Video question answering with limited supervision

Deniz Engin

COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

Université de Rennes

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES

ÉCOLE DOCTORALE N° 601
Mathématiques, Télécommunications, Informatique, Signal, Systèmes, Électronique
Spécialité : *Signal, Image, Vision*

Par
Deniz ENGIN

Video question answering with limited supervision

Thèse présentée et soutenue à Rennes, le 11 June 2024
Unité de recherche : Centre Inria de l'Université de Rennes

Rapporteurs avant soutenance :

Ivan LAPTEV — Visiting Professor, MBZUAI
Matthieu CORD — Professor, Sorbonne University

Composition du Jury :

Président :	Luce MORIN	Professor, INSA Rennes
Examinateurs :	Josef SIVIC	Distinguished Researcher, Czech Technical University
	Karteek ALAHARI	Directeur de recherche, Inria Grenoble
Rapporteurs :	Ivan LAPTEV	Visiting Professor, MBZUAI
	Matthieu CORD	Professor, Sorbonne University
Co-dir. de thèse :	Yannis AVRITHIS	Principal Investigator, IARAI

Invité(s) :

François SCHNITZLER — Senior Scientist, InterDigital
Ngoc Q. K. DUONG — Senior Principal Applied Scientist, Oracle

"To know that we know what we know, and to know that we do not know what we do not know, that is true knowledge."

– NICOLAUS COPERNICUS

ACKNOWLEDGEMENT

My PhD journey began as a CIFRE student at InterDigital and concluded exclusively at Inria, marking a transformative and memorable period in my life. This experience enabled me to advance my research and grow professionally and personally. Each phase of this journey brought unique challenges and opportunities, collectively shaping a truly incredible experience.

Firstly, I would like to express my sincere gratefulness to my advisors, Yannis Avrithis and Teddy Furon, for their dedicated support and mentorship. Yannis, your deep knowledge and insightful guidance have greatly influenced my research and broadened my perspectives. Your constant patience and strong belief in my abilities have both inspired and motivated me. The thoughtful questions you pose have challenged me to deepen my understanding of my research while teaching me the art of questioning. Teddy, your consistent support throughout these years has been invaluable and deeply appreciated. Your willingness to help in every situation has truly made a difference.

Special thanks to my initial supervisors, François and Ngoc, during my time at InterDigital. Their early guidance laid the crucial groundwork for my research, and their ongoing support was invaluable.

Many thanks to Josef and Karteek, members of my CSI committee. Their support and insightful feedback have been invaluable to my research.

I am grateful to Ivan and Matthieu for agreeing to review my thesis. Many thanks to all my jury members, including Josef, Karteek, and Luce, as well as Ivan and Matthieu, for their willingness to participate in my defense.

I am grateful to both former and current members of the Linkmedia team, especially Shashanka, Duc Hau, Guillaume, Carolina, Kassem, Victor, Hugo, Suresh, Cyrielle, Cheikh Brahim, Mateusz, Yann, and Hanwei, for their support during challenging times, making this academic journey both memorable and enjoyable.

I am deeply thankful to my family and friends, especially my sister, mother, and father, for their unwavering support and encouragement. Their guidance and motivation have been invaluable sources of comfort during difficult times.

Last but not least, I owe a huge thank you to Ali for his unwavering support, patience, and love. His sincere belief in me has continuously provided strength and inspiration throughout this journey. He has been there for me through every step, and I could not have done this without him.

CONTENTS

ABBREVIATIONS

AI Artificial Intelligence

ASR Automatic Speech Recognition

CNN Convolutional Neural Network

CV Computer Vision

GPU Graphics Processing Unit

GRU Gated Recurrent Unit

HMM Hidden Markov Model

HOG Histogram of Oriented Gradients

LSTM Long Short-Term Memory

MLM Masked Language Modeling

NLP Natural Language Processing

RNN Recurrent Neural Networks

SIFT Scale-Invariant Feature Transform

VideoQA Video Question Answering

LIST OF SYMBOLS

$\mathbf{w}$	weights
A	adapter layer
b	bias
d	embedding dimension
E^v	video frame features
f	language model
f^m	visual mapping network
f^t	text embedding layer
f^v	video encoder
g	classifier head
i	set $\{1, \ldots, i\}$ for $i \in \mathbb{N}$
j	set $\{1, \ldots, j\}$ for $j \in \mathbb{N}$
K	key
P^t	learnable text prompts
P^v	learnable visual prompts
Q	query
V	value
X^a	answer
X^q	question
X^s	subtitle
X^t	input text

X^v input video

Z^t text embeddings

Z^v video embeddings

List of Figures

List of Tables

ABSTRACT

Video content has significantly increased in volume and diversity in the digital era, and this expansion has highlighted the necessity for advanced video understanding technologies that transform vast volumes of unstructured data into practical insights by learning from data. Driven by this necessity, this thesis explores semantically understanding videos, leveraging multiple perceptual modes similar to human cognitive processes and efficient learning with limited supervision similar to human learning capabilities. Multimodal semantic video understanding synthesizes visual, audio, and textual data to analyze and interpret video content, facilitating comprehension of underlying semantics and context. This thesis specifically focuses on video question answering to understand videos as one of the main video understanding tasks. Our first contribution addresses long-range video question answering, which involves answering questions about long videos, such as TV show episodes. These questions require an understanding of extended video content. While recent approaches rely on human-generated external sources, we present processing raw data to generate video summaries. Our following contribution explores zero-shot and few-shot video question answering, aiming to enhance efficient learning from limited data. We leverage the knowledge of existing large-scale models by eliminating challenges in adapting pre-trained models to limited data, such as overfitting, catastrophic forgetting, and bridging the cross-modal gap between vision and language. We introduce a parameter-efficient method that combines multimodal prompt learning with a transformer-based mapping network while keeping the pre-trained vision and language models frozen. We demonstrate that these contributions significantly enhance the capabilities of multimodal video question-answering systems, where specifically human-annotated labeled data is limited or unavailable.

RÉSUMÉ

Cette thèse explore la compréhension sémantique multimodale des vidéos et son apprentissage efficace à partir d'un nombre limité de données. Elle se concentre sur la réponse aux questions à propos d'une vidéo spécifique. Ce résumé présente les motivations, les objectifs, les défis, le plan de la thèse et les contributions.

Motivation

Le média vidéo a considérablement augmenté en volume et en diversité à l'ère numérique, notamment sous l'impulsion des réseaux sociaux, des services de streaming et des plateformes de contenu généré par les utilisateurs. L'expansion de la vidéo a mis en lumière la nécessité de technologies de compréhension vidéo basées sur l'IA, qui transforment d'énormes volumes de données non structurées en informations pratiques apprises à partir des données. Ces technologies sont spécialisées dans plusieurs tâches, y compris le questionnement vidéo, le sous-titrage vidéo, et la récupération vidéo-texte. Les systèmes de *questionnement vidéo* permettent aux utilisateurs d'interagir avec le contenu vidéo en posant des questions et en recevant des réponses. Le *sous-titrage vidéo* automatise la génération de descriptions textuelles pour les scènes vidéo, fournissant des informations contextuelles qui enrichissent l'expérience visuelle. La *récupération vidéo-texte* permet de rechercher et de récupérer des segments vidéo spécifiques via des requêtes textuelles, comblant ainsi le fossé entre les informations visuelles et textuelles.

Ces tâches démontrent un impact significatif dans divers domaines. Par exemple, le sous-titrage vidéo améliore l'accessibilité pour les personnes malvoyantes en fournissant des descriptions audio des éléments visuels dans les vidéos. Les systèmes de questionnement vidéo permettent aux gens de s'engager davantage avec les vidéos à des fins éducatives et de divertissement, leur permettant de poser des questions et de recevoir des réponses. La récupération vidéo-texte transforme les médias et le journalisme en aidant à localiser des segments spécifiques au sein de vastes archives vidéo. Elle profite également à la recherche éducative en offrant un accès rapide au contenu vidéo pertinent pour des requêtes spécifiques. Elle facilite aussi le stockage et la récupération de données vidéo avec des systèmes d'indexation avancés

pour la gestion des données. De plus, les technologies de compréhension de la vidéo améliorent la sûreté publique grâce aux systèmes de surveillance, soutiennent la conduite autonome en fournissant des données de navigation essentielles, aident à surveiller et contrôler les dispositifs dans les maisons intelligentes et les environnements industriels, et renforcent l'engagement des clients grâce à des publicités interactives. Ces exemples d'applications illustrent comment les technologies de compréhension de la vidéo rendent le contenu numérique plus accessible et interactif dans divers domaines. Au fur et à mesure que cette technologie progresse, le champ d'application devrait s'élargir, offrir des innovations et transformer de multiples aspects de la vie quotidienne.

Bien que la recherche en compréhension de la vidéo ait montré des promesses, son application généralisée dans les systèmes IA reste limitée en raison de plusieurs contraintes, y compris les limitations matérielles en termes de demandes computationnelles pour le traitement des données vidéo en temps réel, et le besoin de modèles plus robustes et fiables. Notre objectif est d'améliorer la capacité et l'efficacité des méthodes de compréhension vidéo en recherche pour combler le fossé entre les limitations actuelles et leur potentiel d'impact sociétal. Nous visons à développer des technologies innovantes qui auront un impact positif sur la société en facilitant l'accès à l'information, en améliorant les interactions et en rendant le contenu vidéo accessible à tous.

Objectifs

Notre objectif principal est de comprendre la sémantique des vidéos avec une supervision limitée en tirant parti des modalités perceptuelles des processus cognitifs humains : la vue, le son et la parole. La compréhension sémantique multimodale des vidéos implique l'analyse et l'interprétation du contenu vidéo par la synthèse des données visuelles, audio et textuelles. Cette approche facilite la compréhension de la sémantique et du contexte sous-jacents, de manière similaire à la façon dont les humains interprètent les interactions entre personnes, objets et actions dans leur environnement visuel et auditif. De plus, nous visons à imiter les capacités d'apprentissage humaines en apprenant à partir de données limitées en nombre.

Pour atteindre ces objectifs, nous nous concentrons sur la *compréhension sémantique multimodale* et l'*apprentissage efficace à partir d'un nombre limité de données* en développant des architectures de modèles et des méthodologies de formation. Ces méthodes peuvent être évaluées avec des tâches de compréhension vidéo, y compris le questionnement vidéo sur la vidéo, la

récupération texte-vidéo et le sous-titrage vidéo. La récupération texte-vidéo implique de faire correspondre des requêtes textuelles avec du contenu vidéo mais elle peut ne pas capter pleinement l'intention de l'utilisateur ou ne pas fournir des indices pertinents sur le contenu vidéo. Bien que le sous-titrage vidéo transforme la vidéo en descriptions textuelles facilement compréhensibles par les humains, il produit souvent des légendes qui limitent une interaction plus poussée ou une compréhension plus profonde. Malgré ces limitations, il est précieux pour convertir les vidéos en contenu interprétable sous forme de texte. D'autre part, le questionnement vidéo offre une approche plus dynamique et interactive, où les utilisateurs peuvent poser des questions sur n'importe quel aspect du contenu vidéo, ce qui en fait un moyen non restreint de comprendre sémantiquement les vidéos. Par conséquent, notre recherche se spécialise principalement dans le questionnement vidéo.

Plus spécifiquement, le questionnement vidéo implique de créer une réponse en comprenant une question, le contenu visuel d'une vidéo, ainsi que tout langage parlé accompagnant, qui peut être présenté sous forme de texte, obtenu via l'ASR, ou traité directement sous sa forme parlée. Les méthodes de questionnement vidéo sont conçues pour traiter des entrées multimodales afin d'atteindre une compréhension sémantique des vidéos. Elles visent soit à générer une réponse à partir d'un vocabulaire spécifié, soit à en sélectionner une parmi un ensemble d'options à choix multiples.

Nous étudions spécifiquement le questionnement vidéo sur de longues vidéos en nous concentrant sur la compréhension sémantique multimodale, en utilisant uniquement les données brutes sans s'appuyer sur des sources générées par l'humain supplémentaires. Cette tâche exige une compréhension globale du contexte plus large d'une longue vidéo, allant au-delà de l'analyse de courts extraits limités à quelques secondes ou minutes.

De plus, nous enquêtons sur des stratégies d'apprentissage efficaces avec des données limitées en nombre. Nous nous concentrons spécifiquement sur le questionnement vidéo en *zero-shot* et en *few-shot*, où la disponibilité des données pour cette tâche spécifique est limitée. Notre objectif est de tirer parti des capacités des grands modèles pré-entraînés de vision et de langue qui ont été formés sur de grands ensembles de données et ont déjà acquis des connaissances. De plus, nous visons à étudier comment les paires vidéo-légende, disponibles en ligne, peuvent être efficacement utilisées pour former des méthodes vidéo-langage et améliorer notre approche du questionnement vidéo.

Défis

Réponse à des questions sur de longues vidéos. Lorsque les gens regardent des clips vidéo qui font partie d'une vidéo plus longue, ils peuvent avoir des questions dont les réponses se trouvent dans des segments antérieurs de la vidéo. Par exemple, en regardant une émission de télévision, un spectateur pourrait s'interroger sur des événements ou des détails survenus plus tôt dans l'émission. Cela indique que la réponse à des questions sur de longues vidéos nécessite une compréhension de haut niveau de l'ensemble de la vidéo. Les systèmes actuels de réponse à des questions vidéo s'appuient souvent sur les modalités vidéo et audio. L'audio peut être traduit sous forme de texte comme des sous-titres. Ils utilisent également des sources générées par l'humain telles que des résumés d'intrigue, des scripts, des descriptions de vidéos ou des bases de connaissances pour faciliter la réponse à des questions sur de longues vidéos. Cependant, notre objectif est d'éliminer la nécessité d'annotations humaines. Le premier défi ici est de parvenir à une compréhension de haut niveau des vidéos de longue durée, telles que les films et les émissions de télévision, à partir de données brutes. Un autre défi est de déterminer quels segments spécifiques contiennent les informations nécessaires pour répondre à une question, rendant essentielle l'analyse de l'ensemble de la vidéo. Cela nécessite un traitement de contexte long et une localisation précise des réponses pertinentes.

Pour relever ces défis, nous proposons d'abord l'utilisation de résumé de dialogue pour générer un résumé vidéo de haut niveau. Cette approche tire parti des dialogues disponibles comme données brutes dans les émissions de télévision pour produire un récit condensé du contenu sous forme textuelle, éliminant ainsi le besoin de données supplémentaires annotées par l'humain. De plus, nous introduisons une méthode pour localiser les réponses pertinentes à partir de longues entrées textuelles, permettant une compréhension efficace des vidéos de longue durée. Ces méthodologies sont détaillées dans le chapitre sur la réponse à des questions vidéo basée sur les connaissances.

Combler le fossé des modalités: Intégration de la vision et du langage. L'un des défis clés de l'apprentissage multimodal réside dans le fossé entre les modèles de vision et de langage. Ce fossé provient des propriétés fondamentalement différentes des données visuelles et textuelles et de leurs méthodes de traitement uniques. Les modèles de vision interprètent les images et les vidéos comme spatiales et continues, tandis que les modèles de langage traitent le texte de manière séquentielle et discrète. Combler ce fossé nécessite de combiner ces deux

types d'informations pour permettre une interprétation et des capacités de réponse cohérentes et efficaces dans des systèmes tels que les modèles de réponse aux questions vidéo.

Pour combler le fossé entre modalités, nous explorons deux approches. La première approche consiste à convertir les vidéos en descriptions textuelles, transformant ainsi la modalité vidéo en une forme textuelle qui peut être traitée par des modèles de langage, comme détaillé dans le chapitre sur la réponse aux questions vidéo basée sur les connaissances. Dans la seconde approche, nous employons des modèles de vision et de langage. Initialement, un encodeur visuel convertit le contenu vidéo en une représentation structurée. Cette représentation est ensuite traitée par notre réseau qui adapte efficacement les données visuelles pour être compatibles avec les modèles de langage. La sortie de ce réseau est par la suite introduite dans le modèle de langage, permettant l'interaction entre les modalités visuelles et textuelles, comme décrit dans le chapitre sur la réponse aux questions vidéo en *few-shot* et *zero-shot*.

Adaptation des modèles pré-entraînés de vision et de langage. Les modèles récents de vision et de langage ont montré des progrès remarquables, portés par les *modèles pré-entraînés à grande échelle* basés sur les transformateurs. Ces modèles ont été intégrés dans des méthodes de compréhension vidéo, y compris la réponse à des questions par fusion à partir de *de grands ensembles de données multimodales*. Lorsque nous intégrons des modèles de vision et de langage pour combler le fossé entre modalités, comme mentionné précédemment, nous exploitons ces modèles pré-entraînés même dans des contextes où les données pour la réponse à des questions sur la vidéo sont limitées en nombre. Cependant, adapter ces modèles aux tâches vidéo-langage avec des données limitées présente des défis significatifs. Au-delà du fossé entre modalités, le réglage fin de l'ensemble du modèle sur des données limitées peut entraîner un surapprentissage et l'oubli des connaissances précédemment acquises. Pour atténuer le surapprentissage, des méthodes d'adaptation efficaces en termes de paramètres, par exemple, *l'apprentissage par prompt* et *les couches d'adaptation*, ont été appliquées sur des modèles pré-entraînés figés.

Pour répondre davantage à ces défis dans la réponse à des questions sur la vidéo, nous explorons des méthodes d'adaptation efficaces en termes de paramètres et introduisons pour la première fois l'apprentissage multimodal par prompt. Cette approche préserve les capacités de généralisation des modèles à grande échelle tout en minimisant le nombre de paramètres entraînables, réduisant ainsi les besoins de stockage pour des configurations *few-shot* à partir de divers ensembles de données.

Plan de la thèse et contributions

Cette thèse comporte cinq chapitres, incluant cette introduction, qui présente les motivations, les objectifs, les défis et les contributions.

Dans le **chapter 2**, nous examinons d'abord les modèles de vision et de langage indépendamment, puis discutons des modèles vision-langage, établissant ainsi le contexte pour la réponse aux questions sut la vidéo.

Le **chapter 3** aborde la réponse aux questions sur la vidéo basée sur la connaissance, qui implique de répondre à des questions pouvant être liées à des vidéos étendues, telles que des épisodes entiers de séries télévisées, en plus de courts clips. Cela indique que répondre à des questions sur de longues vidéos nécessite une compréhension complète du contenu vidéo. Les approches récentes reposent sur des sources externes générées par l'humain telles que des résumés d'intrigue provenant d'Internet ou des connaissances fournies par des ensembles de données, alors que nous traitons les données brutes pour générer des résumés d'épisodes. Nous considérons le dialogue comme une source bruitée, que nous convertissons en une description textuelle via la résumé de dialogue. Nous créons un résumé de dialogue de l'épisode entier en combinant les résumés de dialogue de scène. Ainsi, nous remplaçons les connaissances annotées par l'humain par des résumés d'épisodes générés automatiquement. Nous présentons également une méthode de réponse aux questions sur la vidéo basée sur la connaissance qui encode indépendamment différentes entrées textuelles, y compris la description vidéo par des transformateurs, et une méthode de fusion simple qui combine toutes les modalités. De plus, nous proposons une attention temporelle douce pour la localisation sur de longues entrées afin de traiter efficacement les entrées textuelles longues. Notre modèle surpasse les méthodes précédentes sans utiliser d'annotations spécifiques aux questions et générées par l'humain ou de résumés d'intrigue fabriqués par l'humain. Le contenu de ce chapitre est basé sur notre article ICCV 2021, "*On the hidden treasure of dialog in video question answering*" [Engin, 2021b]. Notre code est disponible sur le site [Engin, 2021a].

Le **chapter 4** traite de la réponse aux questions sur la vidéo en *zero-shot* et en *few-shot* en exploitant les grands modèles de vision et de langage pré-entraînés. Adapter ces modèles pré-entraînés sur des données limitées en nombre présente des défis tels que le surapprentissage, l'oubli catastrophique, et le fossé intermodal entre la vision et le langage. Pour relever ces dé-

fis, nous introduisons une méthode efficace en termes de paramètres combinant l'apprentissage par prompt multimodal et un réseau basé sur des transformateurs tout en conservant les modèles pré-entraînés figés. Cette approche nous permet de surmonter les limitations de la rareté des données et d'obtenir de meilleures performances que les méthodes précédentes sur plusieurs ensembles de données pour la réponse aux questions vidéo. Le contenu de ce chapitre est basé sur notre article pour un séminaire ICCV 2023, "*Zero-Shot and Few-Shot Video Question Answering with Multi-Modal Prompts*" [Engin, 2023b]. Notre code et nos modèles sont disponibles à l'adresse [Engin, 2023a].

Dans le **chapter 5**, enfin, nous résumons les contributions de cette thèse et discutons de certaines pistes potentielles pour la recherche future.

CHAPTER 1

INTRODUCTION

Contents

1.1 Motivation

Video content has significantly increased in volume and diversity in the digital era, especially driven by social media, streaming services, and user-generated content platforms. The expansion of video content has highlighted the necessity for video understanding technologies based on Artificial Intelligence (AI), which transform vast volumes of unstructured data into practical insights by learning from data. These technologies are specialized in multiple tasks, including video question answering, video captioning, and video-text retrieval. *Video question answering* systems enable users to interact with video content by posing questions and receiving answers. *Video captioning* automates the generation of textual descriptions for video scenes, providing contextual information that enriches the viewing experience. *Video-text retrieval* allows for searching and retrieving specific video segments through text queries, bridging the gap between visual and textual information.

These tasks demonstrate significant impact across diverse fields. For instance, video captioning enhances accessibility for visually impaired individuals by providing audio descriptions of visual elements in videos. Video question answering systems allow people to engage more with videos for both educational and entertainment purposes, enabling them to ask questions

and receive answers. Video-text retrieval transforms media and journalism by helping to locate specific segments within extensive video archives, and it similarly benefits educational research by providing rapid access to precise video content relevant to specific queries. It also facilitates storing and retrieving video data with advanced indexing systems for data management. Additionally, video understanding technologies enhance public safety through surveillance systems, support autonomous driving by providing essential navigation data, assist in monitoring and controlling devices in smart homes and industrial settings, and enhance customer engagement through interactive advertisements. These example applications highlight how video understanding technologies make digital content more accessible and interactive across diverse domains. As this technology advances, the scope of video applications is expected to broaden, offering innovations and transforming multiple aspects of everyday life.

Although research in video understanding has shown promise, its widespread application in AI systems remains limited due to several constraints, including hardware limitations in terms of the computational demands of processing video data in real time, and the need for more robust and reliable models. Our goal is to enhance the capability and efficiency of video understanding methods in research to bridge the gap between current limitations and their potential for societal impact. We aim to develop innovative technologies that will positively impact society by facilitating access to information, improving interactions, and making video content accessible to everyone.

1.2 Learning from data

In this section, we will establish a foundational background to enhance understanding of the subsequent parts of the thesis by briefly discussing the concept of deep learning.

Deep learning. In the early days of AI, experts manually processed raw data and crafted features for specific tasks. This approach evolved with machine learning models, which could learn representations directly from human-annotated data. Deep learning has significantly advanced this field by automating the feature extraction process, learning representations directly from the data itself. This involves optimizing model parameters based on data, enabling the model to independently identify useful patterns or features.

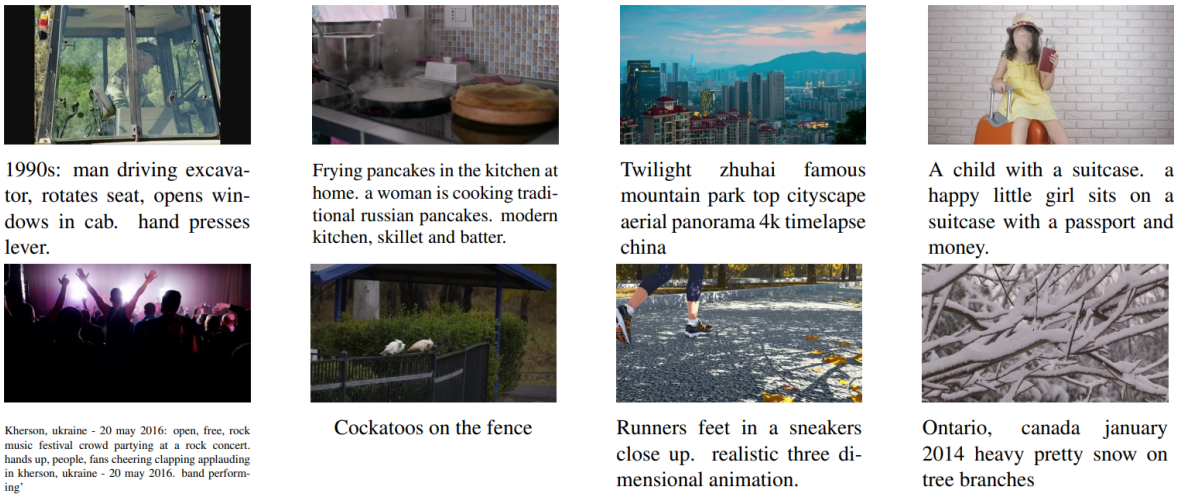


Figure 1.1 – Examples of video-text pairs from WebVid-2M dataset. Image sourced from [Bain, 2021].

Learning paradigms. Initially, the field was dominated by supervised learning, where models learn from human-labeled data. However, since human annotation is costly, self-supervised learning methods, which do not rely on labeled data, have gained attention for improving learned features. The development of these methods has facilitated the widespread adoption of transfer learning techniques. In this approach, a model is first pre-trained on a large dataset of unlabeled data and then fine-tuned on smaller, labeled, task-specific datasets. This two-phase process significantly enhances model performance on downstream tasks, which are specific applications that utilize the pre-trained model to achieve more specialized objectives. For instance, the WebVid-2M dataset [Bain, 2021], illustrated in Figure 1.1, is scraped from the internet and comprises video-caption pairs with manually generated, well-structured sentences that accurately describe the visual content. This dataset is ideal for pre-training models for video understanding tasks. The learned features can then be utilized for specific downstream tasks, either by fine-tuning or adapting in ways suitable for specialized tasks.

Scaling. Early deep learning models, such as AlexNet [Krizhevsky, 2012], which marked a significant breakthrough in the field, contained approximately 60 million parameters. Recently, however, there has been a significant increase in both the complexity and scale of models. Recent models may have billions of parameters and require training on much larger datasets than previously utilized. This expansion in model size and training datasets has not only increased model capabilities but also opened new opportunities for zero-shot transfers, where a

model trained for one objective is applied to solve different tasks. Additionally, even though task-specific and multimodal datasets have grown, they remain limited compared to the datasets used to train large-scale vision or language models. This situation highlights the need to explore parameter-efficient strategies to further enhance pre-trained models for solving diverse tasks with limited data.

1.3 Goals

Our primary goal is to semantically understand videos with limited supervision by leveraging multiple perceptual modes that reflect human cognitive processes, including sight, sound, and speech. Multimodal semantic video understanding involves analyzing and interpreting video content through synthesizing visual, audio, and textual data inputs. This approach facilitates the comprehension of underlying semantics and context, similar to how humans interpret interactions among people, objects, and actions within their visual and auditory environments. Additionally, we aim to mimic human learning capabilities by learning from limited data.

To achieve these goals, we focus on *multimodal semantic understanding* and *efficient learning from limited data* by developing model architectures and training methodologies. These methods can be assessed with video understanding tasks, including video question answering [Xu, 2017; Lei, 2018; Yu, 2019], text-video retrieval [Chen, 2011; Xu, 2016; Rohrbach, 2017; Krishna, 2017a], and video captioning [Xu, 2016; Krishna, 2017a; Zhou, 2018]. Text-video retrieval involves matching textual queries to video content but may not fully capture user intent or provide deep insights into video content. Although video captioning transforms video into text descriptions that are easily understandable by humans, it often produces captions that limit further interaction or deeper understanding. Despite these limitations, it is valuable for converting videos into interpretable content in text form. On the other hand, video question answering offers a more dynamic and interactive approach, where users can ask questions about any aspect of the video content, making it an unrestricted way to understand videos semantically. Therefore, our research primarily specializes in video question answering.

More specifically, video question answering involves creating a response by comprehending a question, the visual content of a video, along with any accompanying spoken language, which may be presented in text form, obtained via Automatic Speech Recognition (ASR), or processed

directly from its speech form. Video question answering methods are designed to process multimodal inputs, achieving a semantic understanding of videos. They aim to either generate an answer from a specified vocabulary or select one from a set of multiple-choice options.

We specifically investigate long video question answering by focusing on multimodal semantic video understanding, utilizing only the raw data without relying on additional human-generated sources. This task demands a comprehensive understanding of the broader context of a long video, moving beyond the analysis of short clips that are limited to just a few seconds or minutes.

Furthermore, we investigate efficient learning strategies with limited data, specifically focusing on zero-shot and few-shot question answering, where data availability for this specific task is constrained. Our goal is to leverage the capabilities of large-scale pre-trained vision and language models that have been trained on large-scale datasets and have acquired knowledge. Additionally, we aim to investigate how video-caption pairs, available online, can be effectively used to train video-language methods and enhance our approach to video question answering.

1.4 Challenges

Developing methods for video question answering with limited supervision presents several challenges in designing and training these methods.

Long-range video question answering. While people are watching video clips that are part of a longer video, they may have questions whose answers lie in earlier segments of the video. For instance, while watching a TV show, a viewer might wonder about events or details from earlier in the episode. This situation is illustrated by an example in Figure 1.2, where an episode contains multiple scenes. The scene on the right discusses a robot competition, and the question asks for the robot's name. This question cannot be answered solely based on this scene because the robot's name is not mentioned. There is another scene from the same episode on the left. In this scene, one of the characters, Sheldon, says their robot's name is Monte; therefore, the question can be answered by using this scene. This indicates that long-range video question answering requires a high-level understanding of the whole video. Current video question answering systems often rely on both video and audio modalities; audio may be provided in text form as subtitles. They also use additional human-generated sources like plot synopses, scripts,

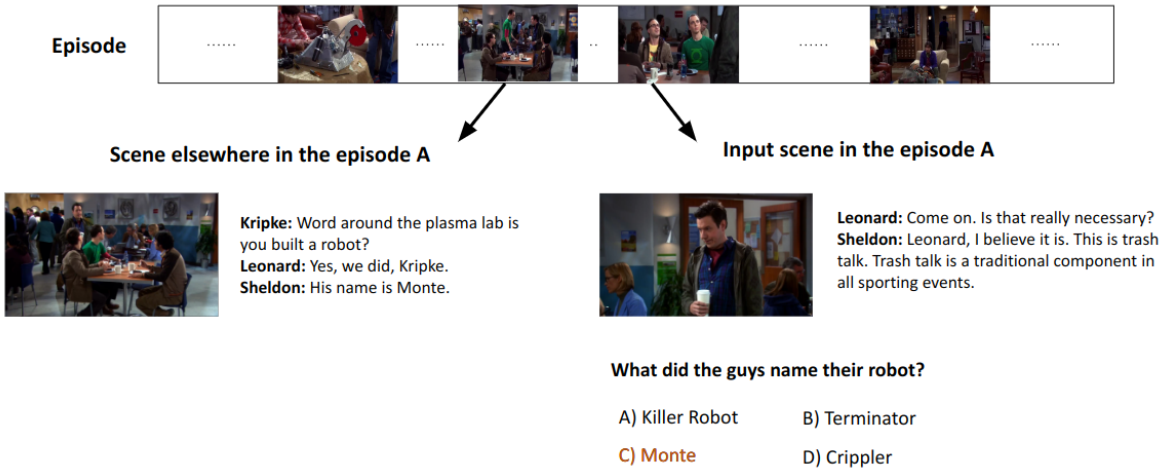


Figure 1.2 – An example of a video question answer pair from the KnowIT-VQA dataset [Garcia, 2020b] based on a TV show illustrates a long-range video question answering. Each episode in a TV show comprises multiple scenes, each paired with a corresponding question. In this example, the question associated with the right scene requires information from the left scene to be answered. This indicates that answering this question requires a high-level understanding of videos and processing long contexts. For visualization purposes, videos are represented using a single frame.

video descriptions, or knowledge bases [Garcia, 2020b; Garcia, 2020a] to facilitate long-range video question answering. However, our goal is to eliminate the requirement of human annotation. The first challenge here is to achieve a high-level understanding of long-range videos, such as movies and TV shows, from raw data. Another challenge is determining which specific segments contain the information necessary to answer a question, making it essential to analyze the entire video. This requires long context processing and precise localization of the relevant answers.

To tackle these challenges, we first propose the use of dialog summarization to generate a high-level video summary. This approach leverages the dialog available as raw data in TV shows to produce a condensed narrative of the episode in text form, eliminating the need for additional human-annotated data. Furthermore, we introduce a method for localizing relevant answers from long text inputs, enabling an effective understanding of long-range videos. These methodologies are detailed in Chapter 3.

Bridging the Modality Gap: Integrating Vision and Language. One of the key challenges in multimodal learning lies in the modality gap between vision and language models. This gap arises from the fundamentally different properties of visual and textual data, as well as their unique processing methods. Vision models interpret images and videos as spatial and continuous, whereas language models process text as sequential and discrete. Bridging this gap necessitates combining these two information types to enable coherent and effective interpretation and response capabilities in systems like video question answering models.

To bridge the modality gap, we explore two approaches. The first approach involves converting videos into textual descriptions, thereby transforming the video modality into a text form that can be processed by language models, as detailed in Chapter 3. In the second approach, we employ vision and language models. Initially, a vision encoder converts video content into a structured representation. This representation is then processed by our proposed mapping network, which effectively adapts the visual data to be compatible with language models. The output from the mapping network is subsequently fed into the language model, enabling interaction between the visual and textual modalities, as described in Chapter 4.

Adapting pre-trained vision and language models. The recent vision and language models have shown remarkable progress, driven by transformer-based *large-scale pre-trained models* [Dosovitskiy, 2021; Liu, 2022b; Devlin, 2019; Liu, 2019a; He, 2021; Radford, 2019; Radford, 2021]. These models have been incorporated into video understanding methods, including video question answering through multimodal fusion on *large-scale multimodal datasets* [Miech, 2019b; Bain, 2021; Zellers, 2021]. When we integrate vision and language models to bridge the modality gap, as mentioned previously, we leverage these pre-trained models even in settings with limited data for video question answering. However, adapting these models to video-language tasks with limited data presents significant challenges. Beyond the modality gap, fine-tuning the entire model on limited data can result in overfitting and forgetting previously acquired knowledge. To mitigate overfitting, parameter-efficient adaptation methods, *e.g.*, *prompt learning* [Li, 2021; Liu, 2021a; Liu, 2022a] and *adapter layers* [Houlsby, 2019] have been applied on frozen pre-trained models.

To further address these challenges in video question answering, we explore parameter-efficient adaptation methods and introduce multimodal prompt learning for the first time. This approach preserves the generalization capabilities of large-scale models while minimizing the

number of trainable parameters, thus reducing the storage requirements for few-shot settings across various datasets.

1.5 Thesis outline and contributions

This thesis has five chapters, including this introduction, which provides motivations, goals, challenges, and contributions.

In **Chapter 2**, we first review vision and language models independently, then discuss vision-language models, establishing the background for video question answering.

Chapter 3 tackles knowledge-based video question answering, which involves answering questions that can be related to extended videos, such as entire TV show episodes, in addition to short clips. This indicates that answering questions over long videos requires a comprehensive understanding of the video content. Recent approaches rely on human-generated external sources like internet-sourced plot summaries or dataset-provided knowledge; however, we process raw data to generate episode summaries. We consider dialog a noisy source, which we convert to a text description via dialog summarization. We obtain the entire episode dialog summary by combining scene dialog summaries; consequently, we replace human-annotated knowledge with automatically generated episode summaries. We also present a knowledge-based video question answering method that independently encodes different text-based inputs, including video description by transformers, and a simple fusion method that combines all modalities. Additionally, we propose soft temporal attention for localization over long inputs to process long text input efficiently. Our model outperforms the previous methods without using question-specific human annotation or humanmade plot summaries. The content of this chapter is based on our ICCV 2021 paper, "*On the hidden treasure of dialog in video question answering*" [Engin, 2021b]. Our code is available at [Engin, 2021a].

Chapter 4 addresses zero-shot and few-shot video question answering by leveraging large-scale pre-trained vision and language models. Adapting pre-trained models on limited data presents challenges such as overfitting, catastrophic forgetting, and the cross-modal gap between vision and language. To address these challenges, we introduce a parameter-efficient method, combining multimodal prompt learning and a transformer-based mapping network while keeping the pre-trained models frozen. This approach enables us to overcome the lim-

itations of the scarcity of data and achieve better performance than previous methods on several datasets for video question answering. The content of this chapter is based on our ICCV 2023 workshop paper, "*Zero-Shot and Few-Shot Video Question Answering with Multi-Modal Prompts*" [Engin, 2023b]. Our code and models are available at [Engin, 2023a].

In **Chapter 5**, finally, we summarize the contributions of this thesis and discuss some potential areas for future research.

Patent Applications. In addition to the theoretical contributions, this thesis has also led to the following patent applications.

1. Patent Application No. EP21305290, "Device and method for question answering", filed on March 10, 2021, "Deniz Engin, Quang-Khanh-Ngoc Duong, François Schnitzler, Yannis Avrithis", which describes a novel method for multimodal question answering. This application directly relates to the methods discussed in Chapter 3.

2. Patent Application No. PCT/EP2022/071087, "System and method for question answering", filed on July 27, 2022, "Quang-Khanh-Ngoc Duong, Deniz Engin, François Schnitzler, Yannis Avrithis", which describes a method to enhance user experience in video question answering. This application is related to the methods discussed in Chapter 3.

CHAPTER 2

BACKGROUND

Contents

This chapter aims to establish a foundational understanding of key concepts in video question answering, which will be explored throughout this thesis. Video question answering involves both language and visual understanding; therefore, we present an overview of language models in Section 2.1 and vision models in Section 2.2. Then, we discuss vision-language models in Section 2.3.

2.1 Language models

Natural Language Processing (NLP) aims to facilitate the comprehension, interpretation, and generation of human language. NLP has diverse applications in both textual and multimodal domains, *e.g.*, *machine translation, sentiment analysis, text summarization, text-based question answering, visual question answering, and image/video captioning.*

This section provides an overview of the evolution of language modeling, highlighting its advancement from the early static models to the neural network approaches in Subsection 2.1.1. Then, Subsection 2.1.2 presents architectures of transformer-based language models, specifically emphasizing employed models in this thesis. Finally, we discuss learning paradigms for transformer-based language models in Subsection 2.1.3.

2.1.1 Evolution of language models

Early language modeling in computational linguistics relied on statistical language models, including n-gram models and Hidden Markov Model (HMM). An n-gram is defined as a consecutive series of n elements, typically words, extracted from a text or speech sample. N-gram models predict the probability of a word based on its preceding words. Their effectiveness is limited by data sparsity, and they have an inability to capture long-range dependencies [Jurafsky, 2023]. Additionally, HMM has been introduced to model language as a sequence of observable outputs, which are generated by a chain of hidden states. This modeling approach effectively represents the stochastic nature of spoken language [Rabiner, 1989] and expands their applications in language processing [Althoff, 2016]. Despite their limitations in context sensitivity and dependency resolution, these early statistical models laid the foundation for developing recent context-aware language models.

Recurrent Neural Networks (RNN) [Rumelhart, 1987] has been employed to address the limitation of the statistical models in capturing longer-range dependencies and contextual information. RNNs process sequential data by maintaining a hidden state to capture information from previous inputs. Moreover, RNNs use current input with hidden states to generate outputs and update the hidden state for future inputs. The ability of RNNs to handle variable-length context eliminates the limited context issue of n-gram models and the fixed context length constraint in feedforward language models. Nevertheless, RNNs face challenges due to the complex function of their hidden layers, which are involved in current decision-making while simultaneously retaining and updating information for future decisions. Additionally, RNNs encounter a training issue known as the vanishing gradient problem, which occurs during backpropagation through time as the gradients tend to diminish due to repeated multiplications in long sequences.

To overcome these challenges, Long Short-Term Memory (LSTM) [Hochreiter, 1997] has been presented by featuring several gates: the forget gate decides which information to keep or discard from the cell state, the input gate controls the flow of new information into the cell

state, and the output gate determines which parts of the cell state are transferred to the output. Subsequently, Gated Recurrent Unit (GRU) [Cho, 2014] simplifies the LSTM by incorporating two essential gates: the update and reset gates. The update gate allows the model to determine how much past information from previous time steps should be kept and carried forward, and the reset gate controls the process of maintaining relevant context while discarding less necessary data.

Word representation serves as a bridge between raw text and neural networks. It has progressed from high-dimensional one-hot encoding to *word embeddings* that condense words into dense, low-dimensional vectors that capture semantic and syntactic relationships in textual data. The introduction of *word embeddings* represents a significant advancement in NLP, playing an important role in enhancing the performance of RNNs. *Word embeddings*, including word2vec [Mikolov, 2013], GloVe [Pennington, 2014], FastText [Bojanowski, 2017], are trained on large-scale text data in a self-supervised manner, allowing them to learn rich, contextual word relationships.

Furthermore, the attention mechanism [Bahdanau, 2015] revolutionizes sequential data processing by enabling more flexible modeling of dependencies, irrespective of their distance in the input or output sequences. This approach tackles the challenge of long-range dependencies, a significant limitation in traditional sequential models, *e.g.*, *RNNs, LSTMs, and GRUs*. The core idea behind attention is to enable the model to give selective focus to different elements of the input sequence while predicting each element of the output sequence.

Building on the attention mechanism, the transformer model [Vaswani, 2017] has been introduced by discarding recurrent processes and relying on the purely attention-based framework. This approach enables comprehensive mapping of global dependencies between input and output, and enhances the parallel processing capabilities of the model. In the context of word representations, the transformers integrate the learning of token embeddings directly into their architecture, utilizing a text embedding layer that evolves and refines these embeddings as part of their comprehensive pre-training process, in contrast to standalone methods *e.g.*, *word2vec*. The subsequent section provides a detailed explanation of transformers as the primary language models employed in this thesis.

2.1.2 Transformer-based language models

Introduction to the transformer architecture. The transformer [Vaswani, 2017] is a network architecture that allows for efficient pairwise interaction between input elements. It was originally developed for machine translation, and its main component is an attention function, which acts as a form of associative memory. The transformer consists of an encoder and a decoder stack, as illustrated in Figure 2.1. The encoder, shown on the left, is composed of a stack of N identical layers, each containing two sub-layers: a multi-head self-attention and a feed-forward neural network. The decoder, represented on the right, also has N identical layers, including additional masked multi-head attention that ensures that the predictions for a particular sequence position can only depend on outputs at preceding positions. This is followed by multi-head attention, which attends to the output of the encoder stack. The final output of the decoder passes through a linear layer and then a softmax layer to produce output probabilities. The decoder employs an *auto-regressive* approach; it predicts one element at a time based on a previously known element. All sub-layers, both in the encoder and decoder, are followed by a residual connection and layer normalization, denoted as "Add & Norm".

A thorough comprehension of the transformer architecture requires understanding the multi-head self-attention mechanism, which enables the processing of different elements of the input sequence simultaneously to capture complex dependencies across the sequence. The multi-head self-attention based on the attention function is defined as follows:

$$\text{Attention}(Q, K, V) := \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{2.1}$$

where Q, K, and V represent queries, keys, and values, respectively. d_k is the dimension of the queries and keys. Large magnitude dot products, a result of high dimensionality d_k, lead to extremely small gradients in the softmax function. To overcome this, the dot products are scaled by $\frac{1}{\sqrt{d_k}}$. The multi-head attention further refines the attention mechanism as follows:

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}\left(\text{head}_1, \ldots, \text{head}_\text{h}\right) W^O \\ \text{where head } &= \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right), \end{aligned} \tag{2.2}$$

where each head independently applies unique linear transformations to Q, K, and V. This approach allows each head to focus on different features of the input sequence, enabling the model to capture a broader range of information and relationships from multiple representation

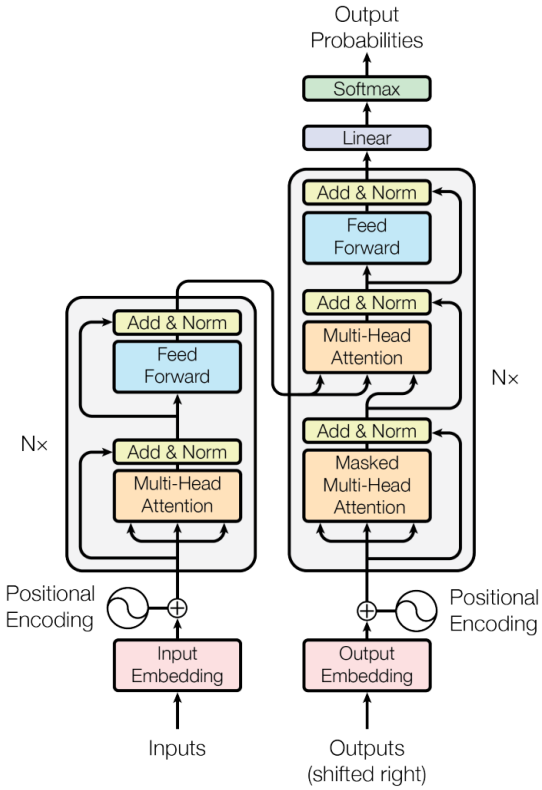


Figure 2.1 – Transformer model architecture. Image sourced from [Vaswani, 2017].

subspaces simultaneously. Then, the outputs of heads are concatenated and transformed through a linear operation with W^O.

After defining the transformer architecture, our focus now shifts to the process of input preparation for models. This initial step, known as tokenization, converts raw text into manageable units, *e.g.*, *words, subwords, or characters*, enabling the model to comprehend and analyze the text effectively. Specifically, sentences are tokenized by utilizing byte-pair encoding (BPE) [Britz, 2017] in the transformer [Vaswani, 2017]. After tokenization is performed on the entire dataset, a vocabulary is constructed once, creating a mapping from tokens to their respective unique integer identifiers. These integer representations are transformed into vectors in a high-dimensional space, typically using a text embedding layer. This embedding layer, often a fully connected layer, maps each token to a d-dimensional vector where d is the size of the embeddings. The embeddings are learned during training, allowing the model to capture semantic relationships between tokens.

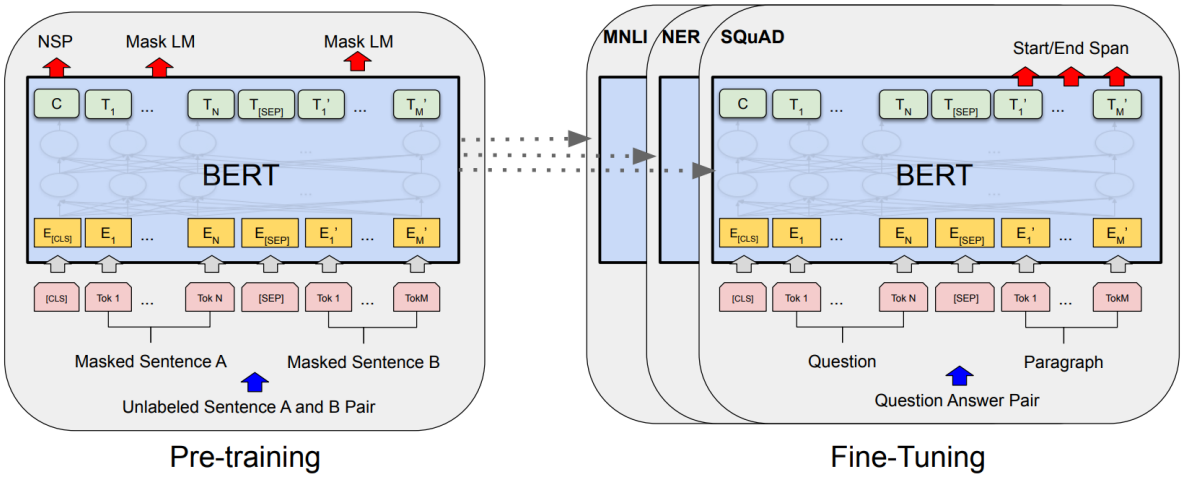


Figure 2.2 – BERT model architecture. The BERT employs the same architecture pre-training and fine-tuning, except for the output layers. BERT incorporates specific tokens for input processing: $[CLS]$ token is placed at the beginning of every input sequence, serves as the representation of the input, and $[SEP]$ token is utilized to indicate the separation of segments in the input. Image sourced from [Devlin, 2019].

Additionally, transformers use positional encodings to incorporate the order of the sequence into the model, as the self-attention mechanism in transformers does not inherently capture sequence order. The transformer model [Vaswani, 2017] uses sinusoidal functions for positional encoding, and the input embeddings are combined with positional encodings. Alternatively, positional encodings can also be learnable, allowing the model to adapt these encodings during training. In summary, to obtain input embeddings, each input sequence is first tokenized, then processed through a text embedding layer, and finally, positional encoding is applied to incorporate the sequence information.

Unidirectional and bidirectional paradigms. Transformer-based language models have revolutionized NLP with their advanced performance in understanding and generating diverse language tasks. These models can be categorized into *unidirectional (causal)*, also known as *autoregressive*, which predicts subsequent words based on preceding context for text generation, and *bidirectional (non-causal)*, which utilizes past and future context for comprehension tasks. Furthermore, *encoder-only* models, *e.g.*, *BERT [Devlin, 2019], RoBERTa [Liu, 2019b], DeBERTa [He, 2021]*, use a bidirectional process, while *decoder-only* models, *e.g.*, *GPT-2 [Radford, 2019], LLaMa [Touvron, 2023]* operate autoregressively. *Encoder-decoder* models, *e.g.*,

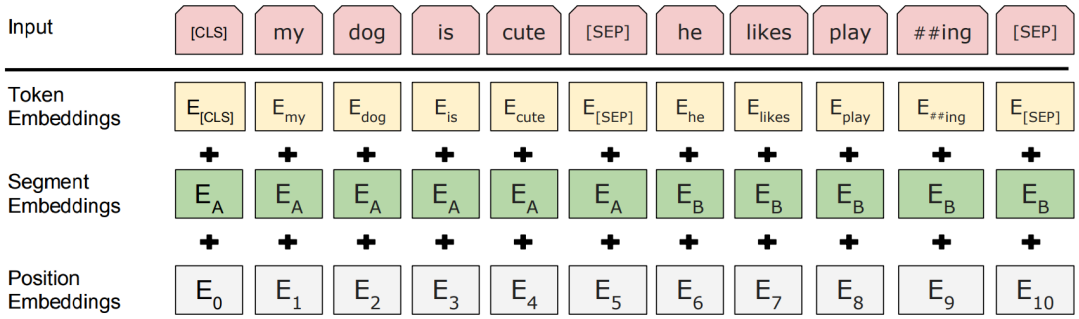


Figure 2.3 – BERT input representation. The input embeddings consist of a combination of token embeddings, segment embeddings, and position embeddings. Image sourced from [Devlin, 2019].

BART [Lewis, 2020], T5 [Raffel, 2020], combine both methodologies, with the encoder analyzing input text bidirectionally to understand its context and the decoder generating text causally.

Overview of employed language models. This section briefly introduces transformer-based language models utilized in this thesis. BERT [Devlin, 2019] is a bidirectional encoder, mapping a sequence of tokens to a sequence of d-dimensional vectors. As demonstrated in Figure 2.2, it includes pre-training and fine-tuning phases. It is pre-trained on a large text corpus with Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) objectives and fine-tuned on supervised downstream tasks, *e.g.*, *question answering, name entity recognition*. BERT processes input text by converting it into a series of tokens, each mapped to a vector in a high-dimensional space. These vectors are learned during training to capture the semantic relationships between words and their context within a sentence. As illustrated in Figure 2.3, the token embeddings are combined with segment and position embeddings to represent input embedding to be fed into the BERT model. Segment embeddings distinguish between sentences in tasks involving multiple sentences, while position embeddings indicate the position of each word within a sentence. This integration enables BERT to comprehend the meaning of words in context, distinguish between sentences, and understand the word sequence in the text. We employ BERT as the backbone of our model architecture in Chapter 3.

DeBERTa [He, 2021] is a bidirectional encoder similar to BERT [Devlin, 2019]. DeBERTa enhances the BERT [Devlin, 2019] architecture by introducing the *disentangled attention* and an *enhanced mask decoder*. The disentangled attention mechanism utilizes distinct vectors to

represent the content and position of words, and it employs disentangled matrices for their contents and relative positions to compute the attention weights. Furthermore, the enhanced mask decoder integrates absolute positions at the decoding layer, after all transformer layers. Similar to BERT [Devlin, 2019], DeBERTa pre-trained on large text corpus with MLM objective. We employ DeBERTa as the language model of our model architecture in Chapter 4.

Sentence-BERT [Reimers, 2019] takes a single sentence as input and is trained by *metric learning* objectives, *e.g.*, *siamese or triplet structure*, facilitating efficient sentence similarity search. It is learned by fine-tuning a pre-trained BERT model on supervised semantic textual similarity. We utilize Sentence-BERT for segmenting dialogs as a part of the dialog summarization method in Chapter 3.

BART [Lewis, 2020] includes an encoder and a decoder. It is pre-trained as an unsupervised denoising autoencoder, *i.e.*, corrupting input text and learning to reconstruct the original, and fine-tuned on supervised classification, generation or translation tasks. It is particularly effective on *text generation*, including abstractive dialog, question answering, and summarization tasks. We utilize BART for summarize dialog in Chapter 3.

2.1.3 Learning paradigms

Pre-training language models. BERT [Devlin, 2019] demonstrated the effectiveness of pre-training a transformer language model on a large text corpus with MLM objective, followed by fine-tuning for specific NLP tasks. This process has notably influenced the practices of pre-training and fine-tuning in the field of transformer-based language models, setting a new standard for NLP models. MLM pre-training strategy for bidirectional language models involves training the model to predict randomly masked tokens using the context of unmasked tokens. This pre-training, a form of a self-supervised learning manner with cross-entropy loss, enables a language model to extract generalization from large-scale text corpora. Additionally, the pre-training strategy for autoregressive-based transformers primarily involves training the model to predict the next token in a sequence based on the preceding tokens, utilizing cross-entropy loss as the objective function. During inference, the next word is selected based on the probabilities of the model, a process known as *decoding*, which can employ various strategies, *e.g.*, *beam search, greedy decoding, and random sampling*. These self-supervised pre-training approaches leverage a text corpus as training data, eliminating the necessity for labeled data.

Adapting language models for downstream tasks. Adapting pre-trained language models for specific applications, known as downstream tasks, is important for leveraging the capabilities of pre-trained models. This adaptation process, also called *transfer learning*, has become a common practice in NLP, beginning with BERT [Devlin, 2019]. As proposed in BERT [Devlin, 2019], the primary strategy of adapting language models to downstream tasks involves a two-step process: pre-training and fine-tuning. A language model is initially pre-trained on a large corpus of unlabeled text data. Then, in the fine-tuning stage, the pre-trained model is further trained on a smaller, task-specific dataset by updating all model parameters [Devlin, 2019].

Recent advancements in NLP have led to the development of large-scale language models, with their performance significantly influenced by three main factors: the size of the model, the size of the dataset, and the computational resources allocated for training. Increasing these factors can enhance model performance, also known as scaling laws. However, the increasing size and complexity of these models introduce challenges when adapting them to downstream tasks, *e.g.*, *overfitting and catastrophic forgetting*. These challenges create a shift from traditional fine-tuning methods to more innovative adaptation strategies.

Prompting is known as one of the earliest adaptation methods for large-scale pre-trained language models, aimed at enabling these models to tackle downstream tasks effectively in zero-shot settings [Brown, 2020a]. This approach involves the integration of human-crafted sentences into the input of the model to guide the model in achieving specific tasks without additional training. Additionally, in-context learning [Brown, 2020a] represents a variation of prompting in which the input prompt contains multiple task examples to guide the language model in understanding and performing the new tasks.

Despite the effectiveness of prompting for leveraging large language models for various tasks, the complexity of designing handcrafted prompts does not always lead to optimal performance for specific tasks and is time-consuming. Consequently, these challenges have led to exploring alternative training-based adaptation strategies to eliminate the need for handcrafted prompts and enhance model performance. The emerged strategies include prompt learning [Li, 2021; Liu, 2021a; Liu, 2022a], adapter layers [Houlsby, 2019], BitFit [Zaken, 2022], low-rank adaptation (LoRA) [Hu, 2022], and QLoRA [Dettmers, 2023].

Prompt learning [Li, 2021; Liu, 2021a; Liu, 2022a] involves learning task-specific prompts while keeping the language model frozen. Adapter layers [Houlsby, 2019] introduce additional trainable modules into each transformer layer, consisting of linear mappings with the residual connection. BitFit [Zaken, 2022] involves updating only all bias terms of the model or a portion of them. LoRA [Hu, 2022] incorporates trainable matrices based on rank decomposition into each transformer layer while keeping the pre-trained model parameters frozen. QLoRA [Dettmers, 2023] utilizes a method where gradients are backpropagated through a frozen, 4-bit quantized pre-trained language model into low-rank adapters to reduce memory requirements for fine-tuning large-scale language models. In this thesis, we utilize prompt learning and adapters while keeping our vision and language models frozen, as detailed in Chapter 4.

2.2 Vision models

Computer Vision (CV) aims to enable machines to process and comprehend visual data similarly to human visual perception. CV has diverse applications in both visual and multimodal domains, *e.g.*, *object recognition, image segmentation, visual tracking, image/video captioning, and visual question answering*. In this thesis, we specifically focus on video question answering, which requires an understanding of the visual semantics in videos. This challenge highlights the key role of vision models, which transform the raw inputs from images or videos into visual representations.

This section provides an overview of the evolution of vision models, which has primarily been centered around enhancing visual representation. Thus, we present an overview of the evolution of image representations in Subsection 2.2.1 due to their significant impact on the video domain. Subsequently, we briefly discuss video representations in Subsection 2.2.2.

2.2.1 Image representations

Evolution of image models. Initial approaches, *e.g.*, *Histogram of Oriented Gradients (HOG) and Scale-Invariant Feature Transform (SIFT) [Lowe, 2004]*, focused on extracting handcrafted features from images as visual representation. HOG computes the magnitude and direction of the gradients at each pixel to capture the edges and textures within an image, grouping these into histograms within small image regions called cells. These histograms are normalized and combined across larger regions to form a feature vector that describes the shape and

appearance of an object. SIFT detects key points in images and generates distinctive feature vectors that are invariant to scale and orientation. SIFT aims to facilitate the comparison and matching of different perspectives of an object or scene. While these methods were pioneering for their time, they were limited by their inability to capture complex patterns and manual feature engineering requirements.

The shift from handcrafted to learned features marked a significant evolution in the field. Notably, LeNet-5 network [LeCun, 1998], a Convolutional Neural Network (CNN) designed for recognizing characters, incorporates convolutional, pooling, and fully connected layers. It demonstrates the feasibility of using CNNs for practical applications, establishing a foundational structure for the evolution of CNNs.

A significant advancement in CV emerged with the introduction of ImageNet dataset [Deng, 2009] and ImageNet Large Scale Visual Recognition Challenge (ILSVRC), with its extensive collection of labeled images. ImageNet dataset [Deng, 2009] provides the necessary data for training deep learning models, particularly CNN. The dataset is coupled with advancements in computational power, especially Graphics Processing Unit (GPU), and training deeper models became feasible. Notably, the introduction of AlexNet [Krizhevsky, 2012] transformed the field by demonstrating the power of CNNs to learn feature representations directly from large-scale data.

Following AlexNet [Krizhevsky, 2012], considerable progress has been achieved in the development of CNN architectures and their training methods. Particularly, GoogleNet [Szegedy, 2015] proposes inception modules by using concatenated filters of different sizes in the same layer to efficiently capture information at various scales. It significantly increases the depth and width of the network without increasing the computational complexity. Moreover, Batch Normalization (BN) [Ioffe, 2015] accelerates the training by normalizing layer inputs for each training mini-batch. This method enables higher learning rates while reducing the importance of initializing network weights. BN also inherently serves as a regularizer, helping to prevent overfitting. Thus, BN is an invaluable technique for optimizing and generalizing deep learning models. Furthermore, ResNet [He, 2016] proposes the concept of residual connections by redefining layers to learn residual functions in relation to their inputs. Residual networks facilitate the training of deeper models by effectively addressing the optimization challenge and enhancing the performance of the model through increased depth.

CNNs have been applied to a wide range of computer vision tasks beyond image classification [Simonyan, 2014b; Szegedy, 2015; He, 2016], including object detection [Ren, 2015; Redmon, 2016], semantic segmentation [Ronneberger, 2015; Badrinarayanan, 2017]. CNNs leverage the spatial hierarchy of features in images, enabling models to learn from simple edges to complex abstract concepts through multiple layers of transformation by learning from data. Consequently, CNNs significantly enhance model performance and generalization capabilities and overcome the constraints of handcrafted feature extraction approaches. Despite their success, CNNs inherently focus on extracting local patterns and textures through their convolutional filters, limiting their ability to process global image context.

Inspired by the success of transformers in NLP, vision transformer architectures have emerged as a powerful alternative to CNNs, demonstrating impressive performance across a variety of CV tasks, including image classification [Dosovitskiy, 2021; Liu, 2021b], object detection [Carion, 2020a; Zhang, 2023], semantic segmentation [Strudel, 2021]. These models utilize self-attention mechanisms to effectively capture global dependencies within an image, addressing the key limitations of CNNs in handling comprehensive visual information. Notably, ViT [Dosovitskiy, 2021] successfully adapts the transformer architecture without relying on convolutional layers to image recognition tasks, while prior methods combine CNNs with transformer architecture or self-attention mechanism [Carion, 2020b; Wu, 2020]. Furthermore, many variants of ViT architecture [Touvron, 2021; Han, 2021; Wang, 2021; Yuan, 2021a; Touvron, 2022; Chu, 2023] have been proposed to address different challenges, including reducing the computational costs of the attention mechanism, increasing data efficiency, and enhancing training procedures. Next, we briefly explain the adaptation of transformers for vision, specifically through the ViT architecture [Dosovitskiy, 2021], as utilized in this thesis.

Adaptation of transformers for vision. The transformer architecture is explained in Subsection 2.1.2. In this section, we introduce vision transformers by highlighting changes to adapting transformers into visual data, specifically describing ViT architecture [Dosovitskiy, 2021] as an employed model in this thesis.

Vision transformers treat images as sequences of patches, similar to text tokens in NLP. As demonstrated in Figure 2.4, each image is divided into smaller patches, which are then flattened and linearly embedded to produce vectors of the same dimensionality as the transformer model. This approach enables the model to process the image sequentially, with positional en-

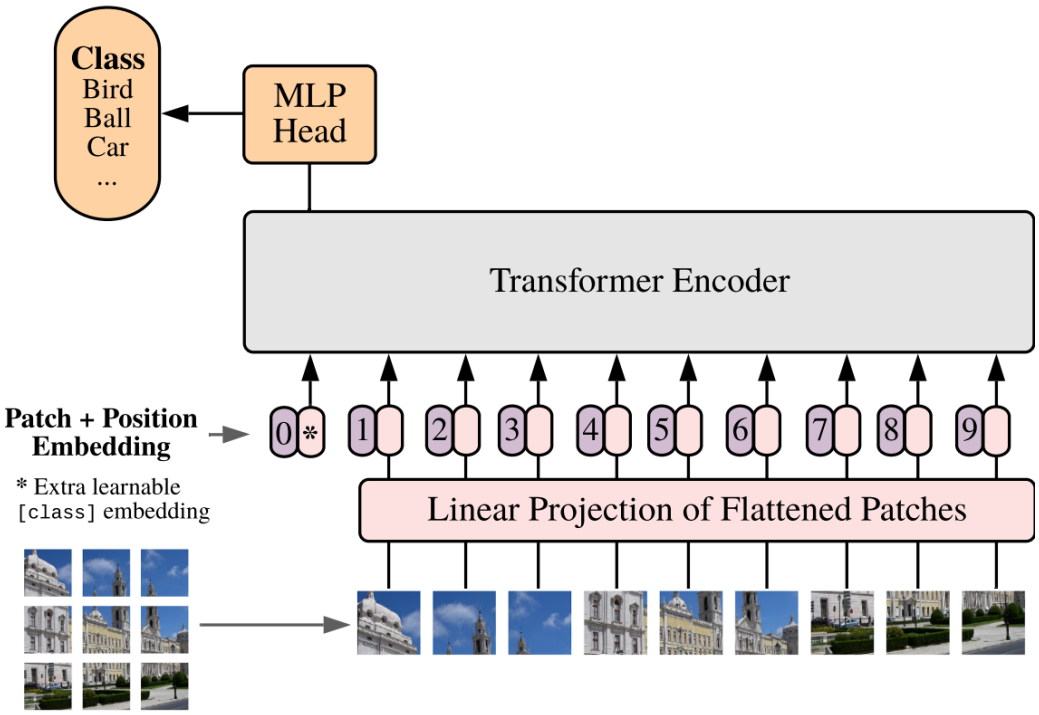


Figure 2.4 – ViT model overview. Image sourced from [Dosovitskiy, 2021].

codings to provide spatial relationships between patches, similar to how positional information is preserved in text sequences. This method allows transformers to interpret images within a sequence, offering a novel approach to visual data analysis.

Similar to BERT [Devlin, 2019], ViT employs a learnable classification token, denoted as $[CLS]$, which is prepended to the sequence of embedded patches. This token is passed through the transformer layers and aggregates information across the image. ViT transformer encoder is based on the original transformer encoder [Vaswani, 2017] as explained in Subsection 2.1.2. Following the transformer encoder, the classification token is used for predictions through an MLP head. ViT trained on image classification through supervised learning.

Learning paradigms. Self-supervised pre-training for image models creates a shift from traditional supervised techniques, facilitating learning of image representations directly from unlabeled images. A variety of approaches are employed for self-supervised learning in image models, including handcrafted pretext tasks, *e.g.*, *predicting image rotations [Gidaris, 2018] or solving jigsaw puzzles [Noroozi, 2016]*, contrastive learning [Chen, 2020c; He, 2020; Chen,

2021], clustering-based methods [Asano, 2020; Caron, 2020], masked image modeling [Bao, 2022; Zhou, 2022a], and generative image modeling [Chen, 2020b]. Notably, SimCLR [Chen, 2020c] utilizes augmented versions of the same image to learn similar representations, emphasizing the contrast between different images to enhance feature learning. MoCo [He, 2020] introduces a dynamic representation dictionary with a query encoder and a momentum-driven encoder, enhancing contrastive learning through a key queue for comparing positive pairs against extensive negative samples. BYOL [Grill, 2020] eliminates the requirement for pairing negative samples in contrastive learning by using two neural networks, called online and target networks. The online network predicts the representations of the same in different views generated by the target network, while the target network is updated by the moving average of the online network. DINO [Caron, 2021] enhances self-supervised learning in vision transformers using a self-distillation approach where the network learns to predict its own softened outputs across different views of the same image. BYOL focuses on minimizing the mean squared error between the feature representations of the online and target networks, while DINO uses cross-entropy loss to align the softmax output distributions of the student and teacher networks. On the other hand, iBOT [Zhou, 2022a] enhances masked image modeling by adopting a knowledge distillation to extract knowledge from the online tokenizer, optimized through momentum update.

Alternatively, visual representations can be learned from natural language supervision [Sariyildiz, 2020; Desai, 2021; Radford, 2021; Jia, 2021]. Since this approach is aligned with multimodal learning, we will further explore in Section 2.3.

2.2.2 Video representations

Introduction and datasets. Video representation requires capturing and understanding temporal dynamics beyond image representation. Advancements in this field have primarily been driven by the task of action recognition and the utilization of benchmark datasets, including HMDB51 [Kuehne, 2011], UCF101 [Soomro, 2012], Sports-1M [Karpathy, 2014], ActivityNet [Caba Heilbron, 2015], Charades [Sigurdsson, 2016], Kinetics [Kay, 2017], Something-Something [Goyal, 2017a], AVA [Gu, 2018]. These datasets can be used to train video models and, importantly, serve as primary benchmarks for evaluating the performance of video representation methods along with other video understanding tasks, *e.g.*, *text-to-video retrieval, action localization, action segmentation*.

Early approaches to CNN-based architectures. Progress in video model architectures has been significantly influenced by image models. Early works relied on handcrafted features to encode appearance and motion [Baccouche, 2010]. After the success of CNNs in image representation learning, 2D CNNs enhanced with temporal modules emerged as a standard architecture for video models [Simonyan, 2014a; Karpathy, 2014; Yue-Hei Ng, 2015; Donahue, 2015; Wang, 2018; Lin, 2019]. In terms of temporal modeling, two stream networks [Simonyan, 2014a] is introduced to designed to process RGB frames and optical flow through two independent streams, before merging their outputs using a late fusion strategy. Additionally, the sequential nature of video data led to the adoption of RNN [Yue-Hei Ng, 2015; Donahue, 2015] to enhance temporal analysis.

Furthermore, 3D CNNs have been proposed for extracting spatial-temporal features directly from video data [Tran, 2015; Carreira, 2017] due to the limitations of 2D CNNs in capturing temporal relationships. However, 3D CNNs have more computational requirements than 2D CNNs, which leads to investigating adaptable and efficient architectural solutions. For instance, SlowFast [Feichtenhofer, 2019] employs dual pathways to analyze videos at different frame rates, effectively capturing both slow and fast motions. In addition, X3D [Feichtenhofer, 2020] presents a scalable architecture that systematically expands standard 2D CNNs into more efficient 3D counterparts.

Transformer-based architectures. Following the introduction of vision transformers for image classification, adapting them for video classification tasks has been promising due to their ability to incorporate temporal data. Specifically, ViViT [Arnab, 2021] and TimeSformer [Bertasius, 2021] explore variants of space-time factorization by adapting ViT architecture [Dosovitskiy, 2021] for video models. Moreover, Video Swin Transformer [Liu, 2022c] extends the Swin Transformer architecture [Liu, 2021b] to video understanding, leveraging hierarchical, shifted window-based self-attention mechanisms for efficient and scalable video processing.

Learning paradigms. Inspired by advancements in image representation learning, self-supervised pre-training of video models has emerged as an alternative to supervised learning methods for video representation learning. This approach allows learning video representations directly from videos, eliminating labeled data requirements. It employs a variety of objectives, including contrastive learning [Dave, 2022], temporal order prediction [Misra, 2016], masked video modeling by prediction of discrete tokens [Wang, 2022], prediction of features [Wei,

2022], or directly reconstructing the pixels [Tong, 2022]. Furthermore, the exploration of training of video models has extended beyond videos [Miech, 2020b; Alayrac, 2020; Akbari, 2021; Han, 2022; Zhao, 2023], incorporating audio, text, or both modalities, where audio modality can also be in text form through ASR. Employing additional modalities to supervise video models extends beyond only video-centered tasks to video and language tasks; therefore, we will further explore in Section 2.3.

2.3 Vision-language models

In this section, we first provide an overview of the evolution of vision-language models in Subsection 2.3.1. Subsequently, we explore recent advancements in vision-language models by focusing on model architectures and learning paradigms in Subsection 2.3.2.

2.3.1 Evolution of vision-language models

Video-language tasks. Vision-language models have significantly broadened the scope of multimodal understanding and generation, introducing diverse tasks that enable bridging the gap between visual information and language. We discuss vision-language models mainly focusing on video-language tasks, specifically, text-video retrieval [Chen, 2011; Xu, 2016; Rohrbach, 2017; Krishna, 2017a], which aims to find relevant video segments corresponding to textual queries, video captioning [Xu, 2016; Krishna, 2017a; Zhou, 2018], which involves generating descriptive text for videos, and video question answering [Xu, 2017; Jang, 2017; Lei, 2018; Yu, 2019; Garcia, 2020b; Yang, 2021], which involves answering questions about videos.

Early approaches with pre-extracted features. Initial approaches to developing video language models heavily relied on extracting video and text features offline, utilizing distinct models for action recognition [Tran, 2015; Carreira, 2017; Xie, 2018], image recognition [He, 2016], and word representations [Mikolov, 2013; Pennington, 2014; Bojanowski, 2017]. Subsequently, these pre-extracted features are fused to address specific video-language tasks. The fusion techniques often employ CNNs for spatial understanding and LSTMs for temporal sequence learning [Venugopalan, 2015b; Venugopalan, 2015a; Na, 2017; Lei, 2018; Liang, 2018].

Attention mechanism and transformers. The introduction of the attention mechanism [Bahdanau, 2015] and transformer architecture [Vaswani, 2017] marked a significant shift in the de-

velopment of vision-language models. Attention mechanisms and transformers are initially utilized along with pre-extracted visual and textual features to allow cross-modal interactions for different video-language tasks [Yu, 2018; Kim, 2018; Kim, 2019a; Kim, 2020b; Iashin, 2020b; Iashin, 2020a]. For instance, JSFusion [Yu, 2018] employs recursively learnable attention modules to measure semantic similarity between multimodal sequence data for video question answering and text-video retrieval. MDVC [Iashin, 2020b] utilizes a transformer architecture to encode pre-extracted feature representations of each modality, and then these encoded representations are fused to perform video captioning.

Pre-training paradigm. Inspired by the success of BERT [Devlin, 2019] in the NLP field, the vision-language domain has widely adopted pre-training as a common approach. This approach generally involves pre-training models on large-scale image-text or video-text pairs datasets to learn rich, joint visual and language representations, rather than relying solely on separately extracted representations. Then, these learned representations can then be fine-tuned for a variety of tasks with minimal adjustments to the architecture of the model. Notably, image-text pre-training approaches [Lu, 2019; Tan, 2019; Chen, 2020d; Zhou, 2020; Radford, 2021; Jia, 2021] have expanded to include video-language pre-training [Sun, 2019; Zhu, 2020; Miech, 2020a]. In the following section, we will discuss pre-training strategies in detail.

2.3.2 Model architectures and learning paradigms

This section provides an overview of the pre-training paradigms for vision-language models by exploring recent developments in model architectures and important datasets. We will discuss common pre-training objectives and parameter-efficient adaptations of large-scale models.

Image-text pre-training. Initial image-text pre-training approaches, *e.g.*, *ViLBERT [Lu, 2019], LXMERT [Tan, 2019], UNITER [Chen, 2020d], VLP [Zhou, 2020]*, have been proposed to learn joint vision-language embeddings and unify various vision-language tasks. Subsequently, larger datasets were utilized for pre-training, *e.g.*, *CLIP [Radford, 2021], ALIGN [Jia, 2021], LIT [Zhai, 2022], Florence [Yuan, 2021b]*, significantly improving their generalization capabilities across a wide range of tasks.

The learning of joint vision and language embeddings not only facilitates tasks that involve both vision and language but also enhances the learning of visual representations for

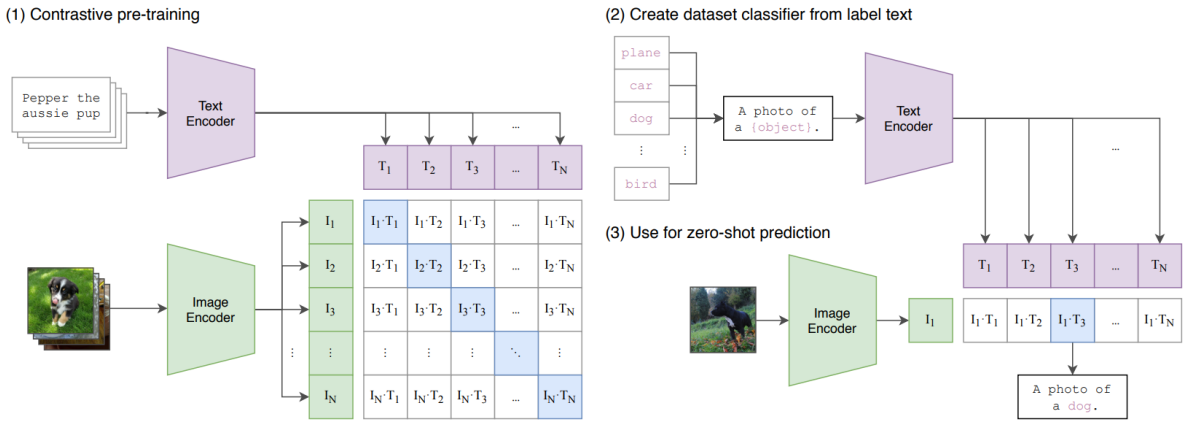


Figure 2.5 – CLIP model overview. Image sourced from [Radford, 2021].

purely vision tasks *e.g.*, *action recongition*, under the language supervision [Sariyildiz, 2020; Desai, 2021; Radford, 2021; Jia, 2021]. This approach serves as an alternative to traditional label supervision and self-supervision, as mentioned in Section 2.2. Particularly, CLIP [Radford, 2021] learns a shared representation space where similar concepts in text and images are closely aligned through a contrastive learning approach. As shown in Figure 2.5, CLIP processes images and text separately through two encoders: a vision encoder utilizing a vision transformer, specifically ViT [Dosovitskiy, 2021], and a text encoder operating as a transformer encoder model. After each encoder produces embeddings, CLIP projects these embeddings into the shared space, allowing them to be directly compared. CLIP is pre-trained on a diverse and large-scale dataset, including 400 million image-text pairs collected from a variety of publicly available sources on the internet. CLIP can perform various image-related tasks without task-specific training data. Specifically, CLIP has the ability to classify images by evaluating similarities between the embedding of images and the embeddings of text descriptions of potential categories. Then, it assigns a class label to an image based on the text embedding that shows the highest similarity, which refers to zero-shot prediction.

In addition to image-based tasks, CLIP features have been employed for video tasks [Lin, 2022; Ju, 2022; Ni, 2022; Luo, 2022; Yang, 2022b], demonstrating their applicability to video analysis. Consequently, we leverage the CLIP vision encoder for frame-level feature extraction from videos in this thesis. Beyond CLIP, we introduce a method to obtain video-level representations from CLIP features, which will be detailed in Chapter 4. This approach allows for a more holistic understanding of video, bridging the gap between image understanding and the dynamic nature of videos.

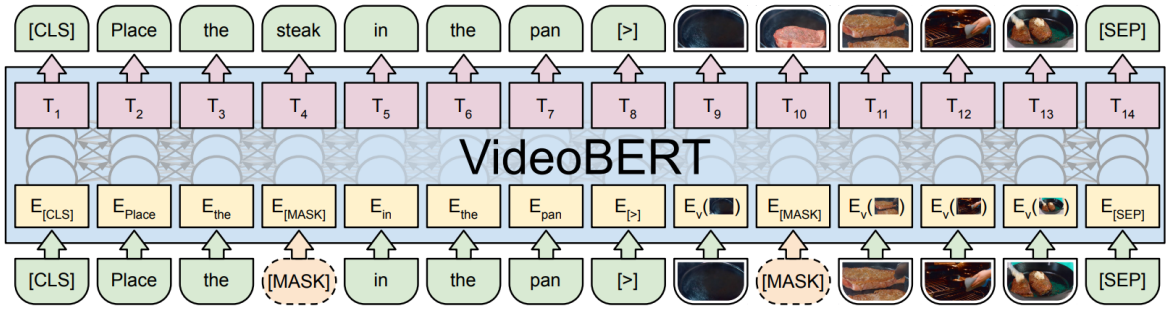


Figure 2.6 – VideoBERT model overview. Image sourced from [Sun, 2019].

Video-text pre-training. Recent works have been proposed to learn joint video-language embeddings and unify multiple video-language tasks, leveraging techniques from image pre-training. VideoBERT [Sun, 2019] is a pioneer work in video-text pre-training, adapting and extending BERT [Devlin, 2019] to video by using a collection of cooking videos from YouTube. The process involves segmenting videos into short clips; each clip is analyzed using a pre-trained S3D network to extract feature vectors. These vectors are subsequently clustered to form a vocabulary of visual tokens. As illustrated in Figure 2.6, a BERT-based encoder processes both textual and visual tokens to predict masked tokens. These learned representations are employed to train a transformer-based method for video captioning.

The introduction of the HowTo100M dataset [Miech, 2019a], a large-scale and noisy collection of video-text pairs, has enhanced the pre-training of video-language models for various video-language tasks, including video question answering [Zhu, 2020; Seo, 2021], text-video retrieval [Patrick, 2021; Zhu, 2020; Gabeur, 2020], and video captioning [Zhu, 2020; Seo, 2022]. Despite providing an extensive collection of video-text pairs, the high noise density of the HowTo100M dataset requires considerable computational resources to achieve competitive results. The MIL-NCE loss [Miech, 2020a], specifically designed to tackle the issue of misaligned narrations within the HowTo100M dataset, facilitates the learning of joint text-video embeddings directly from unlabelled and uncurated narrated videos in an end-to-end manner. Alternatively, due to these high computational demands, CLIPBERT [Lei, 2021] performs pre-training on image-text pairs from COCO Captions [Chen, 2015] and Visual Genome Captions [Krishna, 2017b] for text-video retrieval, and video question answering tasks.

Furthermore, the Howto100M dataset is utilized in the pre-training phase of various works, often relying on pre-trained modules to extract visual features following previous works. For instance, ActBERT [Zhu, 2020], which leverages a transformer block to integrate global actions, local objects, and linguistic descriptions. HERO [Li, 2020] utilizes two types of transformers, which are cross-modal and temporal transformers, to hierarchically encode multimodal inputs, capturing both local and global video contexts. UniVL [Luo, 2020] introduces a transformer-based architecture, including two single-modal encoders, a cross-encoder, and a decoder, to perform multimodal understanding and generation tasks through a two-stage pre-training process, initially focusing on video-text contrastive learning followed by various pre-training objectives. VideoCLIP [Xu, 2021] proposes a unified model trained with contrastive learning that achieves zero-shot capabilities. The model enhances video-text alignment through pre-training with temporally overlapping video-text pairs and improves video-text similarity learning using a retrieval method that incorporates challenging negative pairs from different videos.

Building on the paradigm of end-to-end transformer-based image-text pre-training, which incorporates image encoders for processing raw inputs rather than using pre-extracted visual features, similar approaches have been extended to video-language models. For instance, MERLOT [Zellers, 2021] trains its image encoder from scratch by introducing the YT-Temporal-180M [Zellers, 2021] dataset, which not only includes the HowTo100M dataset but also offers a significantly larger and more diverse range of video content. MERLOT integrates frame-level and video-level objectives to holistically understand and interpret dynamic visual content. This strategy uses masked language modeling, masked frame modeling, and frame order modeling objectives. Similarly, VIOLET [Fu, 2021] introduces an end-to-end video-language framework that integrates a video transformer and a new pre-training task called masked visual-token modeling to effectively capture the temporal dynamics of videos. All-in-one [Wang, 2023] proposes a unified encoder architecture for processing both visual and textual inputs, by introducing a token rolling operation to handle temporal data without adding model complexity. This approach allows the network to be used as a dual-stream framework to process text-video retrievable tasks efficiently. LAVENDER [Li, 2023b] proposes a unified video-language framework by eliminating task-specific architectures using a single masked language modeling head for all pre-training and downstream tasks. Furthermore, UnIVAL [Shukor, 2023b] presents a unified model to process image, video, and audio by leveraging multimodal curriculum learning strategy and weight interpolation.

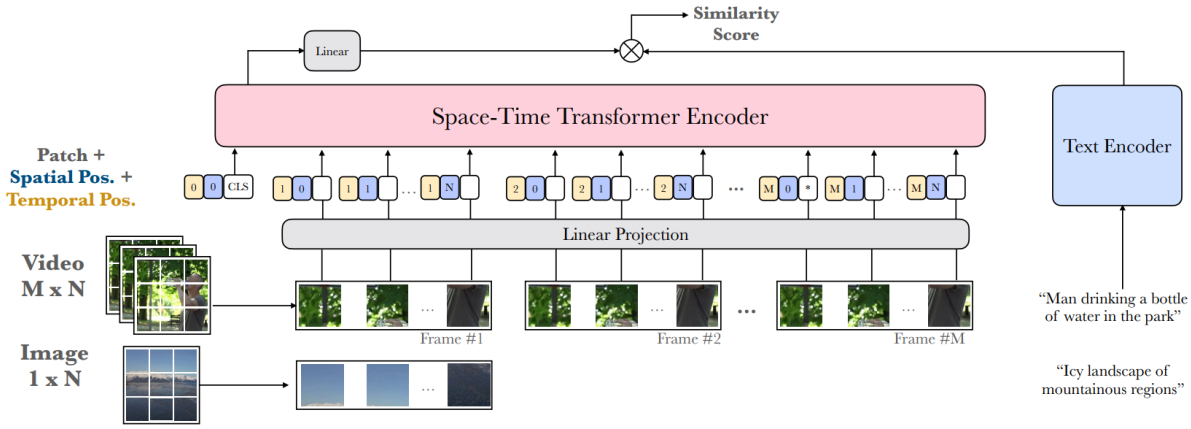


Figure 2.7 – Frozen in Time model overview. Image sourced from [Bain, 2021].

Additionally, the introduction of the WebVid-2M dataset [Bain, 2021], which includes 2.5 million video-text pairs, marks a significant advancement in the field and it has been employed extensively [Bain, 2021; Fu, 2021; Yang, 2022b; Li, 2023b; Wang, 2023]. In this thesis, we also utilize the WebVid-2M dataset. Compared to HowTo100M, the WebVid dataset [Bain, 2021] is considerably smaller in terms of video duration and number of video-text pairs, but it is manually generated, well-formed sentences that are accurately aligned with the video content. In contrast, the HowTo100M relies on automatically generated captions from narrations that often contain incomplete sentences, lack punctuation, and do not always correspond to the visual content, introducing potential mismatches and noise. Furthermore, the OBELICS dataset [Laurençon, 2023] introduces a curated collection of 141 million English web documents, resulting in 353 million images and 115 billion tokens. It enhances multimodal training with long-form text with images, surpassing the typical short or non-grammatical image-text pairs. Moreover, the Cauldron dataset [Laurençon, 2024] comprises a collection of 50 vision-language datasets designed for a variety of tasks, including visual question answering, captioning, and document understanding. Each dataset is formatted into a unified question-answer structure that supports multi-turn conversations. Additionally, it is further improved by text-only instruction datasets for complex instruction following and problem-solving.

Alongside the unified frameworks previously mentioned, various studies adopt the pre-training and fine-tuning paradigm, focusing specifically on individual tasks [Bain, 2021; Luo, 2022]. This thesis similarly employs this paradigm, particularly focusing on video question answering. For instance, Frozen-in-time [Bain, 2021] introduces an end-to-end video-text retrieval

model that utilizes a dual encoder architecture based on transformers. This model can be flexibly trained on both video and image datasets, as demonstrated in Figure 2.7. Additionally, it incorporates curriculum learning by initially training on a smaller number of input frames and then gradually increasing the frame numbers as training progresses, enhancing training efficiency. Additionally, CLIP4Clip [Luo, 2022] leverages the CLIP model for video-text retrieval in an end-to-end manner, integrating two single-modal encoders built upon the pre-trained CLIP model and a similarity calculator to process video-text pairs.

Pre-training objectives. Recent methods utilize different vision-language pre-training objectives, either individually or in combination. The most common objectives include visual-text contrastive learning [Luo, 2020; Radford, 2021; Xu, 2021; Zellers, 2021; Fu, 2021; Wang, 2023] and visual-text matching [Luo, 2020; Fu, 2021; Wang, 2023] to learning video and text representations through alignment and matching mechanisms, respectively. Additionally, masked language modeling [Li, 2020; Luo, 2020; Fu, 2021; Wang, 2023; Li, 2023b; Zellers, 2021], masked frame modeling [Li, 2020; Luo, 2020], and masked video modeling [Fu, 2021], all of which aim to predict masked elements in multimodal data. Moreover, language modeling [Luo, 2020; Shukor, 2023b] predicts subsequent words in sequences. Furthermore, HERO [Li, 2020] introduces new pre-training tasks such as video-subtitle matching, which requires aligning subtitles with video clips globally and locally, and frame order modeling, which involves reordering shuffled video frames. Additionally, VIOLET [Fu, 2021] introduces masked visual-token modeling by employing a discrete variational autoencoder that tokenizes and reconstructs visual inputs into discrete tokens, each corresponding to spatial image patches.

Parameter-efficient adaptations of large-scale models. The emergence of large-scale language models has significantly impacted the field of vision language tasks, particularly in adapting to multimodal tasks with limited data scenarios. As discussed in Subsection 2.1.3, parameter-efficient adaptation methods for language models, including prompt learning [Li, 2021; Liu, 2021a; Liu, 2022a], adapter layers [Houlsby, 2019], BitFit [Zaken, 2022], low-rank adaptation (LoRA) [Hu, 2022], and QLoRA [Dettmers, 2023], are extendable to multimodal approaches. These methods typically involve freezing the parameters of vision and language models while introducing new, learnable parameters. They often employ a mapping strategy to bridge the gap between modalities. By leveraging existing knowledge in both language and visual understanding, these adaptations enhance model capabilities in zero-shot and few-shot learning scenarios for multimodal tasks.

Recent works in multimodal research concentrate on adapting large-scale language models [Tsimpoukelli, 2021; Mokady, 2021; Alayrac, 2022; Yang, 2022b; Han, 2023; Li, 2023a]. Notably, Frozen [Tsimpoukelli, 2021] integrates a vision encoder with a frozen language model. The vision encoder is trained from scratch on image-caption pairs to perform zero-shot image-text tasks, including image question answering. ClipCap [Mokady, 2021] introduces a method for learning a mapping network between vision and language models while keeping the models frozen for the task of image captioning. Moreover, FrozenBiLM [Yang, 2022b] employs a frozen bidirectional language model and vision model, incorporating lightweight, learnable adapter layers to facilitate zero-shot video question answering, including a pre-training stage on video-caption pairs using a masked language modeling objective. Flamingo [Alayrac, 2022] uses a frozen auto-regressive language model with trainable cross-attention layers that incorporate vision and language input, trained on an extreme-scale dataset. Furthermore, BLIP-2 [Li, 2023a] leverages frozen language and vision models by introducing Q-Former as a mapping network, which is pre-trained in representation learning and generative learning stages. It performs various vision-language tasks, including visual question answering, image captioning, and image-text retrieval. Additionally, eP-ALM [Shukor, 2023a] enhances an auto-regressive language model by augmenting it with perception through several modality-specific encoders. This approach strategically focuses on training a minimal number of parameters for downstream multimodal tasks, eliminating the need for extensive multimodal pre-training.

In this thesis, we focus on zero-shot and few-shot video question answering by leveraging frozen vision and language models, as detailed in Chapter 4. Our method incorporates visual inputs to a frozen language model using lightweight learnable adapter layers, similar to FrozenBiLM [Yang, 2022b]. Furthermore, we introduce a novel visual mapping network that summarizes the video input while allowing for temporal interaction. For the first time, we introduce multimodal prompt learning for video question answering, where our architecture incorporates learnable prompts integrated into the visual and textual inputs. Specifically, visual prompts are incorporated into the visual mapping network, and the language model integrates learnable text prompts in the key and value of multi-head attention in each layer of the language model.

CHAPTER 3

LONG-RANGE VIDEO QUESTION ANSWERING

Contents

This chapter addresses long-range Video Question Answering (VideoQA), which involves analyzing videos that extend beyond a few seconds or minutes. For instance, while watching a TV show, a viewer might wonder about events or details from earlier in the episode.

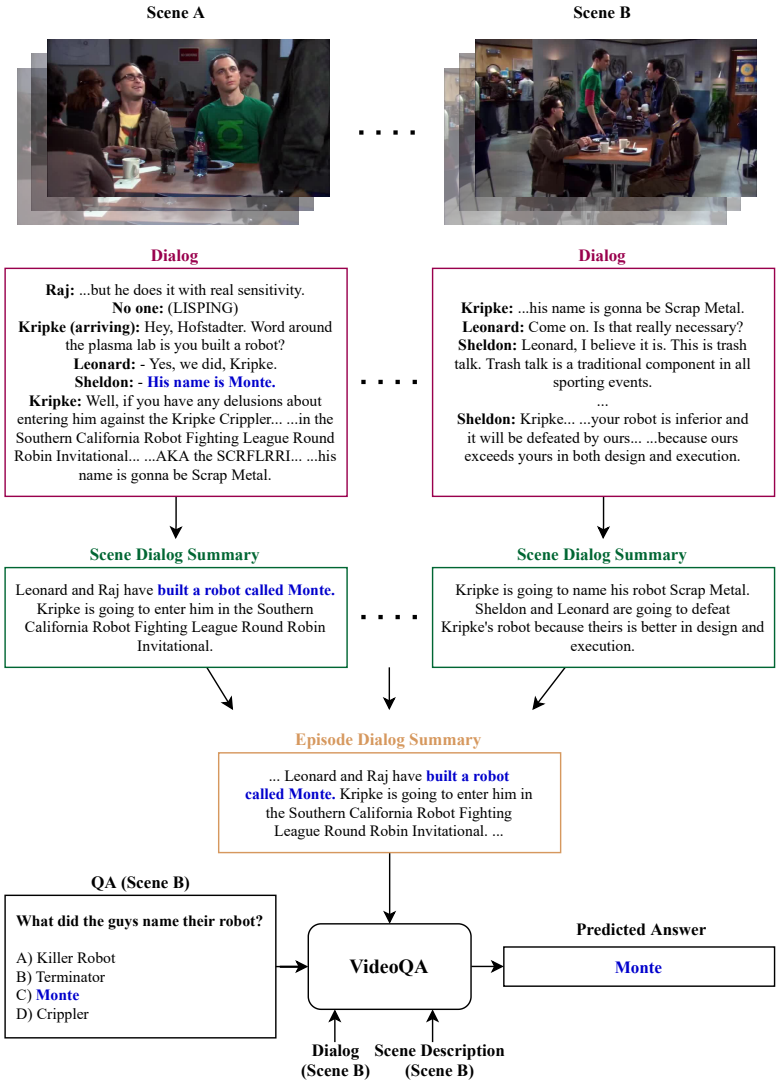


Figure 3.1 – In VideoQA, a question is associated with Scene B, but it can only be answered by information from Scene A. We generate episode dialog summaries from subtitles and give them as input to our VideoQA system, dispensing with the need for external knowledge.

As is illustrated in Figure 3.1, in scene B on the right, a robot competition is discussed, but the robot's name, which is the subject of the question, is not mentioned. However, in scene A from the same episode on the left, a character named Sheldon reveals their robot's name as Monte; thus, the information needed to answer the question can be sourced from this earlier scene. This indicates that long-range video question answering requires a high-level understanding of the entire video, but obtaining narratives from raw data is extremely challenging.

Although the answers could be in the dialog, the raw dialog is often informal and repetitive; searching through all available durations of such a noisy source is error-prone and impractical.

Modern video question answering systems often use additional human-made sources like plot synopses, scripts, or knowledge bases. On the contrary, we present a new approach to understanding the story without external sources. Inspired by the trend of video captioning, we go a step further and apply the same idea to *dialog*: we *summarize* raw dialog, converting it into *text description* for question answering. After generating scene dialogue summaries for all scenes, we combine them to create a comprehensive episode dialogue summary. In this way, we automatically generate narrative stories derived from the video content.

The content of this chapter corresponds to our ICCV 2021 publication, *On the hidden treasure of dialog in video question answering* [Engin, 2021b].

3.1 Introduction

Deep learning has accelerated progress in vision and language tasks. *Visual-semantic embeddings* [Kiros, 2014; Frome, 2013] have allowed zero-shot learning, cross-modal retrieval and generating new descriptions from embeddings. *Image captioning* [Vinyals, 2015] and *VQA* [Antol, 2015] have demonstrated generation of realistic natural language description of images and a great extent of multimodal semantic understanding. The extension to *video captioning* [Krishna, 2017a; Venugopalan, 2015a] and VideoQA [Tapaswi, 2016a; Lei, 2018] has enabled further progress because video requires a higher level of reasoning to understand complex events [Zellers, 2019].

VideoQA systems typically have similar architecture focusing on multimodal embeddings or descriptions, temporal attention and localization, multimodal fusion, and reasoning. While it is often hard to isolate progress in individual components, there are some clear trends. For instance, custom self-attention and memory mechanisms for fusion and reasoning [Na, 2017; Kim, 2018; Fan, 2019] are gradually being streamlined by using *transformer* architectures [Urooj, 2020; Kim, 2020b; Yang, 2020]; while visual embeddings [Tapaswi, 2016a] are being replaced by semantic embeddings [Lei, 2018] and *text descriptions* by captioning [Kim, 2020a; Chadha, 2020].

Datasets are essential for progress in the field, but often introduce bias. For instance, questions from text summaries are less relevant to visual information [Tapaswi, 2016a]; supervised temporal localization [Lei, 2018] biases system design towards two-stage as localization and answering [Lei, 2019; Kim, 2020b]; fixed question structure focusing on temporal localization [Lei, 2018] often results in mere *alignment* of questions with subtitles and *matching* answers with the discovered context [Kim, 2020a], providing little progress on the main objective, which is to study the level of understanding.

Bias can be removed by removing localization supervision and balancing questions over different aspects of comprehension, for instance visual, textual, or semantic [Garcia, 2020b]. However, the requirement of external knowledge, which can be in the form of hints or even ground truth, does not leave much progress in inferring such knowledge from raw data [Garcia, 2020b]. Even weakening this requirement to plain text *human-generated summaries* [Garcia, 2020a], still leaves a system unusable in the absence of such data.

Recent methods often rely on additional human-generated sources like plot synopses, scripts, video descriptions, or knowledge bases [Garcia, 2020b; Garcia, 2020a]. In contrast, we aim to understand videos directly from raw data. The secret lies in the dialog: unlike any prior work, we treat dialog as a noisy source to be converted into text description via *dialog summarization*, much like recent methods treat video. The input of each modality is encoded by a language independently, and a simple fusion method combines all modalities, using soft temporal attention for localization over long inputs. Our model outperforms the state of the art on the KnowIT VQA dataset by a large margin, without using question-specific human annotation or human-made plot summaries. It even outperforms human evaluators who have never watched any whole episode before.

Our finding is astounding: our dialog summary is not only a valid replacement for a human-generated summary in handling questions that require knowledge of a whole story, but it outperforms them by a large margin.

Our contributions can be summarized as follows:

1. We apply *dialog summarization* to video question answering for the first time.
2. Building on a modern VideoQA system, we convert all input sources into *plain text description*.

3. We introduce a weakly-supervised *soft temporal attention* mechanism for localization.
4. We devise a very simple *multimodal fusion* mechanism that has no hyperparameters.
5. We set a new state of the art on KnowIT VQA dataset [Garcia, 2020b] and we beat non-expert humans for the first time, working only with raw data.

3.2 Related work

Video question answering. Progress on video question answering has been facilitated and driven by several datasets and benchmarks. VideoQA by Tapaswi *et al*. [Tapaswi, 2016a] addresses answering questions created from *plot synopses* using a variety of input sources, including video, subtitles, scene descriptions, scripts and the plot synopses themselves. Methods experimenting on MovieQA focus on *memory networks* capturing information from the *whole movie* by videos and subtitles [Na, 2017; Kim, 2019b], scene-based memory attention networks to learn joint representations of frames and captions [Kim, 2018], and LSTM based sequence encoders to learn visual-text embeddings [Liang, 2018].

TVQA [Lei, 2018] and TVQA+ [Lei, 2019] address *scene-based* questions containing *temporal localization* of the answer in TV shows, using video and subtitles. The questions are structured in two parts: one specifying a temporal location in the scene and the other requesting some information from that location. This encourages working with more than one modalities. Methods experimenting on these datasets focus on temporal localization and attention [Lei, 2019; Kim, 2020b], *captioning* [Kim, 2020a; Chadha, 2020] and *transformer*-based pipelines capturing visual-semantic and language information [Yang, 2020; Urooj, 2020].

KnowIT VQA [Garcia, 2020b] is a *knowledge-based* dataset, including questions related to the scene, the episode or the entire story of a TV show, as well as *knowledge annotation* required to address certain questions, in the form of hints. *Transformer*-based methods are proposed to address this task by employing knowledge annotation [Garcia, 2020b] or external human-generated *plot summaries* [Garcia, 2020a]. Our method differs in substituting human-generated knowledge by summaries automatically generated from raw dialog.

Dialog summarization. Dial2Desc dataset [Pan, 2018] addresses generating *high-level short descriptions from dialog* using a transformer-based text generator. SAMSum corpus [Gliwa, 2019] is a human-annotated dialog summarization dataset providing speaker information. Meth-

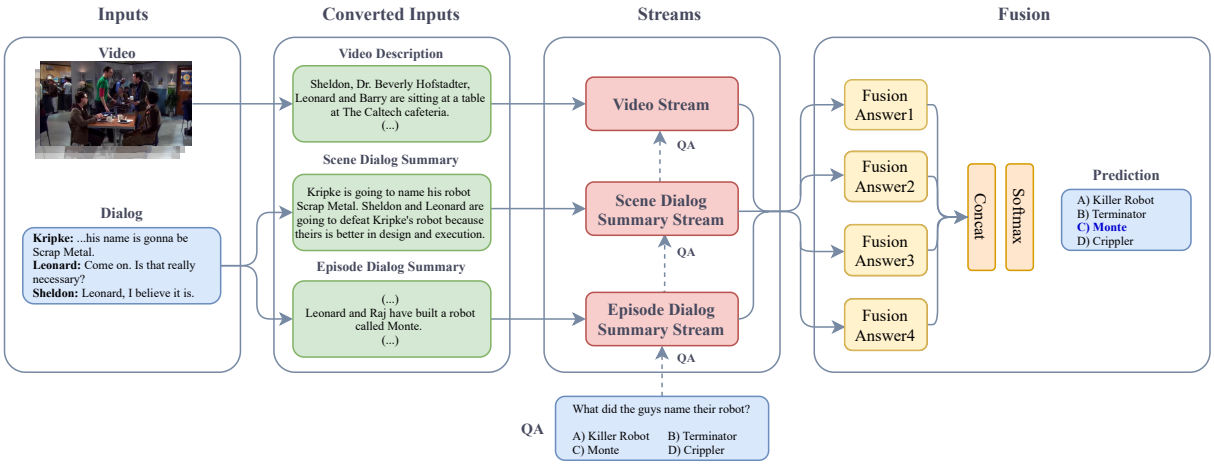


Figure 3.2 – Our VideoQA system converts both video and dialog to text descriptions/summaries, the latter at both scene and episode level. Converted inputs are processed independently in streams, along with the question and each answer, producing a score per answer. Finally, stream embeddings are fused separately per answer and a prediction is made.

ods experimenting on this dataset include existing *document summarization* methods [Gliwa, 2019], *graph neural networks* integrating cross-sentence information flow [Zhao, 2020] and graph construction from utterance and commonsense knowledge [Feng, 2020]. Since dialog differs from structured text and requires extraction of the conversation structure, recent work focuses on representing the dialog from different *views* by sequence to sequence models [Chen, 2020a]. We follow this approach for our dialog summarization.

3.3 Overview

We address knowledge-based video question answering on TV shows. Each episode is split in *scenes*. For each scene, we are given the *video* (frames) and *dialog* (speaker names followed by subtitle text) and a number of *multiple-choice questions*. Certain questions require high-level understanding of the whole episode or show. Garcia *et al.* [Garcia, 2020a] rely on human-generated *plot summaries* (or *plot* for short), which we use only for comparison. Our objective is to extract the required knowledge from raw data.

As shown in Figure 3.2, we first convert inputs into *plain text description*, including both video (by visual recognition) and dialog (by summarization). A number of separate *streams* then map text to embeddings, at the level of both *scene* (video and scene dialog summary) and

episode (episode dialog summary and plot). The question and answers are embedded together with the input text of each stream. A *temporal attention* mechanism localizes relevant intervals from episode inputs. Finally, question answering is addressed both in a *single-stream*, and a *multi-stream* scenario. The latter amounts to *multimodal fusion*.

3.4 Input description

All input sources, *i.e.*, *video*, *dialog* and *plot*, are converted into *plain text description* before being used for question answering. Video is first converted into a *scene graph* by a visual recognition pipeline and then to text description by a set of rules. Importantly, although already in textual form, dialog is also converted into text description by *dialog summarization*. The plot, already in text description form, is used as is, but for comparison only: our main contribution is to replace human-generated plots by automatically generated descriptions.

3.4.1 Dialog

As the main form of human communication, dialog is an essential input source for video understanding and question answering. We use dialog in three ways: *raw dialog* per scene, *dialog summary* per scene and the collection of dialog summary over a whole *episode*.

Raw scene dialog. As in all prior work, we use the raw dialog associated to the scene of the question, *as is*. Although in textual form, it is *not* a text description. It may still contain more information than dialog summary, which is important to investigate.

Scene dialog summary. Given the dialog associated with the scene of the question, we convert this input source into text description by *dialog summarization*. Despite being of textual form, dialog is very different from text *description*: conversations are often informal, verbose and repetitive, with few utterances being informative; while a description is a narrative in *third-person* point of view with clear information flow structured in paragraphs [Chen, 2020a]. Identifying the speaking person is also substantial, especially with multiple people in a conversation. Rather than generic document summarization [Gliwa, 2019], we follow a dedicated dialog summarization method [Chen, 2020a], which blends character names with events in the generated summaries.

A dialog is a sequence of *utterances*, each including a *speaker* (character) name and a *sentence* (sequence of tokens). Each utterance is mapped to a vector embedding by Sentence-BERT [Reimers, 2019]. The sequence of embeddings over the entire dialog is segmented according to *topic*, *e.g. greetings*, *today's plan*, *etc.* by C99 [Choi, 2000], as well as *stage*, *e.g. opening*, *intention*, *discussion*, *conclusion* by a *HMM* [Althoff, 2016]. As a result, for each *view* (topic or stage), the dialog is represented by a sequence of *blocks*, each containing several utterances.

Given the above structure, the input is re-embedded, and the summary is generated using an extension of BART [Lewis, 2020]. In particular, there is one *encoder* per view, mapping each block to an embedding. An LSTM [Hochreiter, 1997] follows, aggregating the entire view into one embedding, obtained as its last hidden state. The *decoder* attends over the output of each encoder using a *multi-view attention* layer to weight the contribution of each view. It is *auto-regressive*, using previous tokens from ground truth at training and previously predicted tokens by the encoder at inference.

We train the HMM on the dialog sources of our VideoQA training set; otherwise, we use Sentence-BERT and BART as used/trained by [Chen, 2020a]. Once a scene dialog summary is generated, it is re-embedded by BERT [Devlin, 2019] like all other input sources, as discussed in Section 3.5.

Episode dialog summary. We collect the scene dialog summaries for all scenes of an episode, and we concatenate them into an *episode dialog summary*. Assuming that the episode of the scene of the question is known, we make available the associated episode dialog summary for question answering. This is a long input source and requires *temporal attention*, as discussed in Subsection 3.5.3. Importantly, episode dialog summary is our most important contribution in substituting plot summary with an automatically generated description.

3.4.2 Plot summary

As part of our comparison to [Garcia, 2020a], we use publicly available plot summaries [1], already in text description form. Assuming that the episode of the scene of the question is known, we make available the associated plot as is to help answering *knowledge-based ques-*

1. `https://the-big-bang-theory.com/`

tions. A plot is shorter and higher-level than our episode dialog summary, but it is still long enough to require *temporal attention*. It is important to investigate whether we can dispense such a human-generated input and how much more information it contains relative to what we can extract automatically.

3.4.3 Video

We use a visual recognition pipeline to convert raw input video into text description. Following [Garcia, 2020a], this pipeline comprises four components: *character recognition* [Schroff, 2015], *place recognition* [Zhou, 2017], *object relation detection* [Zhang, 2019], and *action recognition* [Wu, 2019]. The outputs of these components are character, place, object, relation and action *nodes*. A directed *video scene graph* is generated by collecting all nodes along with edges and then a textual *scene description* is obtained according to a set of predefined rules.

3.5 Single-stream question answering

As shown in Figure 3.2, there is one stream per input source, using a language model to map inputs to embeddings. Following [Garcia, 2020a], we first attempt question answering on each stream alone. In doing so, we learn a linear classifier while fine-tuning a language model per stream. Unlike most existing works, this allows adapting to the data at hand, for instance, a particular TV show.

In this section, we explain the language model and its inputs by differentiating *scene* from *episode* inputs. In both input cases, the given question and candidate answer strings are denoted as X^q and X_c^a for $c = 1, \ldots, n_c$, respectively, where n_c is the number of candidate answers.

3.5.1 Language model

We use a language model as the backbone of our model architecture to represent text, using two segments at a time. Each segment is composed of one or more sentences. First, two segments are tokenized into X_A and X_B, and fed into text embedding layer f^t,

$$Z := f^t(X_A, X_B), \tag{3.1}$$

where text embeddings $Z \in \mathbb{R}^{d \times s}$, d represents embedding dimension, and s is the sequence length. In the text embedding layer f^t, tokens are represented by WordPiece embeddings [Schuster, 2012; Wu, 2016], concatenated with *position embeddings* representing their position in the input sequence and *segment embeddings*, where segments are defined according to occurrences of the *separator* token [SEP]. Then, text embeddings Z is fed into language model f

$$y := f(Z), \tag{3.2}$$

where the output vector $y \in \mathbb{R}^d$ corresponding to token [CLS] is an *aggregated representation* of the entire input sequence.

3.5.2 Scene input sources

Scene input sources refer to the scene of the question, *i.e.*, *raw scene dialog*, *scene dialog summary* or *video*. The tokenized input is denoted by X^i. For each $c = 1, \ldots, n_c$, we jointly embed X^i, X^q and X^a_c with text embedding layer f^t (3.1) as follows:

$$Z^c := f^t([X^i\, X^q], X^a_c). \tag{3.3}$$

Then, Z^c is fed into the language model f (3.2) to obtain d-dimensional vector

$$y^c := f(Z^c). \tag{3.4}$$

A linear classifier with parameters $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$ yields a score per candidate answer

$$o^c := \mathbf{w}^\top \cdot y^c + b. \tag{3.5}$$

The *score vector* $o := [o^1 \ \ldots \ o^{n_c}]$ is followed by softmax and cross-entropy loss. At training, we use f as pre-trained, and we fine-tune it while optimizing W, b on the correct answers of the QA training set. At inference, we predict $\arg\max_c o^c$.

3.5.3 Episode input sources

Episode input sources refer to the entire episode of the scene of the question, *i.e.*, *episode dialog summary* and *plot*. Because such input is typically longer than the transformer's maximum sequence length s_{max}, we split it into overlapping parts in a *sliding window* fashion. Each

part contains the question and one answer, so the window length is $s_w = s_{max} - |X^q| - |X^a_c|$. Given an input of length s tokens, the number of parts is $n := \left\lceil \frac{s - s_w}{st} \right\rceil + 1$, where st is the *stride*. Because all inputs in a mini-batch must have the same number of parts n_p to be stacked in a tensor, certain parts are zero-padded if $n < n_p$ and discarded if $n > n_p$.

Embedding. The tokenized input of the parts is denoted by X^p_j for $j = 1, \ldots, n_p$. Each part X^p_j is paired with each candidate answer X^a_c separately for a given question X^q, and all inputs are embedded as follows:

$$Z^c_j := f^t([X^p_j \, X^q], X^a_c). \tag{3.6}$$

Each Z^c_j is fed into language model f which yields the d-dimensional vectors

$$y^c_j := f(Z^c_j) \tag{3.7}$$

for $c = 1, \ldots, n_c$ and $j = 1, \ldots, n_p$. A classifier with parameters $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$ yields a score per candidate answer c and part j:

$$o^c_j := \mathbf{w}^\top \cdot y^c_j + b. \tag{3.8}$$

Temporal attention. At this point, unlike scene inputs (3.5), predictions from (3.8) are not meaningful unless a part j is known, which amounts to *temporal localization* of the part of the input sequence that contains the information needed to answer a question. In TVQA [Lei, 2018] and related work [Lei, 2019; Kim, 2020a; Kim, 2020b], localization ground truth is available, allowing a two-stage localize-then-answer approach. Without such information, the problem is *weakly supervised*.

Previous work [Garcia, 2020a] simply chooses the part j corresponding to the maximum score o^c_j over all answers c and all parts j in (3.8), which is called *hard temporal attention* in the following. Such hard decision may be harmful when the chosen j is incorrect, especially when the predicted answer happens to be correct, because then the model may receive arbitrary gradient signals at training. To alleviate this, we follow a *soft temporal attention* approach.

In particular, let O be the $n_p \times n_c$ matrix with elements o^c_j over all answers c and all parts j (3.8). For each part j, we take the maximum score over answers

$$v_j := \max_c o^c_j, \tag{3.9}$$

giving rise to a vector $v := [v_1 \ \dots \ v_{n_p}]$, containing a single best score for each part. Then, by soft assignment over the rows of O—corresponding to parts—we obtain a score for each answer c, represented by *score vector* $o \in \mathbb{R}^c$:

$$o := \text{softmax}(v/\tau)^\top \cdot O, \tag{3.10}$$

where τ is a temperature parameter. With this definition of o, we generate a single score vector, and we proceed as described in (3.5).

3.6 Multi-stream question answering

Once a separate transformer has been fine-tuned separately for each stream, we combine all streams into a single question answering classifier, which amounts to multimodal fusion. Here, we introduce two new simple solutions.

In both cases, we freeze all transformers and obtain d-dimensional embeddings y^c for each candidate answer c and for each stream. For scene inputs, y^c is obtained directly from (3.4). Episode input streams produce n_p embeddings per answer. Temporal localization is thus required for part selection, similar to single stream training. Again, *hard temporal attention* amounts to choosing the part with the highest score according to (3.8): $y^c := y^c_{j^*}$ where $j^* := \arg\max_j(o^c_j)$ and y^c_j is given by (3.7). Instead, similar to (3.10), we follow *soft temporal attention*:

$$y^c := \text{softmax}(v/\tau)^\top \cdot Y^{emb}_c, \tag{3.11}$$

where Y^{emb}_c is a $n_p \times d$ matrix collecting the embeddings y^c_j (3.7) of all parts j. Finally, for each answer c, the embeddings y^c of all streams are stacked into a $n_s \times d$ *embedding matrix* Y_c, where n_s is the number of streams.

Multi-stream attention. The columns of Y_c are embeddings of different streams. We weight them according to weights $w_c \in \mathbb{R}^{n_s}$ obtained from Y_c itself, using a *multi-stream attention* block, consisting of two fully connected layers followed by softmax:

$$Y^{\text{att}}_c = \text{diag}(w_c) \cdot Y_c. \tag{3.12}$$

For each answer c, a fully connected layer maps the $d \times n_s$ matrix Y^{att}_c to a scalar score. All

n_c scores are followed by softmax and cross-entropy loss, whereby the parameters of all layers are jointly optimized.

Self-attention. Alternatively, Y_c is mapped to $Y_c^{\text{att}} \in \mathbb{R}^{d \times n_s}$ by a single *multi-head self-attention* block, as in transformers [Vaswani, 2017]:

$$Y_c^{\text{att}} = \text{MultiHeadAttention}(Y_c, Y_c, Y_c). \tag{3.13}$$

The remaining pipeline is the same as in the previous case.

3.7 Experiments

3.7.1 Experimental setup

Dataset. The KnowIT VQA [Garcia, 2020b] dataset contains 24,282 human-generated questions associated to 12,087 scenes, each of duration 20 seconds, from 207 episodes of *The Big Bang Theory* TV show. Questions are of four types: *visual* (22%), *textual* (12%), *temporal* (4%) and *knowledge* (62%). Question types are only known for the test set. Knowledge questions require reasoning based on knowledge from the episode or the entire TV show, which differs from other video question answering datasets. Questions are multiple-choice with $n_c = 4$ answers per question and performance is measured by *accuracy*, per question type and overall.

Implementation details. For scene dialog summary generation, we set the minimum sequence length to 30 tokens and the maximum to 100 in the BART [Lewis, 2020] model. With this setting, episode dialog summaries are 2078 tokens long on average, while plot summaries are 659 tokens long.

We fine-tune the language model f as $\text{BERT}_{\text{BASE}}$ [Devlin, 2019] uncased model with 12 transformer blocks, 12 self-attention heads and embedding dimension $d = 768$ for single-stream models. The maximum token length s_{max} is 512 for scene, 200 for plot and 300 for episode dialog summary inputs. The stride st is 100 for plot and 200 for episode dialog summary. The maximum number of parts n_p is 10 for both. The batch size is 8 for all single-stream models and 32 for multi-stream. We use SGD with momentum 0.9 scheduled with initial learning rate 10^{-4} for multi-stream fusions. We use 1 attention head, and 2 stacks for self-attention and multi-stream self-attention methods. The number of streams n_s varies per experiment.

METHOD	KNOWLEDGE	VIS.	TEXT.	TEMP.	KNOW.	ALL
Rookies [Garcia, 2020b]	–	0.936	0.932	0.624	0.655	0.748
Masters [Garcia, 2020b]	✓	0.961	0.936	0.857	0.867	0.896
ROCK_{GT} [Garcia, 2020b]	question GT	0.747	**0.819**	0.756	0.708	0.731
$\text{ROLL}_{\text{human}}$ [Garcia, 2020a]	question GT	0.708	0.754	0.570	0.567	0.620
TVQA [Lei, 2018]	–	0.612	0.645	0.547	0.466	0.522
$\text{ROCK}_{\text{facial}}$ [Garcia, 2020b]	dataset GT	0.654	0.688	0.628	0.646	0.652
ROLL [Garcia, 2020a]	plot	0.718	0.739	0.640	0.713	0.715
Ours	–	**0.755**	0.783	**0.779**	**0.789**	**0.781**
$\text{Ours}_{\text{plot}}$	plot	0.749	0.783	0.721	0.783	0.773

Table 3.1 – *State-of-the-art accuracy* on KnowIT VQA. Ours uses the video and scene dialog summary as well as the episode dialog summary that we generate from the dialog of the entire episode. $\text{Ours}_{\text{plot}}$ also uses human-generated plot summaries, like [Garcia, 2020a]. TVQA uses an LSTM based encoder; all other methods use BERT. Rookies and Masters are humans.

3.7.2 Quantitative results

Table 3.1 compares our method with the state of the art. Rookies and Masters are human evaluators: Masters have watched most of the show, whereas Rookies have never watched an episode before [Garcia, 2020b]. TVQA [Lei, 2018] encodes visual features and subtitles without considering knowledge information; its results are as reported in [Garcia, 2020b]. ROCK [Garcia, 2020b] uses four visual representations (image, concepts, facial, caption); $\text{ROCK}_{\text{facial}}$ is one of its best results. ROCK_{GT} [Garcia, 2020b] and $\text{ROLL}_{\text{human}}$ [Garcia, 2020a] use the human knowledge annotation provided by the dataset [Garcia, 2020b], while ROLL [Garcia, 2020a] uses human-written plot summaries instead. Our method uses scene video and scene dialog summary as well as the episode dialog summary that it automatically generates, without any human annotation. $\text{Ours}_{\text{plot}}$ additionally uses the same plot as [Garcia, 2020a]. TVQA uses LSTM; all other methods are based on BERT.

Our method outperforms the best state of the art method (ROLL [Garcia, 2020a]) by 6.6%, without any human annotation. By using additional human-generated plots, the gain decreases to 5.8%. This indicates that our episode dialog summary captures the required knowledge and removes the requirement of human-generated input; in fact, human-generated input is harmful. On temporal and knowledge questions in particular, we gain 13.9% and 7.6%, respectively, without any human annotation. This implies that our automatically generated episode dialog summary increases the understanding of the episode and helps answering all types of questions.

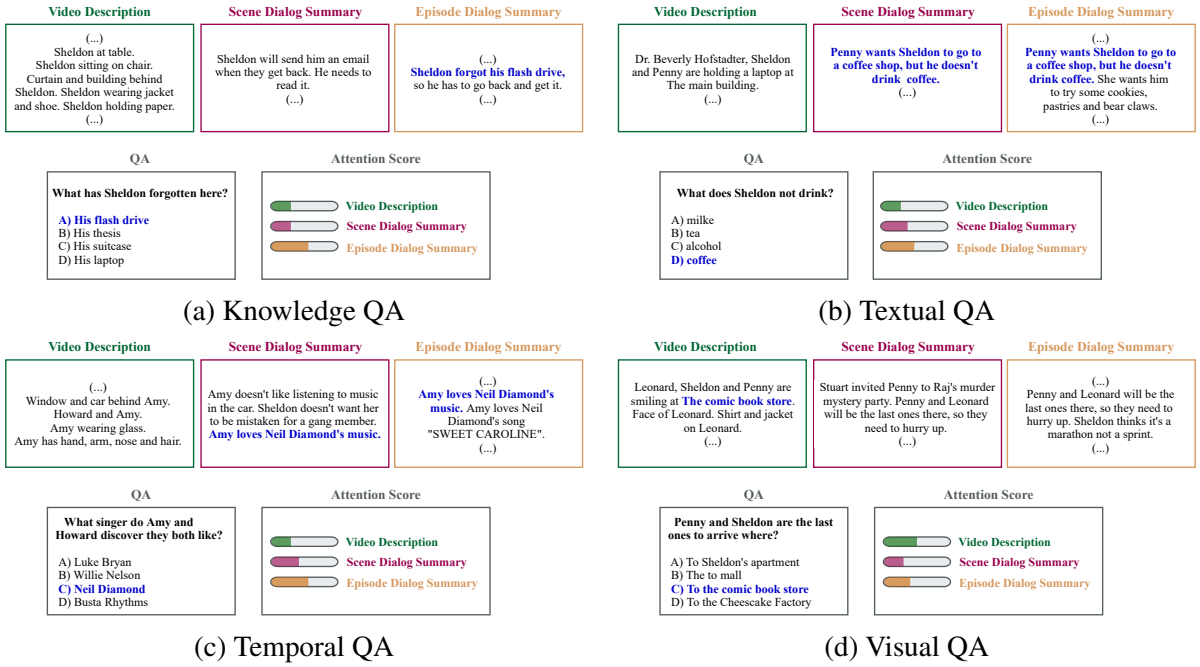


Figure 3.3 – *Multi-stream attention visualization*. We highlight in blue the part of the source text that is relevant to answering the question. The most attended stream is episode dialog summary for (a), (b), (c) and video description for (d).

Despite $\text{ROLL}_{\text{human}}$ [Garcia, 2020a] and ROCK_{GT} [Garcia, 2020b] using ground-truth knowledge, we outperform them by 16.1% and 5.0%, respectively, without any human annotation. We also outperform Rookies, presumably by having access to the dialog of the entire episode. Compared to Masters, there is still room for improvement.

3.7.3 Qualitative analysis

Successful cases. Figure 3.3 visualizes the correct predictions of our method with stream attention scores for different question types. In all examples, the model receives three input sources, question/answers and attention scores over inputs. Figure 3.3(a) shows a *knowledge* question, answered based on episode dialog summary, which has the highest attention score. As shown in Figure 3.3(b), a *textual* question can be answered by using scene dialog summary, but also by episode dialog summary, since the latter includes the former. *Temporal* questions can be answered from scene inputs such as scene dialog summary or video description. According to attention scores, the question in Figure 3.3(c) is answered by episode dialog summary, which includes the correct answer. Finally, Figure 3.3(d) shows a *visual* question answered by video description.

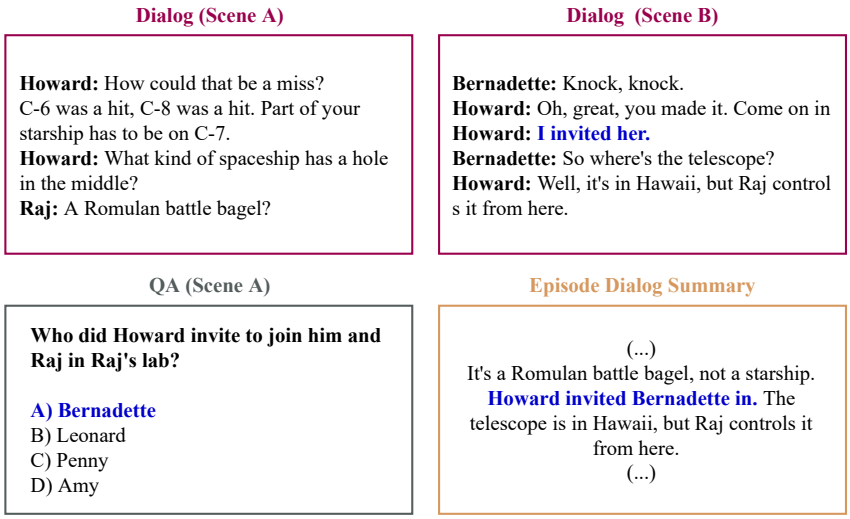


Figure 3.4 – Dialog summarization converts pronouns in dialog to character names in episode dialog summary, supporting question answering. In particular, "I" is substituted by "Howard" and "her" by "Bernadette".

Dialog summarization. In the example of Figure 3.4, Howard says, "I invited *her*." in scene B. Our dialog summarization interprets this sentence by assigning the correct character name: "Howard invited *Bernadette* in." Hence, we can answer the question of scene A, "Who did Howard invite to join him and Raj in Raj's lab?" correctly. Thanks to the episode dialog summary spanning all scenes and the use of character names instead of pronouns, our method can answer character-related questions correctly.

Plot *vs*. episode dialog summary. A comparison of plot summary and episode dialog summary is given in Figure 3.5. There are three different topics in the story line, and each is highlighted with the same color in both summaries. The first topic, highlighted in purple, is "Sheldon's forgotten flash drive." The second, highlighted in yellow, is "Sheldon's grandmother." The third, highlighted in blue, is "Asking Summer out." The plot summary is topic-centered, while the episode dialog summary is following the narrative order. Hence, topics may be fragmented in the latter. The episode dialog summary has more detail than the plot. In particular, it contains enough information to answer the question *Why does Sheldon's grandmother call him Moon Pie?* That is, *because he's nummy-nummy*. This information is missing from the plot summary, which focuses on the main topics/events of an episode. Even though the episode dialog summary is noisy, it contains details that help in question answering.

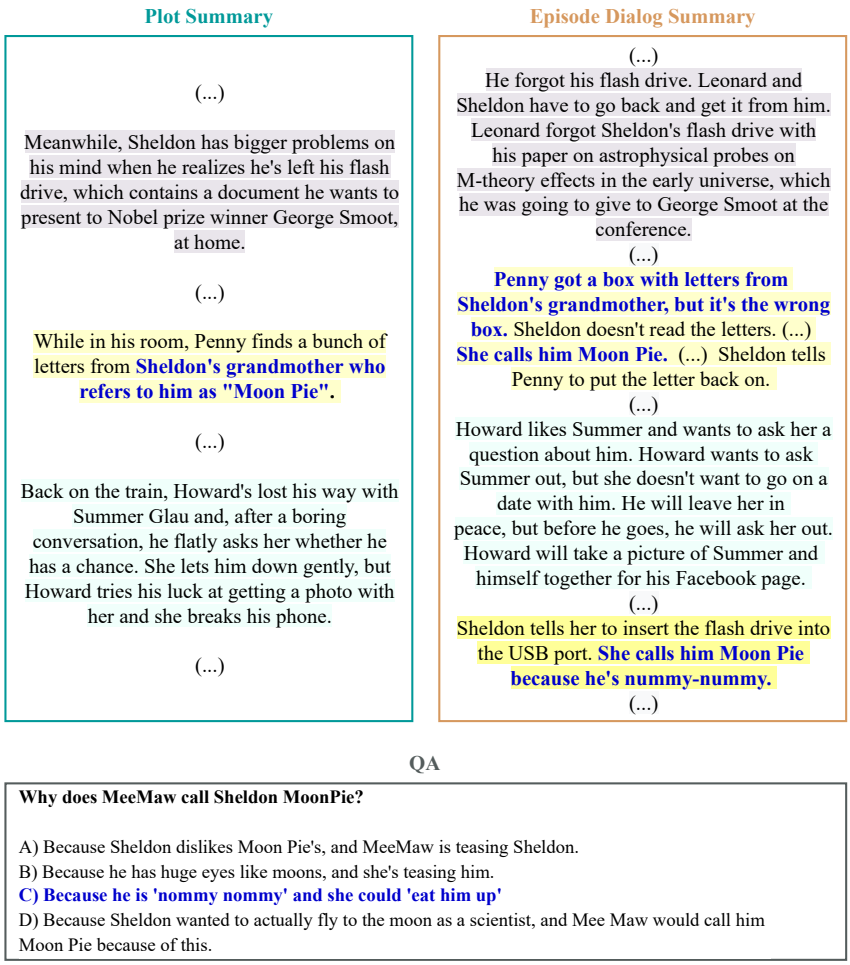
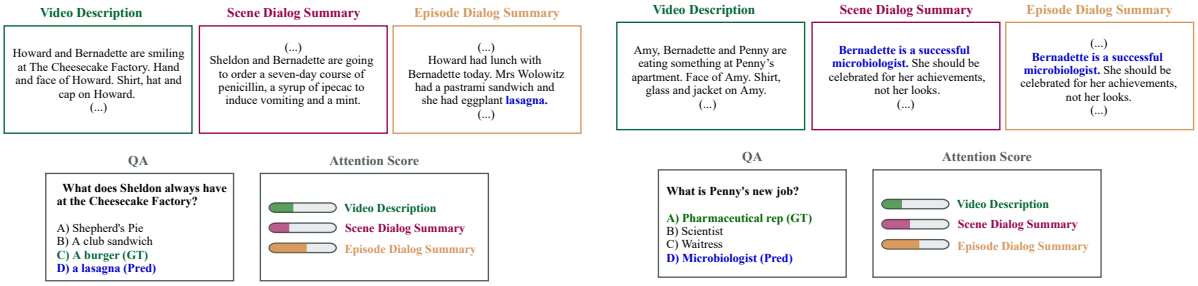


Figure 3.5 – An example of plot summary and episode dialog summary, with each topic highlighted in the same color in both summaries. Phrases relevant to QA in blue. Only the episode dialog summary contains enough information to answer the question.

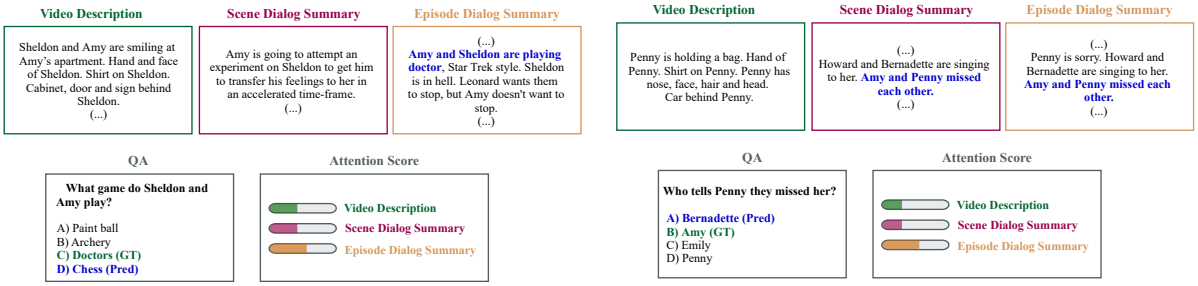
Failure cases. Figure 3.6 shows examples of failed predictions of our model along with stream attention scores for different question types. The model receives three input sources, question/answers and attention scores over inputs.

Figure 3.6(a) refers to a knowledge question, which requires recurrent knowledge of the whole TV show. In other words, the correct answer cannot be found in episode dialog summary. The question is answered as "a lasagna" found in episode dialog summary, even though it is wrong.



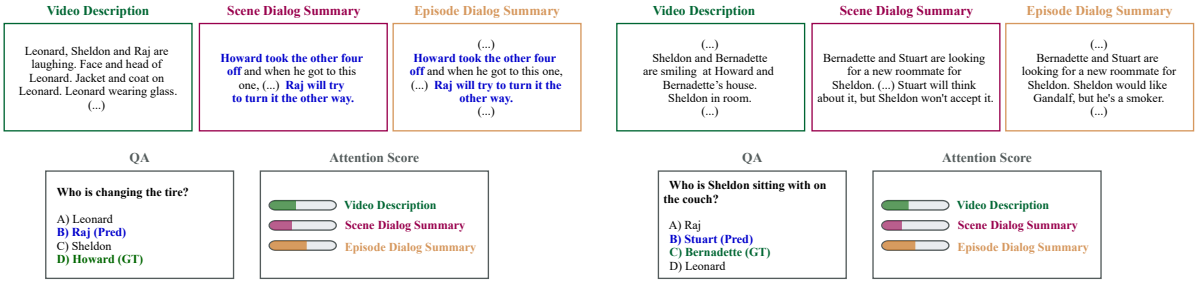
(a) Knowledge QA (1)

(b) Textual QA (1)



(c) Knowledge QA (2)

(d) Textual QA (2)



(e) Temporal QA

(f) Visual QA

Figure 3.6 – *Failed predictions of multi-stream attention*. We highlight in blue the part of the source text that might be relevant to answering the question. "Pred"/blue: model predictions. "GT"/green: ground truth.

Figure 3.6(b) refers to a textual question, which should have been answered by scene dialog summary. However, scene dialog summary does not contain the correct answer. Our model gives most attention to episode dialog summary. The prediction is made according to the highlighted text, which is the same in both sources. However, this prediction refers to the wrong person.

Figure 3.6(c) refers to another knowledge question, which could be answered by the highlighted text in episode dialog summary. Even though episode dialog summary has the most attention, the prediction is incorrect.

METHOD	INPUT	VIS.	TEXT.	TEMP.	KNOW.	ALL
ROLL [Garcia, 2020a]	D	0.656	0.772	0.570	0.525	0.584
	V	0.629	0.424	0.558	0.514	0.530
	P	0.624	0.620	0.570	0.725	0.685
ROLL [Garcia, 2020a]†	D	0.649	**0.801**	0.581	0.543	0.598
	V	0.625	0.431	0.512	0.541	0.546
	P	0.647	0.554	0.674	0.694	0.667
Ours	P	0.666	0.623	0.593	0.735	0.702
	S	0.631	0.746	0.605	0.537	0.585
	E	**0.676**	0.750	**0.779**	**0.785**	**0.756**

Table 3.2 – *Single-stream QA accuracy* on KnowIT VQA. ROLL [Garcia, 2020a]: as reported; [Garcia, 2020a]†: our reproduction. Our model incorporates the scene dialog and video streams of the latter as well as the plot, scene dialog summary and episode dialog summary streams. Plot differs between [Garcia, 2020a]† and our model by our temporal attention and other improvements (Table 3.4). D: dialog; V: video; P: plot; S: scene dialog summary; E: episode dialog summary.

Figure 3.6(d) refers to another textual question, which should have been answered by scene dialog summary. Although both scene dialog summary and episode dialog summary include the correct answer, and episode dialog summary has the most attention, the prediction indicates the wrong person.

Figure 3.6(e) refers to a temporal question. The scene dialog summary and episode dialog summary imply that *Raj* and *Howard* might be changing the tire. The video description is not helpful either. Hence, our model predicts *Raj*, while the correct answer is *Howard*.

Figure 3.6(f) is a visual question. However, the video description fails to convey relevant information to answer the question. The other inputs do not contain relevant information either. One of the character names appearing in episode dialog summary is predicted, which is incorrect.

3.7.4 Ablation studies

Single-stream results. Table 3.2 shows our single-stream QA results. We reproduce [Garcia, 2020a] for dialog, video, and plot inputs. We replace the plot stream by one using our new temporal attention (Subsection 3.5.3) and other improvements (Table 3.4) and we add two new

METHOD	VIS.	TEXT.	TEMP.	KNOW.	ALL
Product	0.743	0.659	0.756	0.751	0.739
Modality weighting [Garcia, 2020a]	0.708	**0.786**	0.767	0.787	0.769
Self-attention	**0.759**	0.764	0.767	0.777	0.771
Multi-stream attention	0.755	0.783	**0.779**	**0.789**	**0.781**
Multi-stream self-attn.	0.755	0.768	0.756	0.777	0.770

Table 3.3 – *Multi-stream QA accuracy* on KnowIT VQA, fusing video, scene dialog summary and episode dialog summary input sources. All fusion methods use soft temporal attention for localization of episode input sources. Top: baseline/competitors. Bottom: ours.

sources automatically generated from dialog: scene dialog summary and episode dialog summary. Due to the dataset having a majority of knowledge questions, episode dialog summary and plot inputs have higher accuracy than other input sources since they span an entire episode. Our episode dialog summary helps in answering questions better than the plot [Garcia, 2020a], bringing an accuracy improvement of 5.4%.

Multi-stream results. We evaluate our two multi-stream QA methods introduced in Section 3.6, namely *multi-stream attention* and *self-attention*, comparing them with the following combinations/baselines/competitors:

1. *Multi-stream self-attention*: combination of multi-stream attention and self-attention: the output of the latter is weighted by the former. The remaining pipeline is the same as in multi-stream attention.
2. *Product*: Hadamard product on embeddings of all streams per answer, followed by a linear classifier per answer. The remaining pipeline is the same.
3. *Modality weighting* [Garcia, 2020a]: a linear classifier (3.5) and loss function is used as in single-stream QA but with transformers frozen for each stream separately. The obtained scores by single-stream classifiers are combined by a multi-stream classifier and another loss function applies. The overall loss all is a linear combination with weight α_ω on the multi-stream loss and $1 - \alpha_\omega$ uniformly distributed over single-stream losses.

Table 3.3 shows results for fusion of video, scene dialog summary and episode dialog summary. For modality weighting, we set $\alpha_\omega = 0.7$ according to the validation set. Our multi-stream attention outperforms other fusion methods. Besides, it does not require tuning of modality

METHOD	VIS.	TEXT.	TEMP.	KNOW.	ALL
ROLL [Garcia, 2020a]†	0.722	0.703	0.709	0.697	0.704
+ Multi-stream attention	0.724	0.721	0.721	0.691	0.703
+ More parts for plot	0.722	0.703	0.651	0.717	0.714
+ New order of plot inputs	0.730	0.710	0.686	0.712	0.715
+ Temporal attention	0.734	0.725	0.663	0.724	0.724
± Replacing P → E	0.753	**0.815**	**0.814**	**0.773**	0.775
± Replacing D → S	**0.755**	0.783	0.779	0.789	**0.781**

Table 3.4 – *Accuracy improvements over* ROLL [Garcia, 2020a]. †: our reproduction. Each row adds a new improvement except the last two, where we replace streams. P: plot; E: episode dialog summary; D: dialog; S: scene dialog summary.

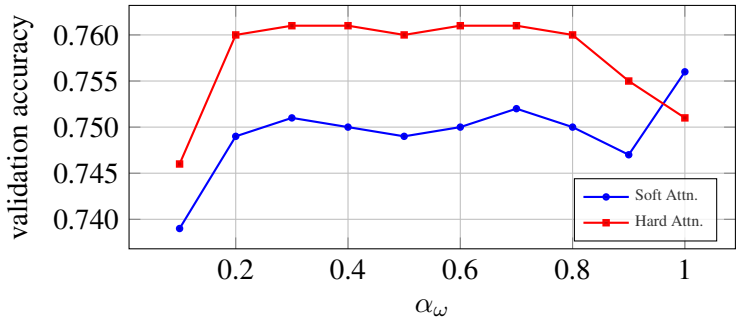


Figure 3.7 – Accuracy *vs.* α_ω for fusion of video, scene dialog summary and episode dialog summary by modality weighting [Garcia, 2020a] on KnowIT VQA validation set.

weight hyperparameter α_ω or selecting the number of heads and blocks for self-attention. Unless specified, we use multi-stream attention for fusion by default.

Improvements over [Garcia, 2020a]. We reproduce ROLL [Garcia, 2020a] using official code by the authors and default parameters. This is our baseline, shown in the first row of Table 3.4. Then, we evaluate our improvements, adding them one at a time. First, we replace modality weighting with *multi-stream attention*. Despite its simplicity, its performance is on par, losing only 0.1%, while requiring no hyperparameter tuning. Then, we increase the *number of parts* of plot summaries from 5 to 10, eliminating information loss by truncation and bringing an accuracy improvement of 1.1%. We change the *order of arguments* of text embedding layer f^t for episode input sources from $f^t(X^q, [X_c^a\ X_j^p])$ to $f^t[(X_j^p\ X^q], X_c^a)$ (3.6), which is consistent with (3.4) and improves only slightly by 0.1%. Our new *temporal attention* mechanism improves accuracy by 0.9%. Replacing plot with episode dialog summary, which is our main contribution, brings an improvement of 5.1%. Finally, the accuracy is improved by 0.6% by us-

ing *scene dialog summary* instead of raw dialog. The overall gain over [Garcia, 2020a] is 7.7%.

Note that the relative improvement of each new idea depends on the order chosen in Table 3.4. For instance, the order of BERT arguments brings improvements of up to 2.3% in experiments including the episode dialog summary.

Hyperparameter validation. *Modality weighting* [Garcia, 2020a] fusion method requires selection of hyperparameter α_ω. Figure 3.7 shows validation accuracy *vs*. α_ω for fusion of video, scene dialog summary and episode dialog summary. We choose $\alpha_\omega = 0.7$ for both soft and hard temporal selection to report results in Table 3.3 and Table 3.6. The remaining weight of $1 - \alpha_\omega$ is evenly distributed over individual stream losses as 0.1 per stream.

Effect of temporal attention on single-stream QA. We investigate the effect of our soft temporal attention (Subsection 3.5.3) on single-stream QA for episode input sources. We also evaluate the effect of single-stream training with soft or hard temporal attention on multi-stream attention, where we use soft temporal attention. According to Table 3.5, temporal attention improves the accuracy of plot and episode dialog summary by 1.9% and 3.3%, respectively. Accordingly, the accuracy of multi-stream QA on the same episode sources as well as video and scene dialog summary increases by 0.6% and 7.0%, respectively. The gain is higher when episode dialog summary is used, since the episode dialog summary is longer than plot.

Effect of temporal attention on multi-stream QA. Table 3.6 shows the effect of *soft temporal attention* on multi-stream QA for fusion of video, scene dialog summary and episode dialog summary input sources. We use soft temporal attention for single-stream QA of episode dialog summary. In all fusion methods, the overall accuracy is improved by using soft temporal attention.

Different input combinations. Table 3.7 shows the accuracy of multi-stream QA for different input combinations, where the number of input streams varies in $\{2, 3, 4, 5\}$. Scene dialog summaries improves the accuracy compared with single-stream QA results in Table 3.2. Moreover, using the episode dialog summary always improves the overall accuracy by a large margin. The best overall accuracy of 0.781 is achieved by video, scene dialog summary, and episode dialog summary.

Stream	Inputs	Soft Attn.	Vis.	Text.	Temp.	Know.	All
Single	P	-	0.656	0.594	0.628	0.712	0.683
	P	✓	0.666	0.623	0.593	0.735	0.702
	E	-	0.604	0.721	0.733	0.765	0.723
	E	✓	**0.676**	**0.750**	**0.779**	**0.785**	**0.756**
Multi	V + S + P	-	0.732	0.688	0.674	0.720	0.717
	V + S + P	✓	0.739	0.699	0.628	0.728	0.723
	V + S + E	-	0.707	0.772	0.721	0.700	0.711
	V + S + E	✓	**0.755**	**0.783**	**0.779**	**0.789**	**0.781**

Table 3.5 – *Effect of temporal attention on single-stream QA* on KnowIT VQA. Soft Attn.: soft temporal attention on single-stream training. We use soft temporal attention for multi-stream QA, but this is still affected by the temporal attention used in single-stream training. V: video; S: scene dialog summary; P: plot; E: episode dialog summary.

Method	Soft Attn.	Vis.	Text.	Temp.	Know.	All
Product	-	0.728	0.645	0.744	0.756	0.736
	✓	0.743	0.659	0.756	0.751	0.739
Modality weighting [Garcia, 2020a]	-	0.716	**0.815**	0.791	0.776	0.768
	✓	0.708	0.786	0.767	0.787	0.769
Self-attention	-	0.753	0.804	**0.802**	0.766	0.769
	✓	**0.759**	0.764	0.767	0.777	0.771
Multi-steam attention	-	0.743	0.790	0.779	0.785	0.776
	✓	0.755	0.783	0.779	**0.789**	**0.781**
Multi-steam self attn.	-	0.749	0.797	0.791	0.768	0.768
	✓	0.755	0.768	0.756	0.777	0.770

Table 3.6 – *Effect of temporal attention on multi-stream QA* on KnowIT VQA for fusion of video, scene dialog summary, and episode dialog summary input sources. Soft Attn.: soft temporal attention on multi-stream training. We use soft temporal attention for single-stream QA of episode dialog summary.

Question type $\leftrightarrow$ attention scores. We perform significance testing for the dependence between the question type and attention scores. There are 2 independent variables in the scores of 3 streams, whose values we discretize into 10×10 bins. We form a $4 \times 10 \times 10$ joint histogram of question type (X) and scores (Y) and compute the mutual information $I(X;Y)$. We perform

ANALYSED INPUTS	INPUTS	VIS.	TEXT.	TEMP.	KNOW.	ALL
D V P	D+V	0.693	0.768	0.593	0.554	0.611
	D+P	0.732	0.721	0.674	0.723	0.723
	D+V+P	0.734	0.725	0.663	0.724	0.724
D V P S	D+S	0.664	0.786	0.628	0.548	0.604
	V+S	0.689	0.721	0.581	0.549	0.601
	P+S	0.716	0.710	0.628	0.727	0.719
	D+V+P+S	0.734	0.732	0.663	0.725	0.726
D V P S E	D+E	0.743	0.812	0.779	0.779	0.775
	V+E	0.732	0.761	0.767	0.788	0.772
	P+E	0.716	0.743	0.721	**0.791**	0.766
	D+S+E	0.743	**0.822**	**0.802**	0.771	0.772
	V+S+E	**0.755**	0.783	0.779	0.789	**0.781**
	P+S+E	0.739	0.779	0.733	0.783	0.771
	D+V+P+S+E	0.751	0.797	0.744	0.781	0.775

Table 3.7 – *Multi-stream QA accuracy* on KnowIT VQA: comparison of different input combinations for multi-stream attention. D: dialog; V: video; P: plot; S: scene dialog summary; E: episode dialog summary.

a G-test[2] with $G = 2n_q \cdot I(X;Y)$, where $n_q = 2361$ is the number of test questions. Finally, using a chi-square distribution of $3 \times 9 \times 9$ DoF, we find a p-value of 1.52×10^{-25} for the null hypothesis. This indicates that attention scores depend on question type.

Replacing attention scores with oracle scores determined by question type. Assuming that we know the question type for the test set, we perform an *oracle* experiment where attention scores are based on question type rather than our fusion method. We only consider visual, textual, and knowledge types of question. In particular, we assign visual questions to video input, textual questions to scene dialog summary and knowledge questions to episode dialog summary. We exclude temporal questions since they can be answerable by scene dialog summary or video. Only 3.6% of questions are of temporal type in the test set. We find that our multi-stream attention method (0.781%) is 3.6% better than the oracle experiment (0.745%). This indicates that our fusion mechanism is more effective than a naïve oracle that assumes more knowledge.

2. https://en.wikipedia.org/wiki/G-test

3.8 Conclusion

KnowIT VQA is a challenging dataset where it was previously believed that some form of external knowledge was needed to handle knowledge questions, as if knowledge was yet another modality. Our results indicate that much of this required knowledge was hiding in *dialog*, waiting to be harnessed. It is also interesting that our *soft temporal attention* helps a lot more with our episode dialog summary than human plot summary, which may be due to the episode dialog summary being longer. This may also explain the astounding performance of episode dialog summary, despite its low overall quality: plot summaries are of much higher quality but may be missing a lot of information.

CHAPTER 4

ZERO-SHOT AND FEW-SHOT VIDEO QUESTION ANSWERING

Contents

This chapter focuses on video question answering with less supervision, particularly for zero-shot and few-shot scenarios. Our objective is to leverage the advantages of large-scale models, as recent developments in vision-language models are mainly driven by such models. However, adapting pre-trained models on limited data presents challenges such as overfitting, catastrophic forgetting, and the cross-modal gap between vision and language. We introduce a parameter-efficient method to address these challenges, combining multimodal prompt learning and a transformer-based mapping network, while keeping the pre-trained models frozen.

The content of this chapter corresponds to our ICCV Workshops 2023 publication, *Zero-Shot and Few-Shot Video Question Answering with Multi-Modal Prompts* [Engin, 2023b].

4.1 Introduction

Recent vision-language models have shown remarkable progress, driven by transformer-based *large-scale pre-trained models* [Dosovitskiy, 2021; Liu, 2022b; Devlin, 2019; Liu, 2019a; He, 2021; Radford, 2019; Radford, 2021]. These models have been incorporated into video understanding methods, including VideoQA, through multimodal fusion on *large-scale multi-modal datasets* [Miech, 2019b; Bain, 2021; Zellers, 2021]. However, adapting pre-trained models to video-language tasks on limited data is challenging. This is because of the gap between the visual and language modalities and, more importantly, because fine-tuning the entire model on limited data can lead to overfitting and forgetting previously acquired knowledge.

To address the gap between modalities, transformer-based mapping networks have been employed between frozen vision and language models [Mokady, 2021; Han, 2023; Alayrac, 2022]. These networks map visual features to an appropriate embedding space before they are given as input to the language models. To address overfitting, parameter-efficient adaptation methods have been explored, *e.g.*, *prompt learning* [Li, 2021; Liu, 2021a; Liu, 2022a] and *adapter layers* [Houlsby, 2019] on frozen pre-trained models. These approaches preserve the generalization of large-scale models while reducing the number of trainable parameters.

In this work, we investigate the adaptation of large-scale vision and language models to VideoQA under a scarcity of training data. Inspired by FrozenBiLM [Yang, 2022b], we incorporate visual inputs to a frozen language model using lightweight learnable adapter layers. Beyond that, we introduce a novel *visual mapping network* that summarizes the video input while allowing for temporal interaction, inspired by [Mokady, 2021; Jaegle, 2021]. In addition, we introduce *multimodal prompt learning*, which diminishes the number of stored parameters for the few-shot scenario. We call our model ***Vi**deoQA with Mul**ti**-Modal Prompt**s*** (ViTiS).

We pre-train trainable parameters of ViTiS, *i.e. visual mapping network, adapter layers, visual and text prompts*, under the *MLM* objective on video-text pairs collected from the web, while the vision and language models are kept frozen. We evaluate ViTiS in the zero-shot and few-shot settings. For the latter, we fine-tune the model on downstream VideoQA tasks, using two approaches: (i) fine-tuning all trainable parameters, which are 8% of the total model parameters, (ii) fine-tuning only the prompts, which are 0.8% of all trainable parameters and a mere 0.06% of the total model parameters.

Our extensive experimental results on multiple open-ended VideoQA datasets demonstrate that ViTiS outperforms prior methods while requiring fine-tuning of only a few parameters for each dataset in few-shot settings. In addition, our visual mapping network contributes to better alignment and understanding of multimodal inputs, improving performance in both zero-shot and few-shot settings.

Our contributions can be summarized as follows:

1. We introduce *multimodal prompt learning* to few-shot VideoQA for the first time, fine-tuning as low as 0.06% of model parameters on downstream tasks.
2. We introduce a *visual mapping network* to VideoQA, mapping video input to the text embedding space, while supporting temporal interaction.
3. We experimentally demonstrate the strong performance of ViTiS on multiple VideoQA datasets in both zero-shot and few-shot settings.

4.2 Related work

Video question answering. Recent advances in vision-language models benefit from pre-trained foundation models, including vision-only [Dosovitskiy, 2021; Liu, 2022b] language-only [Devlin, 2019; Liu, 2019a; He, 2021; Radford, 2019] and vision-language [Radford, 2021]. Recent video understanding methods, including VideoQA, incorporate these models by leveraging large-scale multimodal data [Miech, 2019b; Bain, 2021; Zellers, 2021] with different pre-training objectives, *e.g.*, *masked language modeling*, *masked image modeling*, or *predicting the next word*, to perform single or multiple vision-language tasks [Sun, 2019; Li, 2020; Lei, 2021; Fu, 2021; Yang, 2021; Zellers, 2021; Li, 2022a; Yang, 2022b; Alayrac, 2022; Zellers, 2022; Cheng, 2023; Wang, 2023; Li, 2023b; Huang, 2023; Fu, 2023].

Adapting pre-trained vision-language models to downstream tasks relies on fully supervised fine-tuning on VideoQA datasets in general [Tapaswi, 2016b; Xu, 2017; Jang, 2017; Lei, 2018; Yu, 2019; Li, 2020; Garcia, 2020b]. Few recent works address the challenge of limited data by focusing on zero-shot [Yang, 2021; Yang, 2022a; Yang, 2022b; Alayrac, 2022; Zellers, 2022; Li, 2022b; Li, 2023b] and few-shot [Yang, 2022b; Alayrac, 2022] open-ended VideoQA tasks. Our work is similar to [Yang, 2022b] in leveraging a frozen video encoder and language model

with adapter layers. Beyond that, we introduce a transformer-based visual mapping network between the two models, allowing for temporal interaction. In addition, we incorporate multimodal prompt learning, allowing for efficient fine-tuning in few-shot settings.

Parameter-efficient training. As the size of large-scale pre-trained models grows, adapting them efficiently on limited data without overfitting in an emerging research problem. A common solution is *adapters*, introduced by [Houlsby, 2019] and employed for vision-language tasks [Eichenberg, 2022; Yang, 2022b; Sung, 2022].

Another common solution is *prompting*, referring to inserting tokens to the input to guide pre-trained models on downstream tasks. Prompts can be handcrafted (discrete) [Brown, 2020a] or learned (continuous vectors) [Li, 2021]. Pre-trained language models demonstrate remarkable generalization to zero-shot settings with handcrafted prompts [Brown, 2020a]. Prompt learning is introduced initially in natural language processing tasks [Li, 2021; Lester, 2021; Liu, 2021a; Liu, 2022a; Qin, 2021; Mahabadi, 2022] and subsequently adopted in vision [Jia, 2022; Bahng, 2022] and vision-language models. In the latter case, prompts are introduced to text encoders [Zhou, 2022c; Zhou, 2022b], or both text and vision encoders [Khattak, 2023; Wasim, 2023; Lee, 2023; Rasheed, 2023], called *multimodal*. Learnable prompts can be inserted at the input level [Li, 2021] and/or deep layers [Liu, 2022a; Jia, 2022]. Few recent works employ prompt learning for video understanding [Ju, 2022; Zhu, 2022; Sung, 2022] and multimodal prompt learning for video classification [Wasim, 2023; Rasheed, 2023]. We introduce multimodal prompt learning to few-shot VideoQA for the first time.

4.3 Method

The proposed method, ViTiS, is illustrated in Figure 4.1, consisting of a frozen video encoder, a visual mapping network, a frozen text embedding layer and a frozen language model that includes learnable text prompts and adapter layers. Given an input video X^v, represented as a sequence of frames, and a question X^q, represented as a sequence of tokens, the problem is to predict an answer X^a that is another sequence of tokens. The model takes the concatenated sequence $X^t = (X^q, X^a)$ as input text. Parts of X^t may be masked, for example X^a is masked at inference.

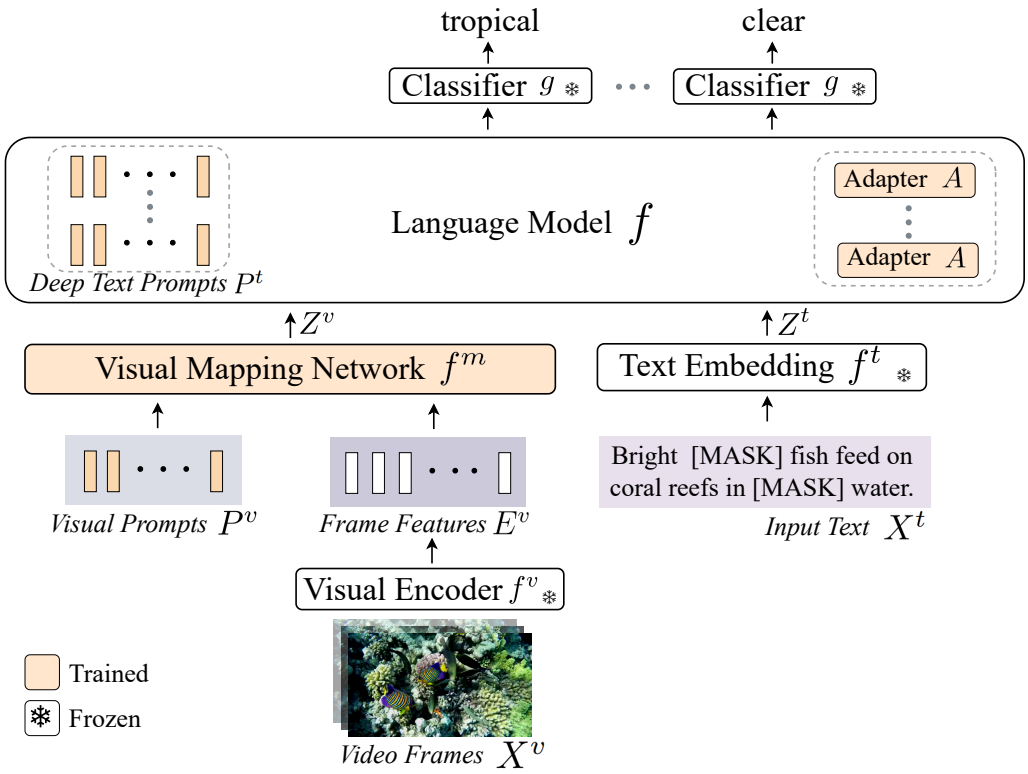


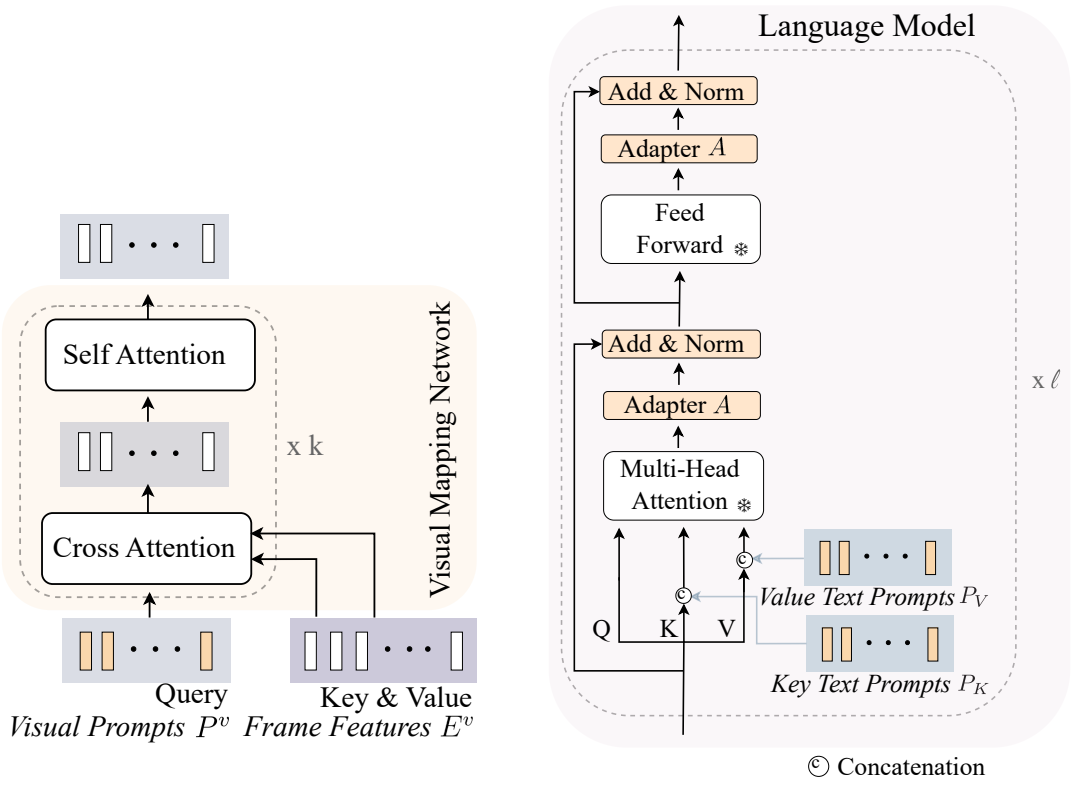
Figure 4.1 – ViTiS consists of a frozen video encoder f^v, a visual mapping network f^m, a frozen text embedding layer f^t, a frozen language model f and a frozen classifier head g. Given input video frames X^v and text X^t, f^v extracts frame features and f^m maps them to the same space as the text embeddings obtained by f^t. Then, f takes the video and text embeddings Z^v, Z^t as input and predicts the masked input tokens. Visual mapping network f^m and language model f are further detailed in Figure 4.2.

Video encoder. The input video is represented by a sequence of c frames, $X^v = [x_1^v \ \dots \ x_c^v]$. This sequence is encoded into the *frame features*

$$E^v := f^v(X^v) = [e_1^v \ \dots \ e_c^v] \in \mathbb{R}^{d \times c} \tag{4.1}$$

by a frozen pre-trained *video encoder* f^v, where d is the embedding dimension.

Visual mapping network. A *visual mapping network* f^m maps the frame features E^v to the same space as the text embeddings. The mapping is facilitated by a set of u *learnable visual*



(a) visual mapping network (b) language model

Figure 4.2 – Our visual mapping network f^m and our language model f are illustrated in detail, following their initial introduction in the method overview (Figure 4.1). (a) Our *visual mapping network* consists of a number of layers, each performing cross-attention between learnable visual prompts and video frame features followed by self-attention. (b) The *language model* incorporates learnable text prompts in the key and value of multi-head-attention and adapter layers after each self-attention and feed-forward layer, before LayerNorm.

prompts $P^v \in \mathbb{R}^{d \times u}$, which are given as input along with E^v, to obtain the *video embeddings*

$$Z^v := f^m(P^v, E^v) \in \mathbb{R}^{d \times u}. \tag{4.2}$$

As shown in Figure 4.2(a), the architecture of f^m is based on Perceiver [Jaegle, 2021], where the latent array corresponds to our learnable visual prompts P^v. It consists of k blocks defined as

$$Z_i := \mathrm{SA}_i(\mathrm{CA}_i(Z_{i-1}, E^v)) \in \mathbb{R}^{d \times u} \tag{4.3}$$

for $i = 1, \ldots, k$, with input $Z_0 = P^v$. Each block i maps the latent vectors Z_{i-1} first by cross attention CA_i with the frame features E^v and then by self-attention SA_i to obtain Z_i. In cross

attention, Z_{i-1} serves as query and E^v as key and value. We thus iteratively extract information from the frame features E^v into the latent vectors, which are initialized by the visual prompts. The output video embeddings are $Z^v = Z_k \in \mathbb{R}^{d \times u}$. To allow modeling of temporal relations within the video, learnable *temporal position embeddings* are added to E^v before f^m.

Text embedding. The input text is tokenized into a sequence of s_t tokens, represented as $X^t = [x_1^t \ \dots \ x_{s_t}^t]$, where X^t is the concatenated sequence of question X^q and answer X^a. This sequence is mapped by a frozen *text embedding layer* f^t to the text embedding space,

$$Z^t := f^t(X^t) = [z_1^t \ \dots \ z_{s_t}^t] \in \mathbb{R}^{d \times s_t}. \tag{4.4}$$

One or more tokens are masked, in which case they are replaced by a learnable mask token.

Language model. We concatenate video and text embeddings into a single input sequence $[Z^v Z^t] \in \mathbb{R}^{d \times s}$ of length $s = u + s_t$. We then feed this sequence to a transformer-based bidirectional language model f to obtain an output sequence

$$f([Z^v \, Z^t]) \in \mathbb{R}^{d \times s} \tag{4.5}$$

of the same length. Finally, a classifier head $g : \mathbb{R}^{d \times s} \to \mathbb{R}^{|U|}$ maps the output sequence to logit vectors over a vocabulary U. The logit vectors corresponding to masked tokens are selected to apply the loss function at training or make predictions at inference. Both f and g are pre-trained and kept frozen. However, as shown in Figure 4.2(b), f is adapted by means of learnable deep text prompts and adapters, described next.

Text prompts. To reduce the number of fine-tuned parameters at downstream tasks, we introduce attention-level text prompts in self-attention blocks at each layer of the language model, referred to as *deep text prompt learning* [Liu, 2022a]. Given a sequence $Z \in \mathbb{R}^{d \times s}$ of token embeddings as input to a self-attention layer of the language model f, we prepend two sequences of *learnable text prompts* $P_K, P_V \in \mathbb{R}^{r \times d}$ to the key and value respectively:

$$Q := W_Q Z \quad K := [P_K \ \ W_K Z] \quad V := [P_V \ \ W_V Z], \tag{4.6}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$ are the query, key and value projections respectively. The output sequence length does not change since it is determined by the query, where we do not prepend

prompts. There is one pair of variables P_K, P_V for each layer of f, collectively denoted as P^t. These variables are either defined as parameters directly or parametrized by means of projections as discussed next.

Text prompt parametrization. Instead of defining text prompts as parameters directly, we discuss here an alternative parametrization using projections. We first generate a sequence of input prompts $P^i \in \mathbb{R}^{d' \times r}$ and then we project it as follows:

$$P^t := W P^i \in \mathbb{R}^{2\ell d \times r}, \tag{4.7}$$

where $W \in \mathbb{R}^{2\ell d \times d'}$, ℓ is the number of layers of the language model f and d its embedding dimension. Then, P^t can be reshaped as a $2 \times \ell \times d \times r$ tensor, representing one pair of sequences $P_K, P_V \in \mathbb{R}^{d \times r}$ for every layer of f. After training, the input sequence P^i and projection matrix W are discarded and we only keep P^t. This allows us to fine-tune fewer parameters at downstream tasks, which is beneficial when data is limited.

Adapters. Following [Yang, 2022b], we add adapter layers to the language model f. Given a sequence $Z \in \mathbb{R}^{d \times s}$ of token embeddings, an adapter layer A maps it through a bottleneck dimension d with a residual connection:

$$A(Z) := Z + W_2 h(W_1 Z) \in \mathbb{R}^{d \times s}, \tag{4.8}$$

where $W_1 \in \mathbb{R}^{d'' \times d}$, $W_2 \in \mathbb{R}^{d \times d''}$, and h is the relu activation function. We insert an adapter module after the self-attention layer and the feed-forward layer, preceding LayerNorm in each layer of f.

Training and inference. Our model is trained using the *masked language modeling* (MLM) objective, where one or more tokens of X^t are masked and the corresponding outputs are predicted over a vocabulary U. The parameters of the visual encoder f^v, text embedding layer f^t, language model f and classifier head g are pre-trained and kept frozen. Only the newly introduced parameters, that is, visual prompts P^v, visual mapping network f^v, text prompts P^t and adapter layers, are optimized on video-text pairs. We then fine-tune these parameters or a smaller subset on downstream video question answering tasks, where $X^t = (X^q, X^a)$ consists of a question-answer pair and masking applies to the X^a only. At inference, X^a is masked and the corresponding output yields a prediction.

Dataset	Videos	QA Pairs			
		Train	Val	Test	Total
MSRVTT-QA [Xu, 2017]	10k	159k	12k	73k	244k
MSVD-QA [Xu, 2017]	2k	31k	6.5k	13k	50.5k
ActivityNet-QA [Yu, 2019]	5.8k	32k	18k	8k	58k
TGIF-QA [Jang, 2017]	40k	39k	–	13k	53k

Table 4.1 – Downstream dataset statistics.

4.4 Experiments

4.4.1 Datasets

Pre-training. We use WebVid2M [Bain, 2021] for pre-training, consisting of 2.5M video-caption pairs scraped from the internet. The domain is open and the captions are manually generated. The average video duration is 18 seconds and the average caption word count is 12.

Downstream tasks. Downstream dataset statistics are given in Table 4.1. Following [Yang, 2022b], we use 1% of the training data for fine-tuning in the few-shot setting. MSRVTT-QA [Xu, 2017] is an extension of MSR-VTT [Xu, 2016], where question-answer pairs are automatically generated from video descriptions. MSVD-QA [Xu, 2017] is based on MSVD [Chen, 2011] and question-answers pairs are automatically generated as in MSRVTT-QA. ActivityNet-QA [Yu, 2019] is derived from ActivityNet [Caba Heilbron, 2015]. The average video duration is 180s. TGIF-QA [Jang, 2017] comprises several tasks, including FRAME-QA, where the question can be answered from one of the frames in a GIF. In this work, TGIF-QA refers only to Frame-QA.

4.4.2 Implementation details

Architecture details. The *frozen video encoder* is CLIP ViT-L/14 [Dosovitskiy, 2021; Radford, 2021], trained with contrastive loss on 400M image-text pairs. We uniformly sample $c = 10$ frames located at least 1 second apart and each frame is resized to 224×224 pixels; if the video is shorter than 10 seconds, we zero-pad up to $c = 10$ frames. The encoder then extracts one feature vector per frame of the dimension of 768, followed by a linear projection to $d = 1536$ dimensions.

The *visual mapping network* has $k = 2$ layers, each with a cross-attention and a self-attention, having 8 heads and embedding dimension $d = 1536$. We use $u = 10$ learnable visual prompt vectors of dimension $d = 1536$.

The *text tokenizer* is based on SentencePiece [Kudo, 2018] with a vocabulary U of size 128k.

The *frozen language model* is DeBERTa-V2-XLarge [He, 2021], trained using MLM on 160G text data, following [Yang, 2022b]. The model has $\ell = 24$ layers, 24 attention heads, and embedding dimension $d = 1536$, resulting in 900M parameters.

For the *adapter layers* [Houlsby, 2019], we set $d'' = d/8 = 192$ by following [Yang, 2022b].

For *text prompts*, we use $r = 10$ learnable text prompt vectors, $d' = d/8 = 192$, and $\ell = 24$.

Downstream input design. We limit the length of text sequences to $s_t = 256$ tokens for pre-training and zero-shot experiments and $s_t = 128$ tokens for downstream experiments. We adopt the input design of [Yang, 2022b] as follows: "[CLS] Question: <Question>? Answer: [MASK]. Subtitles: <Subtitles> [SEP]". Subtitles are optional and if available, their token sequence X^s is incorporated into the input. In this case, the text input sequence becomes $X^t = (X^q, X^a, X^s)$.

Answer vocabulary. The answer vocabulary U is constructed by selecting the top 1k most frequent answers from the training set for the zero-shot setting, following [Yang, 2022b; Zellers, 2021]. Another vocabulary is formed by including answers that occur at least twice in the training set for the few-shot setting, as defined in [Yang, 2022b]. Questions with answers outside the vocabulary are excluded from the training process and are assessed as incorrect during evaluation. To report results for the few-shot setting, we choose the vocabulary that yields the best performance on the validation set.

Answer embedding. The classifier head of the frozen language model includes more tokens than required for downstream training. To address this, by following [Yang, 2022b], we define a task-specific classification head by keeping the weights of the pre-trained head associated with the answer vocabulary. At inference, we provide one mask token at the input, regardless of the ground truth answer length, and we obtain one output logit vector. For multi-token answers, we

#	AD	MAP	PR	TRAINED PARAM	MSRVTT -QA	MSVD -QA	ANET -QA	TGIF -QA
1		Linear		1M	18.0	30.5	27.1	44.4
2		Linear	✓	15M	36.3	46.2	32.7	54.3
3	✓	Linear		30M	35.0	45.0	32.4	53.9
4	✓	Linear	✓	44M	36.4	47.2	32.9	54.7
5		VPN		58M	24.5	37.0	26.1	50.1
6		VPN	✓	72M	36.1	47.4	34.1	55.8
7	✓	VPN		86M	34.7	46.0	32.4	54.4
8	✓	VPN	✓	101M	**36.5**	**47.8**	**37.2**	**55.9**

Table 4.2 – Effect of our proposed components on few-shot top-1 accuracy on the validation set. Pre-training on WebVid2M [Bain, 2021] followed by fine-tuning all trainable parameters on downstream datasets, using 1% of training data. AD: Adapters; MAP: mapping network; PR: text prompts; VPN: our visual mapping network. ANET-QA: ActivityNet-QA.

take the average of the logits corresponding to the ground truth words from the vocabulary.

Evaluation Metrics. We report top-1 accuracy on public test sets for all downstream tasks, except TGIF-QA where we report on the validation set unless otherwise specified.

Training settings. We use the Adam optimizer [Kingma, 2015] with $\beta = (0.9, 0.95)$ in all experiments. We decay the learning rate using a linear schedule with the warm-up in the first 10% of the iterations. We use dropout with probability 0.1 in the language model, adapter layers, text prompts, and visual mapping network. We adopt automatic mixed precision training for all experiments.

We *pre-train* for 10 epochs on WebVid2M with a batch size of 128 on 8 NVIDIA Tesla V100 GPUs, amounting to 20 hours total training time. The base learning rate is 2×10^{-5} and the learning rate for visual and text prompts is separately set to 10^{-3}.

For *fine-tuning* on each downstream dataset, we train for 20 epochs with a batch size of 32 on 4 NVIDIA Tesla V100 GPUs. The base learning rate is searched over 5 values in the interval $[10^{-5}, 5 \times 10^{-5}]$, while the learning rate for visual and text prompts is kept at 10^{-3}. For *prompt-only fine-tuning*, the base learning rate is searched over 3 values in the interval $[10^{-2}, 3 \times 10^{-2}]$.

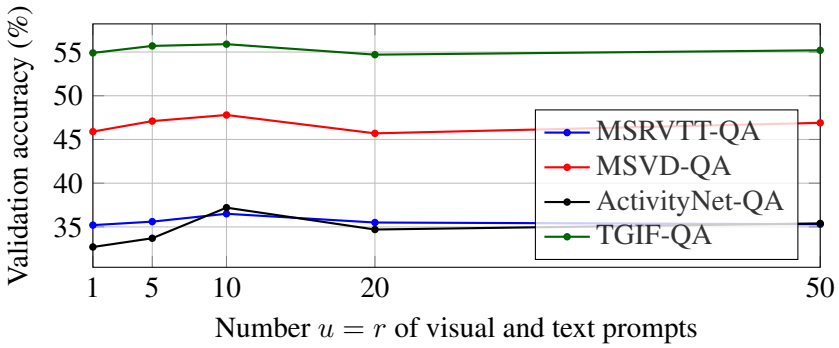


Figure 4.3 – Few-shot top-1 validation accuracy *vs*. number $u = r$ of *visual and text prompts* for different downstream datasets, using 1% of training data.

4.4.3 Ablation

We conduct an extensive ablation study to analyze the proposed method.

Model design. In Table 4.2, we analyze the effect of different components in the model design. We observe that changing the baseline from a linear layer to *our visual mapping network* without adapters increases the performance by a large margin in most datasets (row 1→5). By adding *text prompts* to any model design (row 1→2, 3→4, 5→6, 7→8), the performance increases for all datasets. The improvement is vast in the absence of adapters.

The model design that includes a linear mapping network and adapter layers (row 3) corresponds to FrozenBiLM [Yang, 2022b] trained on WebVid2M. While using only our visual mapping network and text prompts (row 6) already works better than FrozenBiLM trained on WebVid2M, we further improve performance by incorporating adapter layers: our full model (row 8) achieves the best performance overall.

Prompt length. Figure 4.3 shows the effect of the number of prompts on few-shot performance, referring to both visual (u) and text (r) prompts, *i.e.*, $u = r$. Because the space and time complexity of the model is quadratic in the number of prompts, we limit this number to 50. We find that accuracy is consistently best on all downstream benchmarks for $u = r = 10$ prompts, which we choose as default.

VPN Layers	MSRVTT -QA	MSVD -QA	ANET -QA	TGIF -QA
1	36.0	47.0	36.1	55.9
2	**36.5**	**47.8**	**37.2**	**55.9**

Table 4.3 – Effect of number k of layers of our visual mapping network on few-shot top-1 validation accuracy, using 1% of training data. VPN: Visual Mapping Network. ANET-QA: ActivityNet-QA.

REPARAM	MSRVTT -QA	MSVD -QA	ANET -QA	TGIF -QA
	35.6	47.4	34.0	55.1
✓	**36.5**	**47.8**	**37.2**	**55.9**

Table 4.4 – Effect of reparametrization of text prompts on few-shot top-1 validation accuracy, using 1% of training data. REPARAM: Reparametrization. ANET-QA: ActivityNet-QA.

Number of layers of visual mapping network. Table 4.3 shows the effect of the number k of layers of our visual mapping network on few-shot performance. We only experiment with up to 2 layers to avoid an excessive number of parameters and complexity of our model. We find that $k = 2$ works best, which we choose as default.

Reparametrization of text prompts. In Table 4.4, we investigate the impact of the reparametrization of text prompts on few-shot performance. We find that reparametrization consistently improves performance on all downstream benchmarks. Even though the number of trainable parameters increases from 87M to 101M during pre-training and fine-tuning, we do not need to store the additional parameters at inference.

Handcrafted prompts. We explore the use of handcrafted prompts in the input text. In Table 4.5 and Table 4.6, we consider four different input designs for zero-shot and few-shot settings, respectively: (i) no handcrafted prompts, (ii) placed before the question, (iii) placed just before the [MASK] token (answer), and (iv) placed just before the question, answer and subtitles.

In *zero-shot*, handcrafted prompts are beneficial due to the absence of task-specific learning for downstream tasks. As shown in Table 4.5, the absence of handcrafted prompts drastically reduces the performance (row 1), highlighting their necessity. Moreover, the position of the

#	Input Design	MSRVTT -QA	MSVD -QA	ANET -QA	TGIF -QA
1	"[CLS] <Question>? [MASK]. <Subtitles> [SEP]"	13.2	30.2	19.8	29.8
2	"[CLS] **Answer the question:** <Question>? [MASK]. <Subtitles> [SEP]"	7.8	22.3	14.3	35.3
3	"[CLS] <Question>? **Answer:** [MASK]. <Subtitles> [SEP]"	17.7	37.2	**25.8**	45.1
4	"[CLS] **Question:** <Question>? **Answer:** [MASK]. **Subtitles:** <Subtitles> [SEP]"	**18.0**	**38.2**	24.9	**45.5**

Table 4.5 – Effect of handcrafted prompt placement on *zero-shot* top-1 validation accuracy. ANET-QA: ActivityNet-QA.

#	Input Design	MSRVTT -QA	MSVD -QA	ANET -QA	TGIF -QA
1	"[CLS] <Question>? [MASK]. <Subtitles> [SEP]"	36.3	47.0	35.8	55.8
2	"[CLS] **Answer the question**: <Question>? [MASK]. <Subtitles> [SEP]"	36.3	46.8	35.1	55.8
3	"[CLS] <Question>? **Answer:** [MASK]. <Subtitles> [SEP]"	**36.5**	47.1	35.9	55.8
4	"[CLS] **Question:** <Question>? **Answer:** [MASK]. **Subtitles:** <Subtitles> [SEP]"	**36.5**	**47.8**	**37.2**	**55.9**

Table 4.6 – Effect of handcrafted prompt placement on *few-shot* top-1 validation accuracy, using 1% of training data. ANET-QA: ActivityNet-QA.

handcrafted prompt has a significant impact on the performance. More specifically, the location of the "Answer" prompt affects the results by a large margin (row 2→3), even leading to worse performance than the absence of handcrafted prompts (row 1→2). The presence of an "Answer" prompt just before the [MASK] token yields better performance in two input designs (rows 3 & 4).

Although the impact of using handcrafted text prompts is relatively small in *few-shot* experiments compared to zero-shot experiments, they still improve enhances, particularly on MSRVTT-QA and TGIF-QA datasets, as shown in Table 4.6. Placing handcrafted prompts at the beginning (row 2), as is the case for learnable text prompts, leads to lower performance. The best performance is achieved when handcrafted prompts are placed just before the question, answer, and subtitles (row 4). Therefore, we choose to place handcrafted prompts according to row 4 for both settings.

By contrast, *learnable prompts* are all placed at the beginning. We empirically observe that other choices, *e.g.* placing half at the beginning of the input and half just before the [MASK] token, are inferior.

Method	Sub	#Training			MSRVTT-QA	MSVD-QA	ANet-QA	TGIF-QA
		Img	Vid	VQA				
CLIP* [Radford, 2021]		400M	-		2.1	7.2	1.2	3.6
Reserve [Zellers, 2022]	✓	-	20M		5.8	-	-	-
Lavender [Li, 2023b]		3M	2.5M		4.5	11.6	-	16.7
Flamingo-3B [Alayrac, 2022]		2.3B	27M		11.0	27.5	-	-
Flamingo-9B [Alayrac, 2022]		2.3B	27M		13.7	30.2	-	-
Flamingo [Alayrac, 2022]		2.3B	27M		17.4	35.6	-	-
FrozenBiLM [Yang, 2022b]	✓	-	10M		16.7	33.8	**25.9**	41.9
Just Ask [Yang, 2021]		69M	-	✓	2.9	7.5	12.2	-
Just Ask [Yang, 2022a]		69M	3M	✓	5.6	13.5	12.3	-
BLIP [Li, 2022b]		129M	-	✓	19.2	35.2	-	-
ViTiS (Ours)		-	2.5M		**18.2**	**36.2**	25.0	**45.5**
ViTiS (Ours)	✓	-	2.5M		18.1	36.1	25.5	**45.5**

Table 4.7 – *Zero-shot VideoQA* top-1 accuracy on test sets, except TGIF-QA on the validation set. Number of pre-training data: image-text/video-text pairs. Sub: subtitle input. VQA: visual question answer pairs. ANet-QA: ActivityNet-QA. CLIP: CLIP ViT-L/14. Flamingo: Flamingo-80B. We gray out methods trained on VQA pairs, which are not directly comparable. *: CLIP results taken from [Yang, 2022b].

4.4.4 Results

Zero-shot. A comparison with state-of-the-art methods on open-ended zero-shot VideoQA is given in Table 4.7. We observe an outstanding performance of our method across different VideoQA benchmarks, despite using significantly less pre-training data compared to other methods. Our performance on ActivityNetQA [Yu, 2019] is on par with FrozenBiLM [Yang, 2022b]. Lavender [Li, 2023b] employs a multi-task training approach, transforming different vision-language tasks into MLM. Reserve [Zellers, 2022] uses GPT-3 [Brown, 2020b] to convert questions into masked sentences. Flamingo [Alayrac, 2022] uses a frozen auto-regressive language model trained on an extreme-scale dataset. By contrast, our method leverages a lighter frozen language model trained on 2.5M video-text pairs.

BLIP [Li, 2022b] is pre-trained on the VQA dataset [Goyal, 2017b], which is not directly comparable as our setting does not involve training on QA pairs. Similarly, Just Ask [Yang, 2021; Yang, 2022a] uses automatically generated visual question answering datasets. Although these datasets are not annotated by humans, the model is still trained on the specific task.

METHOD	TRAINED MODULES	#TRAINED PARAMS	MSRVTT -QA	MSVD -QA	ANET -QA	TGIF -QA
FrozenBiLM [Yang, 2022b]	ATP	30M	36.0	46.5	33.2	55.1
ViTiS (Ours)	ATP	101M	36.5	47.6	33.1	55.7
ViTiS (Ours)	Prompts	0.75M	**36.9**	**47.8**	**34.2**	**56.2**

Table 4.8 – *Few-shot VideoQA* top-1 accuracy on test sets, except TGIF-QA on the validation set. Number of trained parameters: fine-tuned on the downstream dataset, using 1% of training data. ATP: All trainable parameters. ANET-QA: ActivityNet-QA.

METHOD	#SHOT	#PRE-TRAINING IMG	#PRE-TRAINING VID	#PRE-TRAINING #PARAM	MSRVTT-QA	MSVD-QA	ANET-QA	TGIF-QA
Flamingo-3B [Alayrac, 2022]	32	2.3B	27M	1.4B	25.6	42.6	–	–
Flamingo-9B [Alayrac, 2022]	32	2.3B	27M	1.8B	29.4	47.2	–	–
Flamingo-80B [Alayrac, 2022]	32	2.3B	27M	10B	31.0	52.3	–	–
ViTiS (Ours)	32	–	2.5M	101M	27.0±1.0	41.9±0.8	28.7±1.3	52.2±1.2

Table 4.9 – *Few-shot VideoQA in-context learning*. Mean and standard deviation of top-1 accuracy on test sets, except TGIF-QA on the validation set, over 10 32-shot tasks drawn at random. Only our model involves parameter updates; we fine-tune 0.75M params. Number of pre-training data: image-text/video-text pairs. ANET-QA: ActivityNet-QA.

Few-shot. We fine-tune our method on 1% of the training data by following [Yang, 2022b], which introduced the few-shot VideoQA task in this form. Table 4.8 compares our method with [Yang, 2022b]. We use two strategies, fine-tuning (i) all trainable parameters and (ii) only prompts. The latter works best, consistently outperforming [Yang, 2022b] while diminishing the number of fine-tuned parameters.

Few-shot in-context learning. An alternative approach for few-shot VideoQA is *in-context learning* [Alayrac, 2022], using few, *e.g.* 32, labeled examples. To compare, we draw 10 tasks of 32 examples at random from 1% of training data of each downstream dataset; we fine-tune the prompt vectors, that is, 0.75M parameters, on each task for 5 epochs and report mean and standard deviation. This can be considered as *test-time prompt tuning* [Shu, 2022] using task-specific annotated data.

Table 4.9 shows the results of few-shot in-context learning. Flamingo [Alayrac, 2022] uses a frozen auto-regressive language model with trainable cross-attention layers that incorporate vision and language input, trained on an extreme-scale dataset. The Flamingo-3B, Flamingo-

9B, and Flamingo-80B have 1.4B, 1.8B, and 10B learned parameters, respectively, in addition to the frozen language model. By contrast, our method uses a lighter frozen language model and lighter adaptation modules, resulting in only 101M parameters to learn, and our training data is a relatively small amount of video-text pairs. Despite this, our method outperforms Flamingo-3B [Alayrac, 2022] on MSRVTT-QA and is on par with MSVD-QA.

4.5 Conclusion

We explored the adaptation of large-scale pre-trained vision and language models for VideoQA under a scarcity of data. We introduced multimodal prompt learning and a visual mapping network to address challenges in such adaptation. Our method consistently outperforms prior works while requiring minimal parameter fine-tuning in few-shot VideoQA.

CHAPTER 5

CONCLUSIONS

Contents

This chapter summarizes our contributions in Section 5.1 and discusses potential future work related to this thesis in Section 5.2.

5.1 Summary of contributions

This thesis advances the field of video understanding by focusing on multimodal semantic understanding and efficient learning from limited data, particularly in video question answering.

Our first contribution addresses long-range video question answering, which involves responding to questions about long videos, such as TV show episodes. These questions require an understanding of extended video content. While recent approaches rely on human-generated external sources, we rely on raw data and generate episode summaries. We also present a video question-answering method that encodes different text-based inputs independently, including video description, and employs a simple fusion method to combine all modalities. Additionally, we propose soft temporal attention for localization over long inputs to process long text input efficiently. Our model outperforms the previous methods without using question-specific human annotation or human-made plot summaries.

Our following contribution explores zero-shot and few-shot video question answering, aiming to enhance efficient learning from limited data. We leverage the knowledge of existing large-scale models; however, adapting pre-trained models to limited data presents challenges, such as overfitting, catastrophic forgetting, and bridging the cross-modal gap between vision and lan-

guage. To address these challenges, we introduce a parameter-efficient method that combines multimodal prompt learning with a transformer-based mapping network, while keeping the pre-trained vision and language models frozen. This approach allows us to overcome the limitations posed by data scarcity and to improve performance in video question answering, especially in zero-shot and few-shot scenarios.

These contributions significantly enhance the capabilities of multimodal video question answering systems, making them more robust and adaptive in scenarios where specifically human-annotated labeled data is limited or unavailable.

5.2 Future work

Multimodal long video summarization. We generate text-based summaries for long videos, but a combined process of summarizing both video and dialog into a single output could be implemented, effectively advancing multimodal fusion from its usual roles in question answering or memory networks to an earlier stage in text description. In this way, question-answering can be handled entirely by language models from extended video summaries. Additionally, we utilize the dialog summarization model as pre-trained by [Chen, 2020a]; but there is a margin for improvement by fine-tuning it on extended plot summaries, which are available, *e.g.*, for certain TV shows.

Parameter-efficient methods. We explore prompt learning [Li, 2021; Liu, 2021a; Liu, 2022a], and adapter layers [Houlsby, 2019] as parameter-efficient methods for video question answering in this thesis. Future work could explore incorporating a broader range of adaptation strategies, including low-rank adaptation (LoRA) [Hu, 2022], and QLoRA [Dettmers, 2023]. These strategies could be integrated with multimodal prompt learning or applied independently to enhance efficiency and adaptability to establish benchmark work for video question answering. For instance, a recent work [Jia, 2022] explores various parameter-efficient strategies for image classification tasks, relying solely on the vision modality. Additionally, future work could involve utilizing different visual and language models. Exploring the application of this approach in video understanding tasks beyond question answering also presents a valuable direction for further research.

Audio modality. While we use speech in the form of subtitles in this thesis, we have not directly incorporated the audio modality in its natural form. Future work could benefit from exploring the inclusion of audio, specifically the emotional nuances conveyed by sound, to enhance the understanding of video content. Subtitles obtained via speech-to-text can be noisy and sometimes sparse, potentially missing emotional context. This approach could be particularly beneficial in applications where understanding sentiment and emotional intent is crucial, such as in detecting sarcastic speech and conducting sentiment analysis.

Potential harms and ethical considerations. Employing transformer-based models in various applications raises ethical concerns and potential harms, including introducing biases and stereotypes driven by the training data, spreading misinformation, and compromising privacy through the leakage of sensitive information. Addressing these issues requires a comprehensive effort in model development; future work can focus on ensuring these technologies benefit society while minimizing harm.

Furthermore, the substantial environmental impact due to the high computational demands of training these models poses a significant challenge. This challenge calls for future research into more energy-efficient computing methods to reduce the carbon footprint associated with model training processes. In this thesis, we employ parameter-efficient adaptation methods that reduce computational demand, but these methods can be further improved.

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*. *Proc. NeurIPS*. 2021 (cit. on p. 26).

[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*. *Proc. NeurIPS*. 2022 (cit. on pp. 33, 62, 63, 75–77).

[Alayrac, 2020] Jean-Baptiste Alayrac, Adria Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, et al. *Self-supervised multimodal versatile networks*. *Proc. NeurIPS*. 2020 (cit. on p. 26).

[Althoff, 2016] Tim Althoff, Kevin Clark, and Jure Leskovec. *Large-scale analysis of counseling conversations: An application of natural language processing to mental health*. *Trans. ACL* 4 (2016), pp. 463–476 (cit. on pp. 12, 42).

[Antol, 2015] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, et al. *VQA: Visual Question Answering*. *Proc. ICCV*. 2015 (cit. on p. 37).

[Arnab, 2021] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. *ViViT: A video vision transformer*. *Proc. ICCV*. 2021 (cit. on p. 25).

[Asano, 2020] YM Asano, C Rupprecht, and A Vedaldi. *Self-labelling via simultaneous clustering and representation learning*. *Proc. ICLR*. 2020 (cit. on p. 24).

[Baccouche, 2010] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. *Action classification in soccer videos with long short-term memory recurrent neural networks*. *Proc. ICANN*. 2010 (cit. on p. 25).

[Badrinarayanan, 2017] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. *SegNet: A deep convolutional encoder-decoder architecture for image segmentation*. *TPAMI* (2017) (cit. on p. 22).

[Bahdanau, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. *Neural machine translation by jointly learning to align and translate*. 2015 (cit. on pp. 13, 26).

[Bahng, 2022] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. *Exploring visual prompts for adapting large-scale models*. *arXiv preprint arXiv:2203.17274* (2022) (cit. on p. 64).

[Bain, 2021] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. *Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval*. *Proc. ICCV*. 2021 (cit. on pp. 3, 7, 31, 62, 63, 69, 71).

[Bao, 2022] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. *BEiT: BERT Pre-Training of Image Transformers*. *Proc. ICLR*. 2022 (cit. on p. 24).

[Bertasius, 2021] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. *Is space-time attention all you need for video understanding? Proc. ICML*. 2021 (cit. on p. 25).

[Bojanowski, 2017] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. *Enriching Word Vectors with Subword Information*. *TACL* 5 (2017), pp. 135–146 (cit. on pp. 13, 26).

[Britz, 2017] Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. *Massive Exploration of Neural Machine Translation Architectures*. *Proc. EMNLP*. 2017 (cit. on p. 15).

[Brown, 2020a] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, et al. *Language models are few-shot learners*. *Proc. NeurIPS*. 2020 (cit. on pp. 19, 64).

[Brown, 2020b] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, et al. *Language models are few-shot learners*. *Proc. NeurIPS*. 2020 (cit. on p. 75).

[Caba Heilbron, 2015] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. *ActivityNet: A large-scale video benchmark for human activity understanding*. *Proc. CVPR*. 2015 (cit. on pp. 24, 69).

[Carion, 2020a] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. *End-to-end object detection with transformers*. *Proc. ECCV*. 2020 (cit. on p. 22).

[Carion, 2020b] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. *End-to-end object detection with transformers*. *Proc. ECCV*. 2020 (cit. on p. 22).

[Caron, 2020] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. *Unsupervised learning of visual features by contrasting cluster assignments*. *Proc. NeurIPS*. 2020 (cit. on p. 24).

[Caron, 2021] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, et al. *Emerging Properties in Self-Supervised Vision Transformers*. *Proc. ICCV*. 2021 (cit. on p. 24).

[Carreira, 2017] Joao Carreira and Andrew Zisserman. *Quo vadis, action recognition? a new model and the kinetics dataset*. *Proc. CVPR*. 2017 (cit. on pp. 25, 26).

[Chadha, 2020] Aman Chadha, Gurneet Arora, and Navpreet Kaloty. *iPerceive: Applying Common-Sense Reasoning to Multi-Modal Dense Video Captioning and Video Question Answering*. *arXiv preprint arXiv:2011.07735* (2020) (cit. on pp. 37, 39).

[Chen, 2011] David Chen and William Dolan. *Collecting Highly Parallel Data for Paraphrase Evaluation*. *Proc. ACL*. 2011 (cit. on pp. 4, 26, 69).

[Chen, 2020a] Jiaao Chen and Diyi Yang. *Multi-View Sequence-to-Sequence Models with Conversational Structure for Abstractive Dialogue Summarization*. *Proc. EMNLP*. 2020 (cit. on pp. 40–42, 80).

[Chen, 2020b] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, et al. *Generative pretraining from pixels*. *Proc. ICML*. 2020 (cit. on p. 24).

[Chen, 2020c] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. *A simple framework for contrastive learning of visual representations*. *Proc. ICML*. 2020 (cit. on pp. 23, 24).

[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 p. 29).

[Chen, 2021] Xinlei Chen, Saining Xie, and Kaiming He. *An empirical study of training self-supervised vision transformers. Proc. ICCV*. 2021 (cit. on p. 23).

[Chen, 2020d] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, et al. *UNITER: Universal image-text representation learning*. *Proc. ECCV*. 2020 (cit. on p. 27).

[Cheng, 2023] Feng Cheng, Xizi Wang, Jie Lei, David Crandall, Mohit Bansal, and Gedas Bertasius. *VindLU: A recipe for effective video-and-language pretraining*. *Proc. CVPR*. 2023 (cit. on p. 63).

[Cho, 2014] Kyunghyun Cho, B van Merrienboer, Caglar Gulcehre, F Bougares, H Schwenk, and Yoshua Bengio. *Learning phrase representations using RNN encoder-decoder for statistical machine translation*. *Proc. EMNLP*. 2014 (cit. on p. 13).

[Choi, 2000] Freddy Y. Y. Choi. *Advances in domain independent linear text segmentation*. *Proc. NAACL*. 2000 (cit. on p. 42).

[Chu, 2023] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, and Chunhua Shen. *Conditional Positional Encodings for Vision Transformers*. *Proc. ICLR*. 2023 (cit. on p. 22).

[Dave, 2022] Ishan Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. *TCLR: Temporal contrastive learning for video representation*. *CVIU* (2022) (cit. on p. 25).

[Deng, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. *ImageNet: A large-scale hierarchical image database*. *Proc. CVPR*. 2009 (cit. on p. 21).

[Desai, 2021] Karan Desai and Justin Johnson. *Virtex: Learning visual representations from textual annotations*. *Proc. CVPR*. 2021 (cit. on pp. 24, 28).

[Dettmers, 2023] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. *QLoRA: Efficient finetuning of quantized LLMs*. *Proc. NeurIPS*. 2023 (cit. on pp. 19, 20, 32, 80).

[Devlin, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. *Proc. NAACL*. 2019 (cit. on pp. 7, 16–19, 23, 27, 29, 42, 47, 62, 63).

[Donahue, 2015] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, et al. *Long-term recurrent convolutional networks for visual recognition and description*. *Proc. CVPR*. 2015 (cit. on p. 25).

[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*. *Proc. ICLR*. 2021 (cit. on pp. 7, 22, 23, 25, 28, 62, 63, 69).

[Eichenberg, 2022] Constantin Eichenberg, Sidney Black, Samuel Weinbach, Letitia Parcalabescu, and Anette Frank. *MAGMA–Multimodal Augmentation of Generative Models through Adapter-based Finetuning*. *Proc. Findings of EMNLP*. 2022 (cit. on p. 64).

[Engin, 2021a] Deniz Engin. *On the hidden treasure of dialog in video question answering - Project Webpage*. `https://engindeniz.github.io/dialogsummary-videoqa`. 2021 (cit. on pp. xxviii, 8).

[Engin, 2023a] Deniz Engin. *Zero-Shot and Few-Shot Video Question Answering with Multi-Modal Prompts - Project Webpage*. `https://engindeniz.github.io/vitis`. 2023 (cit. on pp. xxix, 9).

[Engin, 2023b] Deniz Engin and Yannis Avrithis. *Zero-Shot and Few-Shot Video Question Answering with Multi-Modal Prompts*. *Proc. ICCV Workshops*. 2023 (cit. on pp. xxix, 9, 61).

[Engin, 2021b] Deniz Engin, François Schnitzler, Ngoc QK Duong, and Yannis Avrithis. *On the hidden treasure of dialog in video question answering*. 2021 (cit. on pp. xxviii, 8, 37).

[Fan, 2019] Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. *Heterogeneous Memory Enhanced Multimodal Attention Model for Video Question Answering*. *Proc. CVPR*. 2019 (cit. on p. 37).

[Feichtenhofer, 2020] Christoph Feichtenhofer. *X3D: Expanding architectures for efficient video recognition*. *Proc. CVPR*. 2020 (cit. on p. 25).

[Feichtenhofer, 2019] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. *Slowfast networks for video recognition*. *Proc. ICCV*. 2019 (cit. on p. 25).

[Feng, 2020] Xiachong Feng, Xiaocheng Feng, Bing Qin, and Ting Liu. *Incorporating Commonsense Knowledge into Abstractive Dialogue Summarization via Heterogeneous Graph Networks*. *arXiv preprint arXiv:2010.10044* (2020) (cit. on p. 40).

[Frome, 2013] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, et al. *DeViSE: A Deep Visual-Semantic Embedding Model*. *Proc. NIPS*. 2013 (cit. on p. 37).

[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. 30–32, 63).

[Fu, 2023] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, et al. *An empirical study of end-to-end video-language transformers with masked visual modeling*. *Proc. CVPR*. 2023 (cit. on p. 63).

[Gabeur, 2020] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. *Multi-modal transformer for video retrieval*. *Proc. ECCV*. 2020 (cit. on p. 29).

[Garcia, 2020a] Noa Garcia and Yuta Nakashima. *Knowledge-Based Video Question Answering with Unsupervised Scene Descriptions*. *Proc. ECCV*. 2020 (cit. on pp. 6, 38–40, 42, 43, 45, 48, 49, 53–57).

[Garcia, 2020b] Noa Garcia, Mayu Otani, Chenhui Chu, and Yuta Nakashima. *KnowIT VQA: Answering Knowledge-Based Questions about Videos*. *Proc. AAAI*. 2020 (cit. on pp. 6, 26, 38, 39, 47–49, 63).

[Gidaris, 2018] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. *Unsupervised Representation Learning by Predicting Image Rotations*. *Proc. ICLR*. 2018 (cit. on p. 23).

[Gliwa, 2019] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. *SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization*. *Proceedings of the 2nd Workshop on New Frontiers in Summarization*. ACL, 2019 (cit. on pp. 39–41).

[Goyal, 2017a] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, et al. *The something something video database for learning and evaluating visual common sense*. *Proc. ICCV*. 2017 (cit. on p. 24).

[Goyal, 2017b] 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*. *Proc. CVPR*. 2017 (cit. on p. 75).

[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*. *Proc. NeurIPS*. 2020 (cit. on p. 24).

[Gu, 2018] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, et al. *AVA: A video dataset of spatio-temporally localized atomic visual actions*. *Proc. CVPR*. 2018 (cit. on p. 24).

[Han, 2021] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. *Transformer in transformer*. *Proc. NeurIPS*. 2021 (cit. on p. 22).

[Han, 2023] Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. *AutoAD: Movie Description in Context*. *Proc. CVPR*. 2023 (cit. on pp. 33, 62).

[Han, 2022] Tengda Han, Weidi Xie, and Andrew Zisserman. *Temporal alignment networks for long-term video*. *Proc. CVPR*. 2022 (cit. on p. 26).

[He, 2020] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. *Momentum contrast for unsupervised visual representation learning*. *Proc. CVPR*. 2020 (cit. on pp. 23, 24).

[He, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Deep residual learning for image recognition*. *Proc. CVPR*. 2016 (cit. on pp. 21, 22, 26).

[He, 2021] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. *DeBERTa: Decoding-enhanced BERT with Disentangled Attention*. *Proc. ICLR*. 2021 (cit. on pp. 7, 16, 17, 62, 63, 70).

[Hochreiter, 1997] Sepp Hochreiter and Jürgen Schmidhuber. *Long short-term memory*. *Neural Computation* 9.8 (1997), pp. 1735–1780 (cit. on pp. 12, 42).

[Houlsby, 2019] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, et al. *Parameter-efficient transfer learning for NLP*. *Proc. ICML*. 2019 (cit. on pp. 7, 19, 20, 32, 62, 64, 70, 80).

[Hu, 2022] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, et al. *LoRA: Low-Rank Adaptation of Large Language Models*. *Proc. ICLR*. 2022 (cit. on pp. 19, 20, 32, 80).

[Huang, 2023] Jingjia Huang, Yinan Li, Jiashi Feng, Xinglong Wu, Xiaoshuai Sun, and Rongrong Ji. *Clover: Towards a unified video-language alignment and fusion model*. *Proc. CVPR*. 2023 (cit. on p. 63).

[Iashin, 2020a] Vladimir Iashin and Esa Rahtu. *A Better Use of Audio-Visual Cues: Dense Video Captioning with Bi-modal Transformer*. *Proc. BMVC*. 2020 (cit. on p. 27).

[Iashin, 2020b] Vladimir Iashin and Esa Rahtu. *Multi-modal dense video captioning*. *Proc. CVPR Workshops*. 2020 (cit. on p. 27).

[Ioffe, 2015] Sergey Ioffe and Christian Szegedy. *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. *Proc. ICML*. 2015 (cit. on p. 21).

[Jaegle, 2021] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. *Perceiver: General perception with iterative attention*. *Proc. ICML*. 2021 (cit. on pp. 62, 66).

[Jang, 2017] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. *TGIF-QA: Toward spatio-temporal reasoning in visual question answering*. *Proc. CVPR*. 2017 (cit. on pp. 26, 63, 69).

[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*. *Proc. ICML*. 2021 (cit. on pp. 24, 27, 28).

[Jia, 2022] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, et al. *Visual prompt tuning*. *Proc. ECCV*. 2022 (cit. on pp. 64, 80).

[Ju, 2022] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. *Prompting visual-language models for efficient video understanding*. *Proc. ECCV*. 2022 (cit. on pp. 28, 64).

[Jurafsky, 2023] Dan Jurafsky and James H Martin. *Speech and Language Processing. 3rd*. `https://web.stanford.edu/~{}jurafsky/slp3/`. 2023 (cit. on p. 12).

[Karpathy, 2014] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. *Large-scale video classification with convolutional neural networks*. *Proc. CVPR*. 2014 (cit. on pp. 24, 25).

[Kay, 2017] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, et al. *The kinetics human action video dataset*. *arXiv preprint arXiv:1705.06950* (2017) (cit. on p. 24).

[Khattak, 2023] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. *MaPLe: Multi-modal prompt learning*. *Proc. CVPR*. 2023 (cit. on p. 64).

[Kim, 2020a] Hyounghun Kim, Zineng Tang, and Mohit Bansal. *Dense-caption matching and frame-selection gating for temporal localization in VideoQA*. *Proc. ACL*. 2020 (cit. on pp. 37–39, 45).

[Kim, 2019a] Junyeong Kim, Minuk Ma, Kyungsu Kim, Sungjin Kim, and Chang D Yoo. *Gaining extra supervision via multi-task learning for multi-modal video question answering*. *Proc. IJCNN*. 2019 (cit. on p. 27).

[Kim, 2019b] Junyeong Kim, Minuk Ma, Kyungsu Kim, Sungjin Kim, and Chang D Yoo. *Progressive attention memory network for movie story question answering*. *Proc. CVPR*. 2019 (cit. on p. 39).

[Kim, 2020b] Junyeong Kim, Minuk Ma, Trung Pham, Kyungsu Kim, and Chang D. Yoo. *Modality Shifting Attention Network for Multi-Modal Video Question Answering*. *Proc. CVPR*. 2020 (cit. on pp. 27, 37–39, 45).

[Kim, 2018] Kyung-Min Kim, Seong-Ho Choi, Jin-Hwa Kim, and Byoung-Tak Zhang. *Multimodal Dual Attention Memory for Video Story Question Answering*. *Proc. ECCV*. 2018 (cit. on pp. 27, 37, 39).

[Kingma, 2015] Diederik P Kingma and Jimmy Ba. *Adam: A method for stochastic optimization*. *Proc. ICLR*. 2015 (cit. on p. 71).

[Kiros, 2014] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. *Unifying visual-semantic embeddings with multimodal neural language models*. *arXiv preprint arXiv:1411.2539* (2014) (cit. on p. 37).

[Krishna, 2017a] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. *Dense-Captioning Events in Videos*. *Proc. ICCV*. 2017 (cit. on pp. 4, 26, 37).

[Krishna, 2017b] 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* (2017) (cit. on p. 29).

[Krizhevsky, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. *ImageNet classification with deep convolutional neural networks*. *Proc. NIPS*. 2012 (cit. on pp. 3, 21).

[Kudo, 2018] Taku Kudo and John Richardson. *SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing*. *Proc. EMNLP: System Demonstrations*. 2018 (cit. on p. 70).

[Kuehne, 2011] Hildegard Kuehne, Hueihan Jhuang, Estibaliz Garrote, Tomaso Poggio, and Thomas Serre. *HMDB: A large video database for human motion recognition*. *Proc. ICCV*. 2011 (cit. on p. 24).

[Laurençon, 2023] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, et al. *OBELICS: An open web-scale filtered dataset of interleaved image-text documents*. *Proc. NeurIPS Datasets and Benchmarks Track*. 2023 (cit. on p. 31).

[Laurençon, 2024] Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. *What matters when building vision-language models? arXiv preprint arXiv:2405.02246* (2024) (cit. on p. 31).

[LeCun, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. *Gradient-based learning applied to document recognition*. *Proc. of the IEEE* (1998) (cit. on p. 21).

[Lee, 2023] Yi-Lun Lee, Yi-Hsuan Tsai, Wei-Chen Chiu, and Chen-Yu Lee. *Multimodal Prompting with Missing Modalities for Visual Recognition*. *Proc. CVPR*. 2023 (cit. on p. 64).

[Lei, 2021] 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*. *Proc. CVPR*. 2021 (cit. on pp. 29, 63).

[Lei, 2018] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. *TVQA: Localized, Compositional Video Question Answering*. *Proc. EMNLP*. 2018 (cit. on pp. 4, 26, 37–39, 45, 48, 63).

[Lei, 2019] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. *TVQA+: Spatio-temporal grounding for video question answering*. *Proc. ACL*. 2019 (cit. on pp. 38, 39, 45).

[Lester, 2021] Brian Lester, Rami Al-Rfou, and Noah Constant. *The Power of Scale for Parameter-Efficient Prompt Tuning*. *Proc. EMNLP*. 2021 (cit. on p. 64).

[Lewis, 2020] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, et al. *BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*. *Proc. ACL*. 2020 (cit. on pp. 17, 18, 42, 47).

[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*. *Proc. CVPR*. 2022 (cit. on p. 63).

[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*. *Proc. ICML*. 2023 (cit. on p. 33).

[Li, 2022b] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. *BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation*. *Proc. ICML*. 2022 (cit. on pp. 63, 75).

[Li, 2020] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. *HERO: Hierarchical Encoder for Video+ Language Omni-representation Pre-training*. *Proc. EMNLP*. 2020 (cit. on pp. 30, 32, 63).

[Li, 2023b] Linjie Li, Zhe Gan, Kevin Lin, Chung-Ching Lin, Zicheng Liu, Ce Liu, et al. *Lavender: Unifying video-language understanding as masked language modeling*. *Proc. CVPR*. 2023 (cit. on pp. 30–32, 63, 75).

[Li, 2021] Xiang Lisa Li and Percy Liang. *Prefix-Tuning: Optimizing Continuous Prompts for Generation*. *Proc. ACL*. 2021 (cit. on pp. 7, 19, 20, 32, 62, 64, 80).

[Liang, 2018] Junwei Liang, Lu Jiang, Liangliang Cao, Li-Jia Li, and Alexander G Hauptmann. *Focal visual-text attention for visual question answering*. *Proc. CVPR*. 2018 (cit. on pp. 26, 39).

[Lin, 2019] Ji Lin, Chuang Gan, and Song Han. *TSM: Temporal shift module for efficient video understanding*. *Proc. ICCV*. 2019 (cit. on p. 25).

[Lin, 2022] Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard De Melo, Xiaogang Wang, et al. *Frozen clip models are efficient video learners*. *Proc. ECCV*. 2022 (cit. on p. 28).

[Liu, 2022a] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, et al. *P-Tuning: Prompt Tuning Can Be Comparable to Fine-tuning Across Scales and Tasks"*. *Proc. ACL*. 2022 (cit. on pp. 7, 19, 20, 32, 62, 64, 67, 80).

[Liu, 2021a] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, et al. *GPT understands, too*. *arXiv preprint arXiv:2103.10385* (2021) (cit. on pp. 7, 19, 20, 32, 62, 64, 80).

[Liu, 2019a] 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. 7, 62, 63).

[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 p. 16).

[Liu, 2021b] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, et al. *Swin transformer: Hierarchical vision transformer using shifted windows*. *Proc. ICCV*. 2021 (cit. on pp. 22, 25).

[Liu, 2022b] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, et al. *Video swin transformer*. *Proc. CVPR*. 2022 (cit. on pp. 7, 62, 63).

[Liu, 2022c] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, et al. *Video swin transformer*. *Proc. CVPR*. 2022 (cit. on p. 25).

[Lowe, 2004] David G Lowe. *Distinctive image features from scale-invariant keypoints*. *IJCV* (2004) (cit. on p. 20).

[Lu, 2019] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. *ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks*. *Proc. NeurIPS*. 2019 (cit. on p. 27).

[Luo, 2020] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, et al. *UniVL: A unified video and language pre-training model for multimodal understanding and generation*. *arXiv preprint arXiv:2002.06353* (2020) (cit. on pp. 30, 32).

[Luo, 2022] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, et al. *CLIP4Clip: An empirical study of clip for end to end video clip retrieval and captioning*. *Neurocomputing* (2022) (cit. on pp. 28, 31, 32).

[Mahabadi, 2022] Rabeeh Karimi Mahabadi, Luke Zettlemoyer, James Henderson, Lambert Mathias, Marzieh Saeidi, Veselin Stoyanov, et al. *PERFECT: Prompt-free and Efficient Few-shot Learning with Language Models*. *Proc. ACL*. 2022 (cit. on p. 64).

[Miech, 2020a] 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*. *Proc. CVPR*. 2020 (cit. on pp. 27, 29).

[Miech, 2020b] 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*. *Proc. CVPR*. 2020 (cit. on p. 26).

[Miech, 2019a] 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*. *Proc. ICCV*. 2019 (cit. on p. 29).

[Miech, 2019b] 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*. *Proc. ICCV*. 2019 (cit. on pp. 7, 62, 63).

[Mikolov, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. *Efficient estimation of word representations in vector space*. *Proc. ICLR Workshop* (2013) (cit. on pp. 13, 26).

[Misra, 2016] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. *Shuffle and learn: Unsupervised learning using temporal order verification*. *Proc. ECCV*. 2016 (cit. on p. 25).

[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. 33, 62).

[Na, 2017] Seil Na, Sangho Lee, Jisung Kim, and Gunhee Kim. *A Read-Write Memory Network for Movie Story Understanding*. *Proc. ICCV*. 2017 (cit. on pp. 26, 37, 39).

[Ni, 2022] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, et al. *Expanding language-image pretrained models for general video recognition*. *Proc. ECCV*. 2022 (cit. on p. 28).

[Noroozi, 2016] Mehdi Noroozi and Paolo Favaro. *Unsupervised learning of visual representations by solving jigsaw puzzles*. *Proc. ECCV*. 2016 (cit. on p. 23).

[Pan, 2018] Haojie Pan, Junpei Zhou, Zhou Zhao, Yan Liu, Deng Cai, and Min Yang. *Dial2desc: End-to-end dialogue description generation*. *arXiv preprint arXiv:1811.00185* (2018) (cit. on p. 39).

[Patrick, 2021] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander Hauptmann, Joao Henriques, et al. *Support-set bottlenecks for video-text representation learning*. *Proc. ICLR* (2021) (cit. on p. 29).

[Pennington, 2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. *GloVe: Global vectors for word representation*. *Proc. EMNLP*. 2014 (cit. on pp. 13, 26).

[Qin, 2021] Guanghui Qin and Jason Eisner. *Learning How to Ask: Querying LMs with Mixtures of Soft Prompts*. *Proc. NAACL*. 2021 (cit. on p. 64).

[Rabiner, 1989] Lawrence R Rabiner. *A tutorial on hidden Markov models and selected applications in speech recognition*. *Proceedings of the IEEE* 77.2 (1989), pp. 257–286 (cit. on p. 12).

[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*. *Proc. ICML*. 2021 (cit. on pp. 7, 24, 27, 28, 32, 62, 63, 69, 75).

[Radford, 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. *Language models are unsupervised multitask learners*. *Technical Report* (2019) (cit. on pp. 7, 16, 62, 63).

[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 p. 17).

[Rasheed, 2023] Hanoona Rasheed, Muhammad Uzair Khattak, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. *Fine-tuned clip models are efficient video learners*. *Proc. CVPR*. 2023 (cit. on p. 64).

[Redmon, 2016] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. *You only look once: Unified, real-time object detection*. *Proc. CVPR*. 2016 (cit. on p. 22).

[Reimers, 2019] Nils Reimers and Iryna Gurevych. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. *Proc. EMNLP-IJCNLP*. 2019 (cit. on pp. 18, 42).

[Ren, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. *Faster R-CNN: Towards real-time object detection with region proposal networks*. *Proc. NIPS*. 2015 (cit. on p. 22).

[Rohrbach, 2017] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, et al. *Movie description*. *IJCV* (2017) (cit. on pp. 4, 26).

[Ronneberger, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional networks for biomedical image segmentation*. *Proc. MICCAI*. 2015 (cit. on p. 22).

[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. 12).

[Sariyildiz, 2020] Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. *Learning visual representations with caption annotations*. *Proc. ECCV*. 2020 (cit. on pp. 24, 28).

[Schroff, 2015] Florian Schroff, Dmitry Kalenichenko, and James Philbin. *FaceNet: A unified embedding for face recognition and clustering*. *Proc. CVPR*. 2015 (cit. on p. 43).

[Schuster, 2012] Mike Schuster and Kaisuke Nakajima. *Japanese and Korean Voice Search*. *Proc. ICASSP*. 2012 (cit. on p. 44).

[Seo, 2022] Paul Hongsuck Seo, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. *End-to-end generative pretraining for multimodal video captioning*. *Proc. CVPR*. 2022 (cit. on p. 29).

[Seo, 2021] Paul Hongsuck Seo, Arsha Nagrani, and Cordelia Schmid. *Look before you speak: Visually contextualized utterances*. *Proc. CVPR*. 2021 (cit. on p. 29).

[Shu, 2022] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, et al. *Test-time prompt tuning for zero-shot generalization in vision-language models*. *Proc. NeurIPS*. 2022 (cit. on p. 76).

[Shukor, 2023a] Mustafa Shukor, Corentin Dancette, and Matthieu Cord. *eP-ALM: Efficient perceptual augmentation of language models*. *Proc. ICCV*. 2023 (cit. on p. 33).

[Shukor, 2023b] Mustafa Shukor, Corentin Dancette, Alexandre Rame, and Matthieu Cord. *UnIVAL: Unified Model for Image, Video, Audio and Language Tasks*. *TMLR* (2023) (cit. on pp. 30, 32).

[Sigurdsson, 2016] Gunnar A Sigurdsson, Gul Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. *Hollywood in homes: Crowdsourcing data collection for activity understanding*. *Proc. ECCV*. 2016 (cit. on p. 24).

[Simonyan, 2014a] Karen Simonyan and Andrew Zisserman. *Two-stream convolutional networks for action recognition in videos*. *Proc. NIPS*. 2014 (cit. on p. 25).

[Simonyan, 2014b] Karen Simonyan and Andrew Zisserman. *Very deep convolutional networks for large-scale image recognition*. *arXiv preprint arXiv:1409.1556* (2014) (cit. on p. 22).

[Soomro, 2012] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. *UCF101: A dataset of 101 human actions classes from videos in the wild*. *arXiv preprint arXiv:1212.0402* (2012) (cit. on p. 24).

[Strudel, 2021] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. *Segmenter: Transformer for semantic segmentation*. *Proc. ICCV*. 2021 (cit. on p. 22).

[Sun, 2019] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. *VideoBERT: A joint model for video and language representation learning*. *Proc. ICCV*. 2019 (cit. on pp. 27, 29, 63).

[Sung, 2022] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. *Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks*. *Proc. CVPR*. 2022 (cit. on p. 64).

[Szegedy, 2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, et al. *Going deeper with convolutions*. *Proc. CVPR*. 2015 (cit. on pp. 21, 22).

[Tan, 2019] Hao Tan and Mohit Bansal. *LXMERT: Learning Cross-Modality Encoder Representations from Transformers*. *Proc. EMNLP*. 2019 (cit. on p. 27).

[Tapaswi, 2016a] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. *MovieQA: Understanding Stories in Movies Through Question-Answering*. *Proc. CVPR*. 2016 (cit. on pp. 37–39).

[Tapaswi, 2016b] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. *MovieQA: Understanding Stories in Movies Through Question-Answering*. *Proc. CVPR*. 2016 (cit. on p. 63).

[Tong, 2022] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. *VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training*. *Proc. NeurIPS*. 2022 (cit. on p. 26).

[Touvron, 2021] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. *Training data-efficient image transformers & distillation through attention*. *Proc. ICML*. 2021 (cit. on p. 22).

[Touvron, 2022] Hugo Touvron, Matthieu Cord, and Hervé Jégou. *DeiT III: Revenge of the ViT*. *Proc. ECCV*. 2022 (cit. on p. 22).

[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 p. 16).

[Tran, 2015] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. *Learning spatiotemporal features with 3d convolutional networks*. *Proc. ICCV*. 2015 (cit. on pp. 25, 26).

[Tsimpoukelli, 2021] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. *Multimodal few-shot learning with frozen language models*. *Proc. NeurIPS*. 2021 (cit. on p. 33).

[Urooj, 2020] Aisha Urooj, Amir Mazaheri, Niels Da Vitoria Lobo, and Mubarak Shah. *MMFT-BERT: Multimodal Fusion Transformer with BERT Encodings for Visual Question Answering*. *Proc. EMNLP*. 2020 (cit. on pp. 37, 39).

[Vaswani, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, et al. *Attention is All you Need*. *Proc. NIPS*. 2017 (cit. on pp. 13–16, 23, 26, 47).

[Venugopalan, 2015a] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. *Sequence to sequence-video to text*. *Proc. ICCV*. 2015 (cit. on pp. 26, 37).

[Venugopalan, 2015b] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. *Translating Videos to Natural Language Using Deep Recurrent Neural Networks*. *Proc. ACL*. 2015 (cit. on p. 26).

[Vinyals, 2015] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. *Show and tell: A neural image caption generator*. *Proc. CVPR*. 2015 (cit. on p. 37).

[Wang, 2023] Alex Jinpeng Wang, Yixiao Ge, Rui Yan, Ge Yuying, Xudong Lin, Guanyu Cai, et al. *All in One: Exploring Unified Video-Language Pre-training*. *Proc. CVPR*. 2023 (cit. on pp. 30–32, 63).

[Wang, 2018] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, et al. *Temporal segment networks for action recognition in videos*. *TPAMI* (2018) (cit. on p. 25).

[Wang, 2022] Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, et al. *BEVT: BERT pretraining of video transformers*. *Proc. CVPR*. 2022 (cit. on p. 25).

[Wang, 2021] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, et al. *Pyramid vision transformer: A versatile backbone for dense prediction without convolutions*. *Proc. ICCV*. 2021 (cit. on p. 22).

[Wasim, 2023] Syed Talal Wasim, Muzammal Naseer, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. *Vita-CLIP: Video and text adaptive CLIP via Multimodal Prompting*. *Proc. CVPR*. 2023 (cit. on p. 64).

[Wei, 2022] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. *Masked feature prediction for self-supervised visual pre-training*. *Proc. CVPR*. 2022 (cit. on p. 25).

[Wu, 2020] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, et al. *Visual transformers: Token-based image representation and processing for computer vision*. *arXiv preprint arXiv:2006.03677* (2020) (cit. on p. 22).

[Wu, 2019] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. *Long-term feature banks for detailed video understanding*. *Proc. CVPR*. 2019 (cit. on p. 43).

[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. 44).

[Xie, 2018] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. *Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification*. *Proc. ECCV*. 2018 (cit. on p. 26).

[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*. *Proc. ACM Multimedia*. 2017 (cit. on pp. 4, 26, 63, 69).

[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*. *Proc. EMNLP*. 2021 (cit. on pp. 30, 32).

[Xu, 2016] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. *MSR-VTT: A large video description dataset for bridging video and language*. *Proc. CVPR*. 2016 (cit. on pp. 4, 26, 69).

[Yang, 2021] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. *Just ask: Learning to answer questions from millions of narrated videos*. *Proc. ICCV*. 2021 (cit. on pp. 26, 63, 75).

[Yang, 2022a] 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. 63, 75).

[Yang, 2022b] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. *Zero-Shot Video Question Answering via Frozen Bidirectional Language Models*. *Proc. NeurIPS*. 2022 (cit. on pp. 28, 31, 33, 62–64, 68–70, 72, 75, 76).

[Yang, 2020] Zekun Yang, Noa Garcia, Chenhui Chu, Mayu Otani, Yuta Nakashima, and Haruo Takemura. *BERT Representations for Video Question Answering*. *Proc. WACV*. 2020 (cit. on pp. 37, 39).

[Yu, 2018] Youngjae Yu, Jongseok Kim, and Gunhee Kim. *A joint sequence fusion model for video question answering and retrieval*. *Proc. ECCV*. 2018 (cit. on p. 27).

[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*. *Proc. AAAI*. 2019 (cit. on pp. 4, 26, 63, 69, 75).

[Yuan, 2021a] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, et al. *Tokens-to-token ViT: Training vision transformers from scratch on ImageNet*. *Proc. ICCV*. 2021 (cit. on p. 22).

[Yuan, 2021b] 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 p. 27).

[Yue-Hei Ng, 2015] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. *Beyond short snippets: Deep networks for video classification*. *Proc. CVPR*. 2015 (cit. on p. 25).

[Zaken, 2022] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. *BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models*. *Proc. ACL*. 2022 (cit. on pp. 19, 20, 32).

[Zellers, 2019] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. *From recognition to cognition: Visual commonsense reasoning*. *Proc. CVPR*. 2019 (cit. on p. 37).

[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*. *Proc. CVPR*. 2022 (cit. on pp. 63, 75).

[Zellers, 2021] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, et al. *MERLOT: Multimodal neural script knowledge models*. *Proc. NeurIPS*. 2021 (cit. on pp. 7, 30, 32, 62, 63, 70).

[Zhai, 2022] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, et al. *LiT: Zero-shot transfer with locked-image text tuning*. *Proc. CVPR*. 2022 (cit. on p. 27).

[Zhang, 2023] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, et al. *DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection*. *Proc. ICLR*. 2023 (cit. on p. 22).

[Zhang, 2019] Ji Zhang, Yannis Kalantidis, Marcus Rohrbach, Manohar Paluri, Ahmed Elgammal, and Mohamed Elhoseiny. *Large-scale visual relationship understanding*. *Proc. AAAI*. 2019 (cit. on p. 43).

[Zhao, 2020] Lulu Zhao, Weiran Xu, and Jun Guo. *Improving Abstractive Dialogue Summarization with Graph Structures and Topic Words*. *Proc. COLING*. 2020 (cit. on p. 40).

[Zhao, 2023] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. *Learning video representations from large language models*. *Proc. CVPR*. 2023 (cit. on p. 26).

[Zhou, 2017] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. *Places: A 10 million image database for scene recognition*. *IEEE Trans. PAMI* 40.6 (2017), pp. 1452–1464 (cit. on p. 43).

[Zhou, 2022a] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, et al. *iBOT: Image BERT Pre-Training with Online Tokenizer*. *Proc. ICLR*. 2022 (cit. on p. 24).

[Zhou, 2022b] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. *Conditional prompt learning for vision-language models*. *Proc. CVPR*. 2022 (cit. on p. 64).

[Zhou, 2022c] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. *Learning to prompt for vision-language models*. *IJCV* (2022) (cit. on p. 64).

[Zhou, 2020] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. *Unified vision-language pre-training for image captioning and vqa*. *Proc. AAAI*. 2020 (cit. on p. 27).

[Zhou, 2018] Luowei Zhou, Chenliang Xu, and Jason Corso. *Towards automatic learning of procedures from web instructional videos*. *Proc. AAAI*. 2018 (cit. on pp. 4, 26).

[Zhu, 2020] Linchao Zhu and Yi Yang. *ActBERT: Learning global-local video-text representations*. *Proc. CVPR*. 2020 (cit. on pp. 27, 29, 30).

[Zhu, 2022] Xizhou Zhu, Jinguo Zhu, Hao Li, Xiaoshi Wu, Hongsheng Li, Xiaohua Wang, et al. *Uni-perceiver: Pre-training unified architecture for generic perception for zero-shot and few-shot tasks*. *Proc. CVPR*. 2022 (cit. on p. 64).

COLLEGE DOCTORAL BRETAGNE
MATHS, TELECOMS
INFORMATIQUE, SIGNAL
SYSTEMES, ELECTRONIQUE

Université de Rennes

Titre : Réponse aux questions sur la vidéo avec supervision limitée

Mot clés : compréhension de la vidéo, réponse aux questions, apprentissage multimodal

Résumé : Le média vidéo a considérablement augmenté en volume et en diversité à l'ère numérique et cette expansion soulève la nécessité de technologies avancées de compréhension des vidéos. Poussée par cette nécessité, cette thèse explore la compréhension sémantique des vidéos en exploitant plusieurs modalités perceptuelles, comme pour le processus cognitif humain, et un apprentissage efficace avec une supervision limitée, semblable aux capacités d'apprentissage humain. Cette thèse se concentre spécifiquement sur la réponse aux questions sur les vidéos comme l'une des principales tâches de compréhension vidéo. Notre première contribution traite de la réponse aux questions sur les vidéos sur le long terme, nécessitant une compréhension du contenu vidéo étendu. Alors que les approches récentes dépendent de sources externes générées par les humains, nous traitons des données brutes pour générer des résumés vidéo. Notre contribution suivante explore la réponse aux questions sur la vidéo en *zero-shot* et en *few-shot* visant à améliorer l'apprentissage efficace à partir de données limitées en nombre. Nous exploitons la connaissance des grands modèles existants en éliminant les défis d'adaptation de ces modèles pré-entraînés à des données limitées en nombre. Nous démontrons que ces contributions améliorent considérablement les capacités des systèmes multimodaux de réponse aux questions sur la vidéo dans les contextes où les données spécifiquement annotées par l'humain sont limitées ou indisponibles.

Title: Video question answering with limited supervision

Keywords: video understanding, video question answering, multimodal learning

Abstract: Video content has significantly increased in volume and diversity in the digital era, and this expansion has highlighted the necessity for advanced video understanding technologies. Driven by this necessity, this thesis explores semantically understanding videos, leveraging multiple perceptual modes similar to human cognitive processes and efficient learning with limited supervision similar to human learning capabilities. This thesis specifically focuses on video question answering as one of the main video understanding tasks. Our first contribution addresses long-range video question answering, requiring an understanding of extended video content. While recent approaches rely on human-generated external sources, we process raw data to generate video summaries. Our following contribution explores *zero-shot* and *few-shot* video question answering, aiming to enhance efficient learning from limited data. We leverage the knowledge of existing large-scale models by eliminating challenges in adapting pre-trained models to limited data. We demonstrate that these contributions significantly enhance the capabilities of multimodal video question-answering systems, where specifically human-annotated labeled data is limited or unavailable.