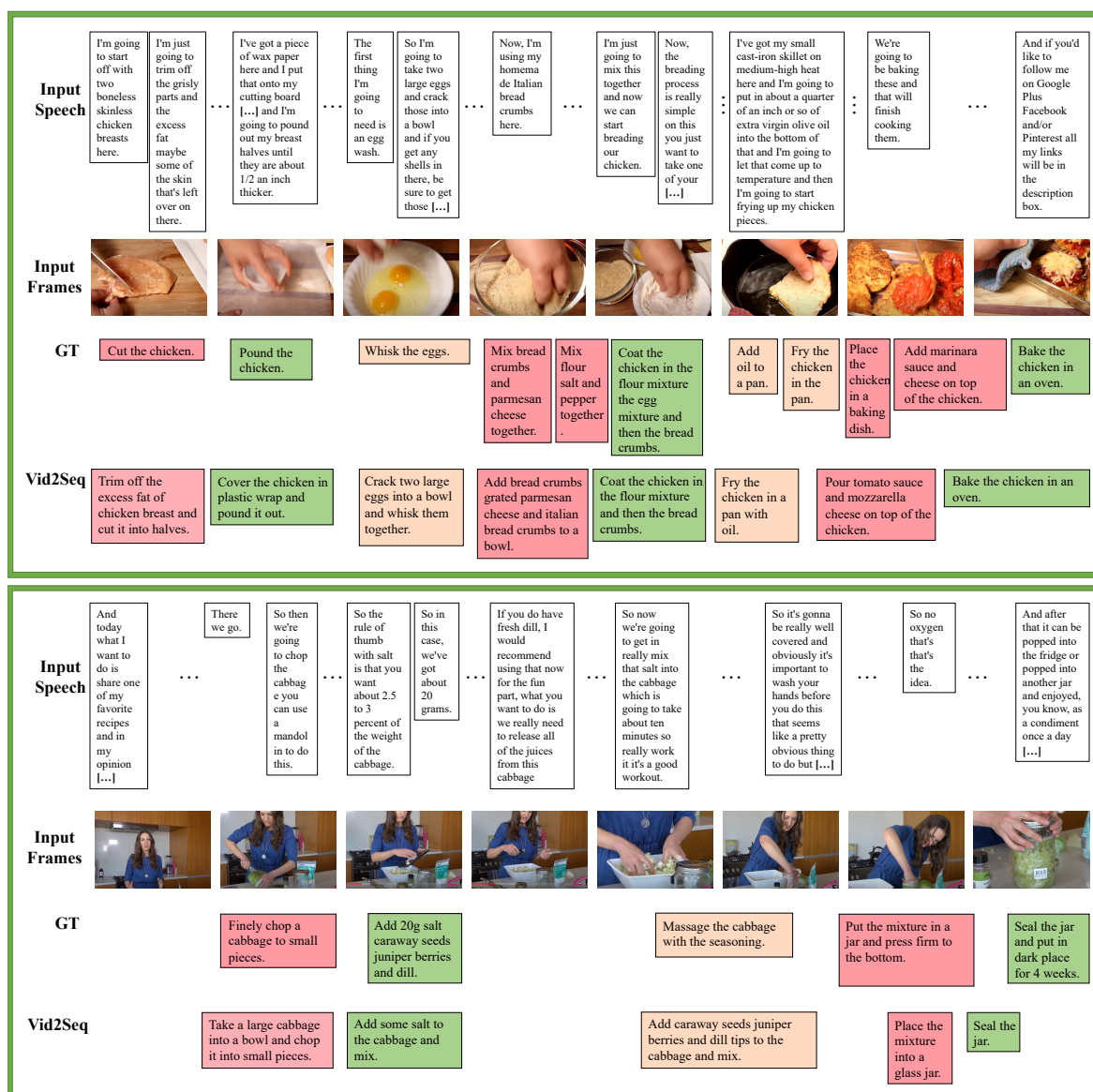


Figure 6.5: Examples of dense event captioning predictions of Vid2Seq on the validation set of YouCook2, compared with ground-truth. We show more examples on our webpage [Yang, 2023a].

in human sports or leisure activities (first four examples in Figure 6.4). The last example in Figure 6.4 illustrates a failure case where the model hallucinates events that are not visually grounded such as ‘one man hats off to the camera’. Moreover, we observe that the predicted captions and boundaries differ considerably from the transcribed speech input (showing the importance of the visual tokens in the input).

6.5 Conclusion

We introduce Vid2Seq, a visual language model that performs dense video captioning by generating a single sequence of tokens including both text and time tokens given multi-modal inputs. We show that Vid2Seq benefits from large-scale pretraining on unlabeled untrimmed

narrated videos by leveraging transcribed speech sentences and corresponding temporal boundaries. Vid2Seq achieves state-of-the-art results on various dense event captioning datasets, as well as multiple video paragraph captioning and standard video clip captioning benchmarks.

Limitations. Vid2Seq can only process 100 video frames at a time, and is not trained end-to-end as it relies on a frozen visual feature extractor. In addition, Vid2Seq cannot make use of raw audio inputs. Moreover, this work mainly focuses on video captioning tasks. However, we believe the sequence-to-sequence design of Vid2Seq has the potential to be extended to a wide range of *other* video tasks such as temporally-grounded video question answering [Lei, 2018a; Li, 2021b; Li, 2020b] or temporal action localization [Liu, 2022a; Zhang, 2022a; Cheng, 2022a].

Chapter 7

VidChapters-7M: Video Chapters at Scale

Segmenting long videos into chapters enables users to quickly navigate to the information of their interest (see Figure 7.1). This important topic has been understudied due to the lack of publicly released datasets. To address this issue, we present VidChapters-7M, a dataset of 817K user-chaptered videos including 7M chapters in total. VidChapters-7M is automatically created from videos online in a scalable manner by scraping user-annotated chapters and hence without any additional manual annotation. We introduce the following three tasks based on this data. First, the video chapter generation task consists of temporally segmenting the video and generating a chapter title for each segment. To further dissect the problem, we also define two variants of this task: video chapter generation given ground-truth boundaries, which requires generating a chapter title given an annotated video segment, and video chapter grounding, which requires temporally localizing a chapter given its annotated title. We benchmark both simple baselines and state-of-the-art video-language models, including the previously presented Vid2Seq model, for these three tasks. We also show that pretraining Vid2Seq on VidChapters-7M transfers well to dense video captioning tasks in both zero-shot and finetuning settings, largely improving over the prior state of the art on the YouCook2 [Zhou, 2018a] and ViTT [Huang, 2020b] benchmarks. Finally, our experiments reveal that downstream performance scales well with the size of the pretraining dataset. Our dataset, code, and models are publicly available at [Yang, 2023b].

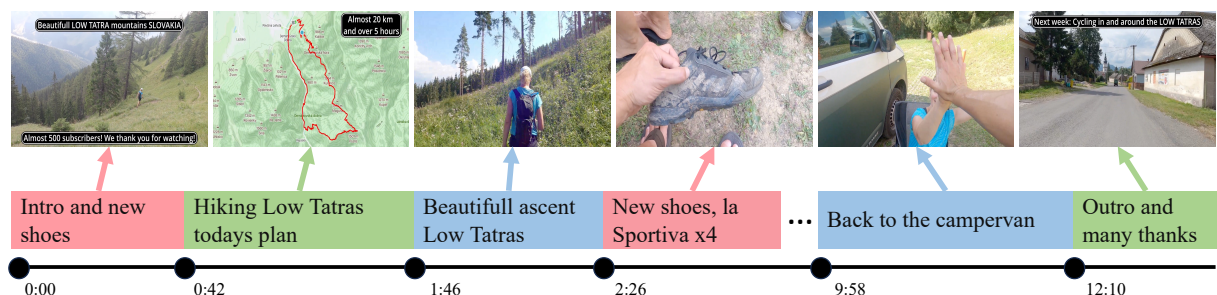


Figure 7.1: **A video with user-annotated chapters in VidChapters-7M:** the video is temporally segmented into chapters, which are annotated with a chapter title in free-form natural language.

7.1 Introduction

As online media consumption grows, the volume of video content available is increasing rapidly. While searching for specific videos is already a challenging problem, searching within a long video is an even *less* explored task. Manual navigation can often be time consuming, particularly for long videos. A compelling solution for organizing content online is to segment long videos into *chapters* (see Figure 7.1). Chapters are contiguous, non-overlapping segments, completely partitioning a video. Each chapter is also labeled with a short description of the chapter content, enabling users to quickly navigate to areas of interest and easily replay different parts of a video. Chapters also give *structure* to a video, which is useful for long videos that contain inherently listed content, such as listicles [Vijgen, 2014], instructional videos [Miech, 2019], music compilations and so on.

Given the plethora of content already online, our goal is to explore automatic solutions related to video chaptering - generating chapters automatically, and grounding chapter titles temporally in long videos. While the benefits of automatically chaptering videos are obvious, data for this task is scarce. Video captioning datasets (such as WebVid-10M [Bain, 2021] and VideoCC [Nagrani, 2022]) consist of short videos (10s in length), and hence are unsuitable. Web datasets consisting of longer videos (HowTo100M [Miech, 2019], YT-Temporal-1B [Zellers, 2022]) come with aligned speech transcripts (ASR), which are only weakly related to visual content, and if used as chapter titles would tend to over-segment videos. Moment retrieval [Gao, 2017a; Hendricks, 2017] or dense video captioning [Krishna, 2017; Zhou, 2018a] datasets are perhaps the most useful, but do not focus on creating explicit *structure*, and instead describe low-level actions comprehensively. Such datasets are also manually annotated, and hence not scalable and small in size (see Table 7.1).

To remedy this, we curate VidChapters-7M, a large-scale dataset of user-annotated video chapters automatically scraped from the Web. Our dataset consists of 7M chapters for over 817K videos. Compared to existing datasets, videos in VidChapters-7M are long (23 minutes on average) and contain rich chapter annotations consisting of a starting timestamp and a title per chapter. Our dataset is also diverse, with 12 different video categories having at least 20K videos each, which itself is the size of existing dense video captioning datasets [Grauman, 2022; Huang, 2020b; Krishna, 2017; Zhou, 2018a]. On top of this dataset we also define 3 video tasks (see Figure 7.2): (i) *video chapter generation* which requires temporally segmenting the video and generating a chapter title for each segment; (ii) *video chapter generation given ground-truth boundaries*, which requires generating a chapter title given an annotated video segment; and (iii) *video chapter grounding*, which requires temporally localizing a chapter given the chapter title. All three tasks involve parsing and understanding *long* videos, and multi-modal reasoning (video and text), and hence are valuable steps towards story understanding.

For all three tasks, we implement simple baselines as well as recent, state-of-the-art video-text methods [Lei, 2021a; Wang, 2021d; Yang, 2023d]. We find that the tasks are far from being solved, demonstrating the value of this problem. Interestingly, we also show that our video chapter generation models trained on VidChapters-7M transfer well to dense video captioning

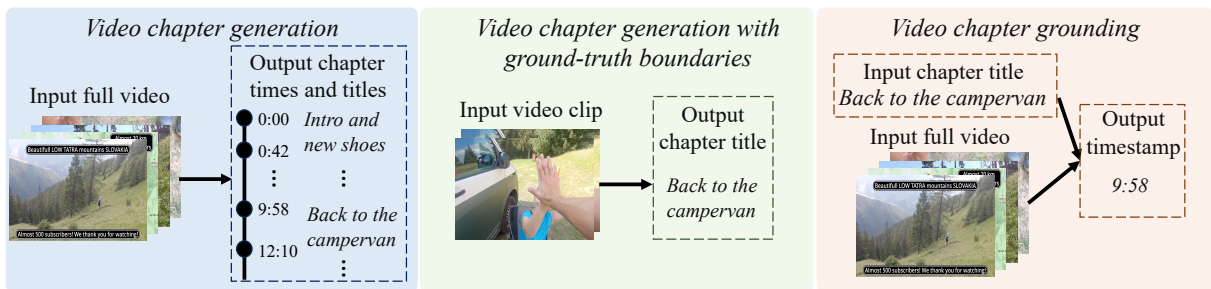


Figure 7.2: Illustration of the three tasks defined for VidChapters-7M.

tasks in both zero-shot and finetuning settings, largely improving over the prior state of the art on the YouCook2 [Zhou, 2018a] and ViTT benchmarks [Huang, 2020b]. Moreover, we show that pretraining using both speech transcripts and chapter annotations significantly outperforms the widely used pretraining method based only on speech transcripts [Miech, 2020; Yang, 2023d; Zellers, 2022]. This demonstrates the additional value of our dataset as a generic video-language *pretraining* set. Interestingly, we also find that the transfer performance scales with the size of the chapter dataset.

In summary, our contributions are:

- (i) We present VidChapters-7M, a large-scale dataset of user-annotated video chapters obtained from the Web consisting of 817K videos and 7M chapters.
- (ii) Based on this dataset, we evaluate a range of simple baselines and state-of-the-art video-language models on the tasks of video chapter generation with and without ground-truth boundaries, and video chapter grounding.
- (iii) We show that video chapter generation models trained on VidChapters-7M transfer well to dense video captioning tasks in both zero-shot and finetuning settings, largely improving over the prior state of the art on the YouCook2 [Zhou, 2018a] and ViTT benchmarks [Huang, 2020b], outperforming prior pretraining methods based on narrated videos [Yang, 2023d], and showing promising scaling behavior.

7.2 Related Work

Large-scale vision-language datasets. The development of powerful multi-modal models [Alayrac, 2022; Chen, 2020b; Gan, 2020; Hu, 2022b; Huang, 2021b; Jia, 2021; Lei, 2021b; Li, 2020a; Li, 2021a; Li, 2022c; Li, 2020d; Lu, 2019; Lu, 2020; Radford, 2021; Singh, 2022; Su, 2019; Tan, 2019; Tsimpoukelli, 2021; Wang, 2022f; Wang, 2022a; Yu, 2020; Yuan, 2021; Zhou, 2020] has been made possible by pretraining on large-scale image-caption datasets scraped from the Web such as SBU [Ordonez, 2011], Conceptual Captions [Sharma, 2018], Conceptual-12M [Changpinyo, 2021], LAIT [Qi, 2020], Wikipedia-ImageText [Srinivasan, 2021], RedCaps [Desai, 2021b] and LAION-5B [Schuhmann, 2022]. Similarly, many strong video-language models [Akbari, 2021; Ge, 2022; Han, 2022; Ko, 2022; Lei, 2021a; Li, 2022a; Li, 2020b; Li, 2023c;

Dataset	Number of videos	Video duration (min)	Number of descriptions	Annotations
HowTo100M [Miech, 2019]	1M	7	136M	Speech transcripts
YT-Temporal-1B [Zellers, 2022]	19M	6	~ 900M	Speech transcripts
HD-VILA-100M [Xue, 2022]	3M	7	103M	Speech transcripts
ActivityNet Captions [Krishna, 2017]	20K	3	100K	Dense Captions
YouCook2 [Zhou, 2018a]	2K	6	15K	Dense Captions
ViTT [Huang, 2020b]	8K	4	56K	Dense Captions
Ego4D [Grauman, 2022]	10K	23	4M	Dense Captions
VidChapters-7M (Ours)	817K	23	7M	Speech transcripts + User-annotated Chapters

Table 7.1: **Comparison of VidChapters-7M with existing datasets.** We consider open-sourced video datasets that contain dense natural language descriptions aligned over time. VidChapters-7M is much larger than current dense video captioning datasets. Compared to datasets with ASR (top 3 rows), it is smaller in the total number of videos but contains longer videos with richer annotations (chapters).

Lin, 2022a; Miech, 2020; Seo, 2021b; Seo, 2022; Sun, 2019b; Sun, 2022; Tang, 2022; Wang, 2023; Wang, 2022b; Xu, 2021; Yang, 2021b; Yang, 2022c; Yang, 2022e; Zhao, 2023] have been pretrained on Web-scraped video-text datasets. These datasets are largely composed of short videos paired with captions, e.g. WebVid-10M [Bain, 2021] and VideoCC [Nagrani, 2022], or narrated videos with speech transcripts aligned over time (ASR), e.g. HowTo100M [Miech, 2019], YT-Temporal-1B [Zellers, 2021; Zellers, 2022] and HD-VILA-100M [Xue, 2022]. Our proposed VidChapters-7M dataset is also downloaded from the Web, via a scalable pipeline without the need for expensive manual annotation. Unlike these datasets, VidChapters-7M consists of long videos with user-annotated chapters aligned over time (see Table 7.1), which significantly differ from ASR (see Section 7.3.3). Furthermore, most videos in VidChapters-7M *also* contain ASR. Finally, VidChapters-7M is also related to the recent ChapterGen dataset [Cao, 2022b], which also consists of user-annotated chapters. However, ChapterGen is several orders of magnitude smaller than VidChapters-7M (10K vs 817K videos) and is not open-sourced at the time of writing.

Video tasks. The video chapter generation task requires temporally segmenting the video into chapters, hence is related to video shot detection [Rasheed, 2003; Rui, 1998; Sidiropoulos, 2011], movie scene segmentation [Chen, 2021c; Rao, 2020], temporal action localization [Chao, 2018; Cheng, 2022a; Liu, 2022a; Shou, 2016; Zhang, 2022a; Zeng, 2019] and temporal action segmentation [Behrmann, 2022; Farha, 2019; Gao, 2021; Lea, 2017; Li, 2021d; Wang, 2020e]. However, unlike these tasks, video chapter generation also requires generating a free-form natural language chapter title for each segment. Hence this task is also related to video captioning [Gao, 2017c; Lin, 2022b; Luo, 2020b; Pan, 2017; Wang, 2018a; Wang, 2018c; Zhang, 2020e], video title generation [Amirian, 2021; Zeng, 2016; Zhang, 2020b], generic event boundary captioning [Wang, 2022d] and dense video captioning [Krishna, 2017; Wang, 2021d; Zhou, 2018c]. Most related to video chapter generation, the dense video captioning task requires temporally localizing and captioning all events in an untrimmed video. In contrast, video chapter generation requires

temporally *segmenting* the video (i.e. the start of the chapter $i + 1$ is the end of chapter i , and the chapters cover the full video), and involves generating a chapter title that is substantially shorter than a video caption. We study in more detail the transfer learning between these two tasks in Section 7.4.4. Finally, the video chapter grounding task is related to temporal language grounding [Hendricks, 2017; Hendricks, 2018; Lei, 2020c; Lei, 2021a; Nan, 2021; Yang, 2022d; Zhang, 2020a; Zhang, 2020c]. However, we here focus on localizing a chapter starting point and not a start-end window. Furthermore, most temporal language grounding methods represent the video only with visual inputs, while we also exhibit the benefits of using speech inputs for localizing chapters in videos (see Section 7.4.3).

7.3 VidChapters-7M: a large-scale dataset of user-chaptered videos

Our goal is to build a large and diverse set of videos annotated with temporarily localized chapter information, consisting of chapter titles and chapter start times. In detail, chapters are contiguous, non-overlapping segments, completely partitioning a video. However manual annotation of chapters is time consuming and expensive and therefore hard to scale. Hence we automatically scrape chapter information from videos available online, as explained in Section 7.3.1. Then, we perform several processing steps on this data, e.g., to extract speech transcripts, as described in Section 7.3.2. The outcome is VidChapters-7M, a dataset of 817K videos with 7M chapter annotations provided by real users online. Finally, we analyze VidChapters-7M in Section 7.3.3. Details are given next.

7.3.1 Data collection

Since early 2020, YouTube users can create chapters for uploaded videos by annotating them in the YouTube description. The YouTube API, however, currently does not enable explicit search for user-chaptered videos. Hence, our data collection procedure consists of: (i) Collecting a large and diverse set of video candidates (characterized by their 11-character YouTube video ID), which do not necessarily contain user-annotated chapters; (ii) For all video candidates, downloading the video description, automatically selecting videos with user-annotated chapters, extracting video chapters and downloading corresponding videos. We next describe the individual steps in more detail.

Video candidates. We start from a large pool of video candidates built from the YT-Temporal-180M dataset [Zellers, 2021], which was constructed to be more diverse than prior large video datasets such as HowTo100M [Miech, 2019]. Note that while the released YT-Temporal-180M dataset consists of only 5M videos, the authors collected a larger set of candidates by using YouTube’s recommendation algorithm to suggest related videos. We obtained this extended list of 92 million video IDs directly from the authors.

Extracting chapters from descriptions. In the description, chapters typically constitute a block with consecutive lines following the format "`<Timestamp>: <Chapter Title>`" or "`<Chapter`

Title>: <Timestamp>", where the chapter title is written in free-form natural language and its corresponding start timestamp is written in `MM:SS` format. The video should contain at least two timestamps listed in ascending order. Hence we download the descriptions for all video candidates and use standard regular expression operations to verify whether a given description contains user-annotated chapters and extract them if so. Note that some videos contain chapters that are automatically generated by YouTube algorithms, however, these generated chapters do not appear in the descriptions and, hence, are excluded by our procedure for data collection. Also note that the video content is only downloaded for user-chaptered videos, which is convenient for both the downloading speed and storage constraints. Finally, we obtain 817K user-chaptered videos, making up 0.9% of all video candidates.

7.3.2 Data processing

We describe below how we process the previously obtained user-chaptered videos to facilitate building efficient video chapter generation models. For reproducibility, we publicly release the resulting speech transcripts and the code for extracting visual features.

ASR extraction. We observed that most user-chaptered videos contain speech. Hence, for all videos, we extract speech transcripts aligned in time with the video content (ASR) by applying the Whisper-Large-V2 model [Radford, 2023] on the audio track, using faster-whisper [Klein, 2023] backend for computational efficiency. We found that the Whisper model provides higher-quality ASR compared to the YouTube API ASR service on several data samples from VidChapters-7M. We further use WhisperX [Bain, 2023] to derive accurate word-level timestamps which we use to segment the speech transcript into sentences. For example, the Whisper-Large-V2 model extracts speech segments like *“Right, we’re gonna do the Synthetics Dirty Race. No we’re not. [...] So we’re gonna put two t-shirts and two pairs of jeans in the”* with timestamps 20.478s and 50.465s, and the corresponding first sentence output by WhisperX is *“Right, we’re gonna do the Synthetics Dirty Race.”* with timestamps 20.538s and 29.26s.

Visual feature extraction. Training end-to-end deep learning models from RGB inputs on minutes-long videos is computationally expensive. Hence we extract visual features with CLIP ViT-L/14 backbone [Dosovitskiy, 2021; Radford, 2021] at resolution 224×224 pixels and 1 FPS. This model has been trained to map images to text descriptions with a contrastive loss on 400M Web-scraped image-text pairs.

7.3.3 Data analysis

The result of the previously described pipeline is VidChapters-7M, a dataset of 817,076 user-chaptered videos containing 6,813,732 chapters in total. We randomly split VidChapters-7M in training, validation, and testing splits with 801K, 8.2K, and 8.2K videos, respectively. We analyze VidChapters-7M below and provide a datasheet [Geburu, 2021] at [Yang, 2023c] (Appendix F).

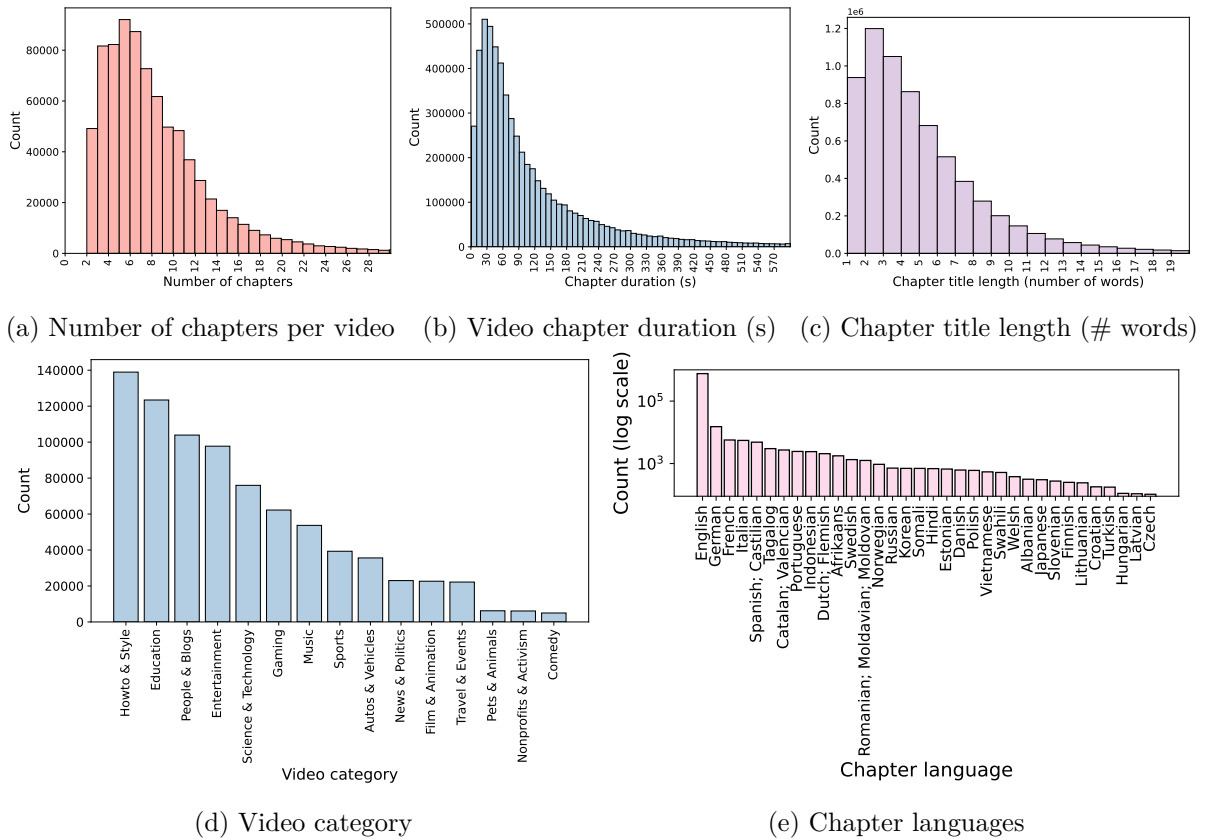


Figure 7.3: Statistics of the VidChapters-7M dataset.

Statistics. VidChapters-7M is highly diverse and contains 4,894,855 distinct chapter titles. On average, a video contains 8.3 chapters, start times of adjacent chapters are separated by 142.0s seconds, a chapter title contains 5.4 words and a video lasts 1354 seconds. The most represented video category (in YouTube’s glossary) is HowTo & Style, making up 17.0% of total videos. The distributions for the number of chapters per video, the video chapter duration, the length of the chapter title, and the video category are illustrated in Figure 7.3, and further show the diversity of VidChapters-7M, e.g., there are 12 different video categories with at least 20K videos in VidChapters-7M.

In Figure 7.4, we also show a histogram of the most common chapter titles and word clouds¹ of the chapters titles and ASR content in VidChapters-7M. A few generic chapter titles that outline the structure of the video (e.g., *Intro*, *Introduction*, *Outro*, *Conclusion* and *Start*) appear more than 10K times. Besides, we notice that many videos include chapters about *Unboxing*, *Review*, or *Tips*. This is consistent with the fact that there are many vlogs and ‘Howto’ videos in VidChapters-7M.

To further measure the text-video alignment in the VidChapters-7M dataset, we compute the CLIP cosine similarity between chapter titles and their corresponding video frames and plot the resulting distribution in Figure 7.5. The average similarity score is 54.6%, and less than 1% of the chapters have a visual-text similarity score below 30%. These statistics demonstrate a

¹To generate the word clouds, we used https://github.com/amueller/word_cloud.

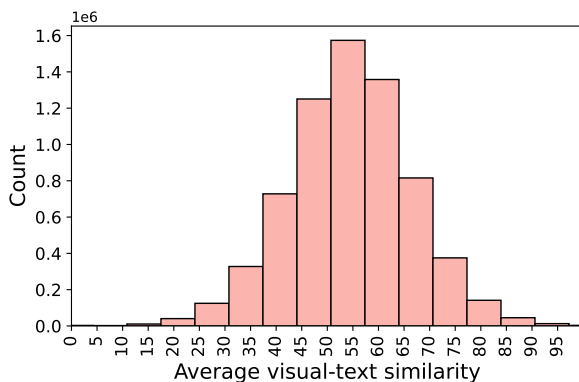


Figure 7.5: Average visual-text similarity between chapter titles and the corresponding video frames as measured by CLIP cosine similarity (rescaled between 0 and 100) in VidChapters-7M.

good video-text alignment in the VidChapters-7M dataset.

Examples of user-chaptered videos. In Figure 7.6, we provide additional examples that complement Figure 7.1. These examples illustrate the diversity of the data in VidChapters-7M, e.g., our dataset includes review videos, cooking videos, clothing fitting videos, ASMR videos, and videos of conversations. These examples also show the multi-modal nature of the chapter data. Indeed, chapters depict visual events (e.g., the mini chicken burgers that appear in the second video), conversations (see the last video), or events in the raw audio (e.g., the sound of the crinkly plastic bag in the penultimate video) in various scenarios.

ASR vs Chapters. 97.3% of videos in VidChapters-7M contain speech transcripts (ASR). However, user-annotated chapters significantly differ from speech transcripts: on average, a video with ASR contains 269.8 speech sentences (vs 8.3 chapter titles), a speech sentence lasts 3.9 seconds (vs 142.0 seconds for chapters) in the video and contains 11.5 words (vs 5.4 words for chapters). In Figure 7.4, we also observe that the most common words in the ASR largely differ from the most common words in the chapter titles, which further shows the difference between these two types of data.

Biases. Using the langdetect [Danilák, 2021] language detection tool, we find that 92.9%/93.9% of total videos in VidChapters-7M have their chapter titles/ASR in English. However, as shown in Figure 7.3 (bottom right), the distribution of chapter languages includes a long tail of languages, e.g., 13 languages appear in more than 1K videos of VidChapters-7M. We also use GenBit [Sengupta, 2021] to measure gender bias in the chapters and ASR. We observe that the percentage of female/male/non-binary gendered words is 19.7%/39.7%/40.7% for the chapters, and 11.6%/35.6%/52.8% for the ASR.

Ethical considerations. We employ several techniques to identify harmful visual or language content. We use a classifier [Schuhmann, 2022] built on top of the previously extracted CLIP

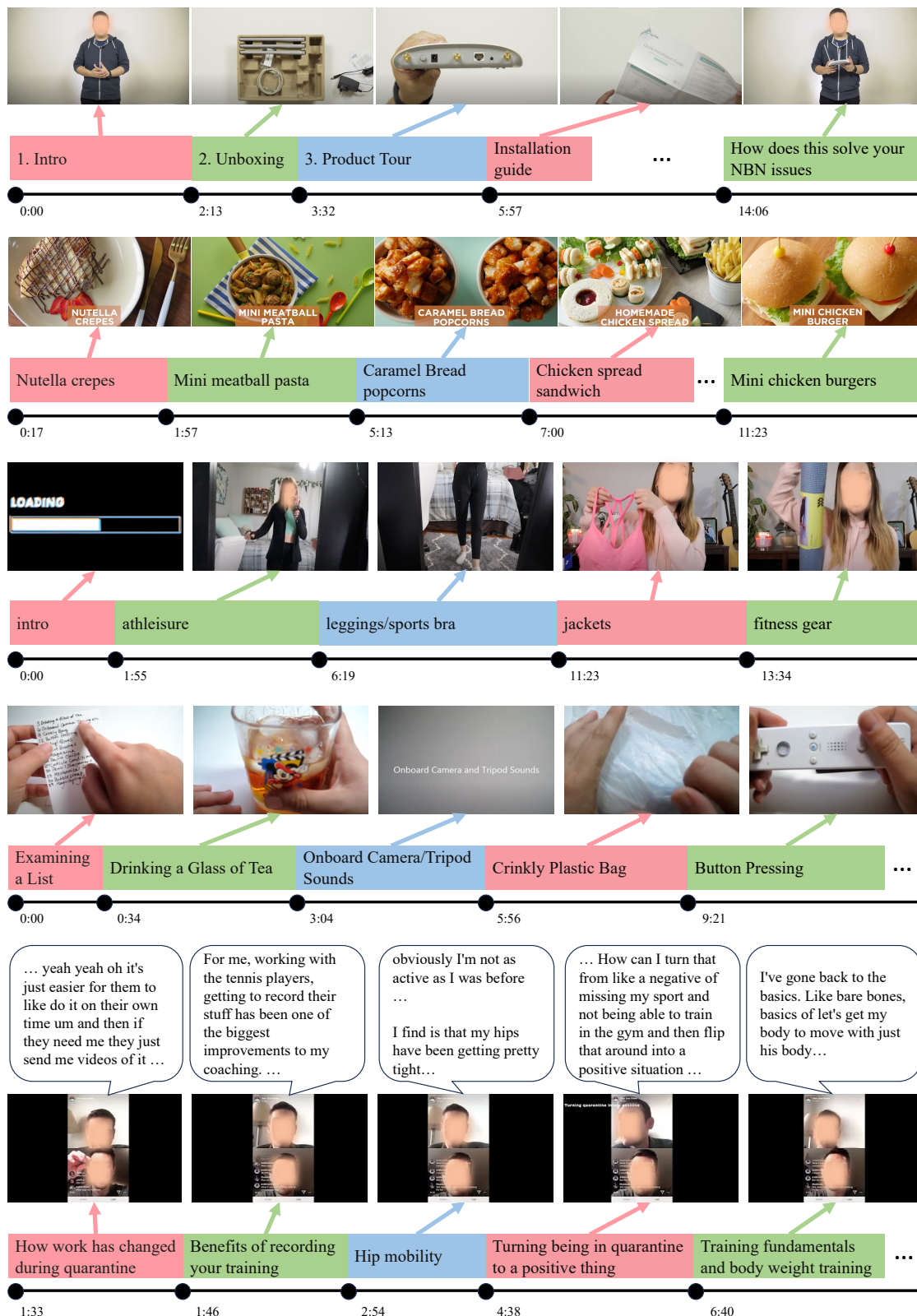


Figure 7.6: Additional examples of videos with user-annotated chapters in VidChapters-7M: Chapters depict visual events (e.g., the mini chicken burgers that appear in the second video), conversations (see the last video), or events in the raw audio (e.g., the sound of the crinkly plastic bag in the penultimate video) in various scenarios.

Type of chapter titles	Percentage
Speech and visual	49
Audio and visual	2
Speech-only	26
Visual-only	3
Audio-only	3
Structure-only	14
Unrelated	3

Table 7.2: **Manual assessment of the informativeness of chapter titles in the VidChapters-7M dataset over a random sample of 100 videos.** Video chapter titles can be based on speech and vision; audio and vision; vision, audio or speech alone; or only on the structure of the video (*e.g.* "step 1", "step 2" etc). In a small number of cases, video chapters are unrelated to the video content.

features to detect not-safe-for-work (NSFW) visual content (such as pornographic and sexualized content). In detail, we compute the NSFW score at every frame (at 1 FPS) and tag videos with an average score above 0.5. Moreover, we use a language model [Hanu, 2020] to detect toxic content in chapter titles and speech transcripts. In detail, we compute the toxicity score at every chapter title / ASR sentence and tag videos where the chapter titles / ASR have an average toxicity score above 0.5. These processes flag 5,716 (0.70%) visually NSFW videos, 355 (0.04%) videos with toxic chapter titles and 1,368 (0.17%) videos with toxic ASR. We assume the relatively low number of flagged videos is due to the regulations performed by the Web platform used to collect our dataset. Following [Schuhmann, 2022], we refrain from removing these samples to encourage research in fields such as dataset curation and tag them instead. Note that these automated filtering techniques are not perfect and that harmful content may pass.

Manual assessment of the quality of annotations. While chapter titles are manually written and uploaded by real users, sometimes chapter titles are not informative about the content of the video at the corresponding timestamps. To assess the quality of chapter title annotations in our dataset, we inspected a random sample of 100 videos in VidChapters-7M. For each video, we checked if the titles are related to the content of the video chapter and if so which video modalities (ASR, visual or raw audio) they are related to, or if they only refer to the structure of the video (*e.g.* chapter titles like "step 1", "step 2" etc). Results are presented in Table 7.2, and show that 83% of videos have chapters related to one or multiple modalities of the video, 14% of videos have chapters only referring to the structure of the video, and 3% of videos have chapters unrelated to the video content.

7.4 Experiments

In this Section, we present the results of models on VidChapters-7M for the full video chapter generation task in Section 7.4.1, the task of video chapter generation given ground-truth bound-

aries in Section 7.4.2 and the video chapter grounding task in Section 7.4.3. Finally, we study transfer learning from video chapter generation to dense video captioning tasks in Section 7.4.4.

Evaluation metrics. To evaluate the quality of the generated chapter titles (without their positions), we use standard metrics used for visual captioning: BLEU [Papineni, 2002] (B), CIDEr [Vedantam, 2015] (C), METEOR [Banerjee, 2005] (M) and ROUGE-L [Lin, 2004] (RL). To evaluate video chapter generation as a whole, including the locations of the generated chapters, we follow standard protocols used for dense video captioning, given the similar nature of the two tasks. We use the standard evaluation tool [Krishna, 2017] which calculates matched pairs between generated events and the ground truth across IoU thresholds of $\{0.3, 0.5, 0.7, 0.9\}$, and compute captioning metrics over the matched pairs. However, these metrics do not take into account the story of the video and give high scores to methods generating many redundant chapters. Hence for an overall evaluation, we also use SODA_c [Fujita, 2020] (S) which first tries to find a temporally optimal matching between generated and reference chapters to capture the story of a video, then computes METEOR scores for the matching and derives F-measure scores from the METEOR scores to penalize redundant chapters. To separately evaluate chapter localization, we report the recall ($R@Ks$, $R@K$) and the precision ($P@Ks$, $P@K$) across various thresholds in terms of the distance to the ground-truth start time or IoU with the ground-truth start-end window. We also report the average recall (R) and average precision (P) across IoU thresholds of $\{0.3, 0.5, 0.7, 0.9\}$.

Implementation details. Unless stated otherwise, for all models, we use the speech transcripts (ASR) and visual features extracted as explained in Section 7.3.2. By default, each model is taken from the corresponding official implementation, and all model hyper-parameters are set according to the original papers. We use the Adam optimizer [Kingma, 2015] for training and select the final model based on the best validation performance. Our experiments are run on 8 NVIDIA A100 80GB GPUs.

7.4.1 Video chapter generation

In this Section, we study the task of video chapter generation that requires temporally segmenting the video and generating a chapter title for each segment.

Models. For the video chapter segmentation subtask, we evaluate two zero-shot approaches (i.e., that are not trained on VidChapters-7M): speech text tiling [Hearst, 1997], which detects subtopic shifts based on the analysis of lexical co-occurrence patterns, and a visual scene change detection algorithm [Tomar, 2006] based on the sum of absolute differences. To derive zero-shot baselines for the full video chapter generation task, we combine text tiling and shot detection with various alternatives that can generate text given text or visual input: a random baseline that predicts a random speech sentence spoken inside the predicted boundaries, LLaMA-7B [Touvron, 2023] (prompted to summarize the speech transcript spoken inside the predicted boundaries) and BLIP-2 [Li, 2023a] (prompted to describe the middle video frame of

Method	Modalities	Pretraining Data	Finetuned	S	B1	B2	B3	B4	C	M	RL
Text tiling [Hearst, 1997] + Random	T	\emptyset	\times	0.4	0.6	0.2	0.1	0.0	0.8	0.7	0.6
Text tiling [Hearst, 1997] + LLaMA [Touvron, 2023]	T	Text mixture	\times	0.2	0.4	0.1	0.1	0.0	0.5	0.3	0.4
Shot detect [Tomar, 2006] + BLIP-2 [Li, 2023a]	V	129M img-txt	\times	0.6	0.7	0.3	0.1	0.1	0.2	0.6	0.8
Vid2Seq [Yang, 2023d]	T+V	C4 + HTM	\times	0.1	0.1	0.0	0.0	0.0	0.1	0.1	0.1
PDVC [Wang, 2021d]	V	\emptyset	\checkmark	6.8	9.4	3.7	1.4	0.9	35.8	9.4	11.4
Vid2Seq [Yang, 2023d]	T	C4	\checkmark	10.2	9.5	6.7	4.0	2.7	48.8	8.5	11.0
Vid2Seq [Yang, 2023d]	T	C4 + HTM	\checkmark	10.5	9.9	7.0	4.2	2.9	50.7	8.7	11.4
Vid2Seq [Yang, 2023d]	V	C4	\checkmark	3.1	2.3	1.5	0.6	0.5	10.9	2.2	2.9
Vid2Seq [Yang, 2023d]	V	C4 + HTM	\checkmark	5.5	4.5	2.8	1.2	0.9	21.1	4.1	5.5
Vid2Seq [Yang, 2023d]	T+V	C4	\checkmark	10.6	9.9	7.0	4.2	2.8	51.3	8.8	11.6
Vid2Seq [Yang, 2023d]	T+V	C4 + HTM	\checkmark	11.4	10.9	7.7	4.6	3.1	55.7	9.5	12.6

Table 7.3: **Video chapter generation (global metrics) on VidChapters-7M test set.** Here, finetuned refers to finetuning on the VidChapters-7M train set. T: Transcribed speech, V: Visual, HTM: HowTo100M [Miech, 2019].

Method	Modalities	Pretraining Data	Finetuned	R@5s	R@3s	R@0.5	R@0.7	P@5s	P@3s	P@0.5	P@0.7
Text tiling [Hearst, 1997]	T	\emptyset	\times	9.4	5.8	23.6	8.9	12.6	7.9	26.0	8.8
Shot detect [Tomar, 2006]	V	\emptyset	\times	31.2	27.4	24.9	12.5	33.2	29.7	18.0	8.7
Vid2Seq [Yang, 2023d]	T+V	C4 + HTM	\times	10.7	9.5	5.8	0.2	23.3	18.5	1.9	0.8
PDVC [Wang, 2021d]	V	\emptyset	\checkmark	21.1	17.8	31.2	22.5	45.3	40.2	47.2	26.9
Vid2Seq [Yang, 2023d]	T	C4	\checkmark	37.8	29.5	44.6	26.1	29.0	23.0	38.0	23.4
Vid2Seq [Yang, 2023d]	T	C4 + HTM	\checkmark	36.7	28.9	46.5	27.2	29.5	23.3	40.4	24.8
Vid2Seq [Yang, 2023d]	V	C4	\checkmark	35.3	26.4	23.6	8.7	17.9	13.6	17.2	7.1
Vid2Seq [Yang, 2023d]	V	C4 + HTM	\checkmark	33.5	25.0	33.0	14.5	19.5	14.7	26.2	12.5
Vid2Seq [Yang, 2023d]	T+V	C4	\checkmark	36.3	28.6	45.8	26.9	29.9	23.8	40.9	24.9
Vid2Seq [Yang, 2023d]	T+V	C4 + HTM	\checkmark	36.4	28.5	48.2	28.5	30.3	24.0	43.1	26.4

Table 7.4: **Video chapter generation (segmentation metrics) on VidChapters-7M test set.** Here, finetuned refers to finetuning on the VidChapters-7M train set. T: Transcribed speech, V: Visual, HTM: HowTo100M [Miech, 2019].

the predicted segment). Finally, we also train and evaluate two state-of-the-art end-to-end dense video captioning models on VidChapters-7M: PDVC [Wang, 2021d] which consists of a visual-only DETR-style [Carion, 2020] architecture and Vid2Seq [Yang, 2023d] which is a multi-modal sequence-to-sequence model pretrained on the C4 text corpus [Raffel, 2020] and on narrated videos with ASR (*e.g.*, YT-Temporal-1B [Zellers, 2022]). For Vid2Seq, we also report zero-shot results after pretraining on narrated videos without finetuning on VidChapters-7M.

Implementation details. We use the text tiling implementation from the NLTK library [Bird, 2009] which tokenizes the text into pseudosentences of size 50. We use the shot detection software from the FFmpeg library [Tomar, 2006] with a confidence threshold of 0.7. For LLaMA, we use the following prompt: `Summarize the following speech transcript in a chapter title. Transcript: <ASR> Chapter title:` where the ASR is the concatenation of all speech sentences spoken during a given video segment. For BLIP-2, we use the 3.4B-parameter variant with FLAN-T5-XL [Wei, 2022a] and CLIP ViT-L/14 [Radford, 2021; Dosovitskiy, 2021], and use the following prompt: `Summarize the image in a chapter title. Chapter title:,` and use the middle frame of the predicted video segment.

PDVC. We use PDVC’s official codebase. PDVC includes a caption decoder that relies on dataset-specific word vocabularies. To adapt PDVC to VidChapters-7M, we construct a vocabulary made with all words that appear at least 50 times in the dataset (33,598 words). We subsample or pad the sequence of frames to 100 frames. We use 100 queries and train with a constant learning rate of $5e^{-5}$, weight decay $1e^{-4}$ and batch size 1 on an NVIDIA V100 32GB (as the official codebase is not compatible with higher batch sizes or multi-gpu training). We train on VidChapters-7M for 5 epochs. The training on VidChapters-7M lasts about a week.

Vid2Seq. We reimplement Vid2Seq (originally released in Jax) in PyTorch. For initialization, we use the T5-Base language model pretrained on the C4 text corpus [Raffel, 2020]. Vid2Seq is originally pretrained on YT-Temporal-1B [Zellers, 2022] using a generative and denoising objective in the speech sequence. Due to computational limitations, we instead pretrain Vid2Seq on the smaller HowTo100M dataset [Miech, 2019] with the same objectives. Then we train Vid2Seq on VidChapters-7M with the next token prediction objective in the chapter sequence and the denoising objective in the speech sequence. We subsample or zero-pad the sequence of frames to 100 frames. The text encoder and decoder sequence are truncated or padded to 1000 and 256 tokens, respectively. For all datasets, we use a learning rate of $3e^{-4}$ warmed up linearly (from 0) for the first 10% of iterations and following a cosine decay (down to 0) for the remaining 90%. We train for 6/10 epochs on HowTo100M/VidChapters-7M. We use a batch size of 64 videos split on 8 NVIDIA A100 80GB for HowTo100M/VidChapters-7M. The training on HowTo100M or VidChapters-7M takes about 2 days.

Results. We report the results for video chapter generation using global metrics and localization-only metrics in Tables 7.3 and 7.4, respectively. We observe that models trained on VidChapters-7M outperform zero-shot baselines, demonstrating the effectiveness of training on VidChapters-7M. In particular, PDVC [Wang, 2021d] has the best precision and Vid2Seq [Yang, 2023d] achieves the best results in terms of overall generation and recall. We also find that Vid2Seq’s speech-only mode outperforms its visual-only mode and that using both speech and visual inputs leads to the best performance. This demonstrates that video chapter generation is a multi-modal task. Finally, we observe that pretraining using ASR in narrated videos from HowTo100M [Miech, 2019] improves the video chapter generation performance of the Vid2Seq model. Specifically, pretraining on HowTo100M is more beneficial for vision-aware models than for the speech-only model.

Results split by language. We report video chapter generation results on the VidChapters-7M dataset split by language for both English and German in Tables 7.5 and 7.6, respectively. We find that training on VidChapters-7M is beneficial for both languages. Interestingly, pretraining on HowTo100M (which is a dataset in English) improves results on English as well as German. We also observe that the quantitative results in German are lower than in English. Finally, we report results of the Vid2Seq model with the multi-lingual language model mT5 [Xue, 2021] pretrained on the multi-lingual dataset mC4 [Xue, 2021]. We find that this variant performs a bit worse on English but slightly better on German compared to the Vid2Seq variant based on

Method	Modalities	Pretraining Data	Finetuned	S	B1	B2	B3	B4	C	M	RL
Text tiling [Hearst, 1997] + Random	T	\emptyset	\times	0.5	0.8	0.2	0.1	0.0	0.9	0.8	0.7
Text tiling [Hearst, 1997] + LLaMA [Touvron, 2023]	T	Text mixture	\times	0.3	0.5	0.2	0.1	0.0	0.5	0.4	0.4
Shot detect [Tomar, 2006] + BLIP-2 [Li, 2023a]	V	129M img-txt	\times	1.3	1.5	0.7	0.4	0.2	4.7	1.4	1.6
PDVC [Wang, 2021d]	V	\emptyset	\checkmark	6.6	9.0	3.8	1.5	1.0	36.0	9.1	11.0
Vid2Seq [Yang, 2023d]	T+V	C4	\checkmark	10.8	10.3	7.6	4.9	3.4	54.8	9.1	11.9
Vid2Seq [Yang, 2023d] w/ mT5	T+V	mC4	\checkmark	10.4	9.9	7.2	4.7	3.3	52.0	8.7	11.3
Vid2Seq [Yang, 2023d]	T+V	C4 + HTM	\checkmark	11.5	11.1	8.1	5.1	3.6	58.8	9.7	12.8

Table 7.5: **Video chapter generation (global metrics) on the VidChapters-7M test set restricted to videos with English chapter titles and ASR.** Here, finetuned refers to finetuning on the VidChapters-7M train set. T: Transcribed speech, V: Visual, HTM: HowTo100M [Miech, 2019].

Method	Modalities	Pretraining Data	Finetuned	S	B1	B2	B3	B4	C	M	RL
Text tiling [Hearst, 1997] + Random	T	\emptyset	\times	0.6	1.7	1.3	1.3	1.1	12.8	1.5	1.6
Text tiling [Hearst, 1997] + LLaMA [Touvron, 2023]	T	Text mixture	\times	0.1	0.3	0.2	0.0	0.0	0.0	0.2	0.2
Shot detect [Tomar, 2006] + BLIP-2 [Li, 2023a]	V	129M img-txt	\times	0.6	0.4	0.2	0.0	0.0	1.3	0.6	0.5
PDVC [Wang, 2021d]	V	\emptyset	\checkmark	5.4	11.6	0.0	0.0	0.0	29.4	12.4	14.9
Vid2Seq [Yang, 2023d]	T+V	C4	\checkmark	9.1	8.4	5.2	1.0	0.9	34.1	6.1	10.1
Vid2Seq [Yang, 2023d] w/ mT5	T+V	mC4	\checkmark	8.8	8.1	5.9	1.7	1.8	38.4	6.1	10.1
Vid2Seq [Yang, 2023d]	T+V	C4 + HTM	\checkmark	10.9	9.6	5.4	1.7	1.7	43.2	8.1	8.1

Table 7.6: **Video chapter generation (global metrics) on the VidChapters-7M test set restricted to videos with German chapter titles and ASR.** Here, finetuned refers to finetuning on the VidChapters-7M train set. T: Transcribed speech, V: Visual, HTM: HowTo100M [Miech, 2019].

T5 pretrained on the C4 corpus.

Qualitative examples. We present qualitative results for video chapter generation in Figures 7.7. Compared with the speech-only model, a key advantage of the speech+visual video chapter generation model is that it can generalize to videos that do not have ASR input, as shown in the first example of Figure 7.7. Compared with the visual-only variant, the multi-modal model can also benefit from speech cues, as seen in the second example in Figure 7.7.

7.4.2 Video chapter generation given ground-truth boundaries

In this Section, we study the task of generating chapter titles provided correct temporal boundaries of video chapters. This task is a simplification of the previously studied task where we assume perfect temporal segmentation.

Models and implementation details. We adopt the same models and implementation details as previously introduced in Section 7.4.1, except for a few differences described next. To adapt the Vid2Seq model pretrained on HowTo100M (see Section 7.4.1) to video chapter generation with ground-truth boundaries, we remove the model weights corresponding to the time tokens (in the token embedding layers and the token prediction layer). We train for 20 epochs

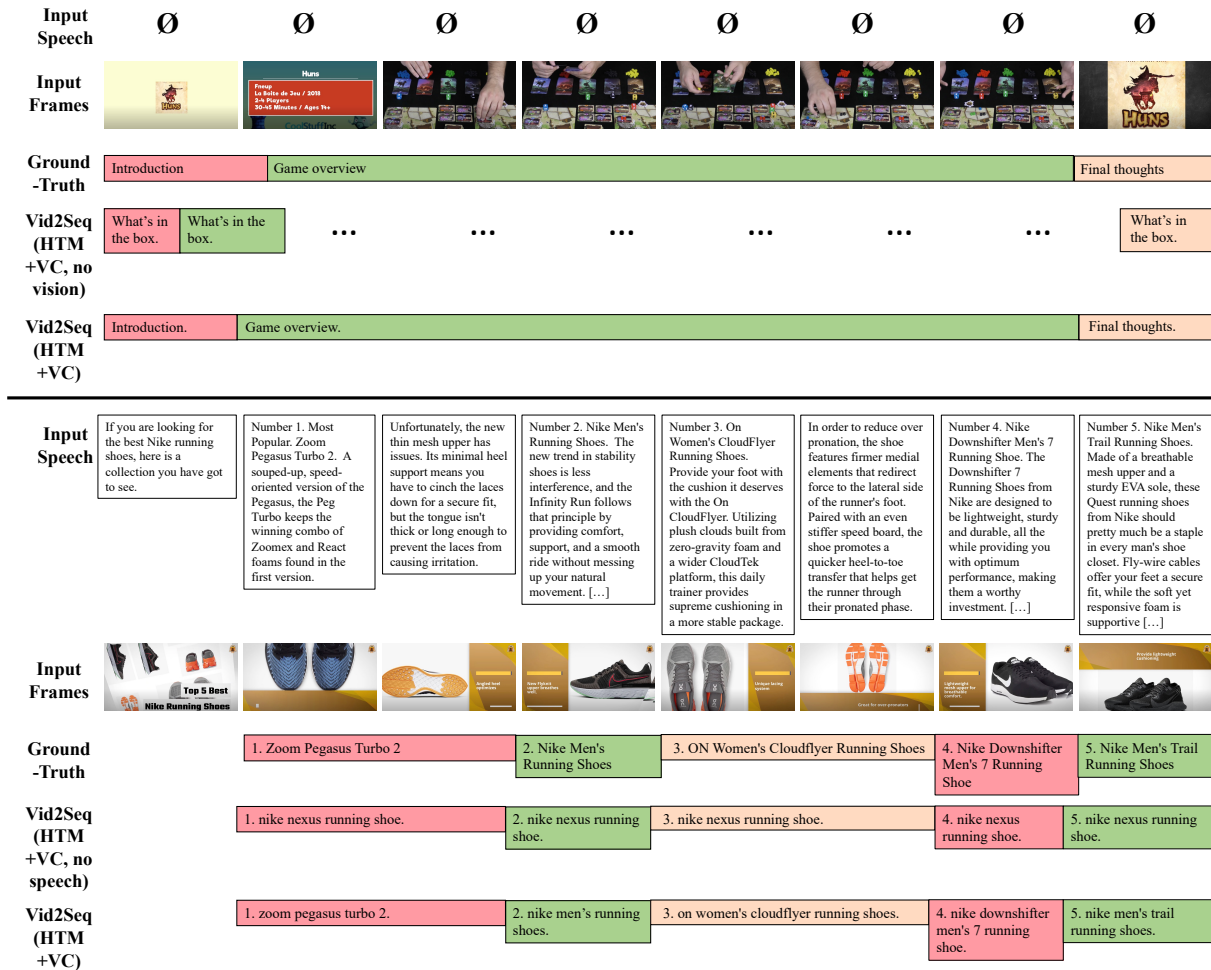


Figure 7.7: Examples of video chapter generation using the Vid2Seq model with different input modalities compared with ground-truth on the test set of VidChapters-7M. The first example shows that the Vid2Seq variant with both speech and visual modalities "Vid2Seq (HTM+VC)" can predict the structure of the input video without the ASR input, unlike the Vid2Seq speech-only variant "Vid2Seq (HTM+VC, no vision)". The second example shows that the Vid2Seq variant with both speech and visual modalities "Vid2Seq (HTM +VC)" can effectively leverage speech cues to detect the names of the depicted and discussed shoes, unlike the Vid2Seq visual-only variant "Vid2Seq (HTM+VC, no speech)".

Method	Modalities	Pretraining Data	Finetuned	B1	B2	B3	B4	C	M	RL
Random	Speech	\emptyset	\times	2.4	1.3	0.9	0.7	10.4	2.2	4.4
LLaMA [Touvron, 2023]	Speech	Text mixture	\times	0.0	0.0	0.0	0.0	0.0	0.1	0.2
BLIP-2 [Li, 2023a]	Visual	129M image-texts	\times	3.1	1.5	0.9	0.7	12.4	2.2	4.5
Vid2Seq [Yang, 2023d]	Speech+Visual	C4 + HowTo100M	\times	2.0	1.2	0.9	0.6	0.9	0.3	0.6
Vid2Seq [Yang, 2023d]	Speech	C4 + HowTo100M	\checkmark	21.0	15.5	12.1	10.0	105.3	11.5	24.5
Vid2Seq [Yang, 2023d]	Visual	C4 + HowTo100M	\checkmark	10.1	5.6	3.5	2.4	47.1	5.1	14.7
Vid2Seq [Yang, 2023d]	Speech+Visual	C4	\checkmark	21.6	15.7	12.3	10.0	110.8	11.5	26.0
Vid2Seq [Yang, 2023d]	Speech+Visual	C4 + HowTo100M	\checkmark	23.5	17.2	13.4	11.0	120.5	12.6	28.3

Table 7.7: **Chapter title generation given ground-truth boundaries on VidChapters-7M test set.** Here, finetuned refers to finetuning on the VidChapters-7M train set, and speech refers to transcribed speech.

on VidChapters-7M using the next token prediction objective in the sequence of tokens corresponding to a single chapter title. We construct training batches by sampling a chapter title with its associated video clip at each iteration (i.e., an epoch corresponds to seeing one chapter title for all videos). The text encoder and decoder sequence are truncated or padded to 256 and 32 tokens, respectively. We use a learning rate of $3e^{-4}$ warmed up linearly (from 0) for the first 10% of iterations and following a cosine decay (down to 0) for the remaining 90%. We use a batch size of 512 videos split on 8 NVIDIA A100 80GB for VidChapters-7M. The training takes about a day.

Results. We report results for video chapter generation given ground-truth boundaries in Table 7.7. Similar to the full video chapter generation task, we observe that solving the task without training on VidChapters-7M is hard. Indeed, LLaMA [Touvron, 2023] struggles to summarize the speech content into a chapter title and underperforms the random baseline. Furthermore, BLIP-2 [Li, 2023a] slightly improves over the random baseline. In addition, Vid2Seq [Yang, 2023d] in zero-shot mode underperforms the random baseline due to the large domain gap between ASR and chapter titles (see Section 7.3.3). In comparison, the performance of models trained on VidChapters-7M is significantly higher. Moreover, Vid2Seq’s speech-only mode outperforms its visual-only mode, and using both speech and visual inputs is beneficial, confirming the benefit of multi-modal reasoning for the task of generating chapter titles. Finally, pretraining on narrated videos from HowTo100M [Miech, 2019] improves the performance of the Vid2Seq model on VidChapters-7M.

7.4.3 Video chapter grounding

In this Section, we study the task of video chapter grounding that requires a model to temporally localize a chapter start time (or start-end window) given an annotated chapter title (query). Hence, compared to the video chapter generation task, we here assume chapter titles to be given and focus on the temporal chapter localization only.

Models. We evaluate three zero-shot alternatives: a random baseline that randomly picks the timestamps of a speech sentence in the video, a BERT [Devlin, 2019] baseline that picks the

Method	Modalities	Pretraining Data	Finetuned	R@10s	R@5s	R@3s	R@1s	R@0.3	R@0.5	R@0.7	R@0.9
Random	T	\emptyset	\times	3.1	1.8	1.2	0.6	0.7	0.3	0.1	0.0
BERT [Devlin, 2019]	T	BookCorpus + Wikipedia	\times	9.0	6.8	5.4	2.9	0.6	0.3	0.1	0.0
CLIP [Radford, 2021]	V	400M img-txt	\times	8.1	5.2	3.7	1.4	10.7	5.2	2.3	0.5
Moment-DETR [Lei, 2021a]	V	5.4K narrated videos [Lei, 2021a]	\times	3.2	1.6	1.1	0.5	11.3	3.6	0.8	0.1
Moment-DETR [Lei, 2021a]	V	\emptyset	\checkmark	21.8	15.5	12.4	8.3	37.4	27.3	17.6	6.4

Table 7.8: **Video chapter grounding on VidChapters-7M test set.** Here, finetuned refers to finetuning on the VidChapters-7M train set. T: Transcribed speech, V: Visual.

timestamps of the speech sentence that has the closest text embedding with the queried chapter title, and a CLIP [Radford, 2021] baseline picking the frames where the query-frame similarity score drops from the highest scoring frame by a certain threshold ϵ . We also train and evaluate on VidChapters-7M a state-of-the-art end-to-end video grounding model: Moment-DETR [Lei, 2021a] which is designed for moment retrieval based on visual inputs. Furthermore, we report zero-shot performance of Moment-DETR obtained with the model checkpoint from Lei et al. [Lei, 2021a] pretrained on 5.4K narrated videos with ASR from the QVHighlights dataset [Lei, 2021a].

Implementation details. We use the [CLS] token sequence embedding for the BERT baseline and a threshold of $\epsilon = 0.05$ for the CLIP baseline. **Moment-DETR.** We use Moment-DETR’s official codebase. We train with the AdamW optimizer [Loshchilov, 2019], a constant learning rate of $3e^{-4}$, and a batch size of 256 videos split on 8 NVIDIA A100 80GB. We use a FPS of 1/3 and subsample or zero-pad the sequence of frames to 1200 frames. We use a maximum number of text query tokens of 77. We train for 50 epochs on VidChapters-7M, where an epoch corresponds to seeing one chapter title for all videos, which takes about 2 days.

Results. We report results for the video chapter grounding task in Table 7.8. We first observe that the simple zero-shot baselines based on ASR can decently find start times, but struggle to predict start-end windows due to the important domain gap between ASR and video chapters (see Section 7.3.3). The CLIP [Radford, 2021] baseline slightly underperforms the BERT baseline [Devlin, 2019] at retrieving start times, but is much better at finding start-end windows. Furthermore, the Moment-DETR model [Lei, 2021a] trained on VidChapters-7M outperform the zero-shot baselines for both localization of start times and start-end windows, which further demonstrates the effectiveness of training on VidChapters-7M. Finally, we note that Moment-DETR cannot handle speech inputs, but hope that our results showing the benefit of this modality on other tasks in VidChapters-7M will foster research in the localization of language queries in untrimmed videos using multi-modal inputs (vision and speech transcripts).

Method	Modalities	Pretraining Data	YouCook2 (val)					ViTT (test)				
			S	C	M	R	P	S	C	M	R	P
PDVC [Wang, 2021d]	V	\emptyset	4.4	22.7	4.7	—	—	—	—	—	—	
E2ESG [Zhu, 2022b]	T+V	C4 + WikiHow	—	25.0	3.5	20.7	20.6	—	25.0	8.1	32.2	32.1
Vid2Seq [Yang, 2023d]	T+V	C4 + HTM	8.3	48.3	9.5	27.1	27.0	—	—	—	—	—
Vid2Seq [Yang, 2023d]	T+V	C4 + YT-Temporal-1B	7.9	47.1	9.3	27.9	27.8	13.5	43.5	8.5	42.6	46.2
PDVC [†]	V	\emptyset	4.8	28.8	5.8	22.6	33.1	9.4	40.6	16.5	19.2	37.4
PDVC [†]	V	VC (Chap.)	5.9	34.7	7.5	28.8	36.4	10.1	41.5	16.1	21.3	37.2
Vid2Seq [†]	T+V	C4 + HTM	8.6	53.2	10.5	29.2	26.2	14.1	44.8	8.7	43.8	44.5
Vid2Seq [†]	T+V	C4 + VC (ASR+Chap.)	9.8	62.9	11.7	32.5	30.1	15.1	50.9	9.6	45.1	46.7
Vid2Seq [†]	T+V	C4 + HTM + VC (ASR)	8.4	50.1	10.3	29.7	26.3	14.3	45.6	8.8	43.7	44.9
Vid2Seq [†]	T+V	C4 + HTM + 1% of VC (ASR+Chap)	8.8	52.7	10.4	29.3	27.6	13.5	41.6	8.2	44.7	42.1
Vid2Seq [†]	T+V	C4 + HTM + 10% of VC (ASR+Chap.)	9.9	63.9	12.1	32.4	31.4	14.5	47.4	9.2	45.3	45.9
Vid2Seq [†]	T+V	C4 + HTM + VC (ASR+Chap.)	10.3	67.2	12.3	34.0	31.2	15.0	50.0	9.5	45.5	46.9

Table 7.9: **Comparison with the state of the art on the YouCook2 and ViTT dense video captioning benchmarks.** T: Transcribed speech, V: Visual, HTM: HowTo100M [Miech, 2019], VC: VidChapters-7M, Chap.: Chapters. [†] denote results of our experiments.

Method	Modalities	Pretraining Data	YouCook2 (val)					ViTT (test)				
			S	C	M	R	P	S	C	M	R	P
Text tiling [Hearst, 1997] + Random	T	\emptyset	0.3	0.9	0.3	3.8	6.6	0.3	0.6	0.6	11.6	24.4
Text tiling [Hearst, 1997] + LLaMA [Touvron, 2023]	T	Text mixture	0.2	0.6	0.2	3.8	6.6	0.2	0.6	0.5	11.6	24.4
Shot detect [Tomar, 2006] + BLIP-2 [Li, 2023a]	V	129M image-texts	0.6	1.0	0.5	8.9	5.5	0.2	0.1	0.2	3.1	13.7
Vid2Seq [Yang, 2023d]	V	C4 + VC (ASR)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.8
Vid2Seq [Yang, 2023d]	V	C4 + VC (Chap.)	0.7	1.1	0.5	21.3	8.6	1.5	1.9	0.6	18.9	10.4
Vid2Seq [Yang, 2023d]	T+V	C4 + HTM	0.0	0.1	0.0	0.5	0.6	0.0	0.0	0.0	0.5	1.0
Vid2Seq [Yang, 2023d]	T+V	C4 + VC (ASR)	0.1	0.1	0.0	1.1	0.9	0.0	0.0	0.0	0.7	0.6
Vid2Seq [Yang, 2023d]	T+V	C4 + VC (Chap.)	0.1	0.2	0.1	0.7	1.4	0.7	1.1	0.3	14.3	12.8
Vid2Seq [Yang, 2023d]	T+V	C4 + VC (ASR+Chap.)	3.2	10.2	2.9	20.6	19.7	9.1	30.2	6.7	33.8	40.8
Vid2Seq [Yang, 2023d]	T+V	C4 + HTM + VC (ASR)	0.0	0.1	0.0	1.2	0.9	0.0	0.0	0.0	0.8	0.7
Vid2Seq [Yang, 2023d]	T+V	C4 + HTM + 1% of VC (ASR+Chap.)	2.7	7.2	2.1	18.1	17.3	5.5	15.5	4.3	31.3	37.1
Vid2Seq [Yang, 2023d]	T+V	C4 + HTM + 10% of VC (ASR+Chap.)	3.2	11.5	3.0	19.4	19.2	6.4	21.6	5.3	31.0	38.2
Vid2Seq [Yang, 2023d]	T+V	C4 + HTM + VC (ASR+Chap.)	3.9	13.3	3.4	22.3	20.1	9.0	28.0	6.5	33.7	40.1

Table 7.10: **Zero-shot dense video captioning on the YouCook2 and ViTT benchmarks.** T: Transcribed speech, V: Visual, HTM: HowTo100M [Miech, 2019], VC: VidChapters-7M, Chap.: Chapters.

7.4.4 Transfer learning on dense video captioning

In this Section, we investigate the pretraining of video-language models on our new VidChapters-7M. To this end, we adopt video chapter generation models trained on VidChapters-7M (see Section 7.4.1) to the tasks of dense video captioning with or without finetuning.

Datasets. We use two dense video captioning datasets. **YouCook2** [Zhou, 2018a] has 2K untrimmed videos of cooking procedures. On average, each video lasts 320s and is annotated with 7.7 temporally-localized sentences. **ViTT** [Huang, 2020b] was created to better reflect the distribution of instructional videos in the wild compared to YouCook2, and consists of 8K untrimmed instructional videos. On average, each video lasts 250s and is annotated with 7.1 temporally-localized short tags. For both datasets, we extract speech transcripts and visual features as described in Section 7.3.2, and follow the standard splits for training, validation and testing. Note that we only use videos available on YouTube at the time of the work, resulting in 10 to 20% less videos than in the original datasets.








Implementation details. We adopt the same models and implementation details as previously introduced in Section 7.4.1, except for a few differences described next.

PDVC. To adapt PDVC to YouCook2/ViTT, we construct a vocabulary made with all words that appear at least 2/3 times in the dataset (3,815/1,607 words). For transfer learning from VidChapters-7M to YouCook2/ViTT, we initialize the downstream dataset-specific word embedding layer with the weights of the corresponding word embedding in the pretrained model. We train on YouCook2/ViTT for 30 epochs.

Vid2Seq. Finetuning on YouCook2/ViTT is done with the next token prediction objective in the dense video captioning sequence and the denoising objective in the speech sequence. We train for 40/20 epochs on YouCook2/ViTT. We use a batch size of 16 videos split on 8 NVIDIA V100 32GB for YouCook2/ViTT.

Results after finetuning. In Table 7.9, we show that pretraining for video chapter generation on VidChapters-7M greatly improves the downstream dense video captioning performance compared to training from scratch or pretraining only with ASR data as done in previous work [Yang, 2023d]. We also find that pretraining both on HowTo100M [Miech, 2019] and VidChapters-7M results in the best overall performance. In particular, the Vid2Seq model pretrained on both HowTo100M and VidChapters-7M largely improves the state of the art on both the YouCook2 and ViTT benchmarks. In detail, on the YouCook2 benchmark, in the setting with C4 + HowTo100M pretraining, we observe that a boost of about 4.9 points in CIDEr is obtained with our reimplementation of Vid2Seq, and that 14.0 additional points in CIDEr are obtained by pretraining on VidChapters-7M. Finally, we report the results of the Vid2Seq model after pretraining on different fractions of VidChapters-7M for a fixed number of iterations. We construct these subsets such that larger subsets include the smaller ones. These results suggest that the scale of the chapter dataset is an important factor in the downstream dense video captioning performance. We conclude that VidChapters-7M opens a promising avenue for multi-modal pretraining. We further show qualitative examples of dense video captioning in Figure 7.8. We observe that the dense video captioning model pretrained on VidChapters-7M is more accurate and hallucinates less than the variant not pretrained on VidChapters-7M.

Zero-shot dense video captioning. In Table 7.10, we report results obtained by directly applying video chapter generation models trained on VidChapters-7M for dense video captioning without finetuning for this task. As far as we know, our work is the first to explore this challenging zero-shot setting where no manual annotation of dense video captions is used for training. The Vid2Seq model trained only using ASR data underperforms the random baseline, due to the large domain difference between speech transcripts and dense captions [Yang, 2023d]. In the visual-only setting, the variant trained on chapter annotations is better than the variant trained on ASR annotations. In the visual+speech settings, only using chapter annotations does not perform well, as training only on chapters (i.e., without speech) does not enable the model to learn how to use the input speech modality at inference. However, using both ASR and chapter annotations results in a largely better zero-shot dense video captioning performance

Input Speech	Okay, so let's get started. So I'm going to heat our pan over medium heat. Once again, keep the temperature low because we are using olive oil. You don't want to cook olive oil over too high of heat.	We'll add in two teaspoons of olive oil to coat the pan. We'll let that heat up, then we're going to add the other ingredients. So come on over.	Okay, now that the pan has been heated over medium heat, let's add in the onions and let's saute these for about two minutes. Get them soft, then we'll add in the other ingredients.	Now that the onions are nice and sauteed and smelling delicious, let's add in our red pepper.	And then we'll add in our sweet potatoes, and then about a tablespoon of water.	So what we're hoping for in that 15 minutes is that these sweet potatoes will become soft and ready to eat.	Whatever you want to call it, it smells delicious, it looks pretty cool. Add this as a side dish to any protein source, so a steak, chicken, or have it in the morning with your eggs and you're good to go.
Input Frames							
Ground -Truth			Add some chopped white onions in a pan under medium heat.	Add in red pepper and sweet potatoes.	Add a spoon of water.	Cook with lid on.	Season it with salt and black pepper.
Vid2Seq (HTM)	Heat a pan with olive oil.	Add 2 tbsp of olive oil to the pan.	Add onions to the pan and saute.	Add red pepper sweet potatoes and water to the pan.	Add red pepper and water to the pan.	Cook the sweet potatoes in the pan.	
Vid2Seq (HTM+VC)	Heat a pan over medium heat.	Add olive oil to the pan.	Add onions to the pan.	Add red pepper to the pan.	Add sweet potatoes and water to the pan.	Cover the pan with a lid.	









Input Speech	When you're making mashed potatoes, if you're using the wrong potato and you're throwing them in a food processor, you're doing it all wrong. The first mistake people tend to make is they pick the wrong potato. [...]	Next you want to peel, rinse, and most important, cut your potatoes into nice, even chunks. That way they'll all cook at the same rate, and you won't get any weird little nasty hard bits in your potato.	Next comes a crucial step. You want to cook these potatoes until they are falling apart. Next comes a crucial step. You want to cook these potatoes until they are falling apart more than if you just put them in boiling water.	You want to salt the water because the salt starts to disintegrate the potatoes too. But the most important thing is that you really cook those potatoes about 25 minutes.	After you bring them to a boil, let them simmer long and slow until the edges start to fall off, and when you stick a knife in the middle, it just all falls apart. Drain the potatoes, and then put them right back into that hot pot and start to dry them	out over very low heat until they really start falling apart, turning white on the edges and practically turning into mashed potatoes right then and there in the pot. To finish the job, you can use a hand masher or you can use a ricer for super smooth, silky potatoes.	So now it's time to finally add your cream or milk. You want to make sure it's warm so it doesn't cool down your potatoes, and the most important thing is you want to make sure you use lots. [...]	Don't worry, it can look like soup at first, but if you keep stirring, you'll see those potatoes just drink up all that cream. The final step for making great mashed potatoes is to add a few fresh herbs, chives are nice [...]
Input Frames								
Ground -Truth		Peel and cut potatoes into chunks.		Put in cold water and cook to a boil and salt the water.	Drain and dry the potatoes.	Mash the potatoes well with a hand masher.	Add milk and stir the potatoes.	Season the potatoes with some chopped parsley leaves.
Vid2Seq (HTM)		Peel rinse and cut the potatoes.	Boil the potatoes in water.	Boil the potatoes in water.	Drain the potatoes and dry them out in a pot.	Add cream and milk to the potatoes.	Add chives parsley and butter to the potatoes.	
Vid2Seq (HTM+VC)		Peel rinse and cut the potatoes.	Boil the potatoes in water and salt the water.		Drain the potatoes and dry them out.	Add milk and mash the potatoes.	Add fresh herbs parsley and butter to the mashed potatoes.	

Figure 7.8: Examples of dense event captioning of the Vid2Seq model pretrained on VidChapters-7M (vs. not pre-trained), compared with ground-truth, on the validation set of YouCook2. We find that the model pretrained on VidChapters-7M "Vid2Seq (HTM+VC)" is more accurate and less prone to hallucination. For instance, in the first example (top), the non-VC-pretrained model "Vid2Seq (HTM)" predicts "Add red pepper sweet potatoes and water to the pan." before the sweet potatoes are actually thrown into the pan. In the second example (bottom), the non-VC-pretrained model "Vid2Seq (HTM)" predicts the event "Boil the potatoes in water." twice and predicts the event "Add chives parsley and butter to the potatoes." before it actually happens. The VC-pretrained model "Vid2Seq (HTM+VC)" produces more accurate predictions.

and outperforms all baselines not trained on VidChapters-7M, demonstrating the complementary nature of the ASR and chapters annotations. Finally, we also observe the benefits of increasing the size of the pretraining dataset of chapters in this setting.

7.5 Conclusion

In this work, we present VidChapters-7M, a large-scale dataset of user-chaptered videos. Furthermore, we evaluate a variety of baselines on the tasks of video chapter generation with and without ground-truth boundaries and video chapter grounding. Finally, we investigate the potential of VidChapters-7M for pretraining video-language models and demonstrate improved performance on the dense video captioning tasks. VidChapters-7M thus provides a new resource to the research community that can be used both as a benchmark for the video chapter generation tasks and as a powerful means for pretraining generic video-language models.

Limitations. As it is derived from YT-Temporal-180M [Zellers, 2021], VidChapters-7M inherits the biases in the distribution of video categories reflected in this dataset. Moreover, the state-of-the-art models [Lei, 2021a; Wang, 2021d; Yang, 2023d] evaluated in this work are originally designed for other tasks. It is possible that models specifically designed for chaptering tasks may perform better.

Chapter 8

Conclusions

In this chapter, we provide a summary of contributions in Section 8.1 and discuss future work in Section 8.2.

8.1 Contributions

We provide below a summary of the contributions presented in this thesis.

1. **Automatic generation of video question answering data (Just Ask).** In Chapter 3, we propose an automatic pipeline that leverages text-only models for generating video question answering triplets from narrated videos. We show that a video-question transformer trained contrastively with an answer transformer on the generated data is capable of answering visual questions in a zero-shot manner (without training on a single manually annotated image or video) better than appropriate baselines. Furthermore, our method achieves competitive results on four existing video question answering benchmarks. Moreover, we extend our data generation approach to web video-caption pairs. For a detailed evaluation, we also introduce iVQA, a new video question answering benchmark with reduced language biases and high-quality redundant manual annotations.
2. **Frozen bidirectional language models for video question answering (Frozen-BiLM).** In Chapter 4, we propose the FrozenBiLM model based on frozen bidirectional language models (BiLM) for zero-shot video question answering. Our method consists in: (i) combining visual inputs with the frozen BiLM using light trainable modules, (ii) training such modules using web-scraped multi-modal data, and finally (iii) performing zero-shot video question answering inference through masked language modeling, where the masked text is the answer to a given question. FrozenBiLM outperforms prior autoregressive models and the prior state of the art for zero-shot video question answering on eight video question answering datasets. It also demonstrates competitive performance in the few-shot and fully-supervised settings.

3. **Transformers for spatio-temporal video grounding (TubeDETR).** In Chapter 5, we propose TubeDETR, an end-to-end transformer-based architecture for spatio-temporal video grounding. Its key components are: (i) an efficient video and text encoder that models spatial multi-modal interactions over sparsely sampled frames and (ii) a space-time decoder that jointly performs spatio-temporal localization. We demonstrate the advantage of our proposed components through an extensive ablation study. With image-text pretraining [Kamath, 2021], TubeDETR improves over the prior state of the art on the challenging VidSTG and HC-STVG benchmarks.
4. **Pretraining a visual language model for dense video captioning (Vid2Seq).** In Chapter 6, we propose Vid2Seq, a visual language model that can densely caption a video by generating a single sequence of tokens. The Vid2Seq architecture augments a language model with special time tokens, allowing it to seamlessly predict event boundaries and textual descriptions in the same output sequence. Vid2Seq can be effectively pretrained on unlabeled narrated videos at scale, by reformulating sentence boundaries of transcribed speech as pseudo event boundaries, and using the transcribed speech sentences as pseudo event captions. The resulting Vid2Seq model pretrained on the YT-Temporal-1B dataset [Zellers, 2022] improves over the prior state of the art on the YouCook2, ViTT and ActivityNet Captions dense video captioning benchmarks. Vid2Seq also generalizes well to the tasks of video paragraph captioning and video clip captioning, and to few-shot settings.
5. **VidChapters7M: a large-scale dataset of user-chaptered videos.** In Chapter 7, we propose VidChapters-7M, a large-scale dataset of user-chaptered videos. VidChapters-7M is automatically created from videos online in a scalable manner by scraping user-annotated chapters and hence without any additional manual annotation. We introduce the tasks of video chapter generation with or without ground-truth boundaries and video chapter grounding. We benchmark both simple baselines as well as state-of-the-art video-language models, including Vid2Seq, on these tasks. We also show that pretraining Vid2Seq on VidChapters-7M transfers well to dense video captioning tasks both in the zero-shot and finetuning settings, largely improving the state of the art on the YouCook2 and ViTT benchmarks. Finally, our experiments reveal that downstream performance scales well with the size of the pretraining dataset.

8.2 Future work

We here discuss several promising future direction for future work related to this thesis.

8.2.1 Localized video dialog

In Chapter 6, we presented a model capable of grounding the captions it generates temporally in the video. However this model cannot be prompted with specific textual instructions as it is

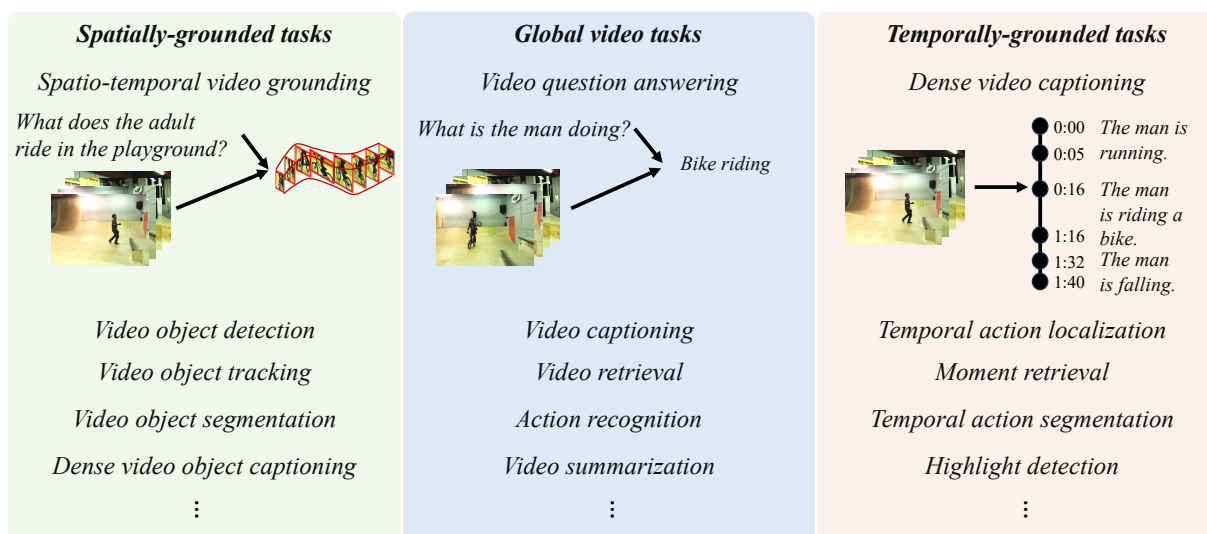


Figure 8.1: Video models are still highly specialized. Can we build a video model that unifies spatially-grounded tasks (left), global video tasks (middle) and temporally-grounded tasks (right) all together?

trained specifically for dense video captioning. Therefore this model cannot be used for dialog applications either, as such applications require understanding past textual context. It would be interesting to develop alternative models that can be prompted, for instance to focus more on specific events in a video (such as sporting events, social events), which may also enable them to be potentially used for dialog applications as well.

From another side, a recent work [Zhou, 2023] has studied the new task of dense video object captioning – detecting, tracking, and captioning trajectories of all objects in a video. The model designed in this work can spatially localize the text it generates, but is not capable of temporal localization. A natural question emerges: can we build a visual language model that can ground the text it generates both spatially and temporally in the video?

All in all, pursuing these research directions may enable to build flexible visual language models capable of dialog while referring to precise spatio-temporal locations in a video. This is unlike most current visual language models which are focused on global visual understanding [Alayrac, 2022; Li, 2023a; Liu, 2023]. Moreover, the consistency between the predicted spatio-temporal locations and the generated text may be used to correct model hallucinations [Alayrac, 2022; Ji, 2023], which is becoming an increasingly important research direction as language models are being more widely used.

8.2.2 Unified video model

Lately, image models have known a great unification. For instance, models like FIBER [Dou, 2022a] and GLIPv2 [Zhang, 2022b] are capable of serving both localization tasks like object detection and vision-language understanding tasks like VQA and image captioning. Moreover, with one suite of parameters, X-Decoder [Zou, 2023] supports all types of image segmentation tasks ranging from open-vocabulary instance/semantic/panoptic segmentation to referring

segmentation, and vision-language tasks including image-text retrieval, and image captioning.

In comparison, video models are still highly specialized. Actually, recent unified video frameworks [Li, 2023c; Wang, 2023; Zellers, 2022] only integrate global video-language tasks like question answering and captioning. Meanwhile, different models [Zhang, 2020a; Zhang, 2020c; Zeng, 2020] are being developed for temporally-grounded tasks like temporal language grounding. Furthermore, no model is capable of doing both temporal action localization [Caba Heilbron, 2015] and dense video captioning [Krishna, 2017] despite the similarities between the two tasks which both require temporal localization.

Therefore an interesting direction consists in designing unified video models that can flexibly handle both localization and understanding tasks, see Figure 8.1. To achieve this, we would certainly need advances in the scalable training of video models and to discover a video representation that is both efficient and compressed.

8.2.3 Processing long videos

Designing unified video models notably requires being able to process videos with varying duration. The standard approach to video modeling - which we use for all models developed in this thesis - consists in sampling temporally equally spaced video frames covering the full video. However, such an approach may miss important details especially when applied to long videos. Recent works have proposed promising alternatives. For instance, [Kim, 2023] apply a video language model to a subset of frames selected with a non-parametric frame retriever conditioned on video features and a text query. [Yu, 2023] use a similar scheme, but the subset of frames is selected using a model specifically trained for this. However, these approaches have mostly been evaluated on video question answering tasks using relatively short videos (several minutes at most). It remains unclear if these approaches would work when tackling other video tasks such as those depicted in Figure 8.1, or processing longer videos like movies. Another promising approach is the concept of memory [Cheng, 2022b; Wu, 2022], which could effectively encode long sequences of video frames. However, there is still work to be done to have a unified framework that works well across tasks.

8.2.4 Model-assisted annotation of video datasets

A key challenge in training video models is to collect the data required to train video models that can generalize well. Additionally, the annotation burden is compounded in the case of long videos. In Chapter 6, we presented an approach that uses language models to generate video question answering training data. The recent improvements in language models like GPT-4 [OpenAI, 2023b] open new possibilities in using language models to generate training data [Peng, 2023]. For instance, [Liu, 2023] generate conversations, detailed descriptions and complex reasoning texts about images using GPT-4. This is done by representing an image as a text that contains human-annotated captions and bounding boxes. A few other works have explored using similar generating similar type of data for visual instruction tuning [Zhang, 2023; Zhu, 2023]. Moreover, recent works [Chen, 2023a; Gao, 2023; Li, 2023b; Yang, 2023e] have also

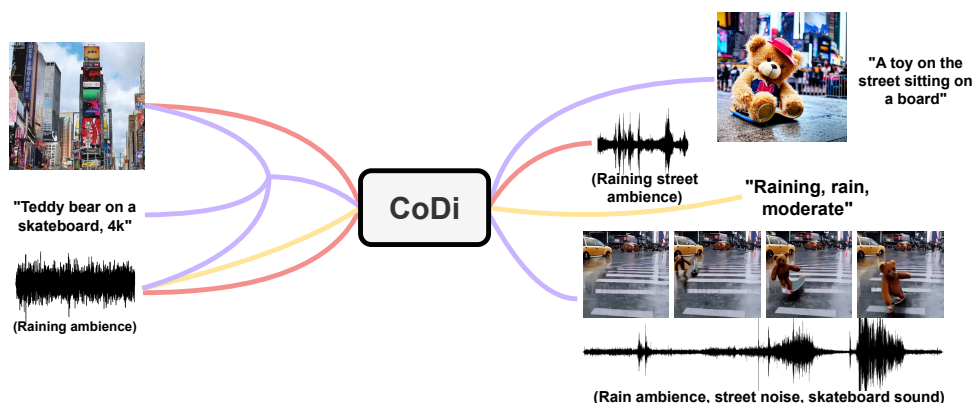


Figure 8.2: This thesis focuses on building models that can take as input video frames and text and output text. Building models that can efficiently generate various output modalities given diverse input modalities is still an open problem. Illustration from [Tang, 2023].

studied the new multi-modal abilities unlocked by large language models, like multi-image understanding or multi-hop document understanding [Yang, 2023e]. From another side, the large-scale collection of segmentation masks in Segment Anything [Kirillov, 2023] has largely been made possible by model-assisted annotation interfaces. Another possibility is to pseudo-label large-scale data using multi-modal models pretrained on small manually annotated datasets, as done in [Ashutosh, 2023]. Therefore future work may leverage large language models or other tools to facilitate or automate partially or entirely the collection of annotations for videos.

8.2.5 Multi-modal generation

The models developed in this thesis cannot take raw audio inputs, and only generate text output. Lately, models generating visual outputs have seen great growth [Chang, 2023; Gafni, 2022; Li, 2023e; Ramesh, 2021; Ramesh, 2022; Saharia, 2022; Singer, 2022; Yu, 2022b]. Hence an interesting research direction consists in building models that can both generate both text and visual outputs, see Figure 8.2. A promising approach in this direction is GILL [Koh, 2023], which is based on a visual language model based on a frozen language model capable of visual dialog and image generation. GILL is trained with a captioning loss to learn to process images, and losses for image retrieval and image generation to learn to produce images. Another promising approach is CoDI [Tang, 2023], a generative model capable of generating any combination of output modalities, such as language, image, video, or audio, from any combination of input modalities. This is achieved despite the absence of training datasets for many combinations of modalities, by training to align modalities in both input and output spaces. However, there is still progress to be done to have a unified model that can generate text and other modalities well.

8.2.6 Ethical considerations

The potential positive or negative impacts of visual language models depend on the application. Many exciting applications have been described in Section 1.1. However, such models may also be

used for video surveillance and hence lead to questionable use. Moreover, visual language models may also reflect biases present in their training data. All models and datasets developed during this thesis have been open-sourced with permissible licenses for research-based use to foster future research in the field. We believe that the open-sourcing of these models and datasets will not only help develop better models, but also help deepen our understanding of their biases and limitations. We acknowledge that the ethical implications and potential societal impacts of such models must be carefully considered and addressed through responsible development and deployment practices. Therefore future work should focus on developing ethical guidelines, mitigating biases, and ensuring fairness and transparency in the use of these models to prevent potential negative consequences.

In addition, more than 500,000 V100 GPU hours have been used during this thesis, consuming roughly 100 MW/h. Given the carbon intensity of France of 70 gCO₂/KWh, the used electricity has produced roughly 30 tCO₂eq. However, we expect that the public release of the developed models and datasets can further amortize this cost. We have also strived to develop efficient learning techniques, for instance computing multi-modal interactions only on a few sampled frames (see Chapter 5) or freezing the language model weights (see Chapter 4). We hope that this inspires future work to consider compute-efficient methods. Finally, we advocate for research institutions and companies to invest in renewable energy sources and implement sustainable practices to mitigate the environmental impact of AI model development and deployment.

Bibliography

- [Akbari, 2021] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, et al. “VATT: Transformers for Multimodal Self-Supervised Learning from Raw Video, Audio and Text”. *NeurIPS* (2021) (cit. on pp. 25, 80, 99, 119).
- [Alayrac, 2022] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, et al. “Flamingo: a visual language model for few-shot learning”. *NeurIPS*. 2022 (cit. on pp. 22, 25, 58, 76, 97, 99, 119, 141).
- [Alberti, 2019a] Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. “Synthetic QA Corpora Generation with Roundtrip Consistency”. *ACL*. 2019 (cit. on p. 34).
- [Alberti, 2019b] Chris Alberti, Jeffrey Ling, Michael Collins, and David Reitter. “Fusion of Detected Objects in Text for Visual Question Answering”. *IJCNLP*. 2019 (cit. on p. 32).
- [Amirian, 2021] Soheyla Amirian, Khaled Rasheed, Thiab R Taha, and Hamid R Arabnia. “Automatic generation of descriptive titles for video clips using deep learning”. *Advances in Artificial Intelligence and Applied Cognitive Computing: Proceedings from ICAI’20 and ACC’20*. 2021 (cit. on p. 120).
- [Amrani, 2021] Elad Amrani, Rami Ben-Ari, Daniel Rotman, and Alex Bronstein. “Noise Estimation Using Density Estimation for Self-Supervised Multimodal Learning”. *AAAI*. 2021 (cit. on pp. 32, 47, 57).
- [Anderson, 2018] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, et al. “Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering”. *CVPR*. 2018 (cit. on pp. 21, 30).
- [Antol, 2015] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, et al. “VQA: Visual Question Answering”. *ICCV*. 2015 (cit. on pp. 21, 41, 72, 73).
- [Arnab, 2021] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. “ViViT: A video vision transformer”. *ICCV*. 2021 (cit. on pp. 19, 78, 80).
- [Ashutosh, 2023] Kumar Ashutosh, Rohit Girdhar, Lorenzo Torresani, and Kristen Grauman. “What You Say Is What You Show: Visual Narration Detection in Instructional Videos”. *CVPR*. 2023 (cit. on p. 143).
- [Ba, 2016] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. “Layer normalization”. *arXiv preprint arXiv:1607.06450* (2016) (cit. on pp. 14, 39, 61, 84).
- [Bahdanau, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural Machine Translation by Jointly Learning to Align and Translate”. *ICLR*. 2015 (cit. on p. 14).
- [Bai, 2020] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. “Boundary content graph neural network for temporal action proposal generation”. *ECCV*. 2020 (cit. on p. 20).
- [Bain, 2023] Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. “WhisperX: Time-Accurate Speech Transcription of Long-Form Audio”. *Interspeech*. 2023 (cit. on p. 122).
- [Bain, 2021] Max Bain, Arsha Nagrai, Gül Varol, and Andrew Zisserman. “Frozen in time: A joint video and image encoder for end-to-end retrieval”. *ICCV*. 2021 (cit. on pp. iii, 2, 7, 10, 19, 25, 27, 29, 32, 43, 51, 52, 57, 59, 61, 63, 80, 99, 118, 120).
- [Banerjee, 2021] Pratyay Banerjee, Tejas Gokhale, Yezhou Yang, and Chitta Baral. “WeaQA: Weak supervision via captions for visual question answering”. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 2021 (cit. on p. 32).
- [Banerjee, 2005] Satyanjeev Banerjee and Alon Lavie. “METEOR: An automatic metric for MT evaluation with improved correlation with human judgments”. *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 2005 (cit. on pp. 106, 128).

- [Bao, 2021] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. “BEiT: Bert pre-training of image transformers”. *arXiv preprint arXiv:2106.08254* (2021) (cit. on p. 17).
- [Bardes, 2022] Adrien Bardes, Jean Ponce, and Yann LeCun. “VICReg: Variance-invariance-covariance regularization for self-supervised learning”. *ICLR*. 2022 (cit. on p. 17).
- [Barlow, 1989] Horace B Barlow. “Unsupervised learning”. *Neural computation* (1989) (cit. on p. 1).
- [Behrmann, 2022] Nadine Behrmann, S Alireza Golestaneh, Zico Kolter, Jürgen Gall, and Mehdi Noroozi. “Unified fully and timestamp supervised temporal action segmentation via sequence to sequence translation”. *ECCV*. 2022 (cit. on p. 120).
- [Beltagy, 2020] Iz Beltagy, Matthew E Peters, and Arman Cohan. “Longformer: The long-document transformer”. *arXiv preprint arXiv:2004.05150* (2020) (cit. on p. 95).
- [Ben-Younes, 2017] Hedi Ben-Younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. “MUTAN: Multi-modal Tucker Fusion for Visual Question Answering”. *CVPR*. 2017 (cit. on p. 30).
- [Bertasius, 2021] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. “Is space-time attention all you need for video understanding?” *ICML*. 2021 (cit. on pp. 19, 78, 80).
- [Bird, 2009] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. O’Reilly Media, Inc., 2009 (cit. on p. 129).
- [Black, 2021] Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. *GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow*. Version 1.0. 2021 (cit. on p. 70).
- [Bojanowski, 2016] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. “Enriching Word Vectors with Subword Information”. *TACL* (2016) (cit. on p. 12).
- [Brown, 2020] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, et al. “Language Models are Few-Shot Learners”. *NeurIPS*. 2020 (cit. on pp. 15, 21, 29, 56, 57).
- [Buch, 2022] Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. “Revisiting the “video” in video-language understanding”. *CVPR*. 2022 (cit. on p. 5).
- [Caba Heilbron, 2015] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. “ActivityNet: A large-scale video benchmark for human activity understanding”. *CVPR*. 2015 (cit. on pp. 20, 64, 142).
- [Cao, 2022a] Meng Cao, Tianyu Yang, Junwu Weng, Can Zhang, Jue Wang, and Yuexian Zou. “Locvtp: Video-text pre-training for temporal localization”. *ECCV*. 2022 (cit. on p. 99).
- [Cao, 2022b] Xiao Cao, Zitan Chen, Canyu Le, and Lei Meng. “Multi-modal Video Chapter Generation”. *BMVC*. 2022 (cit. on p. 120).
- [Carion, 2020] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. “End-to-end object detection with transformers”. *ECCV*. 2020 (cit. on pp. 7, 17, 23, 78, 80, 85, 129).
- [Caron, 2020] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. “Unsupervised Learning of Visual Features by Contrasting Cluster Assignments”. *NeurIPS*. 2020 (cit. on pp. 17, 31).
- [Caron, 2021] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, et al. “Emerging properties in self-supervised vision transformers”. *ICCV*. 2021 (cit. on pp. 17, 31).
- [Carreira, 2017] João Carreira and Andrew Zisserman. “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset”. *CVPR*. 2017 (cit. on pp. 5, 18).
- [Castro, 2020] Santiago Castro, Mahmoud Azab, Jonathan Stroud, Cristina Nougaim, Ruoyao Wang, Jia Deng, et al. “LifeQA: A Real-life Dataset for Video Question Answering”. *LREC*. 2020 (cit. on pp. 31, 57).
- [Chadha, 2021] Aman Chadha, Gurmeet Arora, and Navpreet Kaloty. “iPerceive: Applying Common-Sense Reasoning to Multi-Modal Dense Video Captioning and Video Question Answering”. *WACV*. 2021 (cit. on pp. 27, 31, 57, 98).
- [Chan, 2019] Ying-Hong Chan and Yao-Chung Fan. “A Recurrent BERT-based Model for Question Generation”. *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*. 2019 (cit. on p. 34).
- [Chang, 2023] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, et al. “Muse: Text-to-image generation via masked generative transformers”. *arXiv preprint arXiv:2301.00704* (2023) (cit. on p. 143).

- [Changpinyo, 2021] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. “Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts”. *CVPR*. 2021 (cit. on pp. 18, 119).
- [Chao, 2018] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. “Rethinking the Faster R-CNN architecture for temporal action localization”. *CVPR*. 2018 (cit. on pp. 20, 120).
- [Chen, 2011] David L Chen and William B Dolan. “Collecting highly parallel data for paraphrase evaluation”. *ACL*. 2011 (cit. on pp. 63, 98, 105, 106).
- [Chen, 2018] Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat-Seng Chua. “Temporally grounding natural sentence in video”. *EMNLP*. 2018 (cit. on pp. 78, 79).
- [Chen, 2019a] Jingyuan Chen, Lin Ma, Xinpeng Chen, Zequn Jie, and Jiebo Luo. “Localizing natural language in videos”. *AAAI*. 2019 (cit. on p. 79).
- [Chen, 2021a] Jun Chen, Han Guo, Kai Yi, Boyang Li, and Mohamed Elhoseiny. “VisualGPT: Data-efficient adaptation of pretrained language models for image captioning”. *arXiv preprint arXiv:2102.10407* (2021) (cit. on p. 57).
- [Chen, 2023a] Jun Chen, Deyao Zhu, Kilichbek Haydarov, Xiang Li, and Mohamed Elhoseiny. “Video chatcaptioner: Towards the enriched spatiotemporal descriptions”. *arXiv preprint arXiv:2304.04227* (2023) (cit. on p. 142).
- [Chen, 2019b] Shaoxiang Chen and Yu-Gang Jiang. “Semantic proposal for activity localization in videos via sentence query”. *AAAI*. 2019 (cit. on p. 79).
- [Chen, 2021b] Shaoxiang Chen and Yu-Gang Jiang. “Towards bridging event captioner and sentence localizer for weakly supervised dense event captioning”. *CVPR*. 2021 (cit. on pp. 27, 98).
- [Chen, 2021c] Shixing Chen, Xiaohan Nie, David Fan, Dongqing Zhang, Vimal Bhat, and Raffay Hamid. “Shot contrastive self-supervised learning for scene boundary detection”. *CVPR*. 2021 (cit. on p. 120).
- [Chen, 2021d] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. “History Aware Multimodal Transformer for Vision-and-Language Navigation”. *NeurIPS*. 2021 (cit. on p. 80).
- [Chen, 2020a] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. “A Simple Framework for Contrastive Learning of Visual Representations”. *ICML*. 2020 (cit. on p. 17).
- [Chen, 2022a] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. “Pix2seq: A language modeling framework for object detection”. *ICLR*. 2022 (cit. on pp. 7, 23, 97, 99).
- [Chen, 2022b] Ting Chen, Saurabh Saxena, Lala Li, Tsung-Yi Lin, David J Fleet, and Geoffrey Hinton. “A unified sequence interface for vision tasks”. *NeurIPS*. 2022 (cit. on p. 99).
- [Chen, 1998] Tshuan Chen and Ram R Rao. “Audio-visual integration in multimodal communication”. *Proceedings of the IEEE* (1998) (cit. on p. 25).
- [Chen, 2023b] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, et al. “PaLI: A jointly-scaled multilingual language-image model”. *ICLR*. 2023 (cit. on pp. 22, 97, 99).
- [Chen, 2015] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, et al. “Microsoft COCO Captions: Data Collection and Evaluation Server”. *arXiv preprint arXiv:1504.00325* (2015) (cit. on pp. 16, 18, 20, 21, 23, 32, 48, 72, 85).
- [Chen, 2021e] Xinlei Chen and Kaiming He. “Exploring simple siamese representation learning”. *CVPR*. 2021 (cit. on p. 17).
- [Chen, 2021f] Xinlei Chen, Saining Xie, and Kaiming He. “An empirical study of training self-supervised vision transformers”. *ICCV*. 2021 (cit. on p. 31).
- [Chen, 2020b] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, et al. “UNITER: UNiversal Image-TExt Representation Learning”. *ECCV*. 2020 (cit. on pp. 21, 32, 43, 58, 80, 99, 119).
- [Chen, 2019c] Zhenfang Chen, Lin Ma, Wenhan Luo, and Kwan-Yee K Wong. “Weakly-supervised spatio-temporally grounding natural sentence in video”. *ACL*. 2019 (cit. on p. 79).
- [Chen, 2022c] Zhiyang Chen, Yousong Zhu, Zhaowen Li, Fan Yang, Wei Li, Haixin Wang, et al. “Obj2Seq: Formatting Objects as Sequences with Class Prompt for Visual Tasks”. *NeurIPS*. 2022 (cit. on p. 99).
- [Cheng, 2022a] Feng Cheng and Gedas Bertasius. “TALLFormer: Temporal Action Localization with Long-memory Transformer”. *ECCV*. 2022 (cit. on pp. 20, 116, 120).
- [Cheng, 2022b] Ho Kei Cheng and Alexander G Schwing. “XMem: Long-term video object segmentation with an atkinson-shiffrin memory model”. *ECCV*. 2022 (cit. on p. 142).

- [Cho, 2021] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. “Unifying vision-and-language tasks via text generation”. *ICML*. 2021 (cit. on pp. 21, 99).
- [Cho, 2014] Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, et al. “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation”. *EMNLP*. 2014 (cit. on p. 13).
- [Choi, 2021] Seongho Choi, Kyoung-Woon On, Yu-Jung Heo, Ahjeong Seo, Youwon Jang, Seungchan Lee, et al. “DramaQA: Character-Centered Video Story Understanding with Hierarchical QA”. *AAAI*. 2021 (cit. on pp. 31, 57).
- [Choromanski, 2021] Krzysztof Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, et al. “Rethinking attention with performers”. *ICLR*. 2021 (cit. on p. 95).
- [Chowdhery, 2022] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, et al. “PaLM: Scaling Language Modeling with Pathways”. *ArXiv* (2022) (cit. on p. 15).
- [Chowdhury, 2018] Mithun Chowdhury, Panda Rameswar, Evangelos Papalexakis, and Amit Roy-Chowdhury. “Webly Supervised Joint Embedding for Cross-Modal Image-Text Retrieval”. *ACM International Conference on Multimedia*. 2018 (cit. on p. 21).
- [Colas, 2020] Anthony Colas, Seokhwan Kim, Franck Dernoncourt, Siddhesh Gupte, Daisy Zhe Wang, and Doo Soon Kim. “TutorialVQA: Question Answering Dataset for Tutorial Videos”. *LREC*. 2020 (cit. on pp. 31, 57).
- [Cornia, 2020] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. “Meshed-memory transformer for image captioning”. *CVPR*. 2020 (cit. on p. 80).
- [Dai, 2019] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. “Transformer-XL: Attentive language models beyond a fixed-length context”. *ACL*. 2019 (cit. on pp. 112, 113).
- [Damen, 2018] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, et al. “Scaling Egocentric Vision: The EPIC-KITCHENS Dataset”. *ECCV*. 2018 (cit. on p. 27).
- [Dang, 2021] Long Hoang Dang, Thao Minh Le, Vuong Le, and Truyen Tran. “Object-Centric Representation Learning for Video Question Answering”. *IJCNN*. 2021 (cit. on pp. 30, 47, 57).
- [Danilák, 2021] Michal Danilák. *Language Detection library*. <https://kern-h@ngcolonwd/kern-h@ngcolonwd/github.com/Mimino666/langdetect>. 2021 (cit. on p. 125).
- [Das, 2017] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, et al. “Visual dialog”. *CVPR*. 2017 (cit. on p. 5).
- [Dave, 2022] Ishan Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. “TCLR: Temporal contrastive learning for video representation”. *Computer Vision and Image Understanding*. Elsevier, 2022 (cit. on p. 19).
- [De Lange, 2021] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, et al. “A continual learning survey: Defying forgetting in classification tasks”. *IEEE TPAMI* (2021) (cit. on p. 59).
- [Dehghani, 2022] Mostafa Dehghani, Alexey Gritsenko, Anurag Arnab, Matthias Minderer, and Yi Tay. “Scenic: A JAX Library for Computer Vision Research and Beyond”. *CVPR*. 2022 (cit. on p. 106).
- [Deng, 2021a] Chaorui Deng, Shizhe Chen, Da Chen, Yuan He, and Qi Wu. “Sketch, ground, and refine: Top-down dense video captioning”. *CVPR*. 2021 (cit. on pp. 27, 97, 98, 111).
- [Deng, 2018] Chaorui Deng, Qi Wu, Qingyao Wu, Fuyuan Hu, Fan Lyu, and Mingkui Tan. “Visual grounding via accumulated attention”. *CVPR*. 2018 (cit. on p. 79).
- [Deng, 2021b] Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. “TransVG: End-to-End Visual Grounding with Transformers”. *ICCV*. 2021 (cit. on p. 79).
- [Desai, 2021a] Karan Desai and Justin Johnson. “VirTex: Learning Visual Representations from Textual Annotations”. *CVPR*. 2021 (cit. on pp. 18, 32, 80).
- [Desai, 2021b] Karan Desai, Gaurav Kaul, Zubin Aysola, and Justin Johnson. “RedCaps: Web-curated image-text data created by the people, for the people”. *NeurIPS Datasets and Benchmarks*. 2021 (cit. on pp. 18, 99, 119).
- [Devlin, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. *NAACL-HLT*. 2019 (cit. on pp. 14, 21, 38, 40, 56, 58, 60–63, 70, 78, 133, 134).
- [Ding, 2018] Li Ding and Chenliang Xu. “Weakly-supervised action segmentation with iterative soft boundary assignment”. *CVPR*. 2018 (cit. on p. 20).

- [Donahue, 2015] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, et al. “Long-term recurrent convolutional networks for visual recognition and description”. *CVPR*. 2015 (cit. on p. 18).
- [Dosovitskiy, 2021] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. *ICLR*. 2021 (cit. on pp. 16, 60, 64, 78, 80, 102, 122, 129).
- [Dou, 2022a] Zi-Yi Dou, Aishwarya Kamath, Zhe Gan, Pengchuan Zhang, Jianfeng Wang, Linjie Li, et al. “Coarse-to-fine vision-language pre-training with fusion in the backbone”. *NeurIPS*. 2022 (cit. on pp. 23, 99, 141).
- [Dou, 2022b] Zi-Yi Dou, Yichong Xu, Zhe Gan, Jianfeng Wang, Shuohang Wang, Lijuan Wang, et al. “An empirical study of training end-to-end vision-and-language transformers”. *CVPR*. 2022 (cit. on pp. 21, 99).
- [Du, 2017] Xinya Du, Junru Shao, and Claire Cardie. “Learning to Ask: Neural Question Generation for Reading Comprehension”. *ACL*. 2017 (cit. on p. 32).
- [Edelman, 1987] Gerald M Edelman. *Neural Darwinism: The theory of neuronal group selection*. Basic books, 1987 (cit. on p. 1).
- [Eichenberg, 2021] Constantin Eichenberg, Sidney Black, Samuel Weinbach, Letitia Parcalabescu, and Anette Frank. “MAGMA–Multimodal Augmentation of Generative Models through Adapter-based Finetuning”. *arXiv preprint arXiv:2112.05253* (2021) (cit. on pp. 22, 56, 58).
- [Escorcia, 2016] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. “Daps: Deep action proposals for action understanding”. *ECCV*. 2016 (cit. on p. 98).
- [Fan, 2019a] Chenyou Fan. “EgoVQA - An Egocentric Video Question Answering Benchmark Dataset”. *ICCV Workshops*. 2019 (cit. on p. 31).
- [Fan, 2019b] Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. “Heterogeneous Memory Enhanced Multimodal Attention Model for Video Question Answering”. *CVPR*. 2019 (cit. on pp. 29, 30, 47, 57).
- [Fan, 2021] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, et al. “Multiscale vision transformers”. *ICCV*. 2021 (cit. on p. 19).
- [Farha, 2019] Yazan Abu Farha and Jurgen Gall. “MS-TCN: Multi-stage temporal convolutional network for action segmentation”. *CVPR*. 2019 (cit. on pp. 20, 120).
- [Fathi, 2011] Alireza Fathi, Xiaofeng Ren, and James M Rehg. “Learning to recognize objects in egocentric activities”. *CVPR*. 2011 (cit. on p. 20).
- [Federico, 2012] Marcello Federico, Sebastian Stüker, Luisa Bentivogli, Michael Paul, Mauro Cettolo, Teresa Herrmann, et al. “The IWSLT 2011 Evaluation Campaign on Automatic Talk Translation”. *LREC*. 2012 (cit. on p. 34).
- [Feichtenhofer, 2019] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. “Slowfast networks for video recognition”. *ICCV*. 2019 (cit. on pp. 19, 80).
- [Feichtenhofer, 2021] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. “A large-scale study on unsupervised spatiotemporal representation learning”. *CVPR*. 2021 (cit. on p. 19).
- [Fu, 2021] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, et al. “VIOLET: End-to-End Video-Language Transformers with Masked Visual-token Modeling”. *arXiv preprint arXiv:2111.12681* (2021) (cit. on pp. 58, 73, 74, 99).
- [Fujita, 2020] Soichiro Fujita, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. “SODA: Story oriented dense video captioning evaluation framework”. *ECCV*. 2020 (cit. on pp. 107, 128).
- [Fukui, 2016] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. “Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding”. *EMNLP*. 2016 (cit. on p. 30).
- [Gabeur, 2020] Valentin Gabeur, Chen Sun, Kartteek Alahari, and Cordelia Schmid. “Multi-modal Transformer for Video Retrieval”. *ECCV*. 2020 (cit. on pp. 32, 80).
- [Gafni, 2022] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. “Make-a-scene: Scene-based text-to-image generation with human priors”. *ECCV*. 2022 (cit. on p. 143).
- [Gan, 2020] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. “Large-scale adversarial training for vision-and-language representation learning”. *NeurIPS*. 2020 (cit. on pp. 21, 58, 99, 119).

- [Gao, 2018] Jiyang Gao, Runzhou Ge, Kan Chen, and Ram Nevatia. “Motion-Appearance Co-Memory Networks for Video Question Answering”. *CVPR*. 2018 (cit. on pp. 30, 47, 57).
- [Gao, 2017a] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. “TALL: Temporal activity localization via language query”. *ICCV*. 2017 (cit. on pp. 26, 78, 79, 118).
- [Gao, 2017b] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. “Turn tap: Temporal unit regression network for temporal action proposals”. *ICCV*. 2017 (cit. on p. 98).
- [Gao, 2017c] Lianli Gao, Zhao Guo, Hanwang Zhang, Xing Xu, and Heng Tao Shen. “Video captioning with attention-based LSTM and semantic consistency”. *IEEE Transactions on Multimedia* (2017) (cit. on pp. 98, 120).
- [Gao, 2023] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, et al. “LLaMA-Adapter V2: Parameter-efficient visual instruction model”. *arXiv preprint arXiv:2304.15010* (2023) (cit. on p. 142).
- [Gao, 2021] Shang-Hua Gao, Qi Han, Zhong-Yu Li, Pai Peng, Liang Wang, and Ming-Ming Cheng. “Global2Local: Efficient Structure Search for Video Action Segmentation”. *CVPR*. 2021 (cit. on pp. 20, 120).
- [Garcia, 2020] Noa Garcia, Mayu Otani, Chenhui Chu, and Yuta Nakashima. “KnowIT VQA: Answering knowledge-based questions about videos”. *AAAI*. 2020 (cit. on pp. 31, 57).
- [Ge, 2022] Yuying Ge, Yixiao Ge, Xihui Liu, Dian Li, Ying Shan, Xiaohu Qie, et al. “Bridging Video-Text Retrieval With Multiple Choice Questions”. *CVPR*. 2022 (cit. on pp. 25, 99, 119).
- [Geburu, 2021] Timnit Geburu, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, et al. “Datasheets for datasets”. *Communications of the ACM* (2021) (cit. on p. 122).
- [Ging, 2020] Simon Ging, Mohammadreza Zolfaghari, Hamed Pirsiavash, and Thomas Brox. “COOT: Cooperative Hierarchical Transformer for Video-Text Representation Learning”. *NeurIPS*. 2020 (cit. on p. 80).
- [Girshick, 2015] Ross Girshick. “Fast R-CNN”. *CVPR*. 2015 (cit. on p. 16).
- [Girshick, 2014] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation”. *CVPR*. 2014 (cit. on p. 16).
- [Goyal, 2017] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. “Making the V in VQA matter: : Elevating the Role of Image Understanding in Visual Question Answering”. *CVPR*. 2017 (cit. on pp. 21, 30).
- [Grauman, 2022] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, et al. “Ego4D: Around the World in 3,000 Hours of Egocentric Video”. *CVPR*. 2022 (cit. on pp. 27, 118, 120).
- [Grill, 2020] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, et al. “Bootstrap your own latent—a new approach to self-supervised learning”. *NeurIPS*. 2020 (cit. on p. 17).
- [Han, 2019] Tengda Han, Weidi Xie, and Andrew Zisserman. “Video representation learning by dense predictive coding”. *Workshop on Large Scale Holistic Video Understanding, ICCV*. 2019 (cit. on p. 19).
- [Han, 2020] Tengda Han, Weidi Xie, and Andrew Zisserman. “Self-supervised co-training for video representation learning”. *NeurIPS*. 2020 (cit. on p. 19).
- [Han, 2022] Tengda Han, Weidi Xie, and Andrew Zisserman. “Temporal alignment networks for long-term video”. *CVPR*. 2022 (cit. on pp. 20, 97, 99, 103, 119).
- [Hanu, 2022] Laura Hanu, James Thewlis, Yuki M Asano, and Christian Rupprecht. “VTC: Improving Video-Text Retrieval with User Comments”. *ECCV*. 2022 (cit. on p. 27).
- [Hanu, 2020] Laura Hanu and Unitary team. *Detoxify*. <https://kern-h@ngcolonwd/kern-h@ngcolonwd/github.com/unitaryai/detoxify>. 2020 (cit. on p. 127).
- [He, 2019] Dongliang He, Xiang Zhao, Jizhou Huang, Fu Li, Xiao Liu, and Shilei Wen. “Read, watch, and move: Reinforcement learning for temporally grounding natural language descriptions in videos”. *AAAI*. 2019 (cit. on p. 79).
- [He, 2022] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. “Masked autoencoders are scalable vision learners”. *CVPR*. 2022 (cit. on p. 17).
- [He, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. *CVPR*. 2016 (cit. on pp. 14, 16, 86).
- [He, 2021a] Lu He, Qianyu Zhou, Xiangtai Li, Li Niu, Guangliang Cheng, Xiao Li, et al. “End-to-End Video Object Detection with Spatial-Temporal Transformers”. *Proceedings of the 29th ACM International Conference on Multimedia* (2021) (cit. on p. 80).

- [He, 2021b] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. “DeBERTa: Decoding-enhanced BERT with Disentangled Attention”. *ICLR*. 2021 (cit. on pp. 15, 56, 63, 64, 70).
- [Hearst, 1997] Marti A Hearst. “Text tiling: Segmenting text into multi-paragraph subtopic passages”. *Computational linguistics* (1997) (cit. on pp. 128, 129, 131, 135).
- [Heilbron, 2016] Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. “Fast temporal activity proposals for efficient detection of human actions in untrimmed videos”. *CVPR*. 2016 (cit. on p. 98).
- [Heilman, 2010] Michael Heilman and Noah A Smith. “Good Question! Statistical Ranking for Question Generation”. *ACL*. 2010 (cit. on pp. 32, 36, 37, 49, 50).
- [Hendricks, 2021] Lisa Anne Hendricks, John Mellor, Rosalia Schneider, Jean-Baptiste Alayrac, and Aida Nematzadeh. “Decoupling the role of data, attention, and losses in multimodal transformers”. *TACL*. 2021 (cit. on p. 58).
- [Hendricks, 2017] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. “Localizing Moments in Video with Natural Language”. *ICCV*. 2017 (cit. on pp. 78, 79, 118, 121).
- [Hendricks, 2018] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. “Localizing Moments in Video with Temporal Language”. *EMNLP*. 2018 (cit. on pp. 26, 79, 121).
- [Hendrycks, 2016] Dan Hendrycks and Kevin Gimpel. “Gaussian Error Linear Units (GELUs)”. *arXiv preprint arXiv:1606.08415* (2016) (cit. on p. 39).
- [Hochreiter, 1997] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. *Neural Computation* (1997) (cit. on p. 13).
- [Hoffmann, 2022] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, et al. “Training Compute-Optimal Large Language Models”. *NeurIPS*. 2022 (cit. on pp. 15, 57).
- [Houlsby, 2019] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, et al. “Parameter-efficient transfer learning for NLP”. *ICML*. 2019 (cit. on pp. 56, 58, 60, 64, 67, 70).
- [Hu, 2022a] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, et al. “LoRA: Low-rank adaptation of large language models”. *ICLR*. 2022 (cit. on p. 58).
- [Hu, 2018] Hexiang Hu, Wei-Lun Chao, and Fei Sha. “Learning Answer Embeddings for Visual Question Answering”. *CVPR*. 2018 (cit. on p. 30).
- [Hu, 2017] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. “Modeling relationships in referential expressions with compositional modular networks”. *CVPR*. 2017 (cit. on p. 79).
- [Hu, 2016] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. “Natural Language Object Retrieval”. *CVPR*. 2016 (cit. on pp. 78, 79).
- [Hu, 2022b] Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, et al. “Scaling up vision-language pre-training for image captioning”. *CVPR*. 2022 (cit. on pp. 99, 119).
- [Huang, 2018] De-An Huang, Shyamal Buch, Lucio Dery, Animesh Garg, Li Fei-Fei, and Juan Carlos Niebles. “Finding” it”: Weakly-supervised reference-aware visual grounding in instructional videos”. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5948–5957 (cit. on p. 79).
- [Huang, 2021a] Binbin Huang, Dongze Lian, Weixin Luo, and Shenghua Gao. “Look Before You Leap: Learning Landmark Features for One-Stage Visual Grounding”. *CVPR*. 2021 (cit. on p. 79).
- [Huang, 2020a] Deng Huang, Peihao Chen, Runhao Zeng, Qing Du, Mingkui Tan, and Chuang Gan. “Location-aware Graph Convolutional Networks for Video Question Answering”. *AAAI*. 2020 (cit. on pp. 29, 30, 37, 47, 57).
- [Huang, 2020b] Gabriel Huang, Bo Pang, Zhenhai Zhu, Clara Rivera, and Radu Soricut. “Multimodal pretraining for dense video captioning”. *AAACL-IJCNLP*. 2020 (cit. on pp. iv, 11, 25, 96–99, 101, 103, 105, 107, 117–120, 135).
- [Huang, 2021b] Zhicheng Huang, Zhaoyang Zeng, Yupan Huang, Bei Liu, Dongmei Fu, and Jianlong Fu. “Seeing Out of the Box: End-to-End Pre-training for Vision-Language Representation Learning”. *CVPR*. 2021 (cit. on pp. 21, 32, 99, 119).
- [Huang, 2020c] Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. “Pixel-BERT: Aligning Image Pixels with Text by Deep Multi-Modal Transformers”. *arXiv preprint arXiv:2004.00849* (2020) (cit. on pp. 21, 32, 58, 80, 81).
- [Iashin, 2020a] Vladimir Iashin and Esa Rahtu. “A better use of audio-visual cues: Dense video captioning with bi-modal transformer”. *BMVC*. 2020 (cit. on pp. 26, 97, 98).

- [Iashin, 2020b] Vladimir Iashin and Esa Rahtu. “Multi-modal dense video captioning”. *CVPR Workshops*. 2020 (cit. on pp. 26, 98).
- [Idrees, 2017] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, et al. “The thumos challenge on action recognition for videos “in the wild””. *Computer Vision and Image Understanding* (2017) (cit. on p. 20).
- [Ioffe, 2015] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate”. *arXiv preprint arXiv:1502.03167* (2015) (cit. on p. 16).
- [Jang, 2017] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. “TGIF-QA: Toward Spatio-Temporal Reasoning in Visual Question Answering”. *CVPR*. 2017 (cit. on pp. 5, 24, 30–32, 47, 55, 62, 63, 68).
- [Ji, 2023] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, et al. “Survey of hallucination in natural language generation”. *ACM Computing Surveys* (2023) (cit. on p. 141).
- [Jia, 2021] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, et al. “Scaling up visual and vision-language representation learning with noisy text supervision”. *ICML*. 2021 (cit. on pp. 99, 119).
- [Jiang, 2020a] Jianwen Jiang, Ziqiang Chen, Haojie Lin, Xibin Zhao, and Yue Gao. “Divide and Conquer: Question-Guided Spatio-Temporal Contextual Attention for Video Question Answering”. *AAAI*. 2020 (cit. on pp. 29, 30, 37, 47, 57).
- [Jiang, 2020b] Pin Jiang and Yahong Han. “Reasoning with Heterogeneous Graph Alignment for Video Question Answering”. *AAAI*. 2020 (cit. on pp. 29, 30, 37, 47, 57).
- [Jiang, 2020c] Zhengbao Jiang, Antonios Anastasopoulos, Jun Araki, Haibo Ding, and Graham Neubig. “X-FACTR: Multilingual factual knowledge retrieval from pretrained language models”. *EMNLP*. 2020 (cit. on p. 68).
- [Jin, 2021] Weike Jin, Zhou Zhao, Xiaochun Cao, Jieming Zhu, Xiuqiang He, and Yueting Zhuang. “Adaptive Spatio-Temporal Graph Enhanced Vision-Language Representation for Video QA”. *IEEE Transactions on Image Processing* (2021) (cit. on p. 47).
- [Joshi, 2020] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. “SpanBERT: Improving pre-training by representing and predicting spans”. *TACL*. 2020 (cit. on pp. 15, 56, 68).
- [Kamath, 2021] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. “MDETR - Modulated Detection for End-to-End Multi-Modal Understanding”. *ICCV*. 2021 (cit. on pp. iv, 11, 23, 78, 79, 81, 85, 87, 88, 99, 140).
- [Karpathy, 2015] Andrej Karpathy and Li Fei-Fei. “Deep visual-semantic alignments for generating image descriptions”. *CVPR*. 2015 (cit. on p. 21).
- [Kazakos, 2019] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. “EPIC-Fusion: Audio-Visual Temporal Binding for Egocentric Action Recognition”. *ICCV*. 2019 (cit. on p. 25).
- [Kim, 2020a] Hyounghun Kim, Zineng Tang, and Mohit Bansal. “Dense-Caption Matching and Frame-Selection Gating for Temporal Localization in VideoQA”. *ACL*. 2020 (cit. on pp. 31, 57).
- [Kim, 2020b] Junyeong Kim, Minuk Ma, Trung Pham, Kyungsu Kim, and Chang D Yoo. “Modality Shifting Attention Network for Multi-modal Video Question Answering”. *CVPR*. 2020 (cit. on pp. 31, 57).
- [Kim, 2017] Kyung-Min Kim, Min-Oh Heo, Seong-Ho Choi, and Byoung-Tak Zhang. “Deepstory: Video story qa by deep embedded memory networks”. *IJCAI*. 2017 (cit. on pp. 31, 57).
- [Kim, 2021a] Seonhoon Kim, Seohyeong Jeong, Eunbyul Kim, Inho Kang, and Nojun Kwak. “Self-supervised pre-training and contrastive representation learning for multiple-choice video QA”. *AAAI*. 2021 (cit. on pp. 31, 57).
- [Kim, 2023] Sungdong Kim, Jin-Hwa Kim, Jiyoung Lee, and Minjoon Seo. “Semi-parametric video-grounded text generation”. *arXiv preprint arXiv:2301.11507* (2023) (cit. on p. 142).
- [Kim, 2021b] Wonjae Kim, Bokyung Son, and Ildoo Kim. “ViLT: Vision-and-language transformer without convolution or region supervision”. *ICML*. 2021 (cit. on pp. 58, 80, 99).
- [Kingma, 2015] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. *ICLR*. 2015 (cit. on pp. 44, 64, 106, 128).
- [Kirillov, 2023] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, et al. “Segment anything”. *arXiv preprint arXiv:2304.02643* (2023) (cit. on p. 143).
- [Kitaev, 2020] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. “Reformer: The efficient transformer”. *ICLR*. 2020 (cit. on p. 95).

- [Klein, 2023] Guillaume Klein. *faster-whisper library*. <https://github.com/guillaumekln/faster-whisper>. 2023 (cit. on p. 122).
- [Ko, 2022] Dohwan Ko, Joonmyung Choi, Juyeon Ko, Shinyeong Noh, Kyoung-Woon On, Eun-Sol Kim, et al. “Video-Text Representation Learning via Differentiable Weak Temporal Alignment”. *CVPR*. 2022 (cit. on pp. 97, 99, 103, 119).
- [Koh, 2023] Jing Yu Koh, Daniel Fried, and Ruslan Salakhutdinov. “Generating images with multimodal language models”. *arXiv preprint arXiv:2305.17216* (2023) (cit. on p. 143).
- [Kolesnikov, 2022] Alexander Kolesnikov, André Susano Pinto, Lucas Beyer, Xiaohua Zhai, Jeremiah Harmsen, and Neil Houlsby. “UViM: A Unified Modeling Approach for Vision with Learned Guiding Codes”. *NeurIPS*. 2022 (cit. on p. 99).
- [Koupae, 2018] Mahnaz Koupae and William Yang Wang. “Wikihow: A large scale text summarization dataset”. *arXiv preprint arXiv:1810.09305* (2018) (cit. on p. 99).
- [Krishna, 2017] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. “Dense-Captioning Events in Videos”. *ICCV*. 2017 (cit. on pp. iv, 5, 6, 11, 26, 96–98, 101, 103, 105, 106, 118, 120, 128, 142).
- [Krishna, 2016] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, et al. “Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations”. *IJCV* (2016) (cit. on pp. 16, 21, 23, 32, 48, 72, 85).
- [Krizhevsky, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet classification with deep convolutional neural networks”. *NeurIPS*. 2012 (cit. on pp. 6, 16).
- [Kudo, 2018] Taku Kudo and John Richardson. “Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing”. *ACL*. 2018 (cit. on pp. 63, 68, 101).
- [Kuehne, 2014] Hilde Kuehne, Ali Arslan, and Thomas Serre. “The language of actions: Recovering the syntax and semantics of goal-directed human activities”. *CVPR*. 2014 (cit. on p. 20).
- [Kuehne, 2018] Hilde Kuehne, Alexander Richard, and Juergen Gall. “A hybrid RNN-HMM approach for weakly supervised temporal action segmentation”. *TPAMI* (2018) (cit. on p. 20).
- [Lan, 2020] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations”. *ICLR*. 2020 (cit. on pp. 56, 58).
- [Laptev, 2008] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. “Learning realistic human actions from movies”. *CVPR*. 2008 (cit. on p. 27).
- [Le, 2020a] Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. “Hierarchical Conditional Relation Networks for Video Question Answering”. *CVPR*. 2020 (cit. on pp. 29, 30, 37, 43, 47, 56, 57).
- [Le, 2020b] Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. “Neural Reasoning, Fast and Slow, for Video Question Answering”. *IJCNN*. 2020 (cit. on pp. 30, 57).
- [Le, 2021] Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. “Hierarchical Conditional Relation Networks for Multimodal Video Question Answering”. *IJCV*. 2021 (cit. on p. 74).
- [Lea, 2017] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. “Temporal convolutional networks for action segmentation and detection”. *CVPR*. 2017 (cit. on pp. 20, 120).
- [Lecun, 1998] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. “Gradient-based learning applied to document recognition”. *Proceedings of the IEEE* (1998) (cit. on p. 16).
- [LeCun, 2015] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. *Nature* (2015) (cit. on p. 1).
- [Lei, 2021a] Jie Lei, Tamara L Berg, and Mohit Bansal. “Detecting Moments and Highlights in Videos via Natural Language Queries”. *NeurIPS*. 2021 (cit. on pp. 26, 80, 99, 118, 119, 121, 134, 138).
- [Lei, 2021b] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, et al. “Less is More: ClipBERT for Video-and-Language Learning via Sparse Sampling”. *CVPR*. 2021 (cit. on pp. 29, 30, 47, 58, 74, 80, 81, 99, 119).
- [Lei, 2020a] Jie Lei, Liwei Wang, Yelong Shen, Dong Yu, Tamara L Berg, and Mohit Bansal. “MART: Memory-augmented recurrent transformer for coherent video paragraph captioning”. *ACL*. 2020 (cit. on pp. 98, 106, 112, 113).
- [Lei, 2018a] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. “TVQA: Localized, Compositional Video Question Answering”. *EMNLP*. 2018 (cit. on pp. 5, 24, 25, 31, 55, 56, 62–64, 66, 116).

- [Lei, 2020b] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. “TVQA+: Spatio-Temporal Grounding for Video Question Answering”. *ACL*. 2020 (cit. on pp. 31, 57).
- [Lei, 2020c] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. “TVR: A large-scale dataset for video-subtitle moment retrieval”. *ECCV*. 2020 (cit. on pp. 27, 121).
- [Lei, 2020d] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. “What is more likely to happen next? video-and-language future event prediction”. *EMNLP*. 2020 (cit. on p. 25).
- [Lei, 2018b] Peng Lei and Sinisa Todorovic. “Temporal deformable residual networks for action segmentation in videos”. *CVPR*. 2018 (cit. on p. 20).
- [Lester, 2021] Brian Lester, Rami Al-Rfou, and Noah Constant. “The power of scale for parameter-efficient prompt tuning”. *EMNLP*. 2021 (cit. on p. 60).
- [Li, 2022a] Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven CH Hoi. “Align and Prompt: Video-and-Language Pre-training with Entity Prompts”. *CVPR*. 2022 (cit. on pp. 58, 99, 119).
- [Li, 2020a] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, Daxin Jiang, and Ming Zhou. “Unicoder-VL: A Universal Encoder for Vision and Language by Cross-modal Pre-training”. *AAAI*. 2020 (cit. on pp. 32, 58, 80, 99, 119).
- [Li, 2022b] Guangyao Li, Yake Wei, Yapeng Tian, Chenliang Xu, Ji-Rong Wen, and Di Hu. “Learning to answer questions in dynamic audio-visual scenarios”. *CVPR*. 2022 (cit. on p. 25).
- [Li, 2023a] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. “BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models”. *ICML*. 2023 (cit. on pp. 22, 128, 129, 131, 133, 135, 141).
- [Li, 2022c] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. “BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation”. *ICML*. 2022 (cit. on pp. 22, 57, 72, 99, 119).
- [Li, 2021a] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. “Align before fuse: Vision and language representation learning with momentum distillation”. *NeurIPS*. 2021 (cit. on pp. 21, 58, 99, 119).
- [Li, 2023b] KunChang Li, Yanan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, et al. “VideoChat: Chat-centric video understanding”. *arXiv preprint arXiv:2305.06355* (2023) (cit. on p. 142).
- [Li, 2020b] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. “HERO: Hierarchical Encoder for Video+Language Omni-representation Pre-training”. *EMNLP*. 2020 (cit. on pp. 24, 25, 28, 30–32, 43, 47, 55, 58, 62–64, 74, 80, 99, 116, 119).
- [Li, 2023c] Linjie Li, Zhe Gan, Kevin Lin, Chung-Ching Lin, Zicheng Liu, Ce Liu, et al. “LAVENDER: Unifying video-language understanding as masked language modeling”. *CVPR*. 2023 (cit. on pp. 119, 142).
- [Li, 2021b] Linjie Li, Jie Lei, Zhe Gan, Licheng Yu, Yen-Chun Chen, Rohit Pillai, et al. “VALUE: A Multi-Task Benchmark for Video-and-Language Understanding Evaluation”. *NeurIPS Track on Datasets and Benchmarks*. 2021 (cit. on pp. 66, 116).
- [Li, 2019a] 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 preprint arXiv:1908.03557* (2019) (cit. on pp. 21, 32, 58).
- [Li, 2022d] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, et al. “Grounded language-image pre-training”. *CVPR*. 2022 (cit. on p. 99).
- [Li, 2020c] Shi-Jie Li, Yazan AbuFarha, Yun Liu, Ming-Ming Cheng, and Juergen Gall. “MS-TCN++: Multi-stage temporal convolutional network for action segmentation”. *TPAMI* (2020) (cit. on p. 20).
- [Li, 2022e] Wanhua Li, Zhexuan Cao, Jianjiang Feng, Jie Zhou, and Jiwen Lu. “Label2Label: A Language Modeling Framework for Multi-Attribute Learning”. *ECCV*. 2022 (cit. on p. 99).
- [Li, 2021c] Xiang Lisa Li and Percy Liang. “Prefix-tuning: Optimizing continuous prompts for generation”. *ACL*. 2021 (cit. on p. 60).
- [Li, 2019b] Xiangpeng Li, Jingkuan Song, Lianli Gao, Xianglong Liu, Wenbing Huang, Xiangnan He, et al. “Beyond RNNs: Positional Self-Attention with Co-Attention for Video Question Answering”. *AAAI*. 2019 (cit. on pp. 29, 30, 57).
- [Li, 2020d] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, et al. “Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks”. *ECCV*. 2020 (cit. on pp. 21, 32, 58, 80, 99, 119).

- [Li, 2023d] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. “Scaling language-image pre-training via masking”. *CVPR*. 2023 (cit. on p. 18).
- [Li, 2018a] Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei. “Jointly localizing and describing events for dense video captioning”. *CVPR*. 2018 (cit. on pp. 27, 98).
- [Li, 2018b] Yikang Li, Nan Duan, Bolei Zhou, Xiao Chu, Wanli Ouyang, Xiaogang Wang, et al. “Visual Question Generation as Dual Task of Visual Question Answering”. *CVPR*. 2018 (cit. on p. 31).
- [Li, 2023e] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, et al. “GLIGEN: Open-set grounded text-to-image generation”. *CVPR*. 2023 (cit. on p. 143).
- [Li, 2016] Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, et al. “TGIF: A New Dataset and Benchmark on Animated GIF Description”. *CVPR*. 2016 (cit. on p. 64).
- [Li, 2021d] Zhe Li, Yazan Abu Farha, and Jurgen Gall. “Temporal Action Segmentation From Timestamp Supervision”. *CVPR*. 2021 (cit. on p. 120).
- [Li, 2022f] Liunian Harold Li*, Pengchuan Zhang*, Haotian Zhang*, Jianwei Yang, Chunyuan Li, Yiwu Zhong, et al. “Grounded Language-Image Pre-training”. *CVPR*. 2022 (cit. on p. 23).
- [Liao, 2020] Yue Liao, Si Liu, Guanbin Li, Fei Wang, Yanjie Chen, Chen Qian, et al. “A real-time cross-modality correlation filtering method for referring expression comprehension”. *CVPR*. 2020 (cit. on p. 79).
- [Lin, 2004] Chin-Yew Lin. “ROUGE: a Package for Automatic Evaluation of Summaries.” *Proceedings of the Workshop on Text Summarization Branches Out (WAS)*. 2004 (cit. on p. 128).
- [Lin, 2020a] Chuming Lin, Jian Li, Yabiao Wang, Ying Tai, Donghao Luo, Zhipeng Cui, et al. “Fast learning of temporal action proposal via dense boundary generator”. *AAAI*. 2020 (cit. on p. 98).
- [Lin, 2021a] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, et al. “Learning salient boundary feature for anchor-free temporal action localization”. *CVPR*. 2021 (cit. on p. 20).
- [Lin, 2022a] Kevin Qinghong Lin, Alex Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Zhongcong Xu, et al. “Egocentric Video-Language Pretraining”. *NeurIPS*. 2022 (cit. on pp. 99, 120).
- [Lin, 2022b] Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, et al. “SwinBERT: End-to-end transformers with sparse attention for video captioning”. *CVPR*. 2022 (cit. on pp. 97, 98, 113, 120).
- [Lin, 2019] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. “BMN: Boundary-matching network for temporal action proposal generation”. *ICCV*. 2019 (cit. on pp. 20, 98).
- [Lin, 2017] Tianwei Lin, Xu Zhao, and Zheng Shou. “Single shot temporal action detection”. *Proceedings of the 25th ACM international conference on Multimedia*. 2017 (cit. on p. 20).
- [Lin, 2018] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. “BSN: Boundary sensitive network for temporal action proposal generation”. *ECCV*. 2018 (cit. on pp. 20, 98).
- [Lin, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, et al. “Microsoft COCO: Common Objects in Context”. *ECCV*. 2014 (cit. on p. 31).
- [Lin, 2021b] Xudong Lin, Gedas Bertasius, Jue Wang, Shih-Fu Chang, Devi Parikh, and Lorenzo Torresani. “VX2TEXT: End-to-End Learning of Video-Based Text Generation From Multimodal Inputs”. *CVPR*. 2021 (cit. on pp. 25, 31, 57).
- [Lin, 2020b] Zhijie Lin, Zhou Zhao, Zhu Zhang, Qi Wang, and Huasheng Liu. “Weakly-supervised video moment retrieval via semantic completion network”. *AAAI*. 2020 (cit. on p. 79).
- [Liu, 2023] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. “Visual instruction tuning”. *NeurIPS*. 2023 (cit. on pp. 141, 142).
- [Liu, 2017] Jingyu Liu, Liang Wang, and Ming-Hsuan Yang. “Referring expression generation and comprehension via attributes”. *ICCV*. 2017 (cit. on p. 79).
- [Liu, 2020a] Jingzhou Liu, Wenhui Chen, Yu Cheng, Zhe Gan, Licheng Yu, Yiming Yang, et al. “VIOLIN: A Large-Scale Dataset for Video-and-Language Inference”. *CVPR*. 2020 (cit. on p. 25).
- [Liu, 2020b] Qinying Liu and Zilei Wang. “Progressive boundary refinement network for temporal action detection”. *AAAI*. 2020 (cit. on p. 20).
- [Liu, 2022a] Xiaolong Liu, Qimeng Wang, Yao Hu, Xu Tang, Shiwei Zhang, Song Bai, et al. “End-to-end temporal action detection with transformer”. *IEEE Transactions on Image Processing*. 2022 (cit. on pp. 116, 120).
- [Liu, 2019a] Xihui Liu, Zihao Wang, Jing Shao, Xiaogang Wang, and Hongsheng Li. “Improving referring expression grounding with cross-modal attention-guided erasing”. *CVPR*. 2019 (cit. on p. 79).

- [Liu, 2019b] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, et al. “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. *arXiv preprint arXiv:1907.11692* (2019) (cit. on pp. 15, 56, 58, 86).
- [Liu, 2021a] Yongfei Liu, Bo Wan, Lin Ma, and Xuming He. “Relation-aware Instance Refinement for Weakly Supervised Visual Grounding”. *CVPR*. 2021 (cit. on p. 79).
- [Liu, 2019c] Yuan Liu, Lin Ma, Yifeng Zhang, Wei Liu, and Shih-Fu Chang. “Multi-granularity generator for temporal action proposal”. *CVPR*. 2019 (cit. on p. 20).
- [Liu, 2021b] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, et al. “Swin transformer: Hierarchical vision transformer using shifted windows”. *ICCV*. 2021 (cit. on pp. 17, 78).
- [Liu, 2022b] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, et al. “Video swin transformer”. *CVPR*. 2022 (cit. on p. 19).
- [Long, 2019] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. “Gaussian temporal awareness networks for action localization”. *CVPR*. 2019 (cit. on p. 20).
- [Lopez, 2020] Luis Enrico Lopez, Diane Kathryn Cruz, Jan Christian Blaise Cruz, and Charibeth Cheng. “Transformer-based End-to-End Question Generation”. *arXiv preprint arXiv:2005.01107* (2020) (cit. on p. 34).
- [Loshchilov, 2019] Ilya Loshchilov and Frank Hutter. “Decoupled weight decay regularization”. *ICLR*. 2019 (cit. on pp. 86, 134).
- [Lu, 2019] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. “ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks”. *NeurIPS*. 2019 (cit. on pp. 21, 32, 43, 58, 79, 80, 99, 119).
- [Lu, 2020] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. “12-in-1: Multi-task vision and language representation learning”. *CVPR*. 2020 (cit. on pp. 32, 58, 80, 99, 119).
- [Lu, 2016] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. “Hierarchical Question-Image Co-Attention for Visual Question Answering”. *NeurIPS*. 2016 (cit. on p. 30).
- [Luo, 2020a] Gen Luo, Yiyi Zhou, Xiaoshuai Sun, Liujuan Cao, Chenglin Wu, Cheng Deng, et al. “Multi-task collaborative network for joint referring expression comprehension and segmentation”. *CVPR*. 2020 (cit. on p. 79).
- [Luo, 2020b] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, et al. “UniViLM: A Unified Video and Language Pre-Training Model for Multimodal Understanding and Generation”. *arXiv preprint arXiv:2002.06353* (2020) (cit. on pp. 32, 97, 107, 120).
- [Luo, 2022] Ziyang Luo, Yadong Xi, Rongsheng Zhang, and Jing Ma. “VC-GPT: Visual Conditioned GPT for End-to-End Generative Vision-and-Language Pre-training”. *arXiv preprint arXiv:2201.12723* (2022) (cit. on p. 57).
- [Mahabadi, 2022] Rabeeh Karimi Mahabadi, Luke Zettlemoyer, James Henderson, Marzieh Saeidi, Lambert Mathias, Veselin Stoyanov, et al. “PERFECT: Prompt-free and Efficient Few-shot Learning with Language Models”. *ACL*. 2022 (cit. on p. 56).
- [Maharaj, 2017] Tegan Maharaj, Nicolas Ballas, Anna Rohrbach, Aaron Courville, and Christopher Pal. “A dataset and exploration of models for understanding video data through fill-in-the-blank question-answering”. *CVPR*. 2017 (cit. on pp. 24, 55, 56, 58, 63).
- [Miech, 2020] 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”. *CVPR*. 2020 (cit. on pp. 20, 31, 32, 38, 78, 97, 99, 103, 119, 120).
- [Miech, 2019] 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”. *ICCV*. 2019 (cit. on pp. iii, 2, 7, 9, 10, 19, 27, 29, 32–34, 38, 40, 43, 52, 57, 61, 63, 64, 99, 103, 109, 118, 120, 121, 129–131, 133, 135, 136).
- [Mikolov, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. “Efficient estimation of word representations in vector space”. *arXiv preprint arXiv:1301.3781* (2013) (cit. on p. 12).
- [Misra, 2016] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. “Shuffle and learn: unsupervised learning using temporal order verification”. *ECCV*. 2016 (cit. on p. 19).
- [Mithun, 2019] Niluthpol Chowdhury Mithun, Sujoy Paul, and Amit K Roy-Chowdhury. “Weakly supervised video moment retrieval from text queries”. *CVPR*. 2019 (cit. on p. 79).

- [Mokady, 2021] Ron Mokady, Amir Hertz, and Amit H Bermano. “ClipCap: Clip prefix for image captioning”. *arXiv preprint arXiv:2111.09734* (2021) (cit. on pp. 56, 58).
- [Mostafazadeh, 2016] Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. “Generating Natural Questions About an Image”. *ACL*. 2016 (cit. on p. 31).
- [Mu, 2022] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. “SLIP: Self-supervision meets language-image pre-training”. *ECCV*. 2022 (cit. on p. 18).
- [Mun, 2017] Jonghwan Mun, Paul Hongsuck Seo, Ilchae Jung, and Bohyung Han. “MarioQA: Answering questions by watching gameplay videos”. *CVPR*. 2017 (cit. on pp. 32, 57).
- [Mun, 2019] Jonghwan Mun, Linjie Yang, Zhou Ren, Ning Xu, and Bohyung Han. “Streamlined dense video captioning”. *CVPR*. 2019 (cit. on pp. 27, 98, 111).
- [Nagaraja, 2016] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. “Modeling context between objects for referring expression understanding”. *ECCV*. 2016 (cit. on pp. 78, 79).
- [Nagrani, 2022] Arsha Nagrani, Paul Hongsuck Seo, Bryan Seybold, Anja Hauth, Santiago Manen, Chen Sun, et al. “Learning Audio-Video Modalities from Image Captions”. *ECCV*. 2022 (cit. on pp. 19, 99, 118, 120).
- [Nagrani, 2021] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. “Attention bottlenecks for multimodal fusion”. 2021 (cit. on p. 25).
- [Nan, 2021] Guoshun Nan, Rui Qiao, Yao Xiao, Jun Liu, Sicong Leng, Hao Zhang, et al. “Interventional video grounding with dual contrastive learning”. *CVPR*. 2021 (cit. on p. 121).
- [Oord, 2018] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. “Representation learning with contrastive predictive coding”. *arXiv preprint arXiv:1807.03748* (2018) (cit. on p. 17).
- [OpenAI, 2023a] OpenAI. *ChatGPT*. <https://chat.openai.com/chat>. 2023 (cit. on p. 16).
- [OpenAI, 2023b] OpenAI. “GPT-4 Technical Report”. *ArXiv* (2023) (cit. on p. 142).
- [Ordonez, 2011] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. “Im2text: Describing images using 1 million captioned photographs”. *NeurIPS*. 2011 (cit. on pp. 18, 119).
- [Ouyang, 2022] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, et al. “Training language models to follow instructions with human feedback”. *NeurIPS*. 2022 (cit. on p. 16).
- [Pan, 2016] Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang. “Hierarchical recurrent neural encoder for video representation with application to captioning”. *CVPR*. 2016, pp. 1029–1038 (cit. on p. 24).
- [Pan, 2017] Yingwei Pan, Ting Yao, Houqiang Li, and Tao Mei. “Video captioning with transferred semantic attributes”. *CVPR*. 2017 (cit. on pp. 98, 120).
- [Papineni, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. “BLEU: a method for automatic evaluation of machine translation”. *ACL*. 2002 (cit. on p. 128).
- [Park, 2019] Jae Sung Park, Marcus Rohrbach, Trevor Darrell, and Anna Rohrbach. “Adversarial inference for multi-sentence video description”. *CVPR*. 2019 (cit. on pp. 98, 112, 113).
- [Park, 2021] Jungin Park, Jiyoung Lee, and Kwanghoon Sohn. “Bridge to Answer: Structure-aware Graph Interaction Network for Video Question Answering”. *CVPR*. 2021 (cit. on pp. 30, 47, 57).
- [Patil, 2020] Suraj Patil. *Question Generation using Transformers*. https://github.com/patil-suraj/question_generation. 2020 (cit. on p. 34).
- [Patrick, 2021] Mandela Patrick, Dylan Campbell, Yuki M Asano, Ishan Misra Florian Metzke, Christoph Feichtenhofer, Andrea Vedaldi, et al. “Keeping Your Eye on the Ball: Trajectory Attention in Video Transformers”. *NeurIPS*. 2021 (cit. on pp. 78, 80).
- [Peng, 2023] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. “Instruction tuning with GPT-4”. *arXiv preprint arXiv:2304.03277* (2023) (cit. on p. 142).
- [Pennington, 2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. “GloVe: Global Vectors for Word Representation”. *EMNLP*. 2014 (cit. on p. 12).
- [Peters, 2018] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, et al. “Deep Contextualized Word Representations”. *NAACL*. 2018 (cit. on p. 13).
- [Plummer, 2015] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. “Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models”. *ICCV*. 2015 (cit. on pp. 23, 85).

- [Qi, 2020] Di Qi, Lin Su, Jia Song, Edward Cui, Taroon Bharti, and Arun Sacheti. “ImageBERT: Cross-modal pre-training with large-scale weak-supervised image-text data”. *arXiv preprint arXiv:2001.07966* (2020) (cit. on p. 119).
- [Qing, 2021] Zhiwu Qing, Haisheng Su, Weihao Gan, Dongliang Wang, Wei Wu, Xiang Wang, et al. “Temporal context aggregation network for temporal action proposal refinement”. *CVPR*. 2021 (cit. on p. 20).
- [Radford, 2021] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, et al. “Learning transferable visual models from natural language supervision”. *ICML*. 2021 (cit. on pp. 1, 7, 18, 31, 60, 69, 70, 72, 78, 102, 109, 119, 122, 129, 134).
- [Radford, 2023] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. “Robust speech recognition via large-scale weak supervision”. *ICML*. 2023 (cit. on pp. 20, 122).
- [Radford, 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. “Language Models are Unsupervised Multitask Learners”. *OpenAI blog* (2019) (cit. on p. 15).
- [Rae, 2021] Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, et al. “Scaling Language Models: Methods, Analysis & Insights from Training Gopher”. *ArXiv abs/2112.11446* (2021) (cit. on p. 15).
- [Raffel, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, et al. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. *JMLR* (2020) (cit. on pp. 7, 15, 21, 33, 34, 56, 57, 97, 103, 104, 129, 130).
- [Rahman, 2019] Tanzila Rahman, Bicheng Xu, and Leonid Sigal. “Watch, listen and tell: Multi-modal weakly supervised dense event captioning”. *ICCV*. 2019 (cit. on pp. 27, 98).
- [Rajpurkar, 2016] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. “SQuAD: 100,000+ Questions for Machine Comprehension of Text”. *arXiv preprint arXiv:1606.05250* (2016) (cit. on pp. 1, 34, 54, 57).
- [Ramachandram, 2017] Dhanesh Ramachandram and Graham W Taylor. “Deep multimodal learning: A survey on recent advances and trends”. *IEEE signal processing magazine* (2017) (cit. on p. 25).
- [Ramesh, 2022] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. “Hierarchical text-conditional image generation with clip latents”. *arXiv preprint arXiv:2204.06125* (2022) (cit. on p. 143).
- [Ramesh, 2021] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, et al. “Zero-shot text-to-image generation”. *ICML*. 2021 (cit. on p. 143).
- [Rao, 2020] Anyi Rao, Linning Xu, Yu Xiong, Guodong Xu, Qingqiu Huang, Bolei Zhou, et al. “A local-to-global approach to multi-modal movie scene segmentation”. *CVPR*. 2020 (cit. on p. 120).
- [Rasheed, 2003] Zeeshan Rasheed and Mubarak Shah. “Scene detection in Hollywood movies and TV shows”. *CVPR*. 2003 (cit. on p. 120).
- [Redmon, 2016] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. “You only look once: Unified, real-time object detection”. *CVPR*. 2016 (cit. on p. 17).
- [Ren, 2015a] Mengye Ren, Ryan Kiros, and Richard Zemel. “Exploring models and data for image question answering”. *NeurIPS*. 2015 (cit. on p. 32).
- [Ren, 2015b] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. “Faster R-CNN: Towards real-time object detection with region proposal networks”. *NeurIPS*. 2015 (cit. on p. 17).
- [Rezatofghi, 2019] Hamid Rezatofghi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. “Generalized Intersection over Union”. *CVPR*. 2019 (cit. on p. 85).
- [Richard, 2016] Alexander Richard and Juergen Gall. “Temporal action detection using a statistical language model”. *CVPR*. 2016 (cit. on p. 20).
- [Rodriguez, 2020] Cristian Rodriguez, Edison Marrese-Taylor, Fatemeh Sadat Saleh, HONGDONG LI, and Stephen Gould. “Proposal-free Temporal Moment Localization of a Natural-Language Query in Video using Guided Attention”. *WACV*. 2020 (cit. on pp. 79, 84, 85).
- [Rohrbach, 2014] Anna Rohrbach, Marcus Rohrbach, Wei Qiu, Annemarie Friedrich, Manfred Pinkal, and Bernt Schiele. “Coherent multi-sentence video description with variable level of detail”. *Pattern Recognition*. 2014 (cit. on p. 26).
- [Rohrbach, 2015] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. “A dataset for movie description”. *CVPR*. 2015 (cit. on pp. 27, 63).
- [Rohrbach, 2017] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, et al. “Movie description”. *IJCV* (2017) (cit. on p. 63).

- [Rui, 1998] Yong Rui, Thomas S Huang, and Sharad Mehrotra. “Exploring video structure beyond the shots”. *IEEE International Conference on Multimedia Computing and Systems*. 1998 (cit. on p. 120).
- [Rumelhart, 1987] David E. Rumelhart and James L. McClelland. “Learning Internal Representations by Error Propagation”. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*. 1987 (cit. on p. 13).
- [Russakovsky, 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, et al. “ImageNet Large Scale Visual Recognition Challenge”. *IJCV* (2015) (cit. on pp. 1, 16).
- [Sadhu, 2021] Arka Sadhu, Kan Chen, and Ram Nevatia. “Video Question Answering with Phrases via Semantic Roles”. *NAACL*. 2021 (cit. on pp. 32, 57).
- [Saharia, 2022] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, et al. “Photorealistic text-to-image diffusion models with deep language understanding”. *NeurIPS*. 2022 (cit. on p. 143).
- [Salazar, 2020] Julian Salazar, Davis Liang, Toan Q Nguyen, and Katrin Kirchhoff. “Masked language model scoring”. *ACL*. 2020 (cit. on p. 68).
- [Sanh, 2019] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. *arXiv preprint arXiv:1910.01108* (2019) (cit. on pp. 15, 37–39, 56).
- [Schick, 2021a] Timo Schick and Hinrich Schütze. “Exploiting cloze questions for few shot text classification and natural language inference”. *EACL*. 2021 (cit. on pp. 56, 61).
- [Schick, 2021b] Timo Schick and Hinrich Schütze. “It’s not just size that matters: Small language models are also few-shot learners”. *NAACL*. 2021 (cit. on p. 56).
- [Schuhmann, 2022] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, et al. “LAION-5B: An open large-scale dataset for training next generation image-text models”. *NeurIPS*. 2022 (cit. on pp. 18, 119, 125, 127).
- [Sengupta, 2021] Kinshuk Sengupta, Rana Maher, Declan Groves, and Chantal Olieman. “GenBiT: measure and mitigate gender bias in language datasets”. *Microsoft Journal of Applied Research* (2021) (cit. on p. 125).
- [Seo, 2021a] Ahjeong Seo, Gi-Cheon Kang, Joonhan Park, and Byoung-Tak Zhang. “Attend What You Need: Motion-Appearance Synergistic Networks for Video Question Answering”. *ACL*. 2021 (cit. on pp. 30, 47).
- [Seo, 2022] Paul Hongsuck Seo, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. “End-to-end Generative Pretraining for Multimodal Video Captioning”. *CVPR*. 2022 (cit. on pp. 25, 57, 97, 99, 107, 113, 120).
- [Seo, 2021b] Paul Hongsuck Seo, Arsha Nagrani, and Cordelia Schmid. “Look Before you Speak: Visually Contextualized Utterances”. *CVPR*. 2021 (cit. on pp. 32, 47, 48, 57, 64, 99, 120).
- [Shah, 2019] Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. “Cycle-Consistency for Robust Visual Question Answering”. *CVPR*. 2019 (cit. on p. 31).
- [Sharma, 2018] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. “Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning”. *ACL*. 2018 (cit. on pp. 18, 21, 32, 79, 119).
- [Shen, 2021] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, et al. “How Much Can CLIP Benefit Vision-and-Language Tasks?” *arXiv preprint arXiv:2107.06383* (2021) (cit. on p. 58).
- [Shen, 2017] Zhiqiang Shen, Jianguo Li, Zhou Su, Minjun Li, Yurong Chen, Yu-Gang Jiang, et al. “Weakly supervised dense video captioning”. *CVPR*. 2017 (cit. on pp. 27, 98).
- [Shi, 2019a] Botian Shi, Lei Ji, Yaobo Liang, Nan Duan, Peng Chen, Zhendong Niu, et al. “Dense procedure captioning in narrated instructional videos”. *ACL*. 2019 (cit. on pp. 27, 98).
- [Shi, 2019b] Jing Shi, Jia Xu, Boqing Gong, and Chenliang Xu. “Not all frames are equal: Weakly-supervised video grounding with contextual similarity and visual clustering losses”. *CVPR*. 2019 (cit. on p. 79).
- [Shou, 2017] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. “CDC: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos”. *CVPR*. 2017 (cit. on p. 20).
- [Shou, 2016] Zheng Shou, Dongang Wang, and Shih-Fu Chang. “Temporal action localization in untrimmed videos via multi-stage CNNs”. *CVPR*. 2016 (cit. on pp. 98, 120).

- [Sidiropoulos, 2011] Panagiotis Sidiropoulos, Vasileios Mezaris, Ioannis Kompatsiaris, Hugo Meinedo, Miguel Bugalho, and Isabel Trancoso. “Temporal video segmentation to scenes using high-level audiovisual features”. *IEEE TCSVT* (2011) (cit. on p. 120).
- [Simonyan, 2015] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. *ICLR*. 2015 (cit. on p. 16).
- [Singer, 2022] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, et al. “Make-A-Video: Text-to-video generation without text-video data”. *arXiv preprint arXiv:2209.14792* (2022) (cit. on p. 143).
- [Singh, 2022] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, et al. “FLAVA: A foundational language and vision alignment model”. *CVPR*. 2022 (cit. on pp. 21, 58, 99, 119).
- [Singh, 2016] Bharat Singh, Tim K Marks, Michael Jones, Oncel Tuzel, and Ming Shao. “A multi-stream bi-directional recurrent neural network for fine-grained action detection”. *CVPR*. 2016 (cit. on p. 20).
- [So, 2021] David R So, Wojciech Mańke, Hanxiao Liu, Zihang Dai, Noam Shazeer, and Quoc V Le. “Primer: Searching for efficient transformers for language modeling”. *NeurIPS*. 2021 (cit. on p. 56).
- [Song, 2018] Xiaomeng Song, Yucheng Shi, Xin Chen, and Yahong Han. “Explore Multi-Step Reasoning in Video Question Answering”. *ACM international conference on Multimedia*. 2018 (cit. on pp. 32, 57).
- [Srinivasan, 2021] Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. “Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning”. *ACM SIGIR Conference on Research and Development in Information Retrieval*. 2021 (cit. on p. 119).
- [Srivastava, 2014] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. *JMLR* (2014) (cit. on pp. 39, 64, 86, 106).
- [Stein, 2013] Sebastian Stein and Stephen J McKenna. “Combining embedded accelerometers with computer vision for recognizing food preparation activities”. *Proceedings of the 2013 ACM international joint conference on pervasive and ubiquitous computing*. 2013 (cit. on p. 20).
- [Steiner, 2022] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. “How to train your vit? data, augmentation, and regularization in vision transformers”. *TMLR*. 2022 (cit. on p. 109).
- [Su, 2021] Rui Su, Qian Yu, and Dong Xu. “STVGBert: A Visual-Linguistic Transformer Based Framework for Spatio-Temporal Video Grounding”. *ICCV*. 2021 (cit. on pp. 26, 78, 79, 90).
- [Su, 2019] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, et al. “VL-BERT: Pre-training of Generic Visual-Linguistic Representations”. *ICLR*. 2019 (cit. on pp. 21, 32, 58, 80, 99, 119).
- [Suhr, 2019] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. “A Corpus for Reasoning about Natural Language Grounded in Photographs”. *ACL*. 2019 (cit. on p. 21).
- [Sun, 2019a] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. “Contrastive Bidirectional Transformer for Temporal Representation Learning”. *arXiv preprint arXiv:1906.05743* (2019) (cit. on p. 32).
- [Sun, 2019b] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. “VideoBERT: A Joint Model for Video and Language Representation Learning”. *ICCV*. 2019 (cit. on pp. 24, 32, 43, 58, 80, 99, 120).
- [Sun, 2022] Yuchong Sun, Hongwei Xue, Ruihua Song, Bei Liu, Huan Yang, and Jianlong Fu. “Long-Form Video-Language Pre-Training with Multimodal Temporal Contrastive Learning”. *NeurIPS*. 2022 (cit. on pp. 99, 120).
- [Sutskever, 2011] Ilya Sutskever, James Martens, and Geoffrey E Hinton. “Generating text with recurrent neural networks”. *ICML*. 2011 (cit. on p. 6).
- [Szegedy, 2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, et al. “Going deeper with convolutions”. *CVPR*. 2015 (cit. on p. 16).
- [Szegedy, 2016] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. “Rethinking the inception architecture for computer vision”. *CVPR*. 2016 (cit. on p. 106).
- [Tam, 2021] Derek Tam, Rakesh R Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. “Improving and simplifying pattern exploiting training”. *EMNLP*. 2021 (cit. on p. 56).
- [Tan, 2019] Hao Tan and Mohit Bansal. “LXMERT: Learning Cross-Modality Encoder Representations from Transformers”. *EMNLP*. 2019 (cit. on pp. 32, 58, 80, 99, 119).

- [Tan, 2021] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. “Relaxed transformer decoders for direct action proposal generation”. *CVPR*. 2021 (cit. on p. 20).
- [Tang, 2022] Zineng Tang, Jaemin Cho, Yixin Nie, and Mohit Bansal. “TVLT: Textless Vision-Language Transformer”. *NeurIPS*. 2022 (cit. on p. 120).
- [Tang, 2023] Zineng Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal. “Any-to-Any Generation via Composable Diffusion”. *NeurIPS*. 2023 (cit. on p. 143).
- [Tang, 2021] Zongheng Tang, Yue Liao, Si Liu, Guanbin Li, Xiaojie Jin, Hongxu Jiang, et al. “Human-centric spatio-temporal video grounding with visual transformers”. *IEEE TCSVT* (2021) (cit. on pp. iv, 11, 26, 77–79, 85, 90).
- [Tapaswi, 2016] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. “MovieQA: Understanding Stories in Movies through Question-Answering”. *CVPR*. 2016 (cit. on pp. 24, 31, 32, 57).
- [Tay, 2021] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, et al. “Long range arena: A benchmark for efficient transformers”. *ICLR*. 2021 (cit. on p. 95).
- [Taylor, 1953] Wilson L Taylor. ““Cloze procedure”: A new tool for measuring readability”. *Journalism quarterly* (1953) (cit. on pp. 56, 61).
- [Teney, 2016] Damien Teney and Anton van den Hengel. “Zero-Shot Visual Question Answering”. *arXiv preprint arXiv:1611.05546* (2016) (cit. on p. 31).
- [Thomee, 2016] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, et al. “YFCC100M: The new data in multimedia research”. *Communications of the ACM* (2016) (cit. on p. 27).
- [Tilk, 2016] Ottokar Tilk and Tanel Alumäe. “Bidirectional Recurrent Neural Network with Attention Mechanism for Punctuation Restoration”. *Interspeech*. 2016 (cit. on p. 34).
- [Tilk, 2017] Ottokar Tilk and Tanel Alumäe. *Punctuator*. <https://github.com/ottokart/punctuator2>. 2017 (cit. on p. 34).
- [Tomar, 2006] Suramya Tomar. “Converting video formats with FFmpeg”. *Linux Journal* (2006) (cit. on pp. 128, 129, 131, 135).
- [Tong, 2022] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. “VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training”. *NeurIPS*. 2022 (cit. on p. 19).
- [Touvron, 2023] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, et al. “LLaMA: Open and efficient foundation language models”. *arXiv preprint arXiv:2302.13971* (2023) (cit. on pp. 15, 128, 129, 131, 133, 135).
- [Tran, 2015] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. “Learning spatiotemporal features with 3D convolutional networks”. *ICCV*. 2015 (cit. on p. 18).
- [Tran, 2018] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. “A Closer Look at Spatiotemporal Convolutions for Action Recognition”. *CVPR*. 2018 (cit. on p. 19).
- [Tsimpoukelli, 2021] Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. “Multimodal few-shot learning with frozen language models”. *NeurIPS*. 2021 (cit. on pp. iv, 11, 22, 56, 58, 69, 70, 73, 99, 119).
- [Vasudevan, 2018] Arun Balajee Vasudevan, Dengxin Dai, and Luc Van Gool. “Object referring in videos with language and human gaze”. *CVPR*. 2018 (cit. on pp. 78, 79).
- [Vaswani, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, et al. “Attention Is All You Need”. *NeurIPS*. 2017 (cit. on pp. 7, 13, 14, 16, 37, 56, 60, 78, 97, 102).
- [Vatashsky, 2020] Ben-Zion Vatashsky and Shimon Ullman. “VQA with no questions-answers training”. *CVPR*. 2020 (cit. on p. 31).
- [Vedantam, 2015] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. “CIDEr: Consensus-based image description evaluation”. *CVPR*. 2015 (cit. on pp. 106, 128).
- [Ventura, 2023] Lucas Ventura, Antoine Yang, Cordelia Schmid, and Gül Varol. “CoVR: Learning Composed Video Retrieval from Web Video Captions”. *arXiv preprint arXiv:2308.14746* (2023) (cit. on p. 9).
- [Venugopalan, 2015] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. “Sequence to sequence-video to text”. *ICCV*. 2015 (cit. on p. 24).
- [Vijgen, 2014] Bram Vijgen et al. “The listicle: An exploring research on an interesting shareable new media phenomenon”. *Studia Universitatis Babeş-Bolyai-Ephemerides* 59.1 (2014), pp. 103–122 (cit. on p. 118).

- [Vinyals, 2015] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. “Show and Tell: A Neural Image Caption Generator”. *CVPR*. 2015 (cit. on p. 21).
- [Wang, 2023] Alex Jinpeng Wang, Yixiao Ge, Rui Yan, Yuying Ge, Xudong Lin, Guanyu Cai, et al. “All in One: Exploring Unified Video-Language Pre-training”. *CVPR*. 2023 (cit. on pp. 58, 73, 74, 99, 120, 142).
- [Wang, 2018a] Bairui Wang, Lin Ma, Wei Zhang, and Wei Liu. “Reconstruction network for video captioning”. *CVPR*. 2018 (cit. on pp. 98, 120).
- [Wang, 2021a] Ben Wang and Aran Komatsuzaki. *GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model*. 2021 (cit. on pp. 57, 70).
- [Wang, 2021b] Jianfeng Wang, Xiaowei Hu, Zhe Gan, Zhengyuan Yang, Xiyang Dai, Zicheng Liu, et al. “UFO: A UniFied TransFormer for Vision-Language Representation Learning”. *arXiv preprint arXiv:2111.10023* (2021) (cit. on pp. 58, 99).
- [Wang, 2022a] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, et al. “GIT: A generative image-to-text transformer for vision and language”. *TMLR*. 2022 (cit. on pp. 22, 119).
- [Wang, 2018b] Jingwen Wang, Wenhao Jiang, Lin Ma, Wei Liu, and Yong Xu. “Bidirectional attentive fusion with context gating for dense video captioning”. *CVPR*. 2018 (cit. on pp. 26, 27, 97, 98).
- [Wang, 2020a] Jingwen Wang, Lin Ma, and Wenhao Jiang. “Temporally grounding language queries in videos by contextual boundary-aware prediction”. *AAAI*. 2020 (cit. on p. 79).
- [Wang, 2022b] Jinpeng Wang, Yixiao Ge, Guanyu Cai, Rui Yan, Xudong Lin, Ying Shan, et al. “Object-aware Video-language Pre-training for Retrieval”. *CVPR*. 2022 (cit. on pp. 99, 120).
- [Wang, 2021c] Liwei Wang, Jing Huang, Yin Li, Kun Xu, Zhengyuan Yang, and Dong Yu. “Improving Weakly Supervised Visual Grounding by Contrastive Knowledge Distillation”. *CVPR*. 2021 (cit. on p. 79).
- [Wang, 2019a] Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel. “Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks”. *CVPR*. 2019 (cit. on p. 79).
- [Wang, 2022c] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, et al. “Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework”. *ICML*. 2022 (cit. on pp. 24, 99).
- [Wang, 2020b] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. “Linformer: Self-attention with linear complexity”. *arXiv preprint arXiv:2006.04768* (2020) (cit. on p. 95).
- [Wang, 2021d] Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. “End-to-end dense video captioning with parallel decoding”. *ICCV*. 2021 (cit. on pp. 7, 26, 27, 97, 98, 106, 111–113, 118, 120, 129–131, 135, 138).
- [Wang, 2020c] Teng Wang, Huicheng Zheng, Mingjing Yu, Qian Tian, and Haifeng Hu. “Event-centric hierarchical representation for dense video captioning”. *IEEE TCSVT* (2020) (cit. on pp. 26, 98, 112).
- [Wang, 2019b] Weining Wang, Yan Huang, and Liang Wang. “Language-driven temporal activity localization: A semantic matching reinforcement learning model”. *CVPR*. 2019 (cit. on p. 79).
- [Wang, 2020d] Weining Wang, Yan Huang, and Liang Wang. “Long video question answering: A Matching-guided Attention Model”. *Pattern Recognition* (2020) (cit. on p. 32).
- [Wang, 2018c] Xin Wang, Wenhao Chen, Jiawei Wu, Yuan-Fang Wang, and William Yang Wang. “Video captioning via hierarchical reinforcement learning”. *CVPR*. 2018 (cit. on pp. 98, 120).
- [Wang, 2022d] Yuxuan Wang, Difei Gao, Licheng Yu, Weixian Lei, Matt Feiszli, and Mike Zheng Shou. “GEB+: A Benchmark for Generic Event Boundary Captioning, Grounding and Retrieval”. *ECCV*. 2022 (cit. on p. 120).
- [Wang, 2022e] Zhenhailong Wang, Manling Li, Ruochen Xu, Luwei Zhou, Jie Lei, Xudong Lin, et al. “Language Models with Image Descriptors are Strong Few-Shot Video-Language Learners”. *NeurIPS*. 2022 (cit. on pp. 56, 58).
- [Wang, 2020e] Zhenzhi Wang, Ziteng Gao, Limin Wang, Zhifeng Li, and Gangshan Wu. “Boundary-aware cascade networks for temporal action segmentation”. *ECCV*. 2020 (cit. on pp. 20, 120).
- [Wang, 2022f] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. “SimVLM: Simple visual language model pretraining with weak supervision”. *ICLR*. 2022 (cit. on pp. 21, 97, 119).
- [Wang, 2022g] Zixu Wang, Yujie Zhong, Yishu Miao, Lin Ma, and Lucia Specia. “Contrastive Video-Language Learning with Fine-grained Frame Sampling”. *AAACL-IJCNLP*. 2022 (cit. on p. 99).

- [Wei, 2022a] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, et al. “Finetuned language models are zero-shot learners”. *ICLR*. 2022 (cit. on pp. 15, 129).
- [Wei, 2022b] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai-hsin Chi, F. Xia, et al. “Chain of Thought Prompting Elicits Reasoning in Large Language Models”. *NeurIPS*. 2022 (cit. on p. 16).
- [Winterbottom, 2020] Thomas Winterbottom, Sarah Xiao, Alistair McLean, and Noura Al Moubayed. “On Modality Bias in the TVQA Dataset”. *BMVC*. 2020 (cit. on p. 31).
- [Wu, 2022] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, et al. “MeMViT: Memory-Augmented Multiscale Vision Transformer for Efficient Long-Term Video Recognition”. *CVPR*. 2022 (cit. on p. 142).
- [Wu, 2017] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. “Sampling matters in deep embedding learning”. *ICCV* (2017) (cit. on p. 21).
- [Wu, 2016] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, et al. “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation”. *arXiv preprint arXiv:1609.08144* (2016) (cit. on p. 37).
- [Wu, 2020] Zhanghao Wu, Zhijian Liu, Ji Lin, Yujun Lin, and Song Han. “Lite transformer with long-short range attention”. *ICLR*. 2020 (cit. on p. 95).
- [Xiao, 2020] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. “Audiovisual slowfast networks for video recognition”. *arXiv preprint arXiv:2001.08740* (2020) (cit. on pp. 25, 80).
- [Xiao, 2017] Fanyi Xiao, Leonid Sigal, and Yong Jae Lee. “Weakly-supervised visual grounding of phrases with linguistic structures”. *CVPR*. 2017 (cit. on p. 79).
- [Xiao, 2021] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. “NExT-QA: Next Phase of Question-Answering to Explaining Temporal Actions”. *CVPR*. 2021 (cit. on pp. 32, 57).
- [Xie, 2018] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. “Rethinking Spatiotemporal Feature Learning: Speed-Accuracy Trade-offs in Video Classification”. *ECCV*. 2018 (cit. on pp. 19, 38).
- [Xie, 2022] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, et al. “SimMIM: A simple framework for masked image modeling”. *CVPR*. 2022 (cit. on p. 17).
- [Xiong, 2016] Caiming Xiong, Stephen Merity, and Richard Socher. “Dynamic Memory Networks for Visual and Textual Question Answering”. *ICML*. 2016 (cit. on p. 30).
- [Xiong, 2018] Yilei Xiong, Bo Dai, and Dahua Lin. “Move forward and tell: A progressive generator of video descriptions”. *ECCV*. 2018 (cit. on p. 113).
- [Xu, 2019] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. “Self-Supervised Spatiotemporal Learning via Video Clip Order Prediction”. *CVPR*. 2019 (cit. on p. 19).
- [Xu, 2017] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, et al. “Video Question Answering via Gradually Refined Attention over Appearance and Motion”. *ACM international conference on Multimedia*. 2017 (cit. on pp. 5, 24, 28, 30, 32, 43, 47, 50, 55, 56, 62, 63, 68).
- [Xu, 2021] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, et al. “VideoCLIP: Contrastive pre-training for zero-shot video-text understanding”. *EMNLP*. 2021 (cit. on pp. 57, 99, 120).
- [Xu, 2016a] Huijuan Xu and Kate Saenko. “Ask, attend and answer: Exploring question-guided spatial attention for visual question answering”. *ECCV*. 2016 (cit. on p. 30).
- [Xu, 2016b] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. “MSR-VTT: A Large Video Description Dataset for Bridging Video and Language”. *CVPR*. 2016 (cit. on pp. 64, 98, 105, 106).
- [Xu, 2022] Mengmeng Xu, Erhan Gundogdu, Maksim Lapin, Bernard Ghanem, Michael Donoser, and Loris Bazzani. “Contrastive Language-Action Pre-training for Temporal Localization”. *arXiv preprint arXiv:2204.12293* (2022) (cit. on p. 99).
- [Xu, 2020] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. “G-TAD: Sub-graph localization for temporal action detection”. *CVPR*. 2020 (cit. on p. 20).
- [Xu, 2015] Ran Xu, Caiming Xiong, Wei Chen, and Jason Corso. “Jointly modeling deep video and compositional text to bridge vision and language in a unified framework”. *AAAI*. 2015 (cit. on p. 24).
- [Xue, 2022] Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, et al. “Advancing High-Resolution Video-Language Representation with Large-Scale Video Transcriptions”. *CVPR*. 2022 (cit. on pp. 19, 99, 120).

- [Xue, 2018] Hongyang Xue, Wenqing Chu, Zhou Zhao, and Deng Cai. “A better way to attend: Attention with trees for video question answering”. *IEEE Transactions on Image Processing* (2018) (cit. on pp. 30, 57).
- [Xue, 2021] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, et al. “mT5: A massively multilingual pre-trained text-to-text transformer”. *NAACL*. 2021 (cit. on p. 130).
- [Yamaguchi, 2017] Masataka Yamaguchi, Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. “Spatio-temporal person retrieval via natural language queries”. *ICCV*. 2017 (cit. on p. 79).
- [Yang, 2021a] Antoine Yang. *Just Ask project webpage*. <https://antoyang.github.io/just-ask.html>. 2021 (cit. on pp. 28, 45).
- [Yang, 2022a] Antoine Yang. *FrozenBiLM project webpage*. <https://antoyang.github.io/frozenbilm.html>. 2022 (cit. on pp. 55, 71).
- [Yang, 2022b] Antoine Yang. *TubeDETR project webpage*. <https://antoyang.github.io/tubedetr.html>. 2022 (cit. on pp. 77, 91).
- [Yang, 2023a] Antoine Yang. *Vid2Seq project webpage*. <https://antoyang.github.io/vid2seq.html>. 2023 (cit. on pp. 96, 115).
- [Yang, 2023b] Antoine Yang. *VidChapters-7M project webpage*. <https://antoyang.github.io/vidchapters.html>. 2023 (cit. on p. 117).
- [Yang, 2021b] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. “Just Ask: Learning To Answer Questions From Millions of Narrated Videos”. *ICCV*. 2021 (cit. on pp. 8, 9, 41, 55–57, 62–64, 68, 70, 72, 80, 99, 120).
- [Yang, 2022c] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. “Learning to Answer Visual Questions from Web Videos.” *IEEE TPAMI* (2022) (cit. on pp. 8–10, 56, 70, 72–74, 99, 120).
- [Yang, 2022d] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. “TubeDETR: Spatio-Temporal Video Grounding With Transformers”. *CVPR*. 2022 (cit. on pp. 8, 9, 99, 121).
- [Yang, 2022e] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. “Zero-shot video question answering via frozen bidirectional language models”. *NeurIPS*. 2022 (cit. on pp. 8, 9, 99, 120).
- [Yang, 2023c] Antoine Yang, Arsha Nagrani, Ivan Laptev, Josef Sivic, and Cordelia Schmid. “VidChapters-7M: Video Chapters at Scale”. *NeurIPS*. 2023 (cit. on pp. 8–10, 122).
- [Yang, 2023d] Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, et al. “Vid2Seq: Large-Scale Pretraining of a Visual Language Model for Dense Video Captioning”. *CVPR*. 2023 (cit. on pp. 8, 9, 118, 119, 129–131, 133, 135, 136, 138).
- [Yang, 2021c] Jianwei Yang, Yonatan Bisk, and Jianfeng Gao. “Taco: Token-aware cascade contrastive learning for video-text alignment”. *ICCV*. 2021 (cit. on p. 99).
- [Yang, 2019a] Sibe Yang, Guanbin Li, and Yizhou Yu. “Cross-modal relationship inference for grounding referring expressions”. *CVPR*. 2019 (cit. on p. 79).
- [Yang, 2019b] Sibe Yang, Guanbin Li, and Yizhou Yu. “Dynamic graph attention for referring expression comprehension”. *ICCV*. 2019 (cit. on p. 79).
- [Yang, 2020a] Zekun Yang, Noa Garcia, Chenhui Chu, Mayu Otani, Yuta Nakashima, and Haruo Takemura. “BERT Representations for Video Question Answering”. *WACV*. 2020 (cit. on pp. 31, 57).
- [Yang, 2020b] Zhengyuan Yang, Tianlang Chen, Liwei Wang, and Jiebo Luo. “Improving one-stage visual grounding by recursive sub-query construction”. *ECCV*. 2020 (cit. on p. 79).
- [Yang, 2021d] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, et al. “Crossing the format boundary of text and boxes: Towards unified vision-language modeling”. *ECCV*. 2021 (cit. on pp. 97, 99).
- [Yang, 2021e] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, et al. “UniTAB: Unifying Text and Box Outputs for Grounded Vision-Language Modeling”. *ECCV*. 2021 (cit. on p. 24).
- [Yang, 2021f] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, et al. “An empirical study of GPT-3 for few-shot knowledge-based VQA”. *arXiv preprint arXiv:2109.05014* (2021) (cit. on pp. 56, 58, 69).
- [Yang, 2019c] Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. “A fast and accurate one-stage approach to visual grounding”. *ICCV*. 2019 (cit. on p. 79).

- [Yang, 2023e] Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, et al. “MM-REACT: Prompting chatgpt for multimodal reasoning and action”. *arXiv preprint arXiv:2303.11381* (2023) (cit. on pp. 142, 143).
- [Yang, 2016] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. “Stacked Attention Networks for Image Question Answering”. *CVPR*. 2016 (cit. on pp. 21, 30).
- [Yao, 2018] Kaichun Yao, Libo Zhang, Tiejian Luo, Lili Tao, and Yanjun Wu. “Teaching Machines to Ask Questions”. *IJCAI*. 2018 (cit. on p. 32).
- [Ye, 2017] Yunan Ye, Zhou Zhao, Yimeng Li, Long Chen, Jun Xiao, and Yueting Zhuang. “Video Question Answering via Attribute-Augmented Attention Network Learning”. *ACM SIGIR*. 2017 (cit. on pp. 32, 57).
- [Yu, 2020] Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, et al. “ERNIE-ViL: Knowledge enhanced vision-language representations through scene graph”. *AAAI*. 2020 (cit. on pp. 58, 99, 119).
- [Yu, 2016] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. “Video paragraph captioning using hierarchical recurrent neural networks”. *CVPR*. 2016 (cit. on p. 24).
- [Yu, 2022a] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. “CoCa: Contrastive Captioners are Image-Text Foundation Models”. *TMLR (2022)* (cit. on pp. 22, 99).
- [Yu, 2022b] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, et al. “Scaling autoregressive models for content-rich text-to-image generation”. *arXiv preprint arXiv:2206.10789* (2022) (cit. on p. 143).
- [Yu, 2018] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, et al. “MAttNet: Modular attention network for referring expression comprehension”. *CVPR*. 2018 (cit. on p. 79).
- [Yu, 2017] Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L Berg. “A joint speaker-listener-reinforcer model for referring expressions”. *CVPR*. 2017 (cit. on p. 79).
- [Yu, 2023] Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. “Self-Chained Image-Language Model for Video Localization and Question Answering”. *NeurIPS*. 2023 (cit. on p. 142).
- [Yu, 2021] Weijiang Yu, Haoteng Zheng, Mengfei Li, Lei Ji, Lijun Wu, Nong Xiao, et al. “Learning from Inside: Self-driven Siamese Sampling and Reasoning for Video Question Answering”. *NeurIPS*. 2021 (cit. on pp. 30, 47, 48, 56, 64, 74).
- [Yu, 2019] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, et al. “ActivityNet-QA: A Dataset for Understanding Complex Web Videos via Question Answering”. *AAAI*. 2019 (cit. on pp. 24, 28, 30, 32, 43, 47, 55, 62–64, 68).
- [Yuan, 2021] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, et al. “Florence: A new foundation model for computer vision”. *arXiv preprint arXiv:2111.11432* (2021) (cit. on pp. 99, 119).
- [Yuan, 2019] Yitian Yuan, Tao Mei, and Wenwu Zhu. “To find where you talk: Temporal sentence localization in video with attention based location regression”. *AAAI*. 2019 (cit. on p. 79).
- [Zadeh, 2019] Amir Zadeh, Michael Chan, Paul Pu Liang, Edmund Tong, and Louis-Philippe Morency. “Social-IQ: A question answering benchmark for artificial social intelligence”. *CVPR*. 2019 (cit. on p. 32).
- [Zaheer, 2020] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, et al. “Big Bird: Transformers for Longer Sequences.” *NeurIPS*. 2020 (cit. on p. 95).
- [Zbontar, 2021] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. “Barlow Twins: Self-supervised learning via redundancy reduction”. *ICML*. 2021 (cit. on p. 17).
- [Zellers, 2019] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. “From Recognition to Cognition: Visual Commonsense Reasoning”. *CVPR*. 2019 (cit. on p. 21).
- [Zellers, 2022] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, et al. “MERLOT Reserve: Neural Script Knowledge through Vision and Language and Sound”. *CVPR*. 2022 (cit. on pp. iv, 11, 19, 25, 56, 57, 70, 72, 74, 76, 96, 99, 103, 105, 109, 118–120, 129, 130, 140, 142).
- [Zellers, 2021] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, et al. “MERLOT: Multimodal Neural Script Knowledge Models”. *NeurIPS*. 2021 (cit. on pp. 19, 24, 25, 47, 48, 56–58, 61, 65, 73, 74, 80, 99, 105, 120, 121, 138).
- [Zeng, 2023] Andy Zeng, Adrian Wong, Stefan Welker, Krzysztof Choromanski, Federico Tombari, Aavek Purohit, et al. “Socratic Models: Composing Zero-Shot Multimodal Reasoning with Language”. *ICLR*. 2023 (cit. on pp. 56, 58).

- [Zeng, 2017] 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”. *AAAI*. 2017 (cit. on pp. 32, 50).
- [Zeng, 2016] Kuo-Hao Zeng, Tseng-Hung Chen, Juan Carlos Niebles, and Min Sun. “Title generation for user generated videos”. *ECCV*. 2016 (cit. on p. 120).
- [Zeng, 2019] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, et al. “Graph convolutional networks for temporal action localization”. *CVPR*. 2019 (cit. on pp. 20, 120).
- [Zeng, 2020] Runhao Zeng, Haoming Xu, Wenbing Huang, Peihao Chen, Mingkui Tan, and Chuang Gan. “Dense regression network for video grounding”. *CVPR*. 2020 (cit. on pp. 26, 79, 142).
- [Zha, 2019] Zheng-Jun Zha, Jiawei Liu, Tianhao Yang, and Yongdong Zhang. “Spatiotemporal-textual co-attention network for video question answering”. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* (2019) (cit. on pp. 30, 57).
- [Zhang, 2022a] Chenlin Zhang, Jianxin Wu, and Yin Li. “ActionFormer: Localizing moments of actions with transformers”. *ECCV*. 2022 (cit. on pp. 20, 116, 120).
- [Zhang, 2018a] Da Zhang, Xiyang Dai, Xin Wang, and Yuan-Fang Wang. “S3D: single shot multi-span detector via fully 3d convolutional networks”. *BMVC*. 2018 (cit. on p. 20).
- [Zhang, 2019a] Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S Davis. “Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment”. *CVPR*. 2019 (cit. on p. 79).
- [Zhang, 2023] Hang Zhang, Xin Li, and Lidong Bing. “Video-LLaMA: An instruction-tuned audio-visual language model for video understanding”. *arXiv preprint arXiv:2306.02858* (2023) (cit. on p. 142).
- [Zhang, 2018b] Hanwang Zhang, Yulei Niu, and Shih-Fu Chang. “Grounding referring expressions in images by variational context”. *CVPR*. 2018 (cit. on p. 79).
- [Zhang, 2020a] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. “Span-based localizing network for natural language video localization”. *ACL*. 2020 (cit. on pp. 26, 79, 121, 142).
- [Zhang, 2022b] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Harold Li, Xiyang Dai, et al. “GLIPv2: Unifying localization and vision-language understanding”. *NeurIPS*. 2022 (cit. on pp. 23, 99, 141).
- [Zhang, 2021a] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, et al. “VinVL: Making Visual Representations Matter in Vision-Language Models”. *CVPR*. 2021 (cit. on p. 21).
- [Zhang, 2022c] Qi Zhang, Yuqing Song, and Qin Jin. “Unifying Event Detection and Captioning as Sequence Generation via Pre-Training”. *ECCV*. 2022 (cit. on pp. 98, 99, 112).
- [Zhang, 2020b] Shengyu Zhang, Ziqi Tan, Zhou Zhao, Jin Yu, Kun Kuang, Tan Jiang, et al. “Comprehensive information integration modeling framework for video titling”. *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020 (cit. on p. 120).
- [Zhang, 2020c] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. “Learning 2D temporal adjacent networks for moment localization with natural language”. *AAAI*. 2020 (cit. on pp. 26, 79, 121, 142).
- [Zhang, 2021b] Yanyi Zhang, Xinyu Li, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, et al. “VidTr: Video Transformer Without Convolutions”. *ICCV*. 2021 (cit. on pp. 78, 80).
- [Zhang, 2019b] Zhu Zhang, Zhijie Lin, Zhou Zhao, and Zhenxin Xiao. “Cross-modal interaction networks for query-based moment retrieval in videos”. *SIGIR*. 2019 (cit. on p. 79).
- [Zhang, 2020d] Zhu Zhang, Zhou Zhao, Yang Zhao, Qi Wang, Huasheng Liu, and Lianli Gao. “Where Does It Exist: Spatio-Temporal Video Grounding for Multi-Form Sentences”. *CVPR*. 2020 (cit. on pp. iv, 5, 7, 11, 25, 26, 77–79, 82, 85, 86, 90).
- [Zhang, 2020e] Ziqi Zhang, Yaya Shi, Chunfeng Yuan, Bing Li, Peijin Wang, Weiming Hu, et al. “Object relational graph with teacher-recommended learning for video captioning”. *CVPR*. 2020 (cit. on pp. 113, 120).
- [Zhao, 2021] Chen Zhao, Ali K Thabet, and Bernard Ghanem. “Video self-stitching graph network for temporal action localization”. *ICCV*. 2021 (cit. on p. 20).
- [Zhao, 2023] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. “Learning Video Representations from Large Language Models”. *CVPR*. 2023 (cit. on pp. 20, 120).
- [Zhao, 2017a] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. “Temporal action detection with structured segment networks”. *ICCV*. 2017 (cit. on pp. 20, 98).

- [Zhao, 2020] Zhou Zhao, Shuwen Xiao, Zehan Song, Chujie Lu, Jun Xiao, and Yueting Zhuang. “Open-ended video question answering via multi-modal conditional adversarial networks”. *IEEE Transactions on Image Processing* (2020) (cit. on pp. 32, 50).
- [Zhao, 2017b] Zhou Zhao, Qifan Yang, Deng Cai, Xiaofei He, Yueting Zhuang, Zhou Zhao, et al. “Video Question Answering via Hierarchical Spatio-Temporal Attention Networks”. *IJCAI*. 2017 (cit. on pp. 32, 50).
- [Zhao, 2018] Zhou Zhao, Zhu Zhang, Shuwen Xiao, Zhou Yu, Jun Yu, Deng Cai, et al. “Open-Ended Long-form Video Question Answering via Adaptive Hierarchical Reinforced Networks”. *IJCAI*. 2018 (cit. on pp. 32, 50).
- [Zhou, 2022] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. “Learning to prompt for vision-language models”. *IJCV* (2022) (cit. on p. 60).
- [Zhou, 2018a] Luwei Zhou, Xu Chenliang, and Jason J. Corso. “Towards Automatic Learning of Procedures from Web Instructional Videos”. *AAAI*. 2018 (cit. on pp. iv, 5, 6, 11, 96, 117–120, 135).
- [Zhou, 2019] Luwei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J Corso, and Marcus Rohrbach. “Grounded video description”. *CVPR*. 2019 (cit. on pp. 98, 106, 112, 113).
- [Zhou, 2020] Luwei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. “Unified Vision-Language Pre-Training for Image Captioning and VQA”. *AAAI*. 2020 (cit. on pp. 32, 58, 80, 99, 119).
- [Zhou, 2018b] Luwei Zhou, Chenliang Xu, and Jason J Corso. “Towards Automatic Learning of Procedures From Web Instructional Videos”. *AAAI*. 2018 (cit. on pp. 25, 97, 98, 101, 103, 105).
- [Zhou, 2018c] Luwei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. “End-to-end dense video captioning with masked transformer”. *CVPR*. 2018 (cit. on pp. 27, 80, 97, 98, 112, 113, 120).
- [Zhou, 2017] Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. “Neural Question Generation from Text: A Preliminary Study”. *National CCF Conference on Natural Language Processing and Chinese Computing*. 2017 (cit. on p. 32).
- [Zhou, 2023] Xingyi Zhou, Anurag Arnab, Chen Sun, and Cordelia Schmid. “Dense Video Object Captioning from Disjoint Supervision”. *arXiv preprint arXiv:2306.11729* (2023) (cit. on p. 141).
- [Zhu, 2022a] Chaoyang Zhu, Yiyi Zhou, Yunhang Shen, Gen Luo, Xingjia Pan, Mingbao Lin, et al. “SeqTR: A Simple yet Universal Network for Visual Grounding”. *ECCV*. 2022 (cit. on p. 99).
- [Zhu, 2023] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. “MiniGPT-4: Enhancing vision-language understanding with advanced large language models”. *arXiv preprint arXiv:2304.10592* (2023) (cit. on p. 142).
- [Zhu, 2020] Linchao Zhu and Yi Yang. “ActBERT: Learning Global-Local Video-Text Representations”. *CVPR*. 2020 (cit. on pp. 25, 32, 43, 58, 80).
- [Zhu, 2022b] Wanrong Zhu, Bo Pang, Ashish Thapliyal, William Yang Wang, and Radu Soricut. “End-to-end Dense Video Captioning as Sequence Generation”. *COLING*. 2022 (cit. on pp. 99, 111, 112, 135).
- [Zhu, 2021] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. “Deformable DETR: Deformable Transformers for End-to-End Object Detection”. *ICLR*. 2021 (cit. on p. 80).
- [Zhu, 2015] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, et al. “Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books”. *ICCV*. 2015 (cit. on pp. 1, 14, 38).
- [Zhuang, 2018] Bohan Zhuang, Qi Wu, Chunhua Shen, Ian Reid, and Anton Van Den Hengel. “Parallel attention: A unified framework for visual object discovery through dialogs and queries”. *CVPR*. 2018 (cit. on p. 79).
- [Zhuang, 2020] Yueting Zhuang, Dejing Xu, Xin Yan, Wenzhuo Cheng, Zhou Zhao, Shiliang Pu, et al. “Multi-channel Attention Refinement for Video Question Answering”. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* (2020) (cit. on pp. 30, 37, 47, 57).
- [Zou, 2023] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, et al. “Generalized decoding for pixel, image, and language”. *CVPR*. 2023 (cit. on p. 141).

RÉSUMÉ

L'objectif de cette thèse est de construire et de former des modèles d'apprentissage automatique combinant la puissance du traitement du langage naturel avec la compréhension visuelle, permettant une compréhension complète et détaillée du contenu des vidéos. Premièrement, nous proposons deux nouvelles méthodes pour développer des modèles de réponses aux questions sur des vidéos sans avoir recours à une annotation manuelle coûteuse. Nous générons automatiquement des données de réponses aux questions sur des vidéos à partir de vidéos commentées à l'aide de modèles de génération de questions utilisant uniquement du texte. Nous montrons ensuite qu'un transformateur multi-modal entraîné de manière contrastée sur les données générées peut répondre aux questions visuelles sans entraînement supplémentaire. Afin de contourner la procédure de génération de données, nous présentons une approche alternative, nommée FrozenBiLM, qui exploite directement des modèles de langage masqué bidirectionnels. Deuxièmement, nous développons TubeDETR, un modèle de transformateur capable de localiser spatialement et temporellement une requête en langage naturel dans une vidéo non découpée. Contrairement aux approches spatio-temporelles antérieures, TubeDETR peut être efficacement entraîné de bout en bout sur des vidéos non rognées. Troisièmement, nous présentons un nouveau modèle et un nouvel ensemble de données pour la compréhension de multiples événements dans les vidéos non découpées. Nous introduisons le modèle Vid2Seq qui génère des descriptions denses en langage naturel et les limites temporelles correspondantes pour tous les événements dans une vidéo non découpée en prédisant une seule séquence de jetons. De plus, Vid2Seq peut être efficacement pré-entraîné sur des vidéos commentées à grande échelle en utilisant les transcriptions de paroles comme pseudo-supervision. Enfin, nous présentons VidChapters-7M, un ensemble de données à grande échelle de vidéos chapitrées par les utilisateurs. Sur la base de cet ensemble de données, nous évaluons des modèles de pointe sur trois tâches, dont la génération de chapitres vidéo. Nous montrons également que les modèles de génération de chapitres vidéo se transfèrent bien au sous-titrage vidéo dense.

MOTS CLÉS

apprentissage automatique, vision par ordinateur, intelligence artificielle, traitement du langage naturel, compréhension de vidéos, apprentissage profond

ABSTRACT

The goal of this thesis is to build and train machine learning models that combine the power of natural language processing with visual understanding, enabling a comprehensive and detailed comprehension of the content within videos. First, we propose two scalable approaches to develop video question answering models without the need for costly manual annotation. We automatically generate video question answering data from narrated videos using text-only question-generation models. We then show that a multi-modal transformer trained contrastively on the generated data can answer visual questions in a zero-shot manner. In order to bypass the data generation procedure, we present an alternative approach, dubbed FrozenBiLM, that directly leverages bidirectional masked language models. Second, we develop TubeDETR, a transformer model that can spatially and temporally localize a natural language query in an untrimmed video. Unlike prior spatio-temporal grounding approaches, TubeDETR can be effectively trained end-to-end on untrimmed videos. Third, we present a new model and a new dataset for multi-event understanding in untrimmed videos. We introduce the Vid2Seq model which generates dense natural language descriptions and corresponding temporal boundaries for all events in an untrimmed video by predicting a single sequence of tokens. Moreover, Vid2Seq can be effectively pretrained on narrated videos at scale using transcribed speech as pseudo-supervision. Finally, we introduce VidChapters-7M, a large-scale dataset of user-chaptered videos. Based on this dataset, we evaluate state-of-the-art models on three tasks including video chapter generation. We also show that video chapter generation models transfer well to dense video captioning in both zero-shot and finetuning settings.

KEYWORDS

machine learning, computer vision, artificial intelligence, natural language processing, video understanding, deep learning