Multimodal Data Efficient Learning

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

Otniel-Bogdan Mercea

aus

Arad, Rumänien

Tübingen

2024

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	16.04.2025
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter/-in:	Prof. Dr. Zeynep Akata
2. Berichterstatter/-in:	Prof. Dr. Andreas Geiger

imprs-is

machine learning
new perspectives
for science

DEPARTMENT OF
COMPUTER SCIENCE

MULTIMODAL DATA EFFICIENT LEARNING

OTNIEL-BOGDAN MERCEA

MSc in Artificial Intelligence and BEng in Computers and Information Technology

Adviser: Zeynep Akata
Full Professor, Technical University of Munich

Co-adviser: Andreas Geiger
Full Professor, University of Tübingen

Examination Committee

Chair: Hildegard Kühne
Full Professor, University of Tübingen

Members: Zeynep Akata
Full Professor, Technical University of Munich

Andreas Geiger
Full Professor, University of Tübingen

Justus Thies
Full Professor, TU Darmstadt

DOCTORATE IN COMPUTER SCIENCE

University of Tübingen and International Max Planck Research School for Intelligent Systems
, 2024

Acknowledgements

My PhD journey has been rewarding, enhancing my research skills and personal growth. Working with influential academics and industry professionals provided valuable perspectives. I am grateful for the support I received and would like to acknowledge those who helped me.

Firstly, I would like to express my gratitude to Prof. Zeynep Akata for her guidance and support during my time at the International Max Planck Research School for Intelligent Systems and the University of Tübingen. Her patience, mentorship, and wisdom have been crucial for my personal growth and helped me develop as a researcher and person. I also want to thank the International Max Planck Research School for Intelligent Systems and the University of Tübingen for providing such an excellent research environment and many resources to do cutting-edge and impactful research.

I want to thank Almut Sophia Koepke for her extensive mentorship during my PhD. I had a great time discussing many research papers and my research with her. She was always supportive and willing to help me. Her insights and expertise were crucial for pushing me on many occasions on the right track.

Furthermore, I want to thank Anurag Arnab, Alexey Gritsenko, and Cordelia Schmid for taking me on an internship at Google Research and Stefano Pellegrini, Jasper Uijlings, and Cordelia Schmid for taking me on another internship at Google DeepMind. These internships significantly broadened my research interests and offered me the rare opportunity to work in such prestigious industrial research labs.

I thank the EML group for providing such an amazing research environment. This group's collective knowledge and support were instrumental in my growth as a researcher. I also had much fun there, especially with Massi, Stephan, and Thomas, who challenged me many times at Foosball and helped me significantly improve my Foosball skills. Special thanks to Thomas, with whom I worked for a significant part of my PhD, who helped with insights and his coding expertise and, in general, made my day better.

Finally, I want to thank my parents, Iudita and Vasile, for their support throughout my life, which helped me get where I am today. Nothing would have been possible without their endless support and them teaching me to be honest and work hard. Thank you!

Abstract

Recently, unimodal models have attained good performance in many tasks. However, using one modality may not provide sufficient information in complex situations. Humans use multimodal input, such as vision and hearing, to act in the real world. Similarly, this thesis proposes systems that use multimodal input for video classification and visual-language learning. However, multimodal models need significant amounts of qualitative paired data, which is costly and time-consuming to gather. At the same time, humans require very few training samples, even for the most complex tasks. Given these aspects, this thesis addresses the problem of multimodal data efficient learning.

Firstly, this thesis studies the audio-visual video classification task in generalized zero- and few-shot learning settings. It introduces new training and evaluation protocols, dataset splits, and baselines. Using transformers to fuse the audio and visual modalities leads to higher performance than prior works. Furthermore, typical full-attention does not lead to the best results, and new attention patterns are developed. New loss functions are essential for increasing the performance of both settings. Moreover, the performance in few-shot learning is further improved by using a diffusion model to generate synthetic audio-visual features for the novel classes.

The second task is video-adverb retrieval, which is studied both when plenty of training data is available and in the zero-shot learning scenario. The goal is to improve the text embeddings using a residual gating mechanism and a new training objective. New zero-shot splits are also introduced to facilitate a more comprehensive evaluation.

Finally, this thesis uses multimodal large language models (MLLMs) to focus on visual-language learning. This task studies the ability of MLLMs to adapt the communication on the fly given a conversation partner by using very few interactions. This work provides a general framework for testing this ability for multiple agents, providing insights about their strengths and weaknesses. It turns out that the ability to adapt the communication to different partners with different comprehension abilities is already present in the current MLLMs.

Zusammenfassung

Dies ist eine „Google Translate“-Übersetzung der englischen Version!

In letzter Zeit haben unimodale Modelle bei vielen Aufgaben eine gute Leistung erzielt. Die Verwendung einer einzigen Modalität liefert jedoch in komplexen Situationen möglicherweise nicht genügend Informationen. Der Mensch nutzt multimodalen Input, wie Sehen und Hören, um in der realen Welt zu handeln. In ähnlicher Weise werden in dieser Arbeit Systeme vorgeschlagen, die multimodalen Input für die Videoklassifizierung und das Lernen von visueller Sprache nutzen. Multimodale Modelle benötigen jedoch große Mengen an qualitativen, gepaarten Daten, deren Erfassung kostspielig und zeitaufwändig ist. Gleichzeitig benötigt der Mensch selbst für die komplexesten Aufgaben nur sehr wenige Trainingsbeispiele. Angesichts dieser Aspekte befasst sich diese Arbeit mit dem Problem des effizienten Lernens multimodaler Daten.

Zunächst wird in dieser Arbeit die audiovisuelle Videoklassifikation in verallgemeinerten Zero- und Little-Shot-Lernsettings untersucht. Es werden neue Trainings- und Evaluierungsprotokolle, Datensatzsplits und Baselines vorgestellt. Die Verwendung von Transformatoren zur Verschmelzung der Audio- und visuellen Modalitäten führt zu einer höheren Leistung als in früheren Arbeiten. Außerdem führt die typische volle Aufmerksamkeit nicht zu den besten Ergebnissen, und es werden neue Aufmerksamkeitsmuster entwickelt. Neue Verlustfunktionen sind wichtig, um die Leistung beider Einstellungen zu verbessern. Darüber hinaus wird die Leistung beim „few-shot learning“ durch die Verwendung eines Diffusionsmodells zur Erzeugung synthetischer audiovisueller Merkmale für die neuen Klassen weiter verbessert.

Die zweite Aufgabe ist die Abfrage von Videoadverbien, die sowohl bei einer großen Anzahl von Trainingsdaten als auch im Zero-Shot-Learning-Szenario untersucht wird. Ziel ist es, die Texteinbettungen mit Hilfe eines Residual-Gating-Mechanismus und eines neuen Trainingsziels zu verbessern. Es werden auch neue Zero-Shot-Splits eingeführt, um eine umfassendere Bewertung zu ermöglichen.

Schließlich werden in dieser Arbeit multimodale große Sprachmodelle (MLLMs) verwendet, um sich auf das Lernen visueller Sprache zu konzentrieren. Bei dieser Aufgabe

wird die Fähigkeit von MLLMs untersucht, die Kommunikation bei einem Gesprächspartner mit sehr wenigen Interaktionen anzupassen. Diese Arbeit bietet einen allgemeinen Rahmen für das Testen dieser Fähigkeit für mehrere Agenten und liefert Erkenntnisse über deren Stärken und Schwächen. Es zeigt sich, dass die Fähigkeit, die Kommunikation an verschiedene Partner mit unterschiedlichen Verständnisfähigkeiten anzupassen, in den aktuellen MLLMs bereits vorhanden ist.

Contents

List of Figures

List of Tables

Listings

1

Introduction

This thesis aims to address the task of multimodal data efficient learning in various settings. Sec. 1.1 briefly presents why multimodal learning is such an important problem, and it provides a summary of the tasks that are tackled throughout this thesis, namely audio-visual learning, video-adverb retrieval and adaptation of communication in the context of multimodal large language models. Next, Sec. 1.2, 1.3 and 1.4 motivate each of these tasks by looking at real-world scenarios along with providing a brief description. Furthermore, Sec. 1.5 introduces the concept of data efficient learning, emphasizing its significance in general and providing additional details on how data efficient learning is employed in each task. Sec. 1.6 emphasizes the main contributions of this thesis. The contributions can be divided into two general categories. One line of contributions is related to defining multiple settings in the context of multimodal data efficient learning by introducing benchmarks, baselines, and training and evaluation protocols. The other line of contributions focuses on developing models to solve these tasks better and attain state-of-the-art performance. Finally, Sec. 1.7 provides an outline of the chapters by briefly mentioning each chapter's content, and the venue where these chapters were published.

1.1 Multimodal learning

Since 2012, with the introduction of AlexNet [116], deep learning has made tremendous progress. However, much of the focus was on unimodal models, which take a single modality as input. While these models can sometimes perform well in unimodal tasks, using unimodal input drastically reduces the amount of information these models can process. On the other hand, multimodal learning holds immense potential. It strives to replicate how humans acquire information about the natural world through multiple senses to better understand the world's state and make better decisions. The popularity of multimodal learning increased significantly with the introduction of CLIP [197], which showed impressive results in many tasks related to vision and language. The current multimodal learning methods are mainly enabled by the introduction of the transformer architecture [241], which takes as input tokens. As these tokens can be obtained from

different modalities, such as audio and visual, the transformer architecture can be considered modality agnostic, processing information from any modality with minimal changes, given that the input is in a token format.

However, most of the past research has focused on visual-language learning, with the visual modality mainly represented by images. In contrast, other modalities, such as video and audio, were underexplored. This thesis aims to tackle multimodal learning while also to focus extensively on these less explored tasks, which are of significant importance in the field. A substantial part of the thesis is dedicated to the video modality in different settings, such as audio-visual learning in the video domain and video-adverb retrieval. Furthermore, the already popular visual-language learning task, using images for the visual modality, is tackled in the novel setting of online communication adaptation in the context of multimodal large language models (MLLMs). Next, Sec. 1.2, 1.3 and 1.4 are going to introduce each of the multimodal tasks presented in this thesis.

1.2 Audio-visual learning in the video domain

1.2.1 Task description

The setting of audio-visual learning in the video domain corresponds to Chapters 2, 3, 4 and 5. In the following, the task will be described briefly, and a motivation of why this task is essential is given.

For Chapters 2, 3, and 5, the problem is effectively solved by learning to match the audio-visual representation of a video to the text representation corresponding to the correct class. Thus, it can be said that this way of learning is similar to a retrieval task, where, given an audio-visual representation, the network needs to retrieve the text representation corresponding to the correct class. This is done by projecting both the audio-visual and the text representations in a shared embedding space, where the two are matched. This is advantageous during test time, as the method can also be applied to classes never seen during training, which is the central assumption in these three chapters.

On the other hand, in Chapter 4, the assumption is that during test time, the method will only encounter classes already seen during training. Thus, the task in this chapter can be treated like a traditional classification problem, where the test classes are predefined at training time. Compared to the previously mentioned chapters, the main change is the output modality, which is not text anymore but is represented by a probability distribution over all classes. However, the text modality is still used as a conditioning input to a diffusion model for synthetic data generation, as explained in Chapter 4.

Furthermore, Chapters 3 and 4 also tackle this problem by using the temporal dimension from the videos, which gives more information about how an action is done, leading to a better performance. On the other hand, Chapters 2 and 5 use temporally averaged features, thus removing the temporal information in the videos but being more computationally efficient.

1.2.2 Motivation

Using a single modality can create ambiguities in many scenarios. Imagine a person playing the cello and producing bird sounds. Just by listening to the audio, it can be argued that the correct class is bird singing. However, once both modalities are available, it can be understood that a person is playing the cello and producing bird sounds. As a result, the video class changes significantly from a bird singing to a person playing a musical instrument by adding a modality that provides more information. This depicts an example of ambiguities arising from missing the visual modality.

Another example would be a self-driving car in front of an ambulance. Depending on the distance between the ambulance and the self-driving car, the visual sensors may not be able to detect the ambulance yet. In the ideal scenario, the self-driving car should use audio sensors to detect the sound of the siren as the ambulance approaches and start making maneuvers that will let the ambulance pass unhindered. However, if the self-driving car cannot detect sounds, it will continue to drive normally until the visual sensors detect the ambulance. At this point, the self-driving vehicle may try to move away from the ambulance, but it may be too late, and the ambulance may be hindered. This is an example where missing the audio modality may give a wrong perspective on the environment state, leading to poor actions.

Often, more than multimodal input is needed. This is especially true for classes that require movement cues. Consider a gymnastics class as an example. Imagine a single photo of a person balancing on their hands. This could represent different activities, such as holding a static handstand, performing a handstand walk, doing handstand push-ups, or executing a handstand pirouette. The audio would provide little information, as these actions are usually based on visual and temporal cues. However, additional frames could provide enough temporal context to make the action obvious.

1.3 Video-adverb retrieval

1.3.1 Task description

The video-adverb retrieval task is addressed in Chapter 6, where given a video and an action, the goal is to retrieve the adverb that best describes the given action in the video. This is done by mapping the video and action-adverb representations in a shared embedding space where the matching occurs, similar to Sec. 1.2. The given action also influences the video representation, as it conditions the attention mechanism to attend to the most relevant frames for that particular action.

1.3.2 Motivation

Nowadays, most works in multimodal and unimodal video understanding are concerned with predicting the correct action in the video. While identifying the proper action is

generally helpful, having additional information about the action can provide significantly more insights to a user or other downstream systems. One example is text-video retrieval, where the user may want to retrieve videos based on a specific concept description, such as running fast. A different example can be related to human-computer interaction, such as VR applications, where understanding fine-grained gestures, such as moving the hand in various manners, could lead to a more nuanced computer control. Similar models could also be employed in elder care, such as fall prevention, which detects if an older adult is staggering, indicating potential health issues that may lead to collapse.

1.4 Communication adaptation on the fly in MLLMs

1.4.1 Task description

The task of communication adaptation on the fly in MLLMs is presented in Chapter 7. This task presents two MLLMs that must understand each other to perform well in referring expression settings. This work tackles two referring expression settings, which will be shown next. In the Referring Expression Identification (REI) setting, the speaker MLLM describes an image from two given images without mentioning which image is described. Given the two pictures and the speaker MLLM's text description, the listener MLLM has to guess which image was described. On the other hand, in the Referring Expression Segmentation (RES) setting, the goal is for the speaker MLLM to describe an object from a given image, and the listener MLLM is tasked with segmenting the described object. In both settings, the goal is to adapt the speaker MLLM's communication based on the listener MLLM's responses so that the speaker MLLM can generate descriptions explicitly tailored for the listener MLLM. This is crucial, as different MLLMs may have different understanding abilities. Moreover, MLLMs may have perceptual weaknesses, which are simulated by blurred vision and color blindness in this work.

1.4.2 Motivation

Communication adaptation is critical, and humans routinely do it in conversations—the communication changes based on the understanding capabilities or visual impairments of the communication partners. For example, when describing something or instructing a person with color blindness, it is essential to avoid focusing on colors, as the interlocutor can not fully understand these aspects depending on the degree of color blindness, which may lead to confusion or ambiguity. To create intelligent agents that can assist humans and be inclusive, one has to embed in these agents the ability to adapt the communication based on the interlocutor. However, it is plausible to envision a future where agents interact with one another to automate many tasks. Different models, with varying levels of intelligence and understanding capabilities, may represent each of these agents. To complete the tasks, these agents will need to learn how to cooperate, and a critical step in this direction is to adapt their communication to each other.

1.5 Multimodal data efficient learning

The previous sections introduced the multimodal setting along with multiple tasks and provided a brief description and motivation for each task. This section aims to look at these multimodal tasks from the perspective of data efficient learning. It provides a short motivation for why data efficient learning is important and details how it is employed in each of these tasks.

Data efficient learning is a crucial topic nowadays as the models that achieve state-of-the-art performance are usually huge and require tremendous amounts of training data. Collecting and annotating such data and gathering enough computational resources to train these large models is expensive. Furthermore, this can only scale for a while, as at some point, these systems will run out of qualitative datasets. On the other hand, humans can accomplish complex tasks and perform excellently without massive amounts of training data. Thus, designing systems that can learn as efficiently as humans while achieving similar or even better performance is appealing and desirable. As a result, this thesis mainly explores multimodal learning in the context of data efficient learning, as will be shown next.

Audio-visual learning in the video domain is studied in generalized zero- and few-shot learning settings. The zero-shot learning setting assumes no knowledge of the test classes during training, and the model needs to generalize to classes never seen during training, called unseen classes. In the generalized zero-shot learning setting, the model must also be able to classify the classes seen during training in addition to the unseen classes. This is a more realistic scenario, as the model needs to be able to classify both types of classes. The difficulty in this setting arises from the fact that the model has a significant bias towards the seen classes, and most of the time, it will classify the samples from unseen classes as belonging to the seen classes. Thus, these systems must be designed to reduce or completely mitigate this strong bias towards the seen classes while correctly classifying both types of classes.

On the other hand, for the few-shot learning setting, it is assumed that there are classes with many training samples called base classes and classes with very few training samples (e.g., 1,5 or 10 samples) called novel classes. During training, the system is tasked with learning as much as possible from the limited number of samples in the novel classes while leveraging the knowledge acquired from the base classes. During inference, the system encounters samples from both base and novel classes in the generalized few-shot case and only samples from the novel classes in the traditional few-shot case. The generalized few-shot learning setting strikes a balance between the generalized zero-shot and plenty-of-data scenarios, and it is the scenario most commonly encountered in real-world applications.

For video-adverb retrieval, the task is studied in both zero-shot learning and the traditional setting, where it is assumed that there is plenty of training data. In the zero-shot learning setting, it is assumed that the adverb and action testing combinations were never observed during training, which helps measure the generalisability of these networks

to unseen concepts. An important aspect to mention here is that the individual adverbs and actions were already observed during training but in different combinations than those encountered during testing in zero-shot. For the traditional case (non-zero-shot), it is assumed that every testing combination of adverb and action was already observed in the training dataset.

The focus of the communication adaptation task in MLLMs is online adaptation. There is no prior training dataset, and the speaker MLLM needs to learn how to adapt the communication on the fly by learning from interactions with the listener MLLM. An interaction is defined by the speaker MLLM describing either the image (in the REI setting) or an object (in the RES setting), and the listener MLLM guessing which image or object was described. As these interactions can be costly and time-consuming, the number of interactions is limited to less than 2000 per experiment. This contrasts with prior works, which adapt MLLMs to human feedback by using tens of thousands of training samples annotated by humans and trained reward models. In light of this fact, this work also goes in the direction of data efficient learning by learning from very few examples in an online fashion.

1.6 Contributions

This thesis primarily concerns multimodal data efficient learning. Various tasks are presented, such as audio-visual generalized zero- and few-shot learning, video-adverb retrieval, and communication adaptation in MLLMs. As previously mentioned, all these tasks are also tackled in a data efficient manner. While most of these works try to improve state-of-the-art performance, significant emphasis is placed on defining new settings and providing a starting point for future research. The following paragraphs discuss two main lines of contributions that can be observed throughout this thesis.

The establishment of multimodal data efficient settings. The first essential step in addressing all tasks presented in this thesis is defining the setting of multimodal data efficient learning. Audio-visual learning is a complex problem that requires paired data for the audio and visual modalities. However, as previously mentioned, this can be very expensive or infeasible. Thus, an objective is to create systems that can learn from as little data as possible. The main settings tested in audio-visual learning are generalized zero- and few-shot learning. At a closer inspection, it was observed that both these settings were underexplored for audio-visual video classification. In particular, for generalized zero-shot learning, to the best of our knowledge, this problem was only tackled by two prior works [152, 185]. However, these previous works had issues with data leakage from the validation set to the test set because they had the same classes in both the validation and the test sets. The hyperparameters were chosen for the validation classes, but in doing so, they leaked information about the unseen classes in the test set, making the unseen classes no longer genuinely unseen. Furthermore, the dataset employed in [152, 185] was relatively small, and these works used a single dataset. As a result, the first step in

tackling the audio-visual generalized zero-shot learning involved fixing these issues by correctly formalizing the setting and providing multiple more extensive datasets.

Moreover, the setting of generalized audio-visual zero-shot learning is tackled in various ways. For each of these ways, specific baselines and benchmarks are provided that allow for an extensive and fair comparison. Some works only focus on temporally averaged features, which is less computationally expensive. However, this presents a significant limitation as it can lead to ambiguities for classes that require temporal information, as explained in Sec. 1.2. On the other hand, some works focus on using temporal features, leading to a much better performance but at a higher computational cost. Furthermore, different features extractors are employed in this setting, ranging from older features extractors, which are more consistent with prior zero-shot learning works in fields such as unimodal zero-shot video classification, to newer ones, such as CLIP [197] for the visual modality and CLAP [153] for the audio modality. The main advantage of using these newer feature extractors is obtaining enhanced embeddings, which leads to better performance. These feature extractors comprise two encoders: a visual or audio encoder and a text encoder. These text encoders were pre-trained with a visual (CLIP) or an audio (CLAP) model, encoding slightly different information specific to the corresponding modality. Combining the text representations provided by these two text encoders can lead to an enhanced text representation, which was never tried before in the audio-visual generalized zero-shot learning setting.

While generalized zero-shot learning is an exciting and extreme setting for benchmarking models' generalizability capabilities, generalized few-shot learning is another exciting setting closer to real-world scenarios. As a result, this thesis proposes to formalize this setting by providing new benchmarks and baselines for extensive evaluation, along with training and evaluation protocols. One important thing to mention is that the generalized zero- and few-shot learning benchmarks introduced in this thesis are compatible. More specifically, the unseen classes in the generalized zero-shot learning benchmark become novel classes in the generalized few-shot learning benchmark. This is very advantageous, as new models can be tested comprehensively and consistently in both scenarios, giving a much better idea of how a model would perform in the general setting of low-shot learning, which is considered to be composed of zero- and few-shot learning.

Many works, when they classify actions in videos, do not specifically tackle the problem of classifying how an action is done, which is quite essential, as shown in Sec. 1.3. This thesis also studies the task of video-adverb retrieval to address this problem. A significant focus is put on the zero-shot learning setting by providing additional zero-shot learning benchmarks. The zero-shot setting becomes indispensable as this task requires fine-grained classification, such as classifying the adverb corresponding to a given video and action. In such a fine-grained classification task, the number of valid concepts can be very high, and depending on the complexity of the problem, all these concepts may never be fully captured in a dataset. Zero-shot learning is likely the case these systems will encounter during real-world usage, and more focus should be put into this setting.

Finally, there has been a significant increase in the relevance of MLLMs and their usefulness in individuals' lives nowadays. However, one significant problem with MLLMs is that they are usually trained in a very general way by using general reward models without taking into account the capabilities or disabilities of the interlocutor. As previously shown in Sec. 1.4, in order to make these technologies more inclusive, such that all the individuals in the society can use them, regardless of their understanding capabilities or disabilities, one needs to be able to adapt these systems to the interlocutor. This adaptation must be seamless, and the system must quickly become relevant to the user. This thesis focuses on making this adaptation possible by using as few interactions as possible and adapting the MLLMs online based on the interactions with another MLLM. Moreover, as this setting is unexplored in the context of MLLMs, this thesis also defines the setting by introducing benchmarks and baselines to facilitate future research.

Multimodal learning. The previous few paragraphs mentioned the contributions of this thesis regarding the establishment of multiple multimodal data efficient settings. However, establishing a new setting without initial research into developing a method that can solve that setting to a certain extent or provide additional insights into the current state-of-the-art is not satisfactory. As a result, along with defining these settings, this thesis also develops methods for attaining state-of-the-art performance. The only exception is the work that studies the communication adaptation task, where the goal is to evaluate the capabilities of current models and adaptation algorithms instead of introducing new ones. The visual, audio, and text modalities represent the three modalities used throughout this thesis. The works in this thesis use audio, visual, and text modalities for audio-visual generalized zero- and few-shot learning. In contrast, only visual and text modalities are used for video-adverb retrieval and communication adaptation in MLLMs. Furthermore, the visual modality is represented by images in the communication adaptation task and by videos in the rest of the works.

The novelty of the methods usually consists of adding novel loss functions and improving the fusion mechanism to better combine the information from the multimodal input. One aspect observed throughout these works is that fusing information from multiple modalities is a complex problem. As the transformer architecture is modality agnostic, one naive solution is to use the transformer architecture as a fusion mechanism with typical full attention. However, the works presented in this thesis show that this is suboptimal, as better attention mechanisms can be developed. In the context of audio-visual generalized zero-shot learning, when using temporally averaged features, the model is constrained to only two tokens (one for each modality), and the only type of attention that can be applied is cross-modal attention. However, for the model that uses temporal features in the audio-visual generalized zero-shot learning, it was observed that using only cross-modal attention is significantly better than full-attention. Even more, for generalized few-shot learning, a hybrid attention composed of intra-modality and full-attention performs better than the typical full-attention. A transformer architecture was sometimes unnecessary

(see Chapter 5), as simpler architectures could achieve state-of-the-art performance. Moreover, for many of the works presented in this thesis, the design of the loss functions was essential to take advantage of different types of attention.

Another architectural aspect studied in some of this thesis's works was obtaining better text representations. In the case of audio-visual generalized zero-shot learning, as mentioned in the previous paragraphs, combining multiple text representations from the CLIP and CLAP text encoders proved to be better. Furthermore, an essential aspect of video-adverb retrieval was combining the action and adverb text representation using a residual gated mechanism. Getting better text embeddings is essential as the classification in many of these tasks can be seen as a retrieval task where the visual or audio-visual representation is matched to a text representation, and both these representations are equally crucial for a good performance.

Finally, no architectural novelty was involved in the communication adaptation task, but the goal was to test different adaptation algorithms and MLLMs to observe their behavior. It was observed that no current adaptation algorithm and MLLM perform the best in every setting. However, from all the experiments, on average, some MLLMs and adaptation algorithms perform better than others, and the ability to adapt the communication of the speaker to the listener is already present in these MLLMs. Finally, the best adaptation performance achieved is still far from the ground truth, which shows that this task is still challenging for the current state-of-the-art, and more research effort needs to be put into making these methods perform better.

1.7 Outline

This section outlines the thesis, emphasizing each chapter's acceptance venue and the main idea. This thesis is composed of 8 chapters, as shown below. Fig. 1.1 provides a conceptual relation between the chapters of this thesis. For a summary of contributions for each author in these works, please check Sec. G.2.

Chapter 1: Thesis introduction.

This chapter motivates this study by explaining why the problem of multimodal data efficient learning is essential and briefly introduces each task. Additionally, this chapter presents the contributions of this thesis along with an outline of the chapters, which can be visualized in Fig. 1.1.

Chapter 2: Audio-visual Generalised Zero-shot Learning with Cross-modal Attention and Language.

This chapter describes the initial work done in audio-visual generalized zero-shot learning. In this work, one goal was to formalize the setting by introducing benchmarks, baselines, and training and evaluation protocols. Moreover, a new transformer architecture was proposed for fusing information from both modalities, and novel loss functions were introduced. This chapter was accepted at CVPR 2022 [157].

Chapter 3: Temporal and cross-modal attention for audio-visual zero-shot learning.

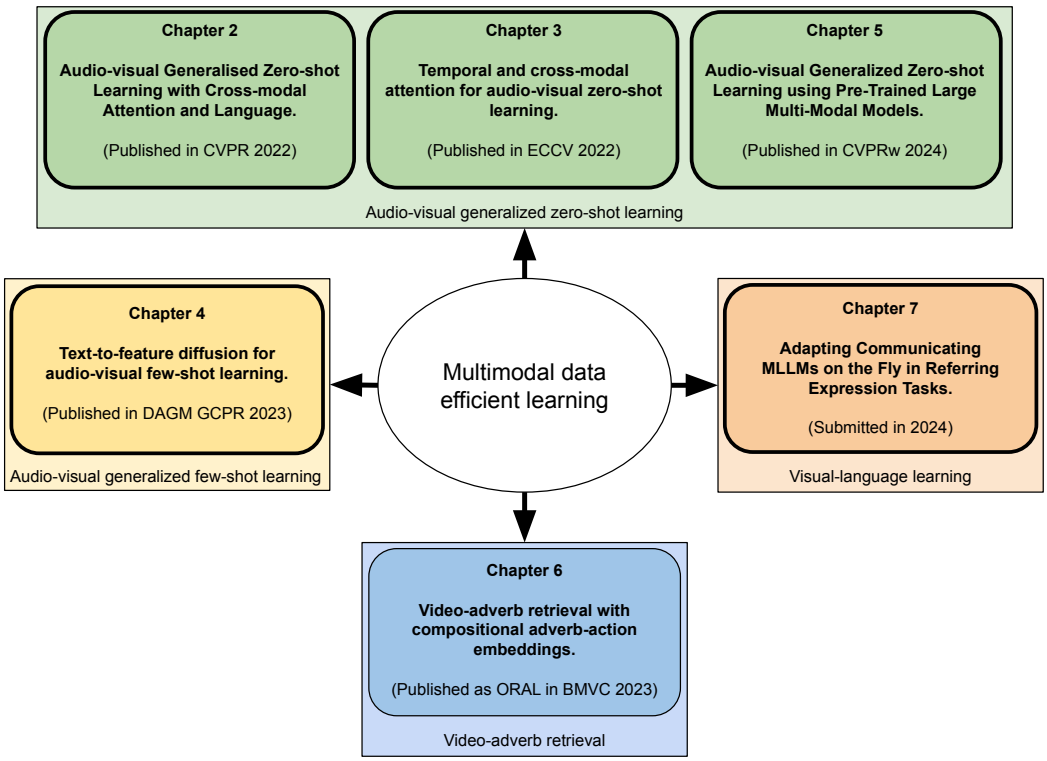


Figure 1.1: This figure depicts the relationships between this thesis's chapters and the venues where they were published. Chapter 2, 3, and 5 tackle the audio-visual generalized zero-shot learning problem. Chapter 4 studies the problem of audio-visual generalized few-shot learning. Chapter 6 deals with the problem of video-adverb retrieval. Finally, Chapter 7 tackles the task of communication adaptation in MLLMs.

This chapter extends the work presented in Chapter 2. The problem studied in this chapter is audio-visual generalized zero-shot learning, with a significant focus on incorporating temporal information into the model. Temporal information aims to reduce ambiguities in the input that may arise for some classes. A new transformer architecture is proposed based on a novel cross-modal attention mechanism, where tokens from one modality can only attend to tokens from the other modality. Furthermore, a novel loss function, which is less complex than the ones used in the previous works, is used to take advantage of this architecture, leading to state-of-the-art performance. This chapter was accepted at ECCV 2022 [155].

Chapter 4: Text-to-feature diffusion for audio-visual few-shot learning.

This chapter switched focus from audio-visual generalized zero-shot learning to audio-visual generalized few-shot learning. To the best of our knowledge, this work is the first to tackle this setting in the video classification domain. As a result, it formalizes the setting and introduces new benchmarks, baselines, and training and evaluation protocols. Moreover, a new transformer model is proposed based on hybrid attention, composed of intra-modality attention for the first few layers and full-attention for the remaining layers. A diffusion model is also employed to generate synthetic audio-visual samples to

augment the training dataset. This chapter was accepted at DAGM GCPR 2023 [156].

Chapter 5: Audio-Visual Generalized Zero-Shot Learning using Pre-Trained Large Multi-Modal Models.

This chapter revisits the problem of audio-visual generalized zero-shot learning presented in Chapter 2. The goal is to introduce a more modern set of audio and visual features based on large image-text and audio-text pre-trained models. As the audio and visual encoders have corresponding pre-trained text encoders, both these text encoders can be used to obtain a better text representation for the class names. Moreover, the proposed architecture takes better advantage of the newly provided features, reaching state-of-the-art performance. This chapter was accepted at CVPR 2024 L3DIVU workshop [119].

Chapter 6: Video-adverb retrieval with compositional adverb-action embeddings.

This chapter studies the problem of video-adverb retrieval. One contribution of this work is introducing a new way of fusing the action and adverb text embeddings based on a residual gated mechanism, along with a new training objective, leading to better performance. Furthermore, another contribution is the introduction of additional zero-shot dataset splits that allow for a more extensive evaluation of the proposed and the prior works in this setting. This chapter was accepted as ORAL at BMVC 2023 [96].

Chapter 7: Adapting Communicating MLLMs on the Fly in Referring Expression Tasks.

This chapter tackles the problem of multimodal learning in the context of communication adaptation between MLLMs using different MLLMs and adaptation algorithms. Insights into these MLLMs are provided, along with experiments designed to test the communication adaptation when the listener MLLM is visually impaired. Finally, this work tackles this problem in the online setting, meaning no prior dataset is available, and the training dataset is composed solely of online interactions between the MLLMs. As these online interactions are costly and time-consuming, only a few interactions are used, aiming for more data efficient training. This chapter was submitted in 2024.

Chapter 8: Thesis discussion and Conclusion.

Finally, this chapter concludes and provides more insights into these works' contributions. Additionally, there is an extended discussion focused on how the contributions from these chapters link to one another. Finally, the limitations of these works are also explored, along with some possible solutions.

2

Audio-visual Generalised Zero-shot Learning with Cross-modal Attention and Language

Learning to classify video data from classes not included in the training data, i.e. video-based zero-shot learning, is challenging. We conjecture that the natural alignment between the audio and visual modalities in video data provides a rich training signal for learning discriminative multi-modal representations. Focusing on the relatively underexplored task of audio-visual zero-shot learning, we propose to learn multi-modal representations from audio-visual data using cross-modal attention and exploit textual label embeddings for transferring knowledge from seen classes to unseen classes. Taking this one step further, in our generalised audio-visual zero-shot learning setting, we include all the training classes in the test-time search space which act as distractors and increase the difficulty while making the setting more realistic. Due to the lack of a unified benchmark in this domain, we introduce a (generalised) zero-shot learning benchmark on three audio-visual datasets of varying sizes and difficulty, VGGSound, UCF, and ActivityNet, ensuring that the unseen test classes do not appear in the dataset used for supervised training of the backbone deep models. Comparing multiple relevant and recent methods, we demonstrate that our proposed AVCA model achieves state-of-the-art performance on all three datasets. Code and data are available at https://github.com/ExplainableML/AVCA-GZSL.

2.1 Introduction

Most zero-shot learning (ZSL) methods developed for image classification [9, 10, 207, 211, 242, 278] and action recognition [28, 34, 87, 280] only use unimodal input, e.g. images. However, humans leverage multi-modal sensory inputs in their everyday activities. Imagine the situation in which the sound of a dog barking is audible but the dog is visually occluded. In this case, we cannot understand the scene when relying on visual information alone. Using multiple modalities, such as vision and sound, allows to gather context and capture complementary information. Similarly, using both visual and audio information

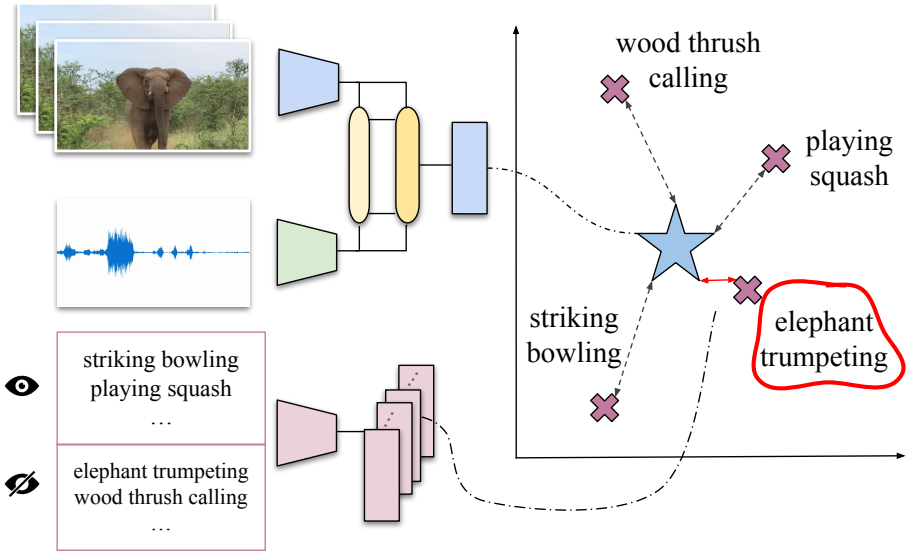


Figure 2.1: Our audio-visual (generalised) ZSL framework aligns an audio-visual embedding with the corresponding textual label embedding via cross-modal attention. It can classify videos from previously unseen classes (e.g. elephant trumpeting) by predicting the class (red) whose textual label embedding (purple cross) is closest to the audio-visual embedding (blue star).

allows for a richer training signal for learning frameworks. This paper investigates the challenging task of (generalised) ZSL with multi-modal audio-visual data by leveraging the natural alignment of audio and visual information in videos.

Recently, [152, 185] have explored the task of zero-shot video recognition using multi-modal visual and audio information as inputs. However, the AudioSetZSL dataset [185] used for this, contains an overlap between the classes used for validation and testing. This results in learning stronger representations for classes overlapping with the training and validation sets (which covers all the classes in this dataset) and hinders the model's capability to learn sufficiently generalisable representations that allow information transfer. In real-world applications, such models perform well on seen classes, but poorly on previously truly unseen classes. In this work, we propose three benchmarks of varying size and difficulty curated from the VGGSound [45], UCF101 [219], and ActivityNet [90] datasets that could act as a unified and challenging playground for Generalised ZSL (GZSL) and ZSL research in the audio-visual domain. We suggest using audio and visual features extracted using SeLaVi [21] pretrained using self-supervision. Throughout this work, we use features that were obtained from training in a self-supervised fashion to reduce the information leakage from supervised pre-training to the zero-shot task which has been identified as a problem in other ZSL benchmarks [34].

We tackle the audio-visual generalised zero-shot learning task with our Audio-Visual Cross-Attention (AVCA) framework which is trained to align a rich learnt audio-visual representation with textual label embeddings. Our multi-stream architecture contains an audio and a visual branch which exchange information using cross-attention between

the two modalities. AVCA is computationally lightweight and efficient since it uses audio and visual features extracted from pretrained networks as inputs instead of raw audio and image data. Our proposed framework is trained using multiple novel loss functions that are based on triplet losses and a regularisation loss that ensures that salient unimodal information is preserved in the learnt multi-modal representations. Our experiments show that AVCA achieves state-of-the-art performance on the three introduced benchmark datasets. We show that using multi-modal input data leads to stronger (G)ZSL performance than using unimodal data.

To summarise, our contributions are as follows: (1) We introduce three novel benchmarks for audio-visual (generalised) zero-shot learning curated from the VGGSound, UCF101, and ActivityNet datasets; (2) We propose AVCA, a cross-modal model for audio-visual (G)ZSL which leverages cross-modal attention between audio and visual information; (3) We show that AVCA yields state-of-the-art performance on all proposed audio-visual (G)ZSL benchmarks, outperforming the state-of-the-art unimodal and multi-modal zero-shot learning methods. Furthermore, we provide a qualitative analysis of the learnt multi-modal embedding space, demonstrating well-separated clustering for both seen and unseen classes.

2.2 Related Work

We review audio-visual learning, ZSL with image, video and audio data, and audio-visual ZSL.

Audio-visual learning. Audio-visual learning has enabled tremendous progress for numerous applications, such as for separating and localising sounds in videos [4, 7, 19, 44, 76, 182, 196, 235, 238, 274, 290, 291, 298], audio-visual synchronisation [43, 54, 67, 108], person-clustering in videos [35], (visual) speech and speaker recognition [5, 6, 170], spotting of spoken keywords [166, 194], audio synthesis using visual information [75, 79, 112, 113, 174, 222, 223, 293], and audio-driven image synthesis [102, 256]. Additionally, the natural alignment between audio and visual data in videos has been leveraged to learn powerful audio-visual representations for video or audio classification [15, 21, 22, 50, 51, 114, 171, 183, 184, 187, 268]. In contrast to those methods, we consider the ZSL setting for classification.

ZSL with images, videos and audio. Recently, numerous image-based generative ZSL methods have been proposed [175, 211, 242, 266, 267, 301, 302]. Their drawback is that the unseen classes need to be known a priori. In contrast, non-generative methods [9, 10, 73, 111, 207, 265, 278, 279] learn a mapping from input features to semantics of the classes (*e.g.* textual class label embeddings). Our AVCA model also learns to map its inputs to textual embeddings, but it leverages cross-attention between the audio and visual input modalities rather than using only visual inputs.

Video-based ZSL has been addressed by multiple recent works [28, 34, 83, 87, 205, 249,

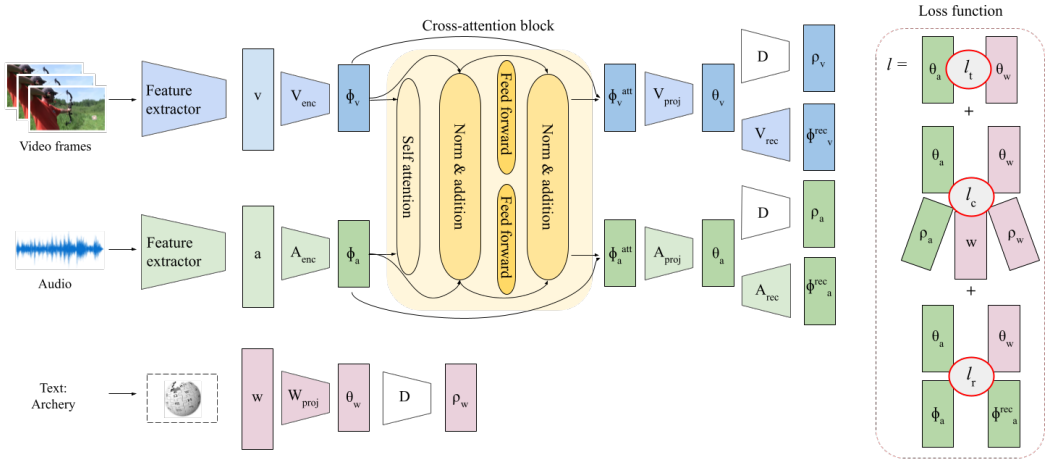


Figure 2.2: Our Audio-Visual Cross Attention (AVCA) model takes visual and audio features as inputs. A cross-attention block allows the sharing of information across modalities. The outputs of the two model branches are trained to be aligned with their corresponding textual label embedding using losses illustrated on the right-hand side. Negative samples for the contrastive loss functions are obtained using visual and audio inputs from different videos which do not share semantic information. We only show losses that involve the audio branch, those for the visual branch are similar. At test time, the class prediction is obtained by determining the class for which θ_w is closest to θ_v.

280]. Using features extracted from pretrained networks results in computationally more feasible frameworks [28, 87, 249] than training end-to-end [34]. Our model also takes pre-extracted audio and visual features as inputs, resulting in a computationally efficient framework. In order to consider a pure ZSL setting when using pre-extracted features, the overlap between classes used for supervised pre-training of the feature extractors and unseen classes has to be removed [34, 83, 205]. This was not done in some of the previous works (*e.g.* [28, 87, 249, 300]). In contrast, we propose three benchmarks for audio-visual (G)ZSL on multi-modal audio-visual video datasets with no overlap between classes used for supervised pre-training and unseen classes.

Methods for zero-shot audio classification [271, 272] also used textual sound class embeddings (*e.g.* word2vec [161], BERT [59], or GloVe [189]) or descriptions. [52] investigate zero-shot music classification and tagging with word2vec embeddings and human-labeled attribute information (e.g. the presence or absence of musical instruments). For our AVCA model, we do not use any attribute information, but instead leverage the semantic alignment between audio and visual information in addition to textual label embeddings.

Audio-visual ZSL. Recently, [152, 185] proposed frameworks that consider the task of GZSL from audio-visual data. AVGZSLNet [152] uses late fusion on the AudioSetZSL dataset [185] to combine information from the two modalities. Instead, and also different to other audio-visual frameworks [234, 282] that use a simple dot-product operation for cross-attention, we use a transformer-based cross-attention mechanism. This allows for early and efficient sharing of multi-modal information, which is further encouraged

by our proposed loss functions. Furthermore, the AudioSetZSL dataset [185] does not include a validation split with unseen validation classes. Hence, [152, 185] select the GZSL hyperparameters directly on the (unseen) test classes. Furthermore, the AudioSetZSL dataset is comparatively small; it uses only 10 test classes as unseen classes. To allow for evaluation of audio-visual ZSL at larger scale and in a pure GZSL setting, we propose new benchmarks on three different audio-visual video datasets. Our proposed benchmarks are suitable for both the GZSL and ZSL tasks.

2.3 Audio-Visual Cross Attention (AVCA)

The goal of audio-visual ZSL from video data is to learn to recognise videos from unseen classes (U), *i.e.* classes that were not seen during training. In the GZSL setting, the test set contains not only samples from unseen classes, but also from seen classes (S). This makes GZSL more challenging and more closely aligned with real-world learning tasks.

More formally, we denote the training set consisting only of samples from seen classes by $S = (v_i^s, a_i^s, y_i^s)_{i\in\{1,\cdots,N\}}$, where v_i^s, a_i^s are visual and audio features respectively, y_i^s is the corresponding ground-truth class j, and N is the number of samples in the training set. We refer to the class-level text embedding for class j as w_j^s. The goal is to learn a function $h : (v_i^s, a_i^s) \mapsto w_j^s$ which can then also be applied to samples from unseen classes $h(v_i^u, a_i^u) = w_j^u$, where $(v_j^u, a_j^u, y_j^u) \in U$ for the set of test samples from unseen classes $U = (v_i^u, a_i^u, y_i^u)_{i\in\{1,\cdots,M\}}$ with M samples.

2.3.1 Model Architecture

Our AVCA model architecture is visualised in Fig. 2.2. For easier readability, we dropped the subscripts i, j, indicating the i-th dataset sample and the ground-truth class j.

AVCA takes audio and visual features $a, v \in \mathbb{R}^{k_{input}}$ as inputs which are extracted using pretrained feature extractors. Those are passed through two different encoder blocks A_{enc} and V_{enc} for the audio and visual modality respectively, giving embeddings

$$A_{enc}(a) = \phi_a \text{ and } V_{enc}(v) = \phi_v \tag{2.1}$$

with $\phi_a, \phi_v \in \mathbb{R}^{k_f}$. The encoder blocks each consist of a sequence of two linear layers f_1^m, f_2^m for $m \in \{a, v\}$, where $f_1^m : \mathbb{R}^{k_{input}} \to \mathbb{R}^{k_{fhidd}}$ and $f_2^m : \mathbb{R}^{k_{fhidd}} \to \mathbb{R}^{k_f}$. f_1^m, f_2^m are each followed by batch normalisation[98], a ReLU [172], and dropout [220] with dropout rate r_{enc}.

Cross-attention block. We propose to use a cross-attention block to share information between the audio and visual representations ϕ_a and ϕ_v. It consists of a multi-head self-attention layer, followed by a fully-connected feed-forward block. Similar to [241], we use a residual connection for the two layers, followed by layer normalisation [23].

The feed-forward blocks for the audio and visual branch each consist of a linear projection layer $f_3^m : \mathbb{R}^{k_f} \to \mathbb{R}^{k_{attnhidd}}$ for $m \in \{a, v\}$, followed by GELU [91], dropout with

dropout rate of r_{enc}, another linear projection layer $f_4^m : \mathbb{R}^{k_{attnhidd}} \rightarrow \mathbb{R}^{k_f}$ for $m \in \{a, v\}$ and finally a dropout with dropout rate of r_{enc}. The outputs of the cross-attention block are $\phi_a^{att}, \phi_v^{att} \in \mathbb{R}^{k_f}$.

A residual connection around the cross-attention block and subsequent projection blocks A_{proj} and V_{proj} give

$$A_{proj}(\phi_a^{att} + \phi_a) = \theta_a \text{ and } V_{proj}(\phi_v^{att} + \phi_v) = \theta_v, \tag{2.2}$$

where $\theta_a, \theta_v \in \mathbb{R}^{k_{proj}}$. The projection blocks each consist of a sequence of two linear layers f_5^m and f_6^m for $m \in \{a, v\}$, where $f_5^m : \mathbb{R}^{k_f} \rightarrow \mathbb{R}^{k_{fhidd}}$ and $f_6^m : \mathbb{R}^{k_{fhidd}} \rightarrow \mathbb{R}^{k_{proj}}$. f_5^m, f_6^m are each followed by batch normalisation, a ReLU, and dropout with dropout rate r_{proj}.

Furthermore, the word2vec class label embeddings w^j for class j are passed through the projection block $W_{proj}(w^j) = \theta_w^j$, where $\theta_w^j \in \mathbb{R}^{k_{proj}}$ (in Fig. 2.2 shown without the superscript j). W_{proj} consists of a sequence of one linear projection layer, batch normalisation, ReLU, and dropout with dropout rate r_{dec}.

At test time, class predictions c are obtained by determining the class c that corresponds to the textual class label embedding that is closest to the multi-modal representation θ_v (in our experiments we found that using θ_a gave slightly weaker results):

$$c = \underset{j}{\text{argmin}}(\|\theta_w^j - \theta_v\|_2). \tag{2.3}$$

2.3.2 Loss Functions

We train our AVCA model using a loss function l consisting of a base triplet loss l_t, a composite triplet and reconstruction loss l_c, and a regularisation loss l_r:

$$l = l_t + l_c + l_r. \tag{2.4}$$

We use the triplet loss function $t(a, p, n) = \max(\|a - p\|_2 - \|a - n\|_2 + \mu)$, where a is the anchor embedding, p and n are embeddings for positive samples and negative samples respectively, and μ is the margin hyperparameter. For triplet losses, we use the superscript $+$ to denote positive samples that match the anchor and $-$ for negative samples that do not semantically match the anchor. For all other losses, we only use matching pairs.

Base triplet loss. In our base triplet loss l_t:

$$\begin{aligned} l_t = t(\theta_a^+, \theta_w^+, \theta_a^-) + t(\theta_v^+, \theta_w^+, \theta_v^-) \\ + t(\theta_w^+, \theta_a^+, \theta_w^-) + t(\theta_w^+, \theta_v^+, \theta_w^-), \end{aligned} \tag{2.5}$$

where θ_m^+ and θ_m^- correspond to positive and negative samples respectively for $m \in \{a, v, w\}$, ensuring that the projected visual and audio features θ_v and θ_a are aligned with the projected textual features θ_w. This is essential, since at test time, the proximity of θ_v (which, despite being the output of the visual branch of AVCA, is a multi-modal embedding containing both audio and visual information) to θ_w for different classes is used to determine the output class.

Composite triplet and reconstruction loss. Inspired by [152], we additionally use a composite triplet and reconstruction loss and explain its components in more detail below:

$$l_c = l_{rec} + l_{ct} + l_w. \tag{2.6}$$

We use a decoder $D : \mathbb{R}^{k_{proj}} \mapsto \mathbb{R}^{k_{w2v}}$, such that $D(\theta_m) = \rho_m$ for $m \in \{a, v, w\}$. D consists of a sequence of one linear projection layer, batch normalisation, a ReLU, and dropout with dropout rate r_{dec}. We employ the mean squared error metric $d(b, c) = \frac{1}{n}\sum_{i=1}^{n}(b_i - c_i)^2$. The reconstruction loss l_{rec} can then be written as:

$$l_{rec} = d(\rho_a, w) + d(\rho_v, w) + d(\rho_w, w). \tag{2.7}$$

This ensures that AVCA is able to decode the pre-extracted textual label embeddings w from the embeddings $\theta_a, \theta_v, \theta_w$. The triplet loss l_{ct} is defined as follows:

$$l_{ct} = t(\rho_w^+, \rho_a^+, \rho_a^-) + t(\rho_w^+, \rho_v^+, \rho_v^-), \tag{2.8}$$

where ρ^+ and ρ^- correspond to positive and negative examples respectively. l_{ct} further encourages the decoded audio and visual features ρ_a, ρ_v to be aligned with the textual features ρ_w that were obtained using the same decoder (with shared weights). The third component l_w of l_c is similar to the base triplet loss in Eq. 2.5 and compares the audio and visual embeddings θ_a, θ_v to θ_w:

$$\begin{aligned} l_w = t(\theta_w^+, \theta_a^+, \theta_a^-) + t(\theta_w^+, \theta_v^+, \theta_v^-) \\ t(\theta_a^+, \theta_w^+, \theta_w^-) + t(\theta_v^+, \theta_w^+, \theta_w^-). \end{aligned} \tag{2.9}$$

Regularisation loss. The final component of our loss l consists of regularisation loss terms which directly encourage the alignment of the audio and visual embeddings with the text embeddings while preserving the information from their respective input modality. For this, we add two reconstruction blocks A_{rec} and V_{rec}, such that $\phi_a^{rec} = A_{rec}(\theta_a)$ and $\phi_v^{rec} = V_{rec}(\theta_v)$, $\phi_a^{rec}, \phi_v^{rec} \in \mathbb{R}^{k_f}$. A_{rec} and V_{rec} each consist of a linear projection layer followed by batch normalisation, ReLU, and dropout with dropout rate r_{dec}:

$$\begin{aligned} l_r = d(\phi_v^{rec}, \phi_v) + d(\phi_a^{rec}, \phi_a) \\ + d(\theta_v, \theta_w) + d(\theta_a, \theta_w). \end{aligned} \tag{2.10}$$

2.4 Experiments

We apply our AVCA model to audio-visual GZSL and ZSL for video classification. In this section, we first describe our proposed benchmark (Sec. 2.4.1). We discuss implementation details (Sec. 2.4.2), and then ablate the choice of different model components and loss functions (Sec. 2.4.5). Finally, we compare AVCA to state-of-the-art baseline methods for (G)ZSL (Sec. 2.4.3), and provide a detailed qualitative analysis of the learnt multi-modal embeddings (Sec. 2.4.4).

Dataset	# classes		# videos				
	all	tr / v(U) / ts(U)	tr	v (S)	v (U)	ts (S)	ts (U)
VGGSound-GZSL	276	138 / 69 / 69	70351	7817	3102	9032	3450
UCF-GZSL	51	30 / 12 / 9	3174	353	1467	555	1267
ActivityNet-GZSL	200	99 / 51 / 50	9204	1023	4307	1615	4199

Table 2.1: Statistics for our VGGSound, UCF, and ActivityNet (G)ZSL datasets, showing the number (#) of classes and videos in our splits (tr: train, v: validation, ts: test; S: seen, U: unseen).

2.4.1 Audio-Visual GZSL Benchmark

In this section, we propose three benchmark datasets for audio-visual GZSL curated from the VGGSound [45], UCF101 [219], and ActivityNet [90] datasets (summarised in Tab. 2.1)[1], and introduce our training and evaluation protocol.

Dataset statistics. For our proposed audio-visual GZSL splits, we include classes contained in the Sports1M [105] dataset only in our seen subsets to allow the use of feature extractors pretrained on Sports1M without leakage of information to unseen classes.

Our GZSL splits for the three datasets consist of a training set (tr), a validation set which is divided into a subset with samples from seen classes (v(S)) and another one with unseen classes (v(U)). Finally, we provide a test set consisting of seen classes (ts(S)) and unseen classes (ts(U)). The training set and the seen validation subset share the same classes with a ratio of 0.9/0.1 with respect to the number of videos. The subsets $\{\text{tr} \cup \text{v(U)} \cup \text{v(S)}\}$ and ts(S) share the same classes and were split to have a ratio of 0.9/0.1 with respect to the number of videos.

VGGSound [45] is a large audio-visual dataset with 309 classes and over 200k videos. The videos can be grouped into the 9 categories *animals*, *home*, *music*, *nature*, *people*, *sports*, *tools*, *vehicle*, and *others*. For our VGGSound-GZSL split, we exclude videos from the *others* category and all samples from v(U) and ts(U) that were used to train SeLaVi [21], resulting in 93,752 videos in 276 classes. The 42 classes that overlap with the Sports1M dataset are only used as training classes for GZSL.

UCF101 [219] is a video action recognition dataset which consists of over 13k videos in 101 classes. We use the subset of UCF101 which contains audio information. This results in a total of 6,816 videos for 51 classes. Previous (visual-only) methods repeatedly split the dataset into random seen and unseen classes. The 30 classes contained in the Sports1M dataset are not selected as unseen classes.

ActivityNet [90] is an action recognition dataset with 20k videos in 200 classes of varying duration. Again, we propose the ActivityNet-GZSL split ensuring that the 99 classes contained in the Sports1M dataset are not selected as unseen classes.

Training and evaluation protocol. We introduce a unified training and evaluation protocol for our GZSL benchmarks. We follow this protocol to train and test all models, including

[1] VGGSound is covered by a Creative Commons license: https://creativecommons.org/licenses/by/4.0/, ActivityNet by the MIT license: https://github.com/activitynet/ActivityNet/blob/master/LICENSE.

Method type	Model	VGGSound-GZSL				UCF-GZSL				ActivityNet-GZSL			
		S	U	HM	ZSL	S	U	HM	ZSL	S	U	HM	ZSL
ZSL	ALE [9]	0.28	5.48	0.53	5.48	57.59	14.89	23.66	16.32	2.63	7.87	3.94	7.90
	SJE [10]	48.33	1.10	2.15	4.06	63.10	16.77	26.50	18.93	4.61	7.04	5.57	7.08
	DEVISE [73]	36.22	1.07	2.08	5.59	55.59	14.94	23.56	16.09	3.45	8.53	4.91	8.53
	APN [278]	7.48	3.88	5.11	4.49	28.46	16.16	20.61	16.44	9.84	5.76	7.27	6.34
	f-VAEGAN-D2 [267]	12.77	0.95	1.77	1.91	17.29	8.47	11.37	11.11	4.36	2.14	2.87	2.40
Audio-visual ZSL	CJME [185]	8.69	4.78	6.17	5.16	26.04	8.21	12.48	8.29	5.55	4.75	5.12	5.84
	AVGZSLNet [152]	18.05	3.48	5.83	5.28	52.52	10.90	18.05	13.65	8.93	5.04	6.44	5.40
	AVCA	14.90	4.00	**6.31**	**6.00**	51.53	18.43	**27.15**	**20.01**	24.86	8.02	**12.13**	**9.13**

Table 2.2: Evaluating our AVCA model and state-of-the-art audio-visual ZSL methods and adapted ZSL methods for GZSL and ZSL on the VGGSound, UCF, and ActivityNet (G)ZSL benchmarks. We report the mean class accuracy on the seen (S) and unseen (U) test classes, and their harmonic mean (HM) for GZSL performance. The ZSL performance is evaluated on the test subset from unseen classes.

AVCA and the baselines that we compare to.

We propose a two-stage training and evaluation protocol for GZSL. In the first stage, we train the models on the training set (tr), using the subsets of seen validation classes (v(S)) and unseen validation classes (v(U)) to determine the GZSL parameters, for instance for calibrated stacking [41].

In the second training stage, we re-train the models on the training (tr) and full validation set $\{\text{v(S)} \cup \text{v(U)}\}$ using the GZSL parameters determined during the first training stage. Our final models are then evaluated on the test set $\{\text{ts(S)} \cup \text{ts(U)}\}$. ts(S) contains samples from the same classes as the training classes with no overlap between training samples for the second stage and the test samples. In particular, there is no class overlap between v(U) and ts(U).

Evaluation metrics. Following [265], we propose to evaluate all models using the mean class accuracy. For GZSL, we evaluate the models on the full test set $\{\text{ts(S)} \cup \text{ts(U)}\}$, and report the averaged performance on the unseen (U) and seen (S) classes. Furthermore, we compute their harmonic mean $HM = \frac{2US}{U+S}$. We report the ZSL performance by evaluating only on the subset ts(U).

2.4.2 Experimental Setting

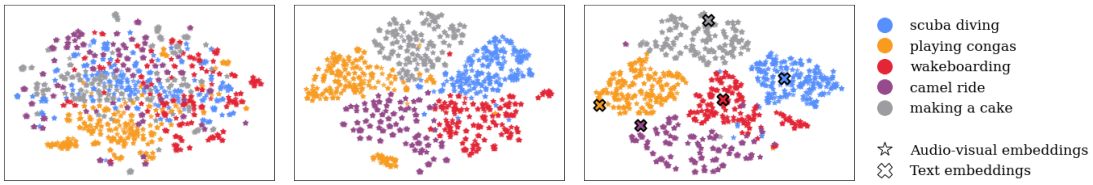
For each video, we use the self-supervised SeLaVi [21] framework pretrained on VGGSound [45] to extract audio and visual features for each second in a video. In our VGGSound-GZSL split, there is no overlap between videos in the unseen test and unseen validation sets and videos that were used for pre-training SeLaVi. We average the per-second features extracted using SeLaVi prior to the two-layer MLP heads to obtain 512-dimensional per-video audio and visual features. We provide additional results for using features extracted from audio and video classification networks in the supplementary material.

All networks were optimised for GZSL performance (HM) and we do not train separate networks for GZSL and ZSL. The training for the first stage was done for 50 epochs. We selected the number of training epochs for the second stage based on the GZSL performance on the validation set in the first stage. To eliminate the bias that the ZSL methods have towards seen classes, we used calibrated stacking [41] on the interval $[0, 3]$ with a step size of 0.2. For AVCA, k_{input} was set to 512 and the size of the word2vec embedding, k_{w2v}, was set to 300. We used dropout rates $r_{dec}/r_{enc}/r_{proj}$ of 0.5/0.2/0.3 for UCF-GZSL, 0.1/0.2/0.2 for Activity-GZSL, and 0.1/0/0 for VGGSound-GZSL. The layer dimensions were set to $k_f = 300$, $k_{fhidd} = 512$, $k_{attnhidd} = 64$, and $k_{proj} = 64$. We used 3 heads for self-attention. The loss margin hyperparameter, μ, was set to 1. We used a batchsize of 256 for UCF-GZSL and ActivityNet-GZSL, and 64 for VGGSound-GZSL. We used the Adam optimiser [110] with an initial learning rate of 0.001 which was reduced by a factor of 0.1 when the GZSL performance plateaued with a patience of 3 epochs.

2.4.3 Comparing with the State of the Art

Compared methods. In our benchmark study, we include four image-based state-of-the-art methods and one generative method for (G)ZSL which we adapt to take audio-visual features as inputs. For this, we concatenate the audio and visual features and use those as inputs instead of image features. Moreover, we compare to current state-of-the-art methods for audio-visual GZSL [152, 185]. Here, we describe each of the methods that we compare to in more detail.

ALE [9] learns a linear mapping between the input features and the ground-truth embeddings, such that the projection of the input features is close to the ground-truth embedding for the corresponding class. For this, it uses a weighted approximate ranking objective [239]. **SJE** [10] computes the dot product between linearly mapped input features and the ground-truth embedding of all negative classes. The highest dot product for each example is chosen and then minimised. **DEVISE** [73] also computes the dot product between the output of a linear projection and the negative class embeddings and it minimises the sum of these dot products. **APN** [278] is the current non-generative state-of-the-art method for image-based ZSL. APN is based on the assumption that the ground-truth embeddings contain visual class attributes. Prototypes are used to map the attributes from the ground-truth embeddings to relevant locations in the image. **f-VAEGAN-D2** [267] is a generative ZSL method which learns to generate synthetic features for unseen classes. Then, a classifier is trained on real examples from seen classes and synthetic examples from unseen classes. **CJME** [185] proposed the task of audio-visual GZSL for video classification on the AudioSetZSL dataset. It embeds audio, video and text into a joint embedding space and uses proximity in the embedding space to select the classification output at test time. **AVGZSLNet** [152] builds on [185] and is the current state-of-the-art method for audio-visual GZSL for video classification. One of the main strengths of this method is its use of triplet losses to leverage information from negative



(a) Input audio embeddings (b) Input visual embeddings (c) Learnt audio-visual embeddings

Figure 2.3: t-SNE visualisation for three seen (*scuba diving, playing congas, wakeboarding*) and two unseen (*camel ride, making a cake*) test classes from ActivityNet-GZSL, showing embeddings extracted with SeLaVi [21] for (a) audio and (b) visual features. (c) Learnt audio-visual embeddings of our model. Projected textual class label embeddings are visualised with a cross with black boundary.

examples.

Results. We compare our AVCA framework to recent methods for (G)ZSL in Tab. 2.2 on the VGGSound-GZSL, UCF-GZSL, and ActivityNet-GZSL datasets. AVCA obtains the best results on all three datasets. On VGGSound-GZSL, AVCA obtains a HM of 6.31% for GZSL and a ZSL performance of 6.00% compared to 6.17% HM for CJME and a ZSL performance of 5.59% for DEVISE. On the UCF-GZSL dataset, our AVCA model outperforms SJE for GZSL with a performance of 27.15% compared to 26.50%, and we obtain a stronger ZSL performance of 20.01% compared to 18.93%. On ActivityNet-GZSL, AVCA outperforms APN, with a GZSL performance of 12.13% compared to 7.27%. For ZSL, AVCA is stronger than DEVISE with a score of 9.13% compared to 8.53%. It can be observed that in some cases U is higher than S. This is due to the use of calibrated stacking [41] as described in [162].

2.4.4 Qualitative Results

We present a qualitative analysis of the learnt multi-modal embeddings in Fig. 2.3. The t-SNE visualisations [148] for a subset of ActivityNet-GZSL classes show the differences between the audio and visual input features and the learnt multi-modal embeddings. We provide additional qualitative results for VGGSound-GZSL and UCF-GZSL in the supplementary material. It can be seen in Fig. 2.3a that the input audio features are not as well-separated and clustered as the visual features shown in Fig. 2.3b. However, the visual features also contain classes, such as *playing congas* and *scuba diving*, which are not clustered cleanly. It can be observed in Fig. 2.3c that our model produces multi-modal features that improve over the clustering of the input embeddings for both, seen and unseen classes. For instance, the cluster separation between the seen class *playing congas* and the unseen class *making a cake* improves significantly, even though the unseen class is not used for training.

Model	VGGSound-GZSL		UCF-GZSL		ActivityNet-GZSL	
	HM	ZSL	HM	ZSL	HM	ZSL
Visual branch	4.83	4.06	20.92	14.16	7.53	6.49
Audio branch	3.84	3.83	11.78	10.78	4.19	4.06
AVCA	**6.31**	**6.00**	**27.15**	**20.01**	**12.13**	**9.13**

Table 2.3: Influence of *training* AVCA with different modalities for GZSL and ZSL on the VGGSound-GZSL, UCF-GZSL and ActivityNet-GZSL datasets measuring the harmonic mean (HM) for GZSL and the mean class accuracy for ZSL. Using both modalities yields the strongest GZSL and ZSL performances.

Model	VGGSound-GZSL		UCF-GZSL		ActivityNet-GZSL	
	HM	ZSL	HM	ZSL	HM	ZSL
W/o x-att	6.02	4.81	26.82	18.37	6.50	5.64
Visual with x-att	**6.63**	4.78	27.11	17.22	9.50	6.89
Audio with x-att	4.93	5.01	18.61	16.05	11.05	8.78
AVCA	6.31	**6.00**	**27.15**	**20.01**	**12.13**	**9.13**

Table 2.4: Using different components of AVCA for GZSL and ZSL on VGGSound-GZSL, UCF-GZSL and ActivityNet-GZSL. Audio (Visual) with x-att uses the visual (audio) modality only for the cross-attention. W/o x-att optimises each branch in isolation and their output predictions are averaged. x-att denotes cross-attention.

2.4.5 Ablation Analysis

Here, we analyse how different architectural choices and loss components for AVCA impact the performances on VGGSound-GZSL, ActivityNet-GZSL, and UCF-GZSL.

Evaluating different modalities. In Tab. 2.3, we compare our multi-modal AVCA model to training our architecture with only unimodal inputs. In this case, we remove the cross-modal attention block and train each unimodal branch in isolation. The visual branch obtains a better performance than the audio branch with a GZSL performance (HM) of 7.53% vs. 4.19% on the ActivityNet-GZSL dataset. A similar pattern can be observed for the ZSL performance with 6.49% vs. 4.06% for the visual and audio branch respectively. This trend is also exhibited on the UCF-GZSL and VGGSound-GZSL datasets, suggesting that the visual input features provide richer information about the video content than the audio inputs. Nevertheless, jointly training AVCA with both input modalities gives significant improvements over using each of them individually with a GZSL performance of 12.13% and a ZSL performance of 9.13% on the ActivityNet-GZSL dataset. This confirms that the complementary information from the audio and visual inputs is highly beneficial for GZSL and ZSL for video classification. We provide the S/U performances for Tab. 2.3 in the supplementary material.

Evaluating the cross-modal attention block. Next, we investigate the effect of using our cross-modal attention block in Tab. 2.4. To obtain results without using cross-attention (W/o x-att), each branch is optimised individually. For evaluation, we compute the distances between the outputs of both branches and θ_w for each class, and then average the distances computed by both branches. The GZSL and ZSL performances drop dramatically when not using the cross-attention block from 12.13% and 9.13% for AVCA to 6.50% and 5.64% for GZSL and ZSL scores respectively on the ActivityNet-GZSL dataset.

Model output	VGGSound-GZSL		UCF-GZSL		ActivityNet-GZSL	
	HM	ZSL	HM	ZSL	HM	ZSL
AVCA (θ_a)	5.18	4.87	25.98	18.25	**12.54**	**9.23**
AVCA (θ_v)	**6.31**	**6.00**	**27.15**	**20.01**	12.13	9.13
AVCA (θ_a, θ_v)	5.90	5.42	25.78	19.30	12.17	8.95
AVCA (min(θ_a, θ_v))	6.10	5.36	25.86	18.39	12.45	9.08

Table 2.5: Influence of using the outputs of the audio and visual branches θ_a and θ_v separately, or using both jointly (θ_a, θ_v) for *evaluation* on VGGSound-GZSL, UCF-GZSL and ActivityNet-GZSL. All models were trained with θ_a and θ_v.

The pattern is similar for VGGSound-GZSL and UCF-GZSL, confirming the importance of our cross-modal attention block for sharing information between the input modalities.

Furthermore, we compare optimising our full AVCA model to using only the visual (Visual with x-att) or only the audio branch (Audio with x-att) for training. Using only the visual branch entails removing A_{rec} and A_{proj} along with their associated losses from the audio branch but keeping the cross-attention. This experiment is repeated for the audio branch by removing the corresponding components from the visual branch. Jointly optimising both branches provides better results than using only one of the branches on ActivityNet-GZSL and UCF-GZSL. On ActivityNet-GZSL, we obtain a GZSL performances of 12.13% compared to 11.05% and 9.50% for using only the audio and visual branches respectively. Interestingly, for the VGGSound-GZSL dataset, the Visual with x-att model yields a slightly stronger GZSL performance than our full AVCA model, with a HM of 6.63% compared to 6.31%. This is in line with the Audio branch performing worse than the Visual branch on VGGSound-GZSL (Tab. 2.3). However, the joint optimisation of AVCA gives the best results.

Evaluating different modalities as output. In Tab. 2.5, we investigate the effect of evaluating our full trained AVCA model using only the output features from the audio (θ_a) or the visual (θ_v) branch, or from both branches together ((θ_a, θ_v) and $min(\theta_a, \theta_v)$). For AVCA(θ_a, θ_v), we compute the distance $|\theta_a - \theta_w|_2 + |\theta_v - \theta_w|_2$. AVCA($min(\theta_a, \theta_v)$) uses the embedding from the modality that has the smallest distance to a word embedding. The class corresponding to the closest text embedding resembles the class prediction.

Using the visual branch gives the strongest performance on VGGSound-GZSL/UCF-GZSL with a HM of 6.31%/27.15% vs 5.18%/25.98% for the audio branch. On ActivityNet-GZSL, the audio branch yields slightly better results (HM of 12.54% vs. 12.13% for the visual branch). Both AVCA(θ_a, θ_v) and AVCA($min(\theta_a, \theta_v)$) obtain lower scores that θ_v. The best results (highest averaged HM) across all three datasets are produced when using the visual branch only. However, as the cross-attention block fuses the audio and visual modalities, both branches contain multi-modal information from both input modalities.

Evaluating different loss functions. Finally, we analyse the impact of using different loss functions for training AVCA on the GZSL and ZSL performance in Tab. 2.6. We observe that using our full loss l provides the strongest GZSL results (HM) on the UCF-GZSL, VGGSound-GZSL, and ActivityNet-GZSL datasets by a large margin. On ActivityNet-GZSL, omitting l_t for training our model ($l - l_t$) provides slightly stronger ZSL results than

Model	VGGSound-GZSL		UCF-GZSL		ActivityNet-GZSL	
	HM	ZSL	HM	ZSL	HM	ZSL
l-l_t	5.06	4.84	18.51	19.17	8.39	**9.54**
l-l_{rec}	5.92	5.22	24.32	17.20	9.59	6.93
l-l_{ct}	6.31	4.87	17.88	17.51	11.20	8.99
l-l_w	5.18	4.93	20.75	16.41	9.08	8.00
l-l_r	6.24	4.43	21.31	14.02	11.14	7.94
l	**6.31**	**6.00**	**27.15**	**20.01**	**12.13**	9.13

Table 2.6: Comparing training AVCA with our full loss function l to removing individual components l_t, l_{rec}, l_{ct}, l_w, or l_r, on the GZSL and ZSL performance on the VGGSound-GZSL, UCF-GZSL and ActivityNet-GZSL datasets.

using our full loss l with a mean class accuracy of 9.54% compared to 9.13%. However, the GZSL performance is significantly better when using l with a HM of 12.13% compared to 8.39% when using $l - l_t$. Our loss ablations confirm that our strong overall performance on all three datasets is only obtained when training with our full proposed loss function.

2.4.6 Limitations and Discussion

Our proposed GZSL benchmark datasets pose an extremely challenging setting, since the underlying datasets span a wide variety of classes (*e.g.* including *wakeboarding* and *making a cake* for the ActivityNet dataset). Our AVCA leverages the varied audio-visual input information effectively, resulting in more robust GZSL performance than the related methods. However, AVCA uses temporally averaged audio-visual input information, and hence does not consider fine semantic details. Furthermore, our model relies on multi-modal input data and cannot be used when only one modality is available.

2.5 Conclusion

We introduced three new benchmarks for audio-visual (generalised) zero-shot learning for video classification on the VGGSound, UCF, and ActivityNet datasets. We proposed a framework for (G)ZSL from audio-visual data which learns to align the audio-visual embeddings with textual label embeddings. Furthermore, we provided baseline performances for seven (G)ZSL methods, and show that our model outperforms them for GZSL and ZSL on our new benchmarks. Finally, we provided a qualitative analysis of the learnt multi-modal embeddings. We hope that our proposed benchmarks will enable and encourage further research into audio-visual zero-shot learning.

3

Temporal and cross-modal attention for audio-visual zero-shot learning

Audio-visual generalised zero-shot learning for video classification requires understanding the relations between the audio and visual information in order to be able to recognise samples from novel, previously unseen classes at test time. The natural semantic and temporal alignment between audio and visual data in video data can be exploited to learn powerful representations that generalise to unseen classes at test time. We propose a multi-modal and Temporal Cross-attention Framework (TCAF) for audio-visual generalised zero-shot learning. Its inputs are temporally aligned audio and visual features that are obtained from pre-trained networks. Encouraging the framework to focus on cross-modal correspondence across time instead of self-attention within the modalities boosts the performance significantly. We show that our proposed framework that ingests temporal features yields state-of-the-art performance on the UCF-GZSL^{cls}, VGGSound-GZSL^{cls}, and ActivityNet-GZSL^{cls} benchmarks for (generalised) zero-shot learning. Code for reproducing all results is available at https://github.com/ExplainableML/TCAF-GZSL.

3.1 Introduction

Learning task-specific audio-visual representations commonly requires a great number of annotated data samples. However, annotated datasets are limited in size and in the labelled classes that they contain. If a model which was trained with supervision on such a dataset is applied in the real world, it encounters classes that it has never seen. To recognise those novel classes, it would not be feasible to train a new model from scratch. Therefore, it is essential to analyse the behaviour of a trained model in new settings. Ideally, a model should be able to transfer knowledge obtained from classes seen during training to previously unseen categories. This ability is probed in the zero-shot learning (ZSL) task. In addition to zero-shot capabilities, a model should retain the class-specific information from seen training classes. This is challenging and is investigated in the so-called generalised ZSL (GZSL) setting which considers the performance on both, seen and unseen classes.

Prior works [152, 157, 185] have proposed frameworks that address the (G)ZSL task for video classification using audio-visual inputs. Those methods learn a mapping from the audio-visual input data to textual label embeddings, enabling the classification of samples from unseen classes. At test time, the class whose word embedding is closest to the predicted audio-visual output embedding is selected. Similar to this, we use the textual label embedding space to allow for information transfer from training classes to previously unseen classes.

However, [152, 157, 185] used temporally averaged features as inputs that were extracted from networks pre-trained on video data. The averaging disregarded the temporal dynamics in videos. We propose a Temporal Cross-attention Framework (TCAF) which builds on [157] and additionally exploits temporal information by using temporal audio and visual data as inputs. This gives a significant boost in performance for the audio-visual (G)ZSL task compared to using temporally averaged input features. Different from computationally expensive methods that operate directly on raw visual inputs [34, 107, 131], our TCAF uses features extracted from networks pre-trained for audio and video classification as inputs. This leads to an efficient setup that uses temporal information instead of averaging across time.

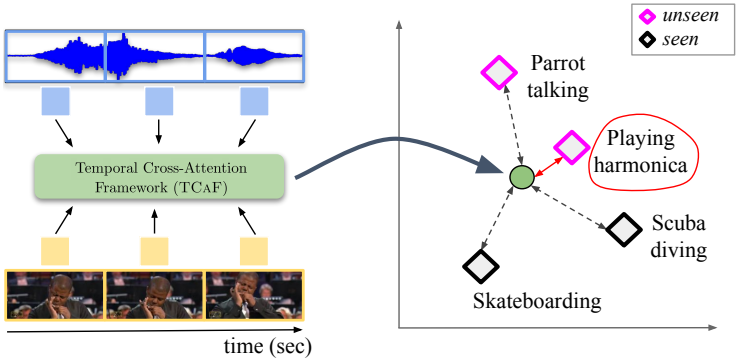


Figure 3.1: Our temporal cross-attention framework for audio-visual (G)ZSL learns a multi-modal embedding (green circle) by exploiting the temporal alignment between audio and visual data in videos. Textual label embeddings (grey squares) are used to transfer information from seen training classes (black) to unseen test classes (pink). The correct class is playing harmonica (red).

The natural alignment between audio and visual information in videos, e.g. a frog being visible in a frame while the sound of a frog croaking is audible, provides a rich training signal for learning video representations. This can be attributed to the semantic and temporal correlation between the audio and visual information when comparing the two modalities. We encourage our TCAF to put special emphasis on the correlation across the two modalities by employing repeated cross-attention. This attention mechanism only allows attention to tokens from the other modality. This effectively acts as a bottleneck which results in cheaper computations and gives a boost in performance over using full self-attention across all tokens from both modalities.

We perform a detailed model ablation study to show the benefits of using temporal inputs and our proposed cross-attention. Furthermore, we confirm that our training objective is well-suited to the task at hand. We also analyse the learnt audio-visual

embeddings with t-SNE visualisations which confirm that training our TCAF improves the class separation for both seen and unseen classes.

To summarise, our contributions are as follows: (1) We propose a temporal cross-attention framework TCAF for audio-visual (G)ZSL. (2) Our proposed model achieves state-of-the-art results on the UCF-GZSL^{cls}, VGGSound-GZSL^{cls}, and ActivityNet-GZSL^{cls} datasets, demonstrating that using temporal information is extremely beneficial for improving the (generalised) zero-shot classification accuracy compared to using temporally averaged features as model inputs. (3) We perform a detailed analysis of the use of enhanced cross-attention across modalities and time, demonstrating the benefits of our proposed model architecture and training setup.

3.2 Related work

Our work relates to several themes in the literature: audio-visual learning, ZSL with side information, audio-visual ZSL with side information, and multi-modal transformer architectures. We discuss those in more detail in the following.

Audio-visual learning. The temporal alignment between audio and visual data in videos is a strong learning signal which can be exploited for learning audio-visual representations. [15, 22, 114, 183, 184, 187]. In addition to audio and video classification, numerous other tasks benefit from audio-visual inputs, such as the separation and localisation of sounds in video data [4, 7, 19, 44, 76, 182, 235], audio-driven synthesis of images [102, 256], audio synthesis driven by visual information [75, 79, 112, 113, 174, 222, 293], and lip reading [5, 6]. Some approaches use class-label supervision between modalities [50, 72] which does not require the temporal alignment between the input modalities. In contrast to full class-label supervision, we train our model only on the subset of seen training classes.

ZSL with side information. Visual ZSL methods commonly map the visual inputs to class side information [9, 10, 73], e.g. word2vec [161] class label embeddings. This allows to determine the class with the side information that is closest at test time as the class prediction. Furthermore, attribute annotations have been used as side information [71, 246, 265, 269]. Recent non-generative methods identify key visual attributes [278], use attention to find discriminative regions [270], or disambiguate class embeddings [141]. In contrast, feature generation methods train a classifier on generated and real features [175, 266, 267, 302]. Unlike methods for ZSL with side information with unimodal (visual) inputs, our proposed framework uses multi-modal audio-visual inputs.

Audio-visual ZSL with side information. The task of GZSL from audio-visual data was introduced by [152, 185] on the AudioSetZSL dataset [185] using class label word embeddings as side information. Recently, [157] proposed the AVCA framework which uses cross-attention to fuse information from the averaged audio and visual input features for audio-visual GZSL. Our proposed framework builds on [157], but instead of using

temporally averaged features as inputs [152, 157, 185], we explore the benefits of using temporal cross-attention information. Unlike [157]'s two-stream architecture, we propose the fusion into a single output branch with a classification token that aggregates multi-modal information. Furthermore, we simplify the training objective, and show that the combination of using temporal inputs, our architecture, and training setup leads to superior zero-shot classification performance.

Multi-modal transformers. The success of transformer models in the language domain [59, 198, 241] has been translated to visual recognition tasks with the Vision Transformer [63]. Multi-modal vision-language representations have been obtained with a masked language modelling objective, and achieved state-of-the-art performance on several text-vision tasks [123, 124, 142, 224, 225, 226, 230]. In this work, we consider audio-visual multi-modality. Transformer-based models that operate on audio and visual inputs have recently been proposed for text-based video retrieval [74, 138, 250], dense video captioning [97], audio-visual event localization [133], and audio classification [31]. Different to vanilla transformer-based attention, our TCAF puts special emphasis on cross-attention between the audio and visual modalities in order to learn powerful representations for the (G)ZSL task.

3.3 TCAF Model

In this section, we describe the problem setting (Sec. 3.3.1), our proposed model architecture (Sec. 3.3.2), and the loss functions used to train TCAF (Sec. 3.3.3).

3.3.1 Problem setting

We address the task of (G)ZSL using audio-visual inputs. The aim of ZSL is to be able to generalise to previously unseen test classes at test time. For GZSL, the model should additionally preserve knowledge about seen training classes, since the GZSL test set contains samples from both, seen and unseen classes.

We denote an audio-visual dataset with N samples and K (seen and unseen) classes by $\mathcal{V} = \{\mathcal{X}_{a[i]}, \mathcal{X}_{v[i]}, y_{[i]}\}_{i=1}^{N}$, consisting of audio data $\mathcal{X}_{a[i]}$, visual data $\mathcal{X}_{v[i]}$, and ground-truth class labels $y_{[i]} \in \mathbb{R}^K$. Naturally, video data contains temporal information. In the following, we use T_a and T_v to denote the number of audio and visual segments in a video clip.

A pre-trained audio classification CNN is used to extract a sequence of audio features $\boldsymbol{a}_{[i]} = \{a_1, \ldots, a_t, \ldots, a_{T_a}\}_i$ to encode the audio information $\mathcal{X}_{a[i]}$. The visual data $\mathcal{X}_{v[i]}$ is encoded into a temporal sequence of features $\boldsymbol{v}_{[i]} = \{v_1, ..., v_t, ..., v_{T_v}\}_i$ by representing visual segments with features extracted from a pre-trained video classification network.

3.3.2 Model architecture

In the following, we describe the architecture of our proposed TCAF (see Fig. 3.2).

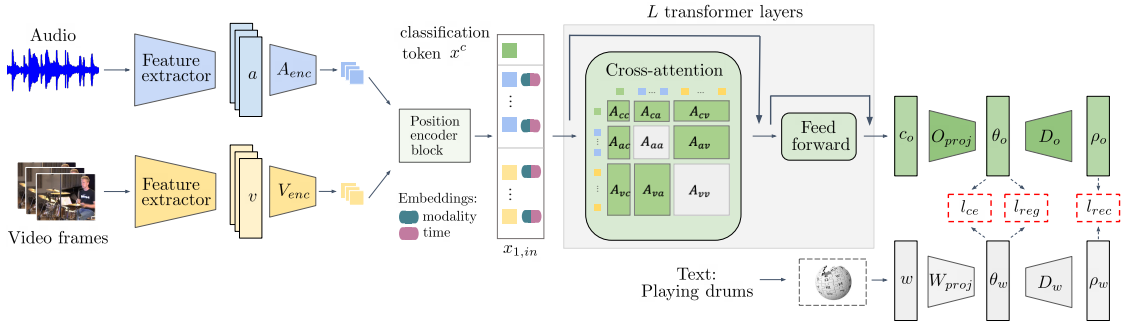


Figure 3.2: TCAF takes audio and visual features extracted from video data as inputs. Those are embedded and equipped with modality and time embeddings before passing through a sequence of L transformer layers with cross-attention. The output classification token c_o is then projected to embedding spaces that are shared with the textual information. The loss functions operate on the joint embedding spaces. At test time, the class prediction c is obtained by determining the word label embedding θ_w^j that is closest to θ_o.

Embedding the inputs and position encoder block. TCAF takes pre-extracted audio and visual features $\boldsymbol{a}_{[i]}$ and $\boldsymbol{v}_{[i]}$ as inputs. For readability, we will drop the subscript i in the following which denotes the $i-$th sample. In order to project audio and visual features to the same feature dimension, $\boldsymbol{a}$ and $\boldsymbol{v}$ are passed through two modality-specific embedding blocks, giving embeddings:

$$\phi_a = A_{enc}(\boldsymbol{a}) \text{ and } \phi_v = V_{enc}(\boldsymbol{v}), \tag{3.1}$$

with $\phi_a \in \mathbb{R}^{T_a * d_{dim}}$ and $\phi_v \in \mathbb{R}^{T_v * d_{dim}}$. The embedding blocks are composed of two linear layers f_1^m, f_2^m for $m \in \{\boldsymbol{a}, \boldsymbol{v}\}$, where $f_1^m : \mathbb{R}^{T_m * d_{in_m}} \rightarrow \mathbb{R}^{T_m * d_{fhidd}}$ and $f_2^m : \mathbb{R}^{T_m * d_{fhidd}} \rightarrow \mathbb{R}^{T_m * d_{dim}}$. f_1^m, f_2^m are each followed by batch normalisation [98], a ReLU [172], and dropout [220] with dropout rate $drop_{enc}$.

The position encoder block adds learnt modality and temporal positional embeddings to the outputs of the modality-specific embedding blocks. We explain this in detail below. To handle different frame rates in the audio and visual modalities, we use Fourier features [232] $pos_t \in \mathbb{R}^{d_{pos}}$ for the temporal embeddings that encode the actual point in time in the video which corresponds to an audio or visual representation. This allows to capture the relative temporal position of the audio and visual features across the modalities.

For an audio embedding ϕ_{a_t} at time t, a linear map $g_a : \mathbb{R}^{d_{pos}+d_{dim}} \rightarrow \mathbb{R}^{d_{dim}}$, and a dropout layer g^D with dropout probability $drop_{prob,pos}$, we obtain position-aware audio feature tokens

$$a_t^p = g^D(g_a(concat(\phi_{a_t}, pos_{at}))) \qquad \text{with} \qquad pos_{at} = pos_a + pos_t, \tag{3.2}$$

with modality and temporal embeddings $pos_a, pos_t \in \mathbb{R}^{d_{pos}}$ respectively. Position-aware visual tokens v_t^p are obtained analogously.

Furthermore, we prepend a learnt classification token $x^c \in \mathbb{R}^{d_{dim}}$ to the sequence of feature tokens. The corresponding output classification token c_o is used by our output projection O_{proj} to obtain the final prediction.

Audio-visual transformer layers. TCAF contains L stacked audio-visual transformer layers that allow for enhanced cross-attention. Each of our transformer layers consists of an attention function $f_{l,Att}$, followed by a feed forward function $g_{l,FF}$. The output of the l-th transformer layer is given as

$$x_{l,out} = x_{l,ff} + x_{l,att} = g_{l,FF}(x_{l,att}) + x_{l,att}, \tag{3.3}$$

with

$$x_{l,att} = f_{l,Att}(x_{l,in}) + x_{l,in}, \tag{3.4}$$

where

$$x_{l,in} = \begin{cases} [x^c, a_1^p, \cdots, a_{T_a}^p, v_1^p, \cdots, v_{T_v}^p] & \text{if } l = 1, \\ x_{l-1,out} & \text{if } 2 \leq l \leq L. \end{cases}$$

We explain the cross-attention used in our transformer layers in the following.

Transformer cross-attention. TCAF primarily exploits cross-modal audio-visual attention to combine the information across the audio and visual modalities. All attention mechanisms in TCAF consist of multi-head attention [241] with H heads and a dimension of d_{head} per head.

We describe the first transformer layer $\mathcal{M}_1$, the transformer layer $\mathcal{M}_l$ operates analogously. We project the position-aware input features x^c, $\{a_t^p\}_{t\in[1,T_a]}$, $\{v_t^p\}_{t\in[1,T_v]}$ to queries, keys, and values with linear maps $g_s : \mathbb{R}^{d_{dim}} \rightarrow \mathbb{R}^{d_{head}H}$ for $s \in \{q, k, v\}$. We can then write the outputs of the projection as zero-padded query, key, and value features. We write those out for the queries below, the keys and values are padded in the same way:

$$\mathbf{q}_c = [g_q(x^c), 0, \cdots, 0], \tag{3.5}$$

$$\mathbf{q}_a = [0, \cdots, 0, g_q(a_1^p), \cdots, g_q(a_{T_a}^p), 0, \cdots, 0], \tag{3.6}$$

$$\mathbf{q}_v = [0, \cdots, 0, g_q(v_1^p), \cdots, g_q(v_{T_v}^p)]. \tag{3.7}$$

The full query, key, and value representations, $\mathbf{q}$, $\mathbf{k}$, and $\mathbf{v}$, are the sums of their modality-specific components

$$\mathbf{q} = \mathbf{q}_c + \mathbf{q}_a + \mathbf{q}_v, \qquad \mathbf{k} = \mathbf{k}_c + \mathbf{k}_a + \mathbf{k}_v, \qquad \text{and } \mathbf{v} = \mathbf{v}_c + \mathbf{v}_a + \mathbf{v}_v. \tag{3.8}$$

The output of the first attention block $x_{1,att}$ is the aggregation of the per-head attention with a linear mapping $g_h : \mathbb{R}^{d_{head}H} \rightarrow \mathbb{R}^{d_{dim}}$, g^{DL} dropout with dropout probability $drop_{prob}$ and layer normalisation g^{LN} [23], such that

$$x_{1,att} = f_{1,Att}(x_{l,in}) = g^{DL}(g_h(f_{1,att}^1(g^{LN}(x_{1,in})), \cdots, f_{1,att}^H(g^{LN}(x_{1,in})))), \tag{3.9}$$

with the attention f_{att}^h for the attention head h. We can write the attention for the head h as

$$f_{att}^h(x_{1,in}) = softmax\left(\frac{\mathbf{A}}{\sqrt{d_{head}}}\right)\mathbf{v}, \tag{3.10}$$

where $\mathbf{A}$ can be split into its cross-attention and self-attention components:

$$\mathbf{A}_c = \mathbf{q}_c\,\mathbf{k}^T + \mathbf{k}\,\mathbf{q}_c^T, \qquad \mathbf{A}_x = \mathbf{q}_a\,\mathbf{k}_v^T + \mathbf{q}_v\,\mathbf{k}_a^T, \tag{3.11}$$

$$\mathbf{A}_{self} = \mathbf{q}_a\,\mathbf{k}_a^T + \mathbf{q}_v\,\mathbf{k}_v^T.$$

We then get

$$\mathbf{A} = \mathbf{A}_c + \mathbf{A}_x + \mathbf{A}_{self} = \begin{pmatrix} A_{cc} & A_{ca} & A_{cv} \\ A_{ac} & \ddots & \vdots \\ A_{vc} & \dots & 0 \end{pmatrix} + \begin{pmatrix} 0 & \dots & 0 \\ \vdots & \ddots & A_{av} \\ 0 & A_{va} & 0 \end{pmatrix} + \begin{pmatrix} 0 & \dots & 0 \\ \vdots & A_{aa} & \vdots \\ 0 & \dots & A_{vv} \end{pmatrix}, \tag{3.12}$$

where the A_{mn} with $m, n \in \{c, a, v\}$ describe the attention contributions from the classification token, the audio and the visual modalities respectively.

Our TCAF uses the cross-attention $\mathbf{A}_c + \mathbf{A}_x$ to put special emphasis on the attention across modalities. Results for different model variants that use only the within-modality self-attention ($\mathbf{A}_c + \mathbf{A}_{self}$) or the full attention which combines self-attention and cross-attention are presented in Sec. 3.4.3.

Feed forward function. The feed forward function $g_{l,FF} : \mathbb{R}^{d_{dim}} \rightarrow \mathbb{R}^{d_{dim}}$ is applied to the output of the attention function

$$x_{l,ff} = g_{l,FF}(x_{l,att}) = g^{DL}(g_{l,F2}(g^{DL}(g^{GD}(g_{l,F1}(g^{LN}(x_{l,att})))))) \tag{3.13}$$

where $g_{l,F1} : \mathbb{R}^{d_{dim}} \rightarrow \mathbb{R}^{d_{ff}}$ and $g_{l,F2} : \mathbb{R}^{d_{ff}} \rightarrow \mathbb{R}^{d_{dim}}$ are linear mappings, g^{GD} is a GELU layer [91] and a dropout layer with dropout probability $drop_{prob}$, g^{DL} is dropout with $drop_{prob}$ and g^{LN} is layer normalisation.

Output prediction. To determine the final class prediction, the audio-visual embedding is projected to the same embedding space as the textual class label representations. We project the output classification token c_o of the temporal cross-attention to $\theta_o = O_{proj}(c_o)$ where $\theta_o \in \mathbb{R}^{d_{out}}$. The projection block is composed of a sequence of two linear layers f_3 and f_4, where $f_3 : \mathbb{R}^{d_{dim}} \rightarrow \mathbb{R}^{d_{fhidd}}$ and $f_4 : \mathbb{R}^{d_{fhidd}} \rightarrow \mathbb{R}^{d_{out}}$. f_3, f_4 are each followed by batch normalisation, a ReLU, and dropout with rate $drop_{proj_o}$. We project the word2vec class label embedding w^j for class j using the projection block $W_{proj}(w^j) = \theta_w^j$, where $\theta_w^j \in \mathbb{R}^{d_{out}}$. W_{proj} consists of a linear projection followed by batch normalisation, ReLU, and dropout with dropout rate $drop_{proj_w}$. The class prediction c is obtained by determining the projected word2vec embedding which is closest to the output embedding:

$$c = \underset{j}{\operatorname{argmin}}(\|\theta_w^j - \theta_o\|_2). \tag{3.14}$$

3.3.3 Loss functions

Our training objective l combines a cross-entropy loss l_{ce}, a reconstruction loss l_{rec}, and a regression loss l_{reg}:

$$l = l_{ce} + l_{rec} + l_{reg}. \tag{3.15}$$

Cross-entropy loss. For the ground-truth label y_i with corresponding class index $k_{gt} \in \mathbb{R}^{K_{seen}}$, the output of our temporal cross-attention θ_{o_i}, and a matrix containing the textual label embeddings for the K_{seen} seen classes $\theta_{w_{seen}}$, we define the cross-entropy loss for n training samples as

$$l_{ce} = -\frac{1}{n}\sum_{i}^{n} y_i \log\left(\frac{\exp\left(\theta_{w_{seen},k_{gt}}\theta_{o_i}\right)}{\sum_{k_j}^{K_{seen}}\exp\left(\theta_{w_{seen},k_j}\theta_{o_i}\right)}\right). \tag{3.16}$$

Regression loss. While the cross-entropy loss updates the probabilities for both the correct and incorrect classes, our regression loss directly focuses on reducing the distance between the output embedding for a sample and the corresponding projected word2vec embedding. The regression loss is based on the mean squared error metric with the following formulation:

$$l_{reg} = \frac{1}{n}\sum_{i=1}^{n}(\theta o_i - \theta_{w_i})^2, \tag{3.17}$$

where θ_{o_i} is the audio-visual embedding, and θ_{w_i} is the projection of the word2vec embedding corresponding to the i-th sample.

Reconstruction loss. The goal of the reconstruction loss is to ensure that the embeddings θ_o and θ_w contain semantic information from the word2vec embedding w. We use $D_u : \mathbb{R}^{d_{out}} \mapsto \mathbb{R}^{d_{dim}}$ with $\rho_u = D_u(\theta_u)$ for $u \in \{o, w\}$. D_w is a sequence of one linear layer, batch normalisation, a ReLU, and dropout with rate $drop_{proj_w}$. D_o is composed of a sequence of two linear layers each followed by batch normalisation, a ReLU, and dropout with dropout rate $drop_{proj_o}$. Our reconstruction loss encourages the reconstruction of the output embedding, ρ_{o_i}, and the reconstruction of the word2vec projection, ρ_{w_i}, to be close to the original word2vec embedding w_i:

$$l_{rec} = \frac{1}{n}\sum_{i=1}^{n}(\rho_{o_i} - w_i)^2 + \frac{1}{n}\sum_{i=1}^{n}(\rho_{w_i} - w_i)^2. \tag{3.18}$$

3.4 Experiments

In this section, we detail our experimental setup (Sec. 3.4.1), and compare to state-of-the-art methods for audio-visual GZSL (Sec. 3.4.2). Furthermore, we present an ablation study in Sec. 3.4.3 which shows the benefits of using our proposed attention scheme and training objective. Finally, we present t-SNE visualisations of our learnt audio-visual embeddings in Sec. 3.4.4.

3.4.1 Experimental setup

Here, we describe the datasets used, the evaluation metrics, and the implementation details for all models.

Datasets. We use the UCF-GZSL^{cls}, VGGSound-GZSL^{cls}, and ActivityNet-GZSL^{cls} datasets [157] for audio-visual (G)ZSL for training and testing all models. [157] introduced benchmarks for two sets of features, the first uses a model pre-trained using self-supervision on the VGGSound dataset from [21], the second takes features extracted from pre-trained VGGish [92] and C3D [237] audio and video classification networks. Since the VGGSound dataset is also used for the zero-shot learning task (VGGSound-GZSL), we selected the second option (using VGGish and C3D) and use the corresponding dataset splits proposed in [157].

In particular, the audio features are extracted using VGGish [92] to obtain one 128-dimensional feature vector for each 0.96 s snippet. The visual features are obtained using C3D [237] pre-trained on Sports-1M [105]. For this, all videos are resampled to 25 fps. A 4096-dimensional feature vector is then extracted for 16 consecutive video frames.

Evaluation metrics. We follow [157, 265] and use the mean class accuracy to evaluate all models. The ZSL performance is obtained by considering only the subset of test samples from the unseen test classes. For the GZSL performance, the models are evaluated on the full test set which includes seen and unseen classes. We then report the performance on the subsets of seen (S) and unseen (U) classes, and also report their harmonic mean (HM).

Implementation details. For TCAF, we use $d_{in_a} = 128$, $d_{in_v} = 4096$, $d_{fhidd} = 512$, $d_{dim} = 300$ and $d_{out} = 64$. Furthermore, TCAF has $L = 6$ transformer layers layers for UCF-GZSL^{cls} and ActivityNet-GZSL^{cls}, and $L = 8$ for VGGSound-GZSL^{cls}. We set $d_{pos} = 64$, $d_{ff} = 128$. For ActivityNet-GZSL^{cls} / UCF-GZSL^{cls} / VGGSound-GZSL^{cls} we use dropout rates $drop_{enc} = 0.1/0.3/0.2$, $drop_{prob,pos} = 0.2/0.2/0.1$, $drop_{prob} = 0.4/0.3/0.5$, $drop_{proj_w} = 0.1/0.1/0.1$, and $drop_{proj_o} = 0.1/0.1/0.2$. All attention blocks use $H = 8$ heads with a dimension of $d_{head} = 64$ per head. We train all models using the Adam optimizer [110] with running average coefficients $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay 0.00001. We use a batch size of 64 for all datasets. In order to efficiently train on ActivityNet-GZSL^{cls}, we randomly trim the features to a maximum sequence length of 60 during training, and we evaluate on features that have a maximum sequence length of 300 and which are centered in the middle of the video. We note, that TCAF can be efficiently trained on a single Nvidia 2080-Ti GPU. All models are trained for 50 epochs. We use a base learning rate of 0.00007 for UCF-GZSL^{cls} and ActivityNet-GZSL^{cls}, and 0.00006 for VGGSound-GZSL^{cls}. For UCF-GZSL^{cls} and ActivityNet-GZSL^{cls} we use a scheduler that reduces the learning rate by a factor of 0.1 when the HM on the validation set has not improved for 3 epochs. To eliminate the bias that the ZSL methods have towards seen classes, we used calibrated stacking [41] on the search space composed of the interval $[0, 3]$ with a step size of 0.2.

We train all models with a two-stage training protocol [157]. In the first stage, we determine the calibrated stacking [41] and the epoch with the best HM performance on the validation set. In the second stage, using the hyperparameters from the first stage, we re-train the models on the union of the training and validation sets. We evaluate the final models on the test set.

3.4.2 Quantitative results

We compare our proposed TCAF to state-of-the-art audio-visual ZSL frameworks and to audio-visual frameworks that we adapted to the ZSL task.

Audio-visual ZSL baselines. We compare our TCAF to three audio-visual ZSL frameworks. **CJME** [185] consists of a relatively simple architecture which maps both input modalities to a shared embedding space. The modality-specific embeddings in the shared embedding space are input to an attention predictor module that determines the dominant modality which is used for the output prediction. **AVGZSLNet** [152] builds on CJME by adding a shared decoder and introducing additional loss functions to improve the performance. AVGZSLNet removes the attention predictor network and replaces it with a simple average between the output from the head of each modality. **AVCA** [157] is a recent state-of-the-art method for audio-visual G(ZSL). It uses a simple cross-attention mechanism on the temporally averaged audio and visual input features to combine the information from the two modalities. Our proposed TCAF improves upon the closely related AVCA framework by additionally ingesting temporal information in the audio and visual inputs with an enhanced cross-attention mechanism that gathers information across time and modalities.

Audio-visual baselines adapted to ZSL. We adapt two attention-based audio-visual frameworks to the ZSL setting. **Attention Fusion** [72] is a method for audio-visual classification which is trained to classify unimodal information. It then fuses the unimodal predictions with learnt attention weights. The **Perceiver** [101] is a scalable multi-modal transformer framework for flexible learning with arbitrary modality information. It uses a latent bottleneck to encode input information by repeatedly attending to the input with transformer-style attention. The Perceiver allows for a comparison to another transformer-based architecture with focus on multi-modality. We adapt the Perceiver to use the same positional encodings and model capacity as TCAF. We use 64 latent tokens and the same number of layers and dimensions as TCAF. Both Attention Fusion and Perceiver use the same input features, input embedding functions A_{enc} and V_{enc}, learning rate and loss functions as TCAF. For Attention Fusion, we temporally average the input features after A_{enc} and V_{enc} to deal with non-synchronous modality sequences due to different feature extraction rates.

All baselines, except for the Perceiver, operate on temporally averaged audio and visual features. This decreases the amount of information contained in the inputs, in particular regarding the dynamics in a video. In contrast to methods that use temporally averaged inputs, TCAF exploits the temporal dimension which boosts the (G)ZSL performance.

Results. We compare the results obtained with our TCAF to state-of-the-art baselines for audio-visual (G)ZSL and for audio-visual learning in Tab. 3.1. TCAF outperforms all previous methods on the VGGSound-GZSL^{cls}, UCF-GZSL^{cls}, and ActivityNet-GZSL^{cls} datasets for both, GZSL performance (HM) and ZSL performance. For ActivityNet-GZSL^{cls}, our proposed model is significantly better than its strongest competitor AVCA,

Model	VGGSound-GZSL^{cls}				UCF-GZSL^{cls}				ActivityNet-GZSL^{cls}			
	S	U	HM	ZSL	S	U	HM	ZSL	S	U	HM	ZSL
Attention Fusion	14.13	3.00	4.95	3.37	39.34	18.29	24.97	20.21	11.15	3.37	5.18	4.88
Perceiver	13.25	3.03	4.93	3.44	46.85	26.82	34.11	28.12	18.25	4.27	6.92	4.47
CJME	10.86	2.22	3.68	3.72	33.89	24.82	28.65	29.01	10.75	5.55	7.32	6.29
AVGZSLNet	**15.02**	3.19	5.26	4.81	**74.79**	24.15	36.51	31.51	13.70	5.96	8.30	6.39
AVCA	12.63	6.19	8.31	6.91	63.15	30.72	41.34	37.72	16.77	7.04	9.92	7.58
TCAF	12.63	**6.72**	**8.77**	**7.41**	67.14	**40.83**	**50.78**	**44.64**	**30.12**	**7.65**	**12.20**	**7.96**

Table 3.1: Performance of our TCAF and of state-of-the-art methods for audio-visual (G)ZSL on the VGGSound-GZSL^{cls}, UCF-GZSL^{cls}, and ActivityNet-GZSL^{cls} datasets. The mean class accuracy for GZSL is reported on the seen (S) and unseen (U) test classes, and their harmonic mean (HM). For the ZSL performance, only the test subset of unseen classes is considered.

with a HM of 12.20% compared to 9.92% and a ZSL performance of 7.96% compared to 7.58%. The CJME and AVGZSLNet frameworks are weaker than the AVCA model. Similar patterns are exhibited for the VGGSound-GZSL^{cls} and UCF-GZSL^{cls} datasets. Interestingly, the GZSL performance for TCAF is improved by a more significant margin than the ZSL performance compared to AVCA across all three datasets. This shows that using temporal information and allowing our model to attend across time and modalities is especially beneficial for the GZSL task.

Furthermore, we observe that the audio-visual Attention Fusion framework and the Perceiver give worse results than AVGZSLNet and AVCA on all three datasets. In particular, our TCAF yields stronger ZSL and GZSL performances than the Perceiver which also takes temporal audio and visual features as inputs, with a HM of 8.77% on VGGSound-GZSL^{cls} for TCAF compared to 4.93% for the Perceiver. Attention Fusion and the Perceiver architecture were not designed for the (G)ZSL setting that uses text as side information. Our proposed training objective, used to also train the Perceiver, aims to regress textual embeddings which might be challenging for the Perceiver given its tight latent bottlenecks.

3.4.3 Ablation study on the training loss and attention variants

Here, we analyse different components of our proposed TCAF. We first compare the performance of our model when trained using different loss functions. We then investigate the influence of the attention mechanisms used in the model architecture on the (G)ZSL performance. Finally, we show that using multi-modal inputs is beneficial and results in outperforming unimodal baselines.

Comparing different training losses. We show the contributions of the different components in our training loss function to the (G)ZSL performance in Tab. 3.2. Using only the regression loss l_{reg} to train our model results in the weakest performance across all datasets, with HM/ZSL performances of 16.25%/30.17% on UCF-GZSL^{cls} compared to 50.78%/44.64% for our full TCAF. Interestingly, the seen performance (S) when using only

Loss	VGGSound-GZSL^{cls}				UCF-GZSL^{cls}				ActivityNet-GZSL^{cls}			
	S	U	HM	ZSL	S	U	HM	ZSL	S	U	HM	ZSL
l_{reg}	0.10	2.41	0.19	2.50	14.30	18.82	16.25	30.17	1.09	0.27	0.43	2.11
$l_{reg} + l_{ce}$	**13.67**	4.06	6.26	4.31	**75.31**	37.15	49.76	41.75	11.36	5.28	7.21	5.31
$l = l_{reg} + l_{ce} + l_{rec}$	12.63	**6.72**	**8.77**	**7.41**	67.14	**40.83**	**50.78**	**44.64**	**30.12**	**7.65**	**12.20**	**7.96**

Table 3.2: Influence of using different components of our proposed training objective for training TCAF on the (G)ZSL performance on the VGGSound-GZSL^{cls}, UCF-GZSL^{cls}, and ActivityNet-GZSL^{cls} datasets.

l_{reg} is relatively weak, likely caused by the calibrated stacking. Similarly, on ActivityNet-GZSL^{cls}, using only l_{reg} yields a low test performance of 0.43% HM. Jointly training with the regression and cross-entropy loss functions ($l_{reg} + l_{ce}$) improves the GZSL and ZSL performance significantly, giving a ZSL performance of 4.31% compared to 2.50% for l_{reg} on VGGSound-GZSL^{cls}. The best results are obtained when training with our full training objective l which includes a reconstruction loss term, giving the best performance on all three datasets.

Comparing different attention variants. We study the use of different attention patterns in Tab. 3.3. In particular, we analyse the effect of using within-modality ($\mathbf{A}_{self}$) and cross-modal ($\mathbf{A}_x$) attention (cf. Eq. 3.11), on the GZSL and ZSL performance. Additionally, we investigate models that use a classification token x^c with corresponding output token c_o (*with class. token*) and models for which we simply average the output of the transformer layers which is then used as input to O_{proj} (*w/o class. token*).

Interestingly, we observe that with no global token, using the full attention $\mathbf{A}_{self} + \mathbf{A}_x$ gives better results than using only cross-attention on UCF-GZSL^{cls} and ActivityNet-GZSL^{cls} for ZSL and GZSL, but is slightly worse on VGGSound-GZSL^{cls}. This suggests that the bottleneck introduced by limiting the information flow in the attention when using only cross-attention is beneficial for (G)ZSL on VGGSound-GZSL^{cls}. When not using the classification token and only self-attention $\mathbf{A}_{self}$, representations inside the transformer are created solely within their respective modalities.

Using a classification token (*with class. token*) and the cross-attention variant ($\mathbf{A}_c + \mathbf{A}_x$) yields the strongest ZSL and GZSL results across all three datasets. The most drastic improvements over full attention can be observed on the UCF-GZSL^{cls} dataset, with a HM of 50.78% for the cross-attention with classification token ($\mathbf{A}_c + \mathbf{A}_x$) compared to 39.18% for the full attention ($\mathbf{A}_c + \mathbf{A}_{self} + \mathbf{A}_x$). Furthermore, when using x_c, cross-attention $\mathbf{A}_x$ instead of self-attention $\mathbf{A}_{self}$ leads to a better performance on all three datasets. For $\mathbf{A}_x$ and x_c, we obtain HM scores of 8.77% and 50.78 % on VGGSound-GZSL^{cls} and UCF-GZSL^{cls} compared to 6.71% and 37.37% with $\mathbf{A}_{self}$ and x_c. This shows that using information from both modalities is important for creating strong and transferable video representations for (G)ZSL. Using the global token relaxes the pure cross-attention setting to a certain extent, since $\mathbf{A}_c$ allows for attention between all tokens from both modalities and the global token. The results in Tab. 3.3 have demonstrated the clear benefits of our cross-attention variant

Model	VGGSound-GZSL^{cls}				UCF-GZSL^{cls}				ActivityNet-GZSL^{cls}			
	S	U	HM	ZSL	S	U	HM	ZSL	S	U	HM	ZSL
w/o class. token												
$\mathbf{A}_{self} + \mathbf{A}_x$	**18.40**	3.78	6.27	4.25	31.70	32.57	32.13	33.26	11.87	3.80	5.75	3.90
$\mathbf{A}_{self}$	16.08	3.56	5.83	4.00	42.59	24.04	30.73	27.49	9.51	4.33	5.95	4.39
$\mathbf{A}_x$	14.62	4.22	6.55	4.59	19.52	29.80	23.62	31.35	1.85	3.50	2.42	3.50
with class. token												
$\mathbf{A}_c + \mathbf{A}_{self} + \mathbf{A}_x$	11.36	5.50	7.41	5.97	36.73	**41.99**	39.18	42.56	17.75	6.79	9.83	6.89
$\mathbf{A}_c + \mathbf{A}_{self}$	12.23	4.63	6.71	5.25	40.14	34.95	37.37	35.74	4.24	3.23	3.67	3.25
$\mathbf{A}_c + \mathbf{A}_x$ (TCAF)	12.63	**6.72**	**8.77**	**7.41**	**67.14**	40.83	**50.78**	**44.64**	**30.12**	**7.65**	**12.20**	**7.96**

Table 3.3: Ablation of different attention variants with and without a classification token on the VGGSound-GZSL^{cls}, UCF-GZSL^{cls}, and ActivityNet-GZSL^{cls} datasets.

Model	VGGSound-GZSL^{cls}				UCF-GZSL^{cls}				ActivityNet-GZSL^{cls}			
	S	U	HM	ZSL	S	U	HM	ZSL	S	U	HM	ZSL
TCAF- audio	5.11	4.06	4.53	4.28	35.51	19.75	25.38	24.24	9.28	4.26	5.84	4.65
TCAF- visual	3.97	3.12	3.50	3.19	38.10	26.84	31.49	27.25	2.75	3.11	2.92	3.11
TCAF	**12.63**	**6.72**	**8.77**	**7.41**	**67.14**	**40.83**	**50.78**	**44.64**	**30.12**	**7.65**	**12.20**	**7.96**

Table 3.4: Influence of using multiple modalities for training and evaluating our proposed model on the (G)ZSL performance on the VGGSound-GZSL^{cls}, UCF-GZSL^{cls}, and ActivityNet-GZSL^{cls} datasets.

used in TCAF.

The influence of multi-modality. We compare using only a single input modality for training TCAF to using multiple input modalities in Tab. 3.4. For the unimodal baselines TCAF- audio and TCAF- visual, we train TCAF only with the corresponding input modality. Using only audio inputs gives stronger GZSL and ZSL results than using only visual inputs on VGGSound-GZSL^{cls} and ActivityNet-GZSL^{cls}. We obtain a HM of 5.84% for audio compared to 2.92% for visual inputs on ActivityNet-GZSL^{cls}. Interestingly this pattern is reversed for the UCF-GZSL^{cls} dataset where using visual inputs only results in a slightly higher performance than using the audio inputs with HM scores of 31.49% compared to 25.38%, and ZSL scores of 27.25% and 24.24%. However, using both modalities (TCAF) increases the HM to 50.78% and ZSL to 44.64% on UCF-GZSL^{cls}. Similar trends can be observed for VGGSound-GZSL^{cls} and ActivityNet-GZSL^{cls} which highlights the importance of the tight multi-modal coupling in our TCAF.

3.4.4 Qualitative results

We present a qualitative analysis of the learnt audio-visual embeddings in Fig. 3.3. For this, we show t-SNE [148] visualisations for the audio and visual input features and for the learnt multi-modal embeddings from 7 classes in the UCF-GZSL^{cls} test set. We averaged the input features for both modalities across time. We observe that the audio and visual input

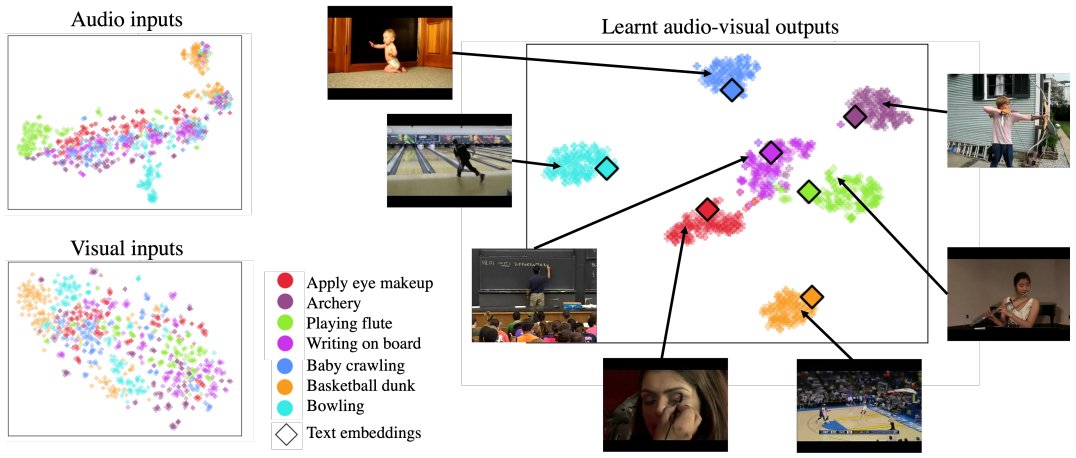


Figure 3.3: t-SNE visualisation for five seen (*apply eye makeup, archery, baby crawling, basketball dunk, bowling*) and two unseen (*playing flute, writing on board*) test classes from the UCF-GZSL^{cls} dataset, showing audio and visual input embeddings extracted with C3D and VGGish, and audio-visual output embeddings learned with TCAF. Textual class label embeddings are visualised with a square.

features are poorly clustered. In contrast, the audio-visual embeddings (θ_o) are clearly clustered for both, seen and unseen classes. This suggests that our network is actually learning useful representations for unseen classes, too. Furthermore, the word2vec class label embeddings (θ_w^j) lie inside the corresponding audio-visual clusters. This confirms that the learnt audio-visual embeddings are mapped to locations that are close to the corresponding word2vec embeddings, showing that our embeddings capture semantic information from the word2vec representations.

3.5 Conclusion

We presented a cross-attention transformer framework that addresses (G)ZSL for video classification using audio-visual input data with temporal information. Our proposed model achieves state-of-the-art performance on the three audio-visual (G)ZSL datasets UCF-GZSL^{cls}, VGGSound-GZSL^{cls}, and ActivityNet-GZSL^{cls}. The use of pre-extracted audio and visual features as inputs results in a computationally efficient framework compared to using raw data. We demonstrated that using cross-modal attention on temporal audio and visual input features and suppressing the contributions from the within-modality self-attention is beneficial for obtaining strong audio-visual embeddings that can transfer information from classes seen during training to novel, unseen classes at test time.

4

Text-to-feature diffusion for audio-visual few-shot learning

Training deep learning models for video classification from audio-visual data commonly requires vast amounts of labeled training data collected via a costly process. A challenging and underexplored, yet much cheaper, setup is few-shot learning from video data. In particular, the inherently multi-modal nature of video data with sound and visual information has not been leveraged extensively for the few-shot video classification task. Therefore, we introduce a unified audio-visual few-shot video classification benchmark on three datasets, i.e. the VGGSound-FSL, UCF-FSL, ActivityNet-FSL datasets, where we adapt and compare ten methods. In addition, we propose AV-Diff, a text-to-feature diffusion framework, which first fuses the temporal and audio-visual features via cross-modal attention and then generates multi-modal features for the novel classes. We show that AV-Diff obtains state-of-the-art performance on our proposed benchmark for audio-visual (generalised) few-shot learning. Our benchmark paves the way for effective audio-visual classification when only limited labeled data is available. Code and data are available at https://github.com/ExplainableML/AVDIFF-GFSL.

4.1 Introduction

The use of audio-visual data can yield impressive results for video classification [171, 187, 268]. The complementary knowledge contained in the two modalities results in a richer learning signal than using unimodal data. However, video classification frameworks commonly rely on significant amounts of costly training data and computational resources. To mitigate the need for large amounts of labeled data, we consider the few-shot learning (FSL) setting where a model is tasked to recognise new classes with only few labeled examples. Moreover, the need for vast computational resources can be alleviated by operating on the feature level, using features extracted from pre-trained visual and sound classification networks.

In this work, we tackle the task of few-shot action recognition in videos from audio and visual data which is an understudied problem in computer vision. In the few-shot setting,

a model has to learn a transferable audio-visual representation which can be adapted to new classes with few annotated data samples. In particular, we focus on the more practical generalised FSL (GFSL) setting, where the aim is to recognise samples from both the base classes, i.e. classes with many training samples, and from novel classes which contain only few examples. Additional modalities, such as text and audio, are especially useful for learning transferable and robust representations from few samples.

To the best of our knowledge, the FSL setting with audio-visual data has only been considered for speech recognition [288], and for learning an acoustic model of 3D scenes [149]. Moreover, existing video FSL benchmarks are not suitable for the audio-visual setting. In particular, the SomethingV2 and HMDB51 benchmarks proposed in [38] and [286] do not contain audio and about 50% of the classes in the UCF101 benchmark from [264] have no sound either. The Kinetics split in [296] suffers from an overlap with the classes used to pre-train the feature extractors [264], and [171, 268] show that the audio modality in Kinetics is less class-relevant than the visual modality. Existing audio-visual zero-shot learning benchmarks [155, 157] cannot directly be used for few-shot learning due to their distinct training and testing protocols. Moreover, the baselines in both settings differ significantly as state-of-the-art few-shot learning methods usually necessitate knowledge of novel classes through classification objectives and generative models, a condition that is not possible in zero-shot learning. Thus, we introduce a new benchmark for generalised audio-visual FSL for video classification that is comprised of three audio-visual datasets and ten methods carefully adapted to this challenging, yet practical task.

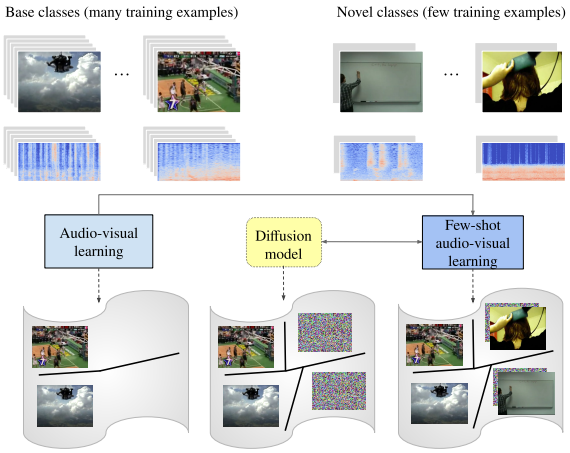


Figure 4.1: AV-DIFF learns to fuse the audio-visual inputs into multi-modal representations in the audio-visual learning stage (left). In the few-shot learning stage (right), the multi-modal representations from the previous stage are used to concurrently train (double arrow line) a text-conditioned diffusion model on all the classes (middle) and a classifier. The classifier is trained on real features from base classes and real and synthetic features from novel classes.

To tackle our new benchmark, we propose AV-DIFF which uses a novel hybrid cross-modal attention for fusing audio-visual information, as shown in Fig. 4.1. Different to various attention fusion techniques in the audio-visual domain [155, 157, 171] which use a single attention type or different transformers for each modality, our model makes use of a novel combination of within-modality and cross-modal attention in a multi-modal transformer. This allows the effective fusion of information from both modalities and

across the temporal dimension of the inputs. Furthermore, we introduce a novel text-conditioned diffusion model for generating audio-visual features to augment the few samples in the novel classes. In the image and video domain, generative adversarial networks (GANs) have been used to generate uni-modal features for data augmentation in the FSL setting [89, 118, 175, 264, 267]. However, we are not aware of prior works that have used diffusion models for multi-modal (audio-visual) feature generation in FSL. Both, cross-modal fusion and the text-to-feature diffusion contribute to significant boosts in performance on our proposed benchmark.

To summarise, our contributions are: 1) We introduce the audio-visual generalised few-shot learning task for video classification and a benchmark on three audio-visual datasets. We additionally adapt and compare ten methods for this task. 2) We propose a hybrid attention mechanism to fuse multi-modal information, and a diffusion model for multi-modal feature generation to augment the training dataset with additional novel-class samples. 3) We obtain state-of-the-art performance across all three datasets, outperforming the adapted multi-modal zero-shot learning and video FSL models.

4.2 Related work

We discuss prior works in learning from audio-visual data, FSL, and feature generation in low-shot learning.

Audio-visual learning. Multi-modal inputs, such as audio and visual data, provide significantly more information than unimodal data, resulting in improved overall performance for video classification and acoustic scene classification [15, 22, 114, 183, 184, 187]. Approaches, such as [50, 72], use class-label supervision between modalities without requiring temporal alignment between the input modalities. Besides audio and video classification, other domains also benefit from multi-modal data, such as lip reading [5, 6], audio synthesis based on visual information [75, 79, 112, 113, 174, 222, 293], and localisation and separation of sounds in videos [4, 7, 19, 44, 76, 182, 235]. Recently, transformer models have gained popularity in audio-visual learning, e.g. for classification [31], event localization [133], dense video captioning [97], and text-based video retrieval [74, 250]. As shown in these works, transformers can effectively process multi-modal input. Thus, our proposed framework fuses audio-visual information using a transformer-based mechanism.

FSL has been explored in the image domain [49, 66, 89, 126, 139, 195, 202, 209, 217, 227, 243, 253, 255, 283] and in the video domain [28, 38, 109, 264, 296]. The popular meta-learning paradigm in FSL [28, 38, 126, 139, 202, 227, 243, 253, 283, 296] has been criticised by recent works [49, 104, 253, 264]. In the video domain, commonly a query and support set is used and each query sample is compared to all the support samples [28, 38, 190, 296]. The number of comparisons grows exponentially with the number of ways and shots. These methods become prohibitively expensive for GFSL, where models are evaluated on both

the base and the novel classes. Hence, we focus on the non-meta learning approach in this work. Some non-meta learning approaches have addressed the more challenging and practical GFSL setting for videos [118, 264] using unimodal visual data. In contrast, we propose to use multi-modal data in our novel (G)FSL benchmark for audio-visual video classification which provides the possibility to test a model in both scenarios (FSL and GFSL).

Feature generation. Due to the progress of generative models, such as GANs [3, 77, 81, 100, 163] and diffusion models [29, 69, 206], different works have tried to adapt these systems to generate features as a data augmentation mechanism. GANs have been used in zero-shot learning (ZSL) and FSL [118, 175, 264, 267] to increase the number and diversity of samples especially for unseen or novel classes. Diffusion models have also been applied to image generation in the feature space, such as [206, 240], but not in the ZSL or FSL setting. It is known that GANs are hard to optimize [210] while diffusion models appear to be more stable, leading to better results [60]. Therefore, our proposed framework uses a text-conditioned diffusion model to generate features for the novel classes in the FSL setting.

4.3 Audio-visual (G)FSL benchmark

We describe the audio-visual (G)FSL setting, present our proposed benchmark that we construct from audio-visual datasets, and explain the methods that we used to establish baselines for this task.

4.3.1 Audio-visual (G)FSL setting

We address the tasks of (G)FSL using audio-visual inputs. The aim of FSL is to recognise samples from classes that contain very few training samples, so-called *novel classes*. In addition, the goal of GFSL is to recognise both *base classes*, which contain a significant amount of samples, and novel classes.

Given an audio-visual dataset $\mathcal{V}$ with M samples and C classes, containing base and novel classes, we have $\mathcal{V} = \{\mathcal{X}_{a[i]}, \mathcal{X}_{v[i]}, y_{[i]}\}_{i=1}^{M}$, where $\mathcal{X}_{a[i]}$ represents the audio input, $\mathcal{X}_{v[i]}$ the video input and $y_{[i]} \in \mathbb{R}^C$ the ground-truth class label. Both the audio and the video inputs contain temporal information. Two frozen, pretrained networks are used to extract features from the inputs, VGGish [92] for the audio features $a_{[i]} = \{a_1, \ldots, a_t, \ldots, a_{F_a}\}_i$ and C3D [237] for video features $v_{[i]} = \{v_1, ..., v_t, ..., v_{F_v}\}_i$. We use these specific feature extractors to ensure that there is no leakage to the novel classes from classes seen when training the feature extractors (Sports1M [105] for the visual and Youtube-8M [1] for the audio modality), similar to [157]. A potential leakage is harmful as it would artificially increase the performance and will not reflect the true performance.

All models are evaluated in the FSL and GFSL settings for k samples in the novel classes (called shots), with $k \in \{1, 5, 10, 20\}$. During inference, in the FSL setting, the class

	# **classes**				# **videos** *stage 1*				# **videos** *stage 2*			
	all	$\mathcal{V}_{B_1}$	$\mathcal{V}_{N_1}$	$\mathcal{V}_{N_2}$	$\mathcal{V}_{B_1}$	$\mathcal{V}_{N_1}$	Val_B	Val_N	$\mathcal{V}_{B_2}$	$\mathcal{V}_{N_2}$	$Test_B$	$Test_N$
(1)	271	138	69	64	70351	345	7817	2757	81270	320	9032	2880
(2)	48	30	12	6	3174	60	353	1407	4994	30	555	815
(3)	198	99	51	48	9204	255	1023	4052	14534	240	1615	3812

Table 4.1: Statistics for our VGGSound-FSL **(1)**, UCF-FSL **(2)**, and ActivityNet-FSL **(3)** benchmark datasets, showing the number of classes and videos in our proposed splits in the 5-shot setting. $\mathcal{V}_{B_1} \cup \mathcal{V}_{N_1}$ are used for training, Val_B and Val_N for validation in the first training stage. $\mathcal{V}_{B_2} \cup \mathcal{V}_{N_2}$ serves as training set in the second stage, and evaluation is done on $Test_B$ and $Test_N$.

search space is composed only of the novel class labels and the samples belonging to these classes. In the GFSL setting, the search space contains both the novel and base class labels and their corresponding samples.

Meta-learning approaches commonly use the notion of episodes, where each episode only uses P novel classes randomly sampled from the total number of novel classes in a dataset, usually $P \in \{1, 5\}$ (coined P-way). However, similar to [264], we suggest to use higher values for P (e.g. all the classes in the dataset), so that the evaluation is closer to the real-world setting, as argued in [89, 264]. In our proposed FSL setting, P corresponds to the total number of novel classes $P = N$, while for GFSL $P = C$. Our evaluation protocol is in line with [89].

4.3.2 Dataset splits and training protocol

We provide training and evaluation protocols for audio-visual (G)FSL along with splits for UCF-FSL, ActivityNet-FSL and VGGSound-FSL. These are based on the UCF-101 [219], ActivityNet [90] and VGGSound [45] datasets.

Our proposed training and evaluation protocol is similar to [89, 155, 157]. The training protocol is composed of two stages, indicated by subscripts $_{1,2}$. In the first stage, a model is trained on the training set $Train_1 = \mathcal{V}_{B_1} \cup \mathcal{V}_{N_1}$ where $\mathcal{V}_{B_1}$ consists of dataset samples from base classes, and $\mathcal{V}_{N_1}$ contains k samples for each of the classes N_1. The trained model is then evaluated on $Val = Val_B \cup Val_N$, where Val is the validation dataset which contains the same classes as $Train_1$. In the first stage, the hyperparameters for the network are determined, such as the number of training epochs and the learning rate scheduler parameters.

In the second stage, the model is retrained on the training set $Train_2$, using the hyperparameters determined in the first stage. Here, $Train_2 = \mathcal{V}_{B_2} \cup \mathcal{V}_{N_2}$ with $\mathcal{V}_{B_2} = Train_1 \cup Val$, and $\mathcal{V}_{N_2}$ contains k samples for the novel classes in the $Test$ set. The final model is evaluated on $Test = Test_B \cup Test_N$ with $Train_2 \cap Test = \emptyset$. With a small number of shots, e.g. $k = 1$, models risk a bias towards the novel samples in $Train_2$. To obtain robust evaluation results, the second stage is repeated three times with k randomly selected, but fixed samples from $\mathcal{V}_{N_2}$. We provide dataset statistics in Tab. 4.1.

4.3.3 Benchmark comparisons

To establish benchmark performances for audio-visual GFSL task, we adapt ten recent state-of-the-art methods for video FSL from visual information only, from audio-visual representation learning, and from audio-visual ZSL.

We provide results with several few-shot video recognition frameworks adapted to the multimodal audio-visual setting.

ProtoGan [118] uses GANs conditioned on the visual prototypes of classes that are obtained by averaging the features of all videos in that class. We adapt it to audio-visual inputs by concatenating the visual and audio features before passing them into the model.

SLDG [30] is a multi-modal video FSL that uses video frames and optical flow as input. It weighs the frame features according to normal distributions. We replace the optical flow in [30] with audio features.

TSL [264] is the current state-of-the-art video FSL which uses a GAN to generate synthetic samples for novel classes. It does not fully use temporal information, as the final score is the average of scores obtained on multiple short segments. We adapt it to the multi-modal setting by concatenating input features from the audio and visual modalities.

Moreover, we have adapted audio-visual representation learning methods to the few-shot task as can be seen below.

Perceiver[101], **Hierarchical Perceiver (HiP)** [39], and **Attention Fusion** [72] are versatile video classification methods and we provide comparisons with them. We use the implementations of the adapted Perceiver and Attention Fusion frameworks provided by [155] and we implement HiP in a similar way.

MBT [171] learns audio-visual representations for video recognition. It uses a transformer for each modality and these transformers can only exchange information using bottleneck attention.

Zorro[203], in contrast to MBT, uses two transformers that do not have access to the bottleneck attention. We adapt it by using a classifier on top of the averaged bottleneck attention tokens.

Finally, we have adapted the state-of-the-art methods in the audio-visual zero-shot learning domain, as shown below.

AVCA [157] is an audio-visual ZSL method which uses temporally averaged features for the audio and visual modalities. We adapt it by using a classifier on the video output, which is the strongest of the two outputs in [157].

TCAF [155] is the state-of-the-art audio-visual ZSL method. It utilizes a transformer architecture with only cross-modal attention, leveraging temporal information in both modalities. As it does not use a classifier, TCAF outputs embeddings, and we determine the class by computing the distance to the semantic descriptors and selecting the closest one.

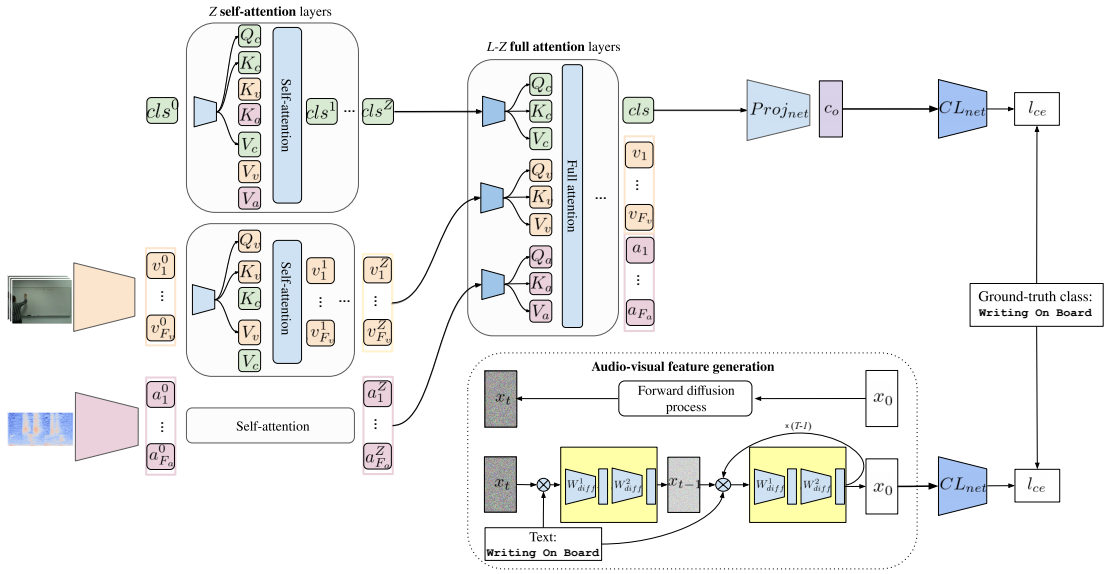


Figure 4.2: Our AV-DIFF model for audio-visual (G)FSL takes audio and visual features extracted from pre-trained audio and video classification models as inputs. During training, the features from both modalities are fused into a classification token, denoted by cls. At the same time, our diffusion model (bottom) generates additional synthetic features for the novel classes (denoted by x_0). Finally, we train our classifier CL_{net} (right) on fused real features c_o of both novel and base classes and synthetic features of novel classes. $\otimes$ is the concatenation operator.

4.4 AV-DIFF framework

In this section, we provide details for our proposed cross-modal AV-DIFF framework which employs cross-modal fusion (Sec. 4.4.1) and a diffusion model to generate audio-visual features (Sec. 4.4.2). Then, we describe the training curriculum in Sec. 4.4.3. Fig. 4.2 illustrates AV-DIFF's full architecture.

4.4.1 Audio-visual fusion with cross-modal attention

Audio-visual fusion. We project the audio $a_{[i]}$ and visual features $v_{[i]}$ to a shared embedding space. Then we use Fourier features [232] as temporal positional embeddings and modality embeddings respectively and obtain positional aware video v_t^E and audio a_t^E tokens for timestep t. We prepend a classification token $cls^0 \in \mathbb{R}^{d_{dim}}$ to the audio and visual tokens. The output token cls corresponding to cls^0 is the final fused audio-visual representation which is input to $Proj_{net}$. Our audio-visual fusion mechanism contains L layers, which are based on multi-head attention [241] Att^l, followed by a feed forward function $\text{FF}^l : \mathbb{R}^{d_{dim}} \rightarrow \mathbb{R}^{d_{dim}}$. The input to the first layer is $x_{in}^1 = [cls^0, a_1^E, \cdots, a_{T_a}^E, v_1^E, \cdots, v_{T_v}^E]$. The output of a layer is:

$$x_{out}^l = \text{FF}^l(\text{Att}^l(x_{in}^l) + x_{in}^l) + \text{Att}^l(x_{in}^l) + x_{in}^l. \tag{4.1}$$

In the following, we describe the first layer of the audio-visual fusion. The other layers work similarly. Our input x_{in}^1 is projected to queries, keys and values with linear maps $s : \mathbb{R}^{d_{dim}} \rightarrow \mathbb{R}^{d_{dim}}$ for $s \in \{q, k, v\}$. The outputs of the projection are written as zero-padded query, key and value features. For the keys we get:

$$\mathbf{K}_c = [k(cls^0), 0, \cdots, 0], \tag{4.2}$$

$$\mathbf{K}_a = [0, \cdots, 0, k(a_1^E), \cdots, k(a_{F_a}^E), 0, \cdots, 0], \tag{4.3}$$

$$\mathbf{K}_v = [0, \cdots, 0, k(v_1^E), \cdots, k(v_{F_v}^E)]. \tag{4.4}$$

The final keys are obtained as $\mathbf{K} = \mathbf{K}_c + \mathbf{K}_a + \mathbf{K}_v$. The queries and values are obtained in a similar way. We define full attention as $\mathbf{A} = \mathbf{A}_c + \mathbf{A}_{cross} + \mathbf{A}_{self}$:

$$\begin{gathered} \mathbf{A}_c = \mathbf{Q}_c \, \mathbf{K}^T + \mathbf{K} \, \mathbf{Q}_c^T, \qquad \mathbf{A}_{cross} = \mathbf{Q}_a \, \mathbf{K}_v^T + \mathbf{Q}_v \, \mathbf{K}_a^T, \\ \mathbf{A}_{self} = \mathbf{Q}_a \, \mathbf{K}_a^T + \mathbf{Q}_v \, \mathbf{K}_v^T. \end{gathered} \tag{4.5}$$

The novelty in the attention mechanism in AV-DIFF is that it exploits a hybrid attention mechanism composed of two types of attention: within-modality self-attention and full-attention. The first Z layers use self-attention $\mathbf{A}_{self} + \mathbf{A}_c$, the subsequent $L - Z$ layers leverage full attention $\mathbf{A}$.

Audio-visual classification. We project cls to $\mathbb{R}^{d_{out}}$ by using a projection network, $c_o = Proj_{net}(cls)$. Then, we apply a classification layer to c_o, $logits = CL_{net}(c_o)$. Given the ground-truth labels gt, we use a cross-entropy loss, $L_{ce} = CE(logits, gt)$ to train the full architecture.

4.4.2 Text-conditioned feature generation

AV-DIFF uses a diffusion process to generate audio-visual features which is based on the Denoising Diffusion Probabilistic Models (DDPM) [93]. In particular, we condition the generation of features for novel classes on a conditioning signal, such as the word embedding (e.g. word2vec [161]) of a class name. The diffusion framework consists of a forward process and a reverse process.

The forward process adds noise to the data sample x_0 for T timesteps:

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}) = \prod_{t=1}^{T} \mathcal{N}\big(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathbf{I}\big), \tag{4.6}$$

where $\beta_1, \ldots, \beta_T$ is the variance schedule.

As the **reverse process** $q(x_{t-1}|x_t)$ is intractable, we approximate it with a parameterised model p_θ:

$$p_\theta(x_{0:T}) = p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) = p_\theta(x_T) \prod_{t=1}^{T} \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \tag{4.7}$$

We condition the model on the timestep t and the class label embedding w,

$$L_{\text{diff},w} = E_{x_0,t,w,\epsilon}[||\epsilon - \epsilon_\theta(\sqrt{\bar{a}_t}x_0 + \sqrt{1-\bar{a}_t}\epsilon, w, t)||^2], \tag{4.8}$$

where ϵ is the noise added at each timestep and ϵ_θ is a model that predicts this noise. The sample at timestep $t-1$ is obtained from timestep t as:

$$p_\theta(x_{t-1}|x_t, w) = \mathcal{N}(x_{t-1}; \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, w, t)), \sigma_t^2\mathcal{I}). \tag{4.9}$$

The input to ϵ_θ at timestep t is obtained by concatenating x_t, w, and t. We optimize $L_{\text{diff},w}$ to learn p_θ.

4.4.3 Training curriculum and evaluation

Each training stage (explained in Sec. 4.3.2) is split into two substages. In the first substage, we train the full architecture (the fusion mechanism, the diffusion model, $Proj_{net}$ and the classifier CL_{net}) on base classes $\mathcal{V}_{B_1}$ (or $\mathcal{V}_{B_2}$ in the second stage) by minimizing $L_{ce} + L_{\text{diff},w}$. The classifier CL_{net} is trained only on real features for the base classes in $\mathcal{V}_{B_1}$ (or $\mathcal{V}_{B_2}$ for the second stage) in the first substage.

During the second substage, we freeze the fusion mechanism and continue to train the diffusion model, $Proj_{net}$ and CL_{net} with the same training objective $L_{ce} + L_{\text{diff},w}$. Here we consider both base and novel classes $\mathcal{V}_{B_1}$ and $\mathcal{V}_{N_1}$ classes (or $\mathcal{V}_{B_2}$ and $\mathcal{V}_{N_2}$ in the second stage), unlike in the first substage where we only used base classes. For each batch composed of real samples from novel classes, we generate a corresponding batch of the same size with synthetic samples using our diffusion model. CL_{net} is then trained on real features from $\mathcal{V}_{B_1}$ (or $\mathcal{V}_{B_2}$ in the second stage) and on real and synthetic features for the classes in $\mathcal{V}_{N_1}$ (or $\mathcal{V}_{N_2}$ in the second stage). Freezing the audio-visual transformer ensures that its fusion mechanism does not overfit to the few samples from the novel classes.

The diffusion model is not used for inference, and the output of the classifier CL_{net} for c_0 provides the predicted score for each class (including the novel classes). The class with the highest score is selected as the predicted class.

4.5 Experiments

In this section, we first provide the implementation details for obtaining the presented results (Sec. 4.5.1). We then report results for our proposed AV-DIFF in our benchmark study (Sec. 4.5.2). Finally, we analyse the impact of different components of AV-DIFF (Sec. 4.5.3).

4.5.1 Implementation details

AV-DIFF uses features extracted from pre-trained audio and visual classification networks as inputs (details provided in the suppl. material). AV-DIFF is trained using $d_{dim} = 300$

Model ↓	VGGSound-FSL						UCF-FSL						ActivityNet-FSL					
	1-shot		*5-shot*		*10-shot*		*1-shot*		*5-shot*		*10-shot*		*1-shot*		*5-shot*		*10-shot*	
	HM	FSL	HM	FSL	HM	FSL	HM	FSL	HM	FSL	HM	FSL	HM	FSL	HM	FSL	HM	FSL
Att. Fusion [72]	15.46	16.37	28.22	31.57	30.73	39.02	37.39	36.88	51.68	47.18	57.91	52.19	4.35	5.82	6.17	8.13	10.67	10.78
Perceiver [101]	17.97	18.51	29.92	33.58	33.65	40.73	44.12	33.73	48.60	40.47	55.33	47.86	17.34	12.53	25.75	21.50	29.88	26.46
MBT [171]	14.70	21.96	27.26	34.95	30.12	38.93	39.65	27.99	46.55	34.53	50.04	39.73	14.26	12.63	23.26	22.38	26.86	26.03
TCAF [155]	19.54	20.01	26.09	32.22	28.95	36.43	44.61	35.90	46.29	37.39	54.19	47.61	16.50	13.01	22.79	21.81	24.78	23.33
ProtoGan [118]	10.74	14.08	25.17	28.87	29.85	34.80	37.95	28.08	42.42	33.63	51.01	40.68	2.77	4.40	2.67	7.81	4.05	8.81
SLDG [30]	16.83	17.57	20.79	25.17	24.11	29.48	39.92	28.91	36.47	28.56	34.31	26.96	13.57	10.30	22.29	19.16	27.81	25.35
TSL [264]	18.73	22.44	19.49	29.50	21.93	31.29	44.51	35.17	51.08	42.42	60.93	55.63	9.53	10.77	10.97	12.77	10.39	12.18
HiP [39]	19.27	18.64	26.82	30.67	29.25	35.13	21.79	34.88	36.44	42.23	50.69	43.29	13.80	10.31	18.10	16.25	19.37	17.06
Zorro [203]	18.88	21.79	29.56	35.17	32.06	40.66	44.35	34.52	51.86	42.59	58.89	49.06	14.56	11.94	23.14	21.94	27.35	26.33
AVCA [157]	6.29	10.29	15.98	20.50	18.08	28.27	43.61	31.24	49.19	36.70	50.53	39.17	12.83	12.22	20.09	21.65	26.02	26.76
AV-DIFF	**20.31**	**22.95**	**31.19**	**36.56**	**33.99**	**41.39**	**51.50**	**39.89**	**59.96**	**51.45**	**64.18**	**57.39**	**18.47**	**13.80**	**26.96**	**23.00**	**30.86**	**27.81**

Table 4.2: **Our benchmark study for audio-visual (G)FSL**: 1,5,10-shot performance of our AV-DIFF and compared methods on (G)FSL. The harmonic mean (HM) of the mean class accuracies for base and novel classes are reported for GFSL. For the FSL performance, only the test subset of the novel classes is considered. Base, novel, and 20-shots performances are included in the suppl. material.

and $d_{out} = 64$. Our fusion network has $L = 5, 4, 8$ transformer layers, the layer after which the attention changes is set to $Z = 3, 2, 5$ on ActivityNet-FSL, UCF-FSL and VGGSound-FSL respectively. We train all models on a single NVIDIA RTX 2080-Ti GPU. The first substage uses 30 epochs while the second one uses 20 epochs. We use the Adam optimizer [110], and $\beta_1 = 0.9, \beta_2 = 0.999$, and weight decay of $1e^{-5}$. We use a learning rate of $7e^{-5}$ for UCF-FSL and ActivityNet-FSL, and $6e^{-5}$ for VGGSound-FSL. For ActivityNet-FSL and UCF-FSL, we use a scheduler that reduces the learning rate by a factor of 0.1 when the performance has not improved for 3 epochs. We use a batch size of 32 for ActivityNet-FSL, and 64 for UCF-FSL and VGGSound-FSL. Each epoch consists of 300 batches. As ActivityNet-FSL has very long videos, we randomly trim the number of features during training to 60. During evaluation, we also trim the videos to a maximum length of 300 features, and the trimmed features are centered in the middle of the video. To reduce the bias towards base classes, we use calibrated stacking [41] on the search space composed of the interval [0,1] with a step size of 0.1. This value is obtained on the validation dataset.

4.5.2 Audio-visual GFSL performance

For each of the models featuring in our benchmark, we report results for three different numbers of shots, i.e. 1-shot, 5-shot, 10-shot on all three datasets in Tab. 4.2. AV-DIFF outperforms all the methods across all shots and datasets for few-shot learning (FSL) and generalised few-shot learning (HM).

For 1-shot, AV-DIFF achieves a HM/FSL of 20.31%/22.95% vs. HM of 19.54% for TCAF and FSL score of 22.44% for TSL on VGGSound-FSL. On 5-shot, our model obtains a HM/FSL of 31.19%/36.56% vs. 29.92% for the Perceiver and FSL of 35.17% for Zorro. Furthermore, AV-DIFF yields slightly better results than the Perceiver in both HM and

FSL for 10 shots, with HM/FSL of 33.99%/41.39% vs. 33.65%/40.73% for the Perceiver. Thus, combining our hybrid attention and the diffusion model is superior to systems that rely solely on powerful attention mechanisms without incorporating generative modeling (Perceiver, TCAF) and systems that incorporate generative modelling, but that do not employ powerful attention mechanisms (TSL, ProtoGan).

Similar trends are observed on UCF-FSL, while on ActivityNet-FSL, the ranking of methods changes dramatically. Methods that perform well on UCF-FSL and VGGSound-FSL, but which do not fully use the temporal information (e.g. Attention Fusion, ProtoGan and TSL) perform weakly on ActivityNet-FSL which contains videos with varying length, including some very long videos, making the setting more challenging. Our AV-DIFF can process temporal information effectively, resulting in robust state-of-the-art results on ActivityNet-FSL.

Interestingly, VGGSound-FSL contains the most classes among the datasets considered, resulting in a significantly lower N (suppl. material, Tab. C.1) than FSL. This also lowers the HM (computed from B, N). On VGGSound-FSL, methods tend to be biased towards novel classes (N $\geq$ B) due to calibration [41]. In this case, HM $\leq$ N $\leq$ FSL. Moreover, some baselines that were also used in audio-visual zero-shot learning [155, 157] (e.g. TCAF) exhibit significant increases in performance even in the 1-shot setting. This is expected as for 1-shot learning, one training example is used from each novel class. This reduces the bias towards base classes, leading to more balanced B and N scores, and thereby better HM and FSL results. Base, novel, and 20-shot performances are included in the suppl. material.

4.5.3 AV-DIFF model ablations

Here, we analyse the benefits of the main components of AV-DIFF, i.e. our proposed audio-visual fusion mechanism, and the diffusion model for feature generation. Furthermore, we analyse the importance of using multiple modalities, and the effect of different semantic representations.

Audio-visual fusion mechanism. Tab. 4.3 ablates our cross-modal fusion mechanism for generating rich audio-visual representations. As shown in Sec. 4.4.1, AV-DIFF uses two types of attention: $\mathbf{A}_{self}$+$\mathbf{A}_c$ for the first few layers and $\mathbf{A}$ for the later layers. For *Alternate AV-DIFF*, we alternate the two types of attention used in AV-DIFF in subsequent layers. We also show our model with $\mathbf{A}_{cross}$+$\mathbf{A}_c$ which is the same attention used by the SOTA audio-visual GZSL framework [155]. On ActivityNet-FSL, AV-DIFF obtains a HM/FSL of 26.96%/23.00% vs. 25.58%/22.65% for $\mathbf{A}_{self}$+$\mathbf{A}_c$. The same trend is seen on UCF-FSL. On VGGSound-FSL we outperform *Alternate AV-DIFF* on HM, but are slightly weaker than $\mathbf{A}_{self}$+$\mathbf{A}_c$ in FSL. Overall, our fusion mechanism is the best across both metrics and datasets.

Feature generation model. In Tab. 4.4, we investigate the impact of different generative models to produce audio-visual features for the novel classes. We compare the diffusion

Model ↓	VGGSound-FSL				UCF-FSL				ActivityNet-FSL			
	B	N	HM	FSL	B	N	HM	FSL	B	N	HM	FSL
A	28.56	31.52	29.98	36.55	78.95	42.07	54.90	43.75	23.10	22.06	22.57	22.53
$\mathbf{A}_{cross} + \mathbf{A}_c$	28.44	32.48	30.33	36.85	82.89	44.33	57.77	47.02	27.02	21.25	23.79	21.98
$\mathbf{A}_{self} + \mathbf{A}_c$	26.68	33.23	29.60	**37.06**	50.10	44.58	47.18	45.03	31.61	21.48	25.58	22.65
Alternate AV-DIFF	27.40	32.60	29.78	36.82	80.25	43.01	56.00	45.81	31.15	21.57	25.49	22.59
AV-DIFF	30.88	31.50	**31.19**	36.56	74.11	50.35	**59.96**	**51.45**	35.84	21.61	**26.96**	**23.00**

Table 4.3: Impact of different audio-visual fusion mechanisms in the 5-shot setting.

Model ↓	VGGSound-FSL				UCF-FSL				ActivityNet-FSL			
	B	N	HM	FSL	B	N	HM	FSL	B	N	HM	FSL
AV-GAN	27.80	31.75	29.64	36.53	83.79	36.20	50.56	37.33	35.12	19.53	25.10	21.35
AV-DIFF	30.88	31.50	**31.19**	**36.56**	74.11	50.35	**59.96**	**51.45**	35.84	21.61	**26.96**	**23.00**

Table 4.4: Influence of using different feature generators in the 5-shot setting.

model in AV-DIFF to a GAN similar to the one used by TSL [264], which optimizes a Wasserstein GAN loss [20]. On ActivityNet-FSL, we observe that AV-DIFF outperforms the GAN variant, with a HM/FSL of 26.96%/23.00% vs. 25.10%/21.35% for the GAN. The same can be seen on UCF-FSL and VGGSound-FSL. This shows that our generative diffusion model is better suited for audio-visual GFSL than a GAN.

Multi-modal input. We explore the impact of using multi-modal inputs for AV-DIFF in Tab. 4.5. For unimodal inputs, we adapt AV-DIFF to only employ full attention which is identical to self-attention in this case. On ActivityNet-FSL, using multi-modal inputs provides a significant boost in performance compared to unimodal inputs, with a HM/FSL of 26.96%/23.00% vs. 19.01%/17.84% when using only visual information. The same trend can be observed on UCF-FSL. In contrast, on VGGSound-FSL, using multi-modal inputs gives stronger GFSL but slightly weaker results in FSL than using the audio modality. This might be due to the focus on the audio modality in the data curation process for VGGSound. As a result, significant portions of the visual information can be unrelated to the labelled class. Overall, the use of multi-modal inputs from the audio and visual modalities significantly boosts the (G)FSL performance for AV-DIFF.

However, one interesting aspect is that using both modalities leads to better B and N performances across all three datasets. For example, on ActivityNet-FSL, AV-DIFF obtains a B score of 35.84% and an N score of 21.61% compared to 20.80% and 17.49% when using only the visual modality. On UCF-FSL, AV-DIFF achieves a score of 74.11% for B and 50.35% for N compared to 67.13% and 39.18% for the visual and audio modalities respectively. Finally, on VGGSound-FSL, AV-DIFF achieves a B score of 30.88% and an N score of 31.50% compared to 28.30% and 30.56% for unimodal audio inputs. This shows that using multi-modal inputs decreases the bias towards either of the metrics, leading to a more robust and balanced system.

Semantic class representations. We consider using different semantic class representations

Model ↓	VGGSound-FSL				UCF-FSL				ActivityNet-FSL			
	B	N	HM	FSL	B	N	HM	FSL	B	N	HM	FSL
Audio	28.30	30.56	29.39	**36.64**	55.31	39.18	45.87	44.44	13.74	15.23	14.45	17.58
Visual	7.83	8.92	8.35	9.51	67.13	30.70	42.14	30.98	20.80	17.49	19.01	17.84
AV-DIFF	30.88	31.50	**31.19**	36.56	74.11	50.35	**59.96**	**51.45**	35.84	21.61	**26.96**	**23.00**

Table 4.5: Influence of using multi-modal input in the 5-shot setting.

Model ↓	VGGSound-FSL				UCF-FSL				ActivityNet-FSL			
	B	N	HM	FSL	B	N	HM	FSL	B	N	HM	FSL
AV-DIFF av_{prot}	25.74	33.00	28.92	35.76	83.38	42.46	56.26	44.78	32.22	21.50	25.79	22.73
AV-DIFF	30.88	31.50	**31.19**	**36.56**	74.11	50.35	**59.96**	**51.45**	35.84	21.61	**26.96**	**23.00**

Table 4.6: Influence of different semantic class representations in the 5-shot setting.

in Tab. 4.6. In FSL, the most common semantic descriptor is word2vec [161] which is used to condition the audio-visual feature generation in AV-DIFF. However, related works (e.g. ProtoGan [118]), use prototypes which average the visual features of all the training videos in a class to obtain the semantic representation of that class. In the multi-modal setting, we can concatenate the audio and visual prototypes to obtain multi-modal prototypes av_{prot} which is used as a conditioning signal for our diffusion model. On ActivityNet-FSL, using word2vec embeddings leads to better results than using the audio-visual prototypes av_{prot}, with a HM/FSL of 26.96%/23.00% vs. 25.79%/22.73% for av_{prot}. The same can be seen on UCF-FSL and VGGSound-FSL, demonstrating that the word2vec embeddings provide a more effective conditioning signal.

4.6 Conclusion

In this work, we propose an audio-visual (generalised) few-shot learning benchmark for video classification. Our benchmark includes training and evaluation protocols on three datasets, namely VGGSound-FSL, UCF-FSL and ActivityNet-FSL, and baseline performances for ten state-of-the-art methods adapted from different fields. Moreover, we propose AV-DIFF which fuses multi-modal information with a hybrid attention mechanism and uses a text-conditioned diffusion model to generate features for novel classes. AV-DIFF outperforms all related methods on the new benchmark. Finally, we provided extensive model ablations to show the benefits of our model's components. We hope that our benchmark will enable significant progress for audio-visual generalised few-shot learning.

5

Audio-Visual Generalized Zero-Shot Learning using Pre-Trained Large Multi-Modal Models

Audio-visual zero-shot learning methods commonly build on features extracted from pre-trained models, e.g. video or audio classification models. However, existing benchmarks predate the popularization of large multi-modal models, such as CLIP and CLAP. In this work, we explore such large pre-trained models to obtain features, i.e. CLIP for visual features, and CLAP for audio features. Furthermore, the CLIP and CLAP text encoders provide class label embeddings which are combined to boost the performance of the system. We propose a simple yet effective model that only relies on feed-forward neural networks, exploiting the strong generalization capabilities of the new audio, visual and textual features. Our framework achieves state-of-the-art performance on VGGSound-GZSL^{cls}, UCF-GZSL^{cls}, and ActivityNet-GZSL^{cls} with our new features. Code and data available at https://github.com/dkurzend/ClipClap-GZSL.

5.1 Introduction

The synergy of audio and visual modalities is a valuable asset for tasks like video classification. Imagine a bustling street captured on camera, where the integration of audio—such as footsteps, car engines, or a dog barking—provides crucial context for interpreting the visual content. In practical deep learning applications, models often encounter new and unseen data, e.g. objects or scenes not present in their training data. This challenge arises due to the vast diversity of real-world data and the impracticality of preparing models for every possible variation. A well-designed deep learning model should exhibit the ability to transfer knowledge from familiar classes to unseen ones.

Audio-visual generalized zero-shot learning (GZSL) aims at classifying videos using audio and visual inputs. Previous work [94, 152, 155, 157, 185] learns to align audio-visual embeddings with corresponding class label embeddings. The class label embedding that is closest to the audio-visual embedding is chosen for the prediction. Existing audio-visual

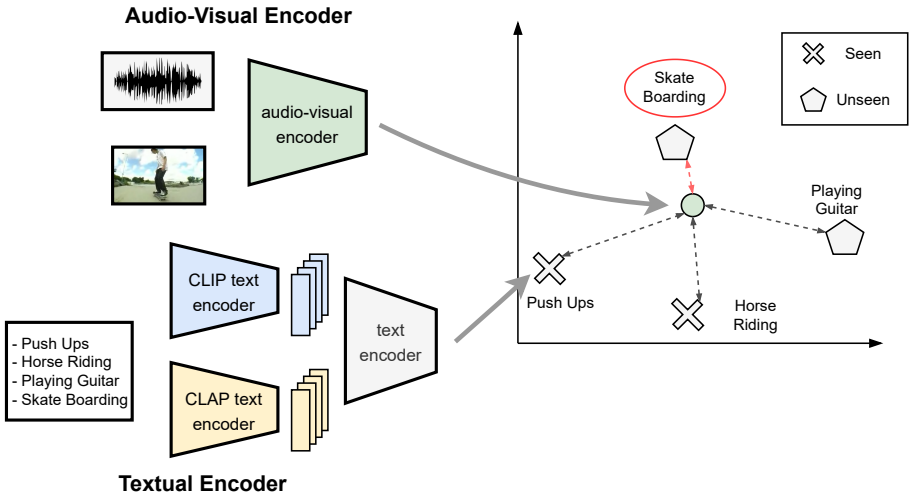


Figure 5.1: Our framework for audio-visual GZSL maps the audio and visual data to embeddings that are aligned with class label embeddings that are obtained from merging CLIP and CLAP embeddings. The class label embedding that is closest to the audio-visual embedding determines the class prediction. At test time, the set of class label embeddings contains both seen and unseen classes.

GZSL methods build on features obtained from pre-trained models for the audio, visual and textual data. However, the feature extraction methods [21, 40, 92, 117, 237] used in most previous works [94, 152, 155, 185] do not reflect the state of the art anymore. In recent years, the transformer architecture [241] has proved successful in many areas such as natural language processing [2, 59, 197], the vision domain [63, 197] or the audio domain [47, 80]. CLIP [197] is a popular vision-language model which contains transformers as the text and image encoders that map to a joint multi-modal embedding space. [153] introduces CLAP, a similar method for the audio-language domain.

In this work, we address audio-visual GZSL by exploring pre-trained multi-modal models to produce audio, visual, and textual input features. We show that the high generalization capabilities of such large pre-trained models are beneficial in the GZSL setting. We use CLIP [197] for visual feature extraction and CLAP [153] for audio feature extraction. Both models contain text encoders which provide input class label embeddings. Consequently, a novel feature of our method compared to prior work (e.g. [94, 152, 155, 157, 185]) is the usage of two class label embeddings that are aggregated into a unified label embedding. Since the textual embeddings are obtained from vision-langauge and audio-language models, the audio and visual input features are already aligned with the corresponding class embeddings. Our proposed model (see Fig. 5.1 for an overview) ingests the aforementioned input features and class label embeddings, only relying on simple feed-forward neural networks in conjunction with a composite loss function. Our contributions can be summarized as follows:

- Our proposed audio-visual GZSL framework builds on features from pre-trained

multi-modal models. Moreover, we exploit the text encoders in the same multi-modal models to provide two class label embeddings that are combined to form a unified and robust textual class label embedding;

- Our simple but effective framework achieves state-of-the-art results on the VGGSound-GZSL^{cls}, UCF-GZSL^{cls} and ActivityNet-GZSL^{cls} datasets when using the new input features;

- Qualitative analysis shows that our approach produces well-separated clusters for the seen and unseen classes in the embedding space.

5.2 Related Work

In this section, we summarize related work concerned with audio-visual learning, zero-shot learning, and audio-visual generalized zero-shot learning.

Audio-Visual Learning. Using audio data for video analysis can significantly enhance the visual representations, for instance for sound source localization [4, 19, 44, 196, 235, 274], sound source separation [76, 238, 298] or both sound source localisation and separation [7, 182, 290, 291]. Various works perform the audio-visual correspondence task [4, 19, 22, 44, 183, 184, 196] or synchronization task [43, 51, 54, 67, 108, 114, 182, 268, 291] to learn representations that contain useful knowledge of both modalities. Moreover, [15, 21, 50, 51, 156, 171, 187, 268] learn rich audio-visual representations. For example, [15, 21, 51, 187, 268] use self-supervision to learn these rich audio-visual representations, while [50] uses knowledge distillation, and [156, 171] use a supervised learning objective along with a transformer specifically designed to merge the audio and the visual modalities. Multiple works combine the audio and visual modalities for speech recognition and lip reading [5, 6, 145, 170]. Other tasks where audio and visual modalities are combined include spotting of spoken keywords [166] and audio synthesis from visual information [75, 79, 112, 113, 174, 222, 223, 293].

Zero-Shot Learning (ZSL) involves training a model to classify new test classes not seen during training, e.g. by learning a mapping between input features and semantic embeddings. Typically, the semantic embeddings are obtained as text embeddings from class labels [9, 10, 73, 177, 263, 266] and from class attributes and descriptions [120, 207, 266]. Other works use generative methods to synthesize data for unseen classes [82, 85, 88, 242, 266]. In [68, 122, 153, 192, 197, 284, 299], ZSL is performed by applying a pre-trained model on new, unseen datasets. Recently, CLIP text and image encoders have been integrated into various ZSL frameworks [61, 86, 125, 130, 143, 178, 247, 260, 276, 295], e.g. for zero-shot / open-vocabulary semantic segmentation [61, 130, 143, 276, 295]. Taking advantage of the strong generalization ability of large pretrained multi-modal models, we use CLIP [197] and CLAP [153] as feature extraction methods for the visual and audio domain.

Audio-visual GZSL was first introduced in [185] which proposed the AudioSetZSL datset. [152, 185] both proposed methods that map the audio, visual, and textual input features (i.e. word2vec [161]) to a joint embedding space. [157] curated several new benchmarks for the audio-visual GZSL task, along with a framework that uses cross-attention between the audio and the visual modalities. [94] additionally uses a hyperbolic alignment loss. While [94, 152, 157, 185] ingest temporally averaged audio and visual features from pre-trained audio and visual classifiers, [155] exploits the inherent temporal structure of videos. In contrast to prior work that fuses the audio and visual information at later stages, our method directly concatenates audio and visual input features before passing them into a feed-forward neural network. Furthermore, our proposed method utilizes two input class label embeddings obtained from CLIP and CLAP.

5.3 Proposed Approach

In this section, we motivate the use of features extracted from large pre-trained multi-modal models, describe the audio-visual GZSL setting and our proposed framework and training objective.

The audio-visual GZSL benchmarks introduced in prior work [155, 157] build on features extracted from audio and video classification networks. However, those feature extraction methods date back to 2017 and 2015 for the audio [92] and visual features [237] respectively. CLIP [197] and CLAP [153] have shown impressive generalization capabilities. We propose to use features extracted from CLIP and CLAP as inputs to our framework, eliminating the need for a complex architecture to adapt to the audio-visual GZSL task. We use text embeddings obtained from CLIP and CLAP which are aligned with corresponding audio / visual features.

Audio-visual GZSL setting. In the ZSL setting, two disjoint sets of classes are considered, i.e. seen and unseen classes S and U with $\mathrm{S} \cap \mathrm{U} = \varnothing$. In ZSL, the model is trained on the seen classes and later evaluated on the test set, which only consists of unseen classes. In the GZSL setting the model is trained on seen classes (S), but the test set contains both seen and unseen classes, making this scenario more realistic.

Formally, the set of data samples that belong to the seen classes is denoted by $\mathrm{S} = (v_i^s, a_i^s, w_i^s, y_i^s)_{i \in \{1,\cdots,N\}}$ where each data point i is a quadruple where a_i^s is the audio feature, v_i^s is the visual feature, y_i^s is the ground-truth class label of sample i and w_i^s is the textual label embedding corresponding to the ground-truth class label. N is the number of samples in S. Likewise, the set of samples from unseen classes of size M is defined as $\mathrm{U} = \left(v_i^u, a_i^u, w_i^u, y_i^u\right)_{i \in \{1,\cdots,M\}}$. In GZSL, the goal is to learn a function $h : \left(v_i^s, a_i^s\right) \mapsto w_i^s$ which for samples from unseen classes fulfills $h\left(v_i^u, a_i^u\right) = w_i^u$. The total number of classes is denoted as K and the class label $j \in \{1, \ , 2 \ \ldots, \ K\}$. The number of seen and unseen classes are denoted as K_s and K_u.

Model architecture. Our proposed model is visualized in Fig. 5.2. It accepts audio and

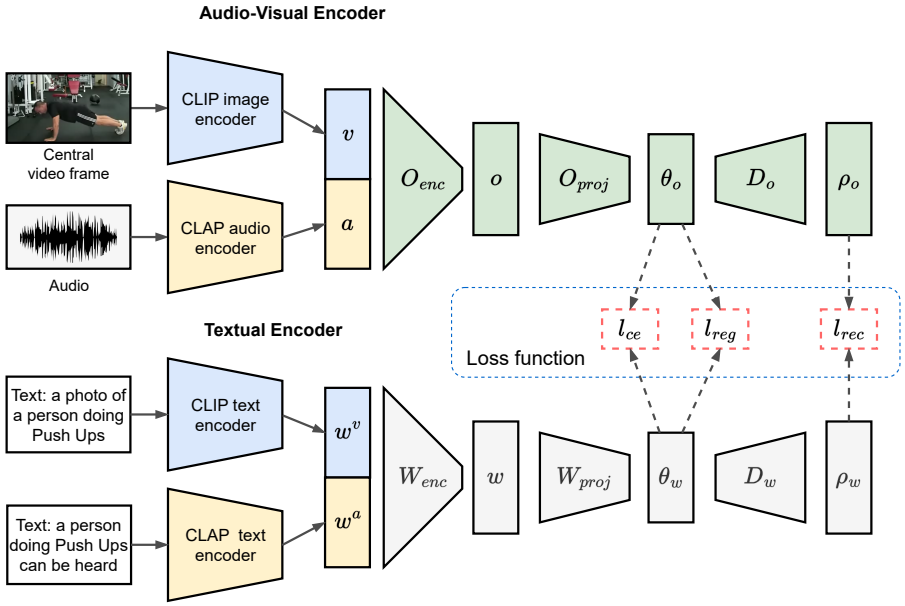


Figure 5.2: The image and audio encoders of CLIP and CLAP are used to extract features from the raw input which are concatenated and passed through multiple feed-forward networks to get an audio-visual output embedding θ_o. Likewise, the text encoders of CLIP and CLAP are used to extract textual label embeddings. They are passed through a series of neural networks to obtain a learned class label embedding θ_w. Both θ_o and θ_w reside in a joint embedding space.

visual features as inputs, denoted as $a \in \mathbb{R}^{d_{in_a}}$ and $v \in \mathbb{R}^{d_{in_v}}$ respectively (for simplicity, subscripts i denoting the i^{th} sample are dropped). These features are obtained using CLAP and CLIP as feature extractors. In addition, our model takes as input text embeddings $w^v \in \mathbb{R}^{d_{in_v}}$ from CLIP and $w^a \in \mathbb{R}^{d_{in_a}}$ from CLAP. CLIP and CLAP merely serve as feature extractors and thus are not optimized when training our framework. Our proposed model consists of a branch for the audio-visual features, and a branch for the textual label embeddings. In the audio-visual branch, the inputs a and v are first concatenated and then passed through an encoder block O_{enc} to produce the audio-visual input

$$o = O_{enc}(\text{concat}\,(v, a)), \tag{5.1}$$

where $o \in \mathbb{R}^{d_{model}}$. O_{enc} consists of a linear layer $f_{O_{enc}} : \mathbb{R}^{(d_{in_v}+d_{in_a})} \rightarrow \mathbb{R}^{d_{model}}$, followed by batch normalization [98], a ReLU activation function [172], and dropout [220]. To get the final audio-visual embedding $\theta_o \in \mathbb{R}^{d_{out}}$ that is used for the prediction, o is passed through a projection network $\theta_o = O_{proj}(o)$, where θ_o is composed of linear layers $f^1_{O_{proj}} : \mathbb{R}^{d_{model}} \rightarrow \mathbb{R}^{d_{hidden}}$ and $f^2_{O_{proj}} : \mathbb{R}^{d_{hidden}} \rightarrow \mathbb{R}^{d_{out}}$. Both layers are followed by batch normalization, ReLU, and dropout.

The textual branch follows a similar structure as the audio-visual branch. First, w^a and w^v are concatenated and are input into an encoder network to generate a unified text embedding $w \in \mathbb{R}^{d_{model}}$,

$$w = W_{enc}(\text{concat}\,(w^v, w^a)). \tag{5.2}$$

W_{enc} contains a linear layer $f_{W_{enc}} : \mathbb{R}^{(d_{in_v}+d_{in_a})} \rightarrow \mathbb{R}^{d_{model}}$ followed by batch normalization, ReLU, and dropout. The output w is further processed by a projection layer $\theta_w = W_{proj}(w)$, where $\theta_w \in \mathbb{R}^{d_{out}}$. W_{proj} is given by a linear layer $f_{W_{proj}} : \mathbb{R}^{d_{model}} \rightarrow \mathbb{R}^{d_{out}}$ with batch normalization, ReLU, and dropout. The goal of the model is to align the projected label embedding θ_w and the audio-visual output embedding θ_o in a joint embedding space of dimension d_{out}, such that θ_o is closest to the θ_w that corresponds to the ground-truth class.

At test time, classification is done by calculating θ_w for all the classes and determining the class label c that is closest to θ_o:

$$c = \underset{j}{\operatorname{argmin}} \left(\left\| \theta_w^j - \theta_o \right\|_2 \right), \tag{5.3}$$

where θ_w^j is the output label embedding θ_w for class j.

Training objective. The loss function l used to train our framework is adopted from [155] and consists of a cross-entropy loss l_{ce}, a reconstruction loss l_{rec}, and a regression loss l_{reg}. The final loss is then given by

$$l = l_{ce} + l_{rec} + l_{reg}. \tag{5.4}$$

The **cross-entropy loss** is given by

$$l_{ce} = -\frac{1}{n} \sum_{i}^{n} \log \left(\frac{\exp\left(\theta_{w_{seen},k_{gt_i}} \theta_{o_i}\right)}{\sum_{k_j}^{K_s} \exp\left(\theta_{w_{seen},k_j} \theta_{o_i}\right)} \right), \tag{5.5}$$

where $\theta_{w_{seen}} \in \mathbb{R}^{K_s \times d_{out}}$ denotes the matrix of the projected class embeddings for the seen classes. $k_{gt} \in \{1, ,2 \ldots, K_s\}$ refers to the ground-truth class index, and thus $\theta_{w_{seen},k_{gt_i}}$ selects the row of $\theta_{w_{seen}}$ that belongs to the target of the current sample i. The number of training samples is denoted by n.

The **reconstruction loss** semantically aligns the output embeddings θ_o and θ_w. For this, two decoder networks, D_o and D_w, are used to obtain $\rho_o = D_o(\theta_o)$ with $\rho_o \in \mathbb{R}^{d_{model}}$. D_o consists of two linear layers $f_{D_o}^1 : \mathbb{R}^{d_{out}} \rightarrow \mathbb{R}^{d_{hidden}}$ and $f_{D_o}^2 : \mathbb{R}^{d_{hidden}} \rightarrow \mathbb{R}^{d_{model}}$ which are both followed by batch normalization, ReLU, and dropout. D_w gets θ_w as input, such that $\rho_w = D_w(\theta_w)$, where $\rho_w \in \mathbb{R}^{d_{model}}$. D_w consists of one linear layer $f_{D_w} : \mathbb{R}^{d_{out}} \rightarrow \mathbb{R}^{d_{model}}$ with batch normalization, ReLU, and dropout.

The reconstruction loss encourages the reconstructions ρ_o and ρ_w to be close to the label embedding w by minimizing

$$l_{rec} = \frac{1}{n} \sum_{i=1}^{n} \left(\rho_{o_i} - w_i\right)^2 + \frac{1}{n} \sum_{i=1}^{n} \left(\rho_{w_i} - w_i\right)^2, \tag{5.6}$$

where n is the number of training samples.

The **regression loss** computes the mean squared error between the output embeddings of the model and the ground-truth label embeddings:

$$l_{reg} = \frac{1}{n} \sum_{i=1}^{n} \left(\theta_{o_i} - \theta_{w_i}\right)^2, \tag{5.7}$$

where θ_{o_i} is the audio-visual embedding for sample i and θ_{w_i} is the corresponding output label embedding.

5.4 Experiments

In this section, we describe our experimental setup Sec. (5.4.1), our results (Sec. 5.4.2), and ablate crucial components of our framework (Sec. 5.4.3).

5.4.1 Experimental setup

Here, we describe the evaluation metrics, the features used, implementation details, and our baselines.

Evaluation metrics. We evaluate our audio-visual GZSL method on the VGGSound-GZSL^{cls}, UCF-GZSL^{cls}, and ActivityNet-GZSL^{cls} datasets introduced in [157] and as suggested in [155] (instead of using the *main split* also introduced in [157]). We follow [155, 157, 265] and report the mean class accuracy scores for the seen classes (acc_{S}) and unseen classes (acc_{U}) separately. For the GZSL performance metric, their harmonic mean is obtained as

$$HM = \frac{2 * acc_{\mathrm{U}} * acc_{\mathrm{S}}}{acc_{\mathrm{U}} + acc_{\mathrm{S}}}. \tag{5.8}$$

In addition, we calculate the zero-shot learning performance as the mean class accuracy acc_{ZSL} for the unseen classes. In this setting, only classes from the subset of unseen test classes can be selected as prediction.

Feature extraction. We do not rely on the same feature extractors as previous work [94, 155, 157]. Instead, visual features v_i and class label embeddings w_j^v are extracted from the videos using **CLIP** [197]. For each video, the middle frame is passed through the image encoder of ViT-B/32 CLIP model, yielding a 512−dimensional feature vector. In addition, for each class label, a 512−dimensional textual embedding is extracted using the CLIP text encoder. Here, we follow [197], which recommends the usage of text prompt ensembles. We provide more details and a concrete list of text prompts in the supplementary materials in D.1.1.

Likewise, audio features a_i and class label embeddings w_j^a are extracted using **CLAP** [153]. The raw audio data is resampled to 32000 Hz and center cropped or zero-padded to 10 seconds, depending on the audio length. 64-dimensional log mel-spectrograms are extracted from the audio by using a 1024-point Hanning window with a hop size of 320. Audio embeddings are obtained from the audio encoder of CLAP, and text embeddings from its text encoder. The joint audio and textual embedding space in CLAP is of size 1024. For the class text embeddings, text prompt ensembles are used similar to CLIP (See more details in the supplementary materials in D.1.2).

Implementation details. To train our framework, we use the Adam optimizer [110] with weight decay $1e^{-5}$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and a batch size of 64. Furthermore, the initial learning rates are $1e^{-4}$ / $7e^{-5}$ / $1e^{-4}$ for VGGSound-GZSL^{cls} / UCF-GZSL^{cls} / and

ActivityNet-GZSL^{cls}. When the validation HM score does not improve for 3 consecutive epochs during training, the learning rate is reduced by a factor of 0.1. In the first training stage, the models for VGGSound-GZSL^{cls} and ActivityNet-GZSL^{cls} were trained for 15 epochs, while for UCF-GZSL^{cls} we used 20 epochs. Calibrated stacking [41] is a scalar which biases the output of the network towards unseen classes, as the network without the calibration is significantly biased towards seen classes. To address the inherent bias of ZSL methods towards seen classes, we use calibrated stacking with search space interval [0, 5] and a step size of 0.07.

We follow the training and evaluation protocol of [94, 155, 157] for our method. The training is divided into two stages. In the first stage, models are trained on the training set. The validation set comprising val(U) and val(S) is used to determine model parameters such as those for calibrated stacking and the best epoch based on the HM score. In the second training stage, the models are trained again using the parameters determined in the first stage, however, this time, the models are trained on the union of the training and validation set {train ∪ val(S) ∪ val(U)}. Finally, the final results on the test set are obtained by evaluating the models trained in the second stage.

The inputs are of size $d_{in_a} = 1024$ and $d_{in_v} = 512$. For the VGGSound-GZSL^{cls}, UCF-GZSL^{cls} and ActivityNet-GZSL^{cls} datasets, the model dimension $d_{model} = 512$ is chosen, and the output dimension is set to $d_{out} = 64$. In addition, for all three datasets, d_{hidden} is set to 512, and the dropout rate is set to 0.1. All models were trained on a single NVIDIA GeForce RTX 2080 Ti GPU.

Baselines. We compare our framework with the state-of-the-art methods CJME [185], AVGZSLNet [152], AVCA [157], and Hyper-multiple [94]. For CJME, AVGZSLNet and AVCA we use the training parameters from [157] and we evaluate Hyper-multiple using the training parameters from [94]. All methods are evaluated using the new CLIP and CLAP features. For fairness, we adjust the baseline methods for our input representation to using two textual input embeddings, by appending an additional layer at the beginning of the network as

$$w_i = W_{enc}(\text{concat}\left(w_i^v, w_i^a\right)), \tag{5.9}$$

where $w_i^v \in \mathbb{R}^{512}$ is the CLIP class label embedding for sample i and $w_i^a \in \mathbb{R}^{1024}$ is the CLAP class label embedding.

5.4.2 Experimental results

In this section, we present quantitative and qualitative results for audio-visual GZSL obtained with our proposed framework.

Quantitative results. On all three datasets, our model outperforms all baseline methods in terms of GZSL performance (HM) as can be seen in Tab. 5.1. For example, on UCF-GZSL^{cls}, we achieve a HM score of 55.97% whereas the next best baseline (AVGZSLNet) achieves 42.67%. Similarly, on ActivityNet-GZSL^{cls}, our model achieves a HM score of 27.93%

Method	VGGSound-GZSL^{cls}				UCF-GZSL^{cls}				ActivityNet-GZSL^{cls}			
	acc_S	acc_U	HM	acc_{ZSL}	acc_S	acc_U	HM	acc_{ZSL}	acc_S	acc_U	HM	acc_{ZSL}
CJME [185]	11.96	5.41	7.45	6.84	48.18	17.68	25.87	20.46	16.06	9.13	11.64	9.92
AVGZSLNet [152]	13.02	2.88	4.71	5.44	56.26	34.37	42.67	35.66	14.81	11.11	12.70	12.39
AVCA [157]	**32.47**	6.81	11.26	8.16	34.90	38.67	36.69	38.67	24.04	19.88	21.76	20.88
Hyper-multiple [94]	21.99	8.12	11.87	8.47	43.52	39.77	41.56	40.28	20.52	**21.30**	20.90	22.18
Ours	29.68	**11.12**	**16.18**	**11.53**	**77.14**	**43.91**	**55.97**	**46.96**	**45.98**	20.06	**27.93**	**22.76**

Table 5.1: Performance of our model compared to state-of-the-art methods for audio-visual (G)ZSL on the VGGSound-GZSL^{cls}, UCF-GZSL^{cls} and ActivityNet-GZSL^{cls} datasets. For a fair comparison, all baselines are also trained and evaluated using both CLIP and CLAP features and class label embeddings. We report the mean class accuracy for seen (acc_S) and unseen (acc_U) classes, along with their harmonic mean (HM) for GZSL performance. In addition, ZSL performance (acc_{ZSL}) is reported.

while Hyer-multiple achieves 20.90%, showing an improvement of 7.03%. Finally, on VGGSound-GZSL^{cls}, our model achieves 16.18%, compared to 11.87% for Hyper-multiple.

Our method also outperforms all the baselines on all three datasets for ZSL (acc_{ZSL}). On VGGSound-GZSL^{cls}, we achieve a acc_{ZSL} score of 11.53% while the second-best method achieves a score of 8.47%. On UCF-GZSL^{cls}, our method achieves a acc_{ZSL} performance of 46.96%, compared to 40.28% for Hyper-multiple. On ActivityNet-GZSL^{cls}, the difference in ZSL performance is very small. Our model obtains a acc_{ZSL} score of 22.76% while Hyper-multiple achieves 22.18%.

In terms of the seen and unseen scores acc_S and acc_U, our method is the best performing model most of the time. On VGGSound-GZSL^{cls} we achieve the second best acc_S of 29.68%, while AVCA achieves 32.47%. For the acc_U, our model performs best with 11.12% vs. 8.12% achieved by Hyper-multiple. On UCF-GZSL^{cls}, our model achieves the highest acc_S / acc_U scores with 77.14% / 43.91% compared to 56.26% / 39.77% achieved by AVGZSLNet / Hyper-multiple. On ActivityNet-GZSL^{cls}, we achieve the best acc_S score with 45.98%, compared to 24.04% achieved by AVCA. For the acc_U performance, only Hyper-multiple performs better than our method with 21.30% vs. 20.06%.

The GZSL and ZSL results show, that when using CLIP and CLAP as feature extraction methods, a rather simple model like our method, is able to outperform methods that use more sophisticated architectural components such as cross-attention used in AVCA and Hyper-multiple, or complex concepts from hyperbolic geometry used in Hyper-multiple.

Our method uses around 2.2 million parameters, whereas CJME and AVGZSLNet use 2.3 million parameters, and AVCA and Hyper-multiple have approximately 2.4 million parameters. These numbers do not include the number of parameters of the feature extraction methods CLIP and CLAP.

Overall, our model achieves significant improvements over the baselines, while it requires marginally fewer parameters. Furthermore, unlike AVCA [157] and Hyper-multiple [94], our method does not require positive and negative samples to calculate a triplet loss during training. This reduces memory requirements and allows for a larger batch size.

Qualitative results. We provide t-SNE visualisations (Fig. 5.3) of our learned output embeddings on the VGGSound-GZSL^{cls}, UCF-GZSL^{cls} and ActivityNet-GZSL^{cls} datasets. Two unseen classes and four seen classes were randomly selected from the test set. For the unseen classes, all samples were used for the visualization, for the seen classes, all samples from the test set were used. This results in a class imbalance in the plots, since some seen classes have only a few test samples.

It can be observed that while the input features are not clustered well for all datasets, the t-SNE plots for the model outputs shows well-separated clusters for all classes. In particular, the unseen classes are very well-separated. This shows that our method learns useful embeddings for both seen and unseen classes. Only on VGGSound-GZSL^{cls}, our model does not separate well the unseen class *wood thrush calling* and the seen class *barn swallow calling*. This might come from the fact, that both classes can be categorized as bird sounds. Finally, all the text embeddings are located inside the cluster of the class they belong to. This shows that our approach effectively learns to assign the audio-visual input features to the correct class.

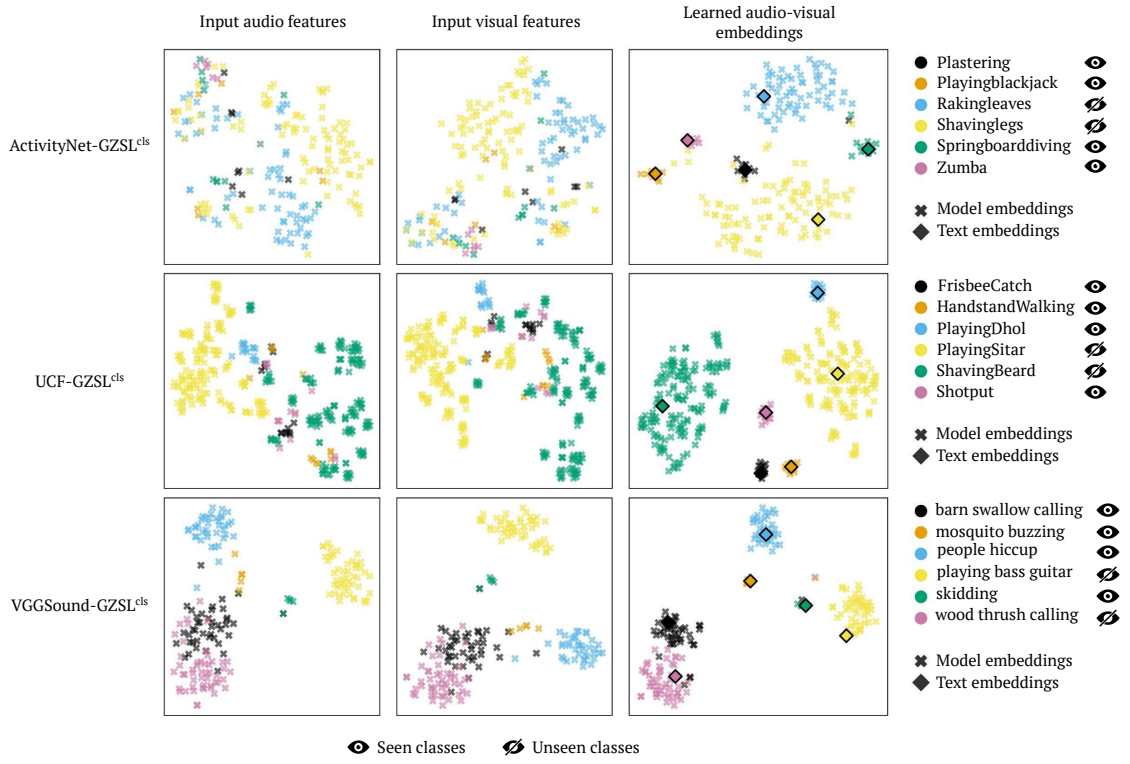


Figure 5.3: t-SNE visualizations for the audio input features (left), visual input features (center), and the learned output embeddings for our model (right) for the ActivityNet-GZSL^{cls} (top), UCF-GZSL^{cls} (center) and VGGSound-GZSL^{cls} (bottom) datasets for two unseen classes and four seen classes. The learned class text embeddings are visualized as diamonds.

Label Embedding	VGGSound-GZSL^{cls}				UCF-GZSL^{cls}				ActivityNet-GZSL^{cls}			
	acc_S	acc_U	HM	acc_{ZSL}	acc_S	acc_U	HM	acc_{ZSL}	acc_S	acc_U	HM	acc_{ZSL}
CLIP (w^v)	28.30	8.75	13.37	9.28	75.91	32.83	45.83	37.47	43.91	**21.04**	**28.45**	**23.18**
CLAP (w^a)	18.71	8.94	12.10	9.09	53.09	39.62	45.38	39.78	35.08	13.03	19.00	14.20
Both (Ours)	**29.68**	**11.12**	**16.18**	**11.53**	**77.14**	**43.91**	**55.97**	**46.96**	**45.98**	20.06	27.93	22.76

Table 5.2: Influence of using the two input label embeddings from CLIP and CLAP on the VGGSound-GZSL^{cls}, UCF-GZSL^{cls} and ActivityNet-GZSL^{cls} datasets.

Modality	VGGSound-GZSL^{cls}				UCF-GZSL^{cls}				ActivityNet-GZSL^{cls}			
	acc_S	acc_U	HM	acc_{ZSL}	acc_S	acc_U	HM	acc_{ZSL}	acc_S	acc_U	HM	acc_{ZSL}
Audio (a)	17.48	9.12	11.99	9.34	35.59	39.69	37.53	41.13	10.72	6.58	8.15	6.75
Visual (v)	15.39	7.00	9.62	7.16	53.65	43.13	47.82	43.98	38.59	**20.40**	26.69	22.58
Both (Ours)	**29.68**	**11.12**	**16.18**	**11.53**	**77.14**	**43.91**	**55.97**	**46.96**	**45.98**	20.06	**27.93**	**22.76**

Table 5.3: Influence of using only one modality or both modalities as inputs for our method on the VGGSound-GZSL^{cls}, UCF-GZSL^{cls} and ActivityNet-GZSL^{cls} datasets.

Loss	VGGSound-GZSL^{cls}				UCF-GZSL^{cls}				ActivityNet-GZSL^{cls}			
	acc_S	acc_U	HM	acc_{ZSL}	acc_S	acc_U	HM	acc_{ZSL}	acc_S	acc_U	HM	acc_{ZSL}
l_{reg}	5.41	9.44	6.87	10.03	22.76	25.49	24.05	28.22	5.01	6.69	5.73	7.11
$l_{reg} + l_{ce}$	**32.64**	**11.91**	**17.45**	**12.47**	76.79	40.30	52.86	43.16	38.93	**20.37**	26.75	22.73
$l_{reg} + l_{ce} + l_{rec}$	29.68	11.12	16.18	11.53	**77.14**	**43.91**	**55.97**	**46.96**	**45.98**	20.06	**27.93**	**22.76**

Table 5.4: Influence of using different components of the loss function on the (G)ZSL performance for the VGGSound-GZSL^{cls}, UCF-GZSL^{cls} and ActivityNet-GZSL^{cls} datasets.

5.4.3 Ablation studies

In this section, we conduct ablations studies on different model design choices. First, we ablate the choice of using two different textual label embeddings. Then, we study the benefit of using multi-modal inputs (audio and visual). Thirdly, we analyze the influence of the different loss components.

Class label embeddings. Our proposed framework uses two class label embeddings as inputs. Here, we compare this strategy to the usage of only a single class label embedding. For this, we train our model only using CLIP text embeddings w^v, or only using CLAP text embeddings w^a. However, we use both audio and visual input features for this experiment. The results are presented in Tab. 5.2.

Using either only w^v or only w^a as input class label embedding leads to very similar results on VGGSound-GZSL^{cls} and UCF-GZSL^{cls}. However, our proposed method that uses both w^v and w^a, outperforms them significantly. On UCF-GZSL^{cls}, using both text embeddings obtains a HM 55.97% vs. 45.83% for w^v, while on VGGSound^{cls} our method obtains a HM of 16.18% vs. 13.37% for w^v. The same trends can be observed for the ZSL performance. On ActivityNet-GZSL^{cls}, using w^v leads to HM / acc_{ZSL} scores of 28.45% / 23.18%, while using both w^v and w^a performs slightly worse, achieving HM / acc_{ZSL} scores of 27.93% / 22.76%.

Finally, using both label emebeddings help significantly in terms of the acc_U score. For VGGSound-GZSL^{cls}, we boost performance from 8.94% for w^a to 11.12%, while on UCF-GZSL^{cls} we improve the performance from 39.62% for w^a to 43.91% when using both

label embeddings (Both). On ActivityNet-GZSL^{cls}, using both embeddings gives slightly lower numbers than using only w^v in terms of the acc_U scores. On the other hand, our method obtains the best acc_S results on all three datasets. Overall, jointly using both w^a and w^v provides a significant boost in performance across all the metrics and datasets.

Multi-modality. In Tab. 5.3, we present the impact of using multi-modal input data. To obtain results for using a single input modality, only the audio or visual input feature (a or v) along with the corresponding text embedding (w^a or w^v) is used.

On VGGSound-GZSL^{cls}, using only the audio modality achieves higher HM and acc_{ZSL} scores compared to using the visual modality with HM / acc_{ZSL} scores of 11.99% / 9.34% vs. 9.62% / 7.16% for the visual modality. This is likely due to VGGSound being curated specifically to include relevant audio information. In contrast, for UCF-GZSL^{cls} and ActivityNet-GZSL^{cls}, using only the visual modality achieves better results than the audio modality on its own. On ActivityNet-GZSL^{cls}, the audio modality results in HM / acc_{ZSL} scores of 8.15% / 6.75% while using visual inputs gives HM / acc_{ZSL} scores of 26.69% / 22.58%.

Across all datasets, the acc_S score is significantly improved when using both modalities compared to using only a or v. On UCF-GZSL^{cls}, our full model (Both) yields a acc_S performance of 77.14% vs. 53.65% for v and 35.59% for a. For the acc_U score, we slightly improve upon the v, and significantly improve over a. The same trend can be observed on VGGSound^{cls} where our model (Both) significantly outperforms both a and v in acc_U and acc_S. On ActivityNet^{cls}, our full model is significantly stronger in terms of the acc_S score, while it is slightly outperformed in terms of acc_U when using only the visual modality v.

Overall, using both modalities as inputs is sound and leads to the best performance. These results highlight the fact that our model effectively exploits cross-modal relationships through the fusion of audio and visual modalities by using linear layers.

Training objective. We present results for using different loss functions in Tab. 5.4. Only using the regression loss l_{reg} yields the poorest performance on all three datasets, with HM scores of 6.87% / 24.05% / 5.73% for VGGSound-GZSL^{cls} / UCF-GZSL^{cls} / ActivityNet-GZSL^{cls}. Using the cross-entropy loss l_{ce} in addition to the regression loss drastically improves the performance with HM scores of 17.45% / 52.86% / 26.75% on VGGSound-GZSL^{cls} / UCF-GZSL^{cls} / ActivityNet-GZSL^{cls}. Finally, adding the reconstruction loss, i.e. when using the full loss function $l_{reg} + l_{ce} + l_{rec}$, we achieve the best overall GZSL results. While the impact of the reconstruction loss is smaller compared to the other two components, it still bring gains in performance. We observe a similar pattern for acc_{ZSL}.

Furthermore, on VGGSound-GZSL^{cls}, acc_S heavily benefits from adding the cross-entropy loss function. For all three datasets, one can observe at least a three-fold improvement. Moreover, we see improvements in the acc_U, where l_{ce} brings significant improvements. On UCF-GZSL^{cls}, our proposed loss function performs best in terms of both acc_S and acc_U. On ActivityNet-GZSL^{cls}, the complete loss function achieves

the best acc_S, while the $l_{reg} + l_{ce}$ loss function gives the best acc_U scores. Finally, on VGGSound-GZSL^{cls}, $l_{reg} + l_{ce}$ obtains a slightly better acc_S and acc_U score than our full loss. Overall, this shows that the full training objective provides the best results across all evaluation metrics.

5.5 Limitations

Our proposed method sets the new state of the art for audio-visual ZSL on three benchmark datasets when using CLIP and CLAP features. However, since the dataset used to train CLIP is not publicly available, we cannot guarantee that no unssen classes were used. Similarly, we did not attempt to remove unseen classes from the *WavCaps* dataset used to train CLAP. However, [151] shows that information leakage from CLIP pre-training to image ZSL is not very significant. Incorporating CLIP encoders into the model architecture is already an established practice in current research in zero-shot / open-vocabulary semantic segmentation [61, 130, 143, 276, 295]. As CLIP and CLAP were not specifically trained for the task of audio-visual GZSL, our problem setting requires significant transfer of knowledge to the new task.

5.6 Conclusion

In this paper, we explored the usage of pre-trained large multi-modal models for audio-visual generalized zero-shot learning. Our proposed framework ingests features extracted from the CLIP [197] and CLAP [153] models. One of the advantages of both of the feature extraction methods is that they are also able to produce textual input embeddings for the class labels. We proposed a simple model that consists of feed-forward neural networks and is trained with a composite loss function. When utilizing input features and both label embeddings obtained from CLIP and CLAP, our method achieves state-of-the-art results on the VGGSound-GZSL^{cls}, UCF-GZSL^{cls}, and ActivityNet-GZSL^{cls} datasets.

6

Video-adverb retrieval with compositional adverb-action embeddings.

Retrieving adverbs that describe an action in a video poses a crucial step towards fine-grained video understanding. We propose a framework for video-to-adverb retrieval (and vice versa) that aligns video embeddings with their matching compositional adverb-action text embedding in a joint embedding space. The compositional adverb-action text embedding is learned using a residual gating mechanism, along with a novel training objective consisting of triplet losses and a regression target. Our method achieves state-of-the-art performance on five recent benchmarks for video-adverb retrieval. Furthermore, we introduce dataset splits to benchmark video-adverb retrieval for unseen adverb-action compositions on subsets of the MSR-VTT Adverbs and ActivityNet Adverbs datasets. Our proposed framework outperforms all prior works for the generalisation task of retrieving adverbs from videos for unseen adverb-action compositions. Code and dataset splits are available at https://hummelth.github.io/ReGaDa/.

6.1 Introduction

Fine-grained video understanding is concerned with the detailed analysis of video content beyond action recognition. This is relevant for improving and potentially accelerating video search and retrieval. While there has been significant progress in action retrieval and recognition in videos [14, 158, 188, 231], the fine-grained understanding of actions remains challenging. In particular, it can be useful to perceive how an action is performed in order to better understand the action itself [64, 65, 165]. For instance, in addition to recognising the action *cutting*, it is useful to understand details about the execution of an action, e.g. *cutting slowly*. Specifically, we consider the bidirectional video-adverb retrieval task where we retrieve adverbs that match an action in a video and vice versa.

For bidirectional video-adverb retrieval, adverbs and action words can be combined in a compositional manner. The same adverb can describe multiple actions, such as *cutting*

slowly or *dancing slowly*. The compositional nature of the adverb-action pairings can also be exploited when learning adverb-action representations. Our proposed REGADA framework for video-adverb retrieval uses a **re**sidual **g**ating mechanism to compose **ad**verb-**a**ction (REGADA) representations for retrieval.

At its core, our framework learns to align adverb representations and video representations in a shared embedding space using a novel training objective which consists of a direct regression loss between the adverb and video representations and triplet losses. To obtain the adverb representation, the adverb and action are jointly embedded using a residual gating mechanism, which we adapted to the video-adverb retrieval task from [244]. It models the composition as a transformation of the adverb embedding based on the action by using a gate and a residual mechanism. The gate facilitates the preservation of meaningful information from the adverb embeddings based on the adverb-action composition. Our final composition is learned as a residual combination on top of the gated adverb embeddings. This allows our composed embeddings to be in the same "feature space" as the original adverb embeddings. Similar to previous works for this task, our model assumes knowledge of the ground-truth action class to perform video-adverb retrieval.

The compositional adverb-action embeddings and our proposed training objective prove beneficial for the retrieval performance, specifically for the retrieval of unseen adverb-action compositions. REGADA obtains state-of-the-art results on the five video-adverb retrieval benchmarks HowTo100M Adverbs [64, 160], VATEX Adverbs [65, 252], ActivityNet Adverbs [65, 90], MSR-VTT Adverbs [65, 275], and Adverbs in Recipes [160, 165]. Furthermore, we propose two additional splits for benchmarking the retrieval of unseen adverb-action compositions on the ActivityNet Adverbs and MSR-VTT Adverbs datasets.

To summarise, we make the following contributions: 1) Our proposed method for video-adverb retrieval uses a text encoder based on a gated residual mechanism and a novel training objective. 2) We evaluate REGADA on the challenging unseen video-adverb retrieval task and introduce new benchmark splits, compliant with zero-shot learning principles, for the retrieval of unseen adverb-action compositions based on the ActivityNet Adverbs and MSR-VTT Adverbs datasets. 3) Our framework outperforms prior work for both the seen and the unseen adverb-action composition retrieval tasks.

6.2 Related work

Fine-grained action understanding in video retrieval. Early works for video understanding extended retrieval approaches for images to videos, by temporally aggregating frames in a video [62, 180, 236, 277]. With the availability of large video-text datasets [17, 25, 115, 160, 179, 252, 275, 294], different methods focused on sentence disambiguation [46, 258], self-supervision [13, 208, 297], weakly supervised learning [158, 160, 188], multiple

embedding experts [74, 140, 159], or the use of large pre-trained models [121, 144, 186, 261]. Video-action retrieval specifically aims at retrieving videos based on an action, e.g. using a verb to describe the same [87, 257]. Moreover, [48, 78, 258, 277, 303] use nouns in addition to verbs for video-text retrieval. In a more general setting, [167] recently proposed to use a large language model to generate modified captions to improve verb understanding in video-language models. Different to these methods, we focus on adverbs in the video-adverb retrieval task.

Video-adverb retrieval. The video-adverb retrieval task was introduced by [64] along with the HowTo100M Adverbs dataset. [64] learns a shared representation between videos and adverbs, modelling adverb information as learned linear transformations on action class label word embeddings, similar to [169] for object attributes. Unlike [64], we choose to utilise semantic information from adverb embeddings in addition to action embeddings for modelling adverb-action compositions. [65] extends [64] to the low-data regime with pseudo-labelling. The recently proposed [165] tackles the task either as a classification or regression problem. Its video encoder builds on [64] with an additional projection following the attention while keeping the text representations frozen. The classification variant is trained with a cross-entropy loss for adverb classification, while the regression variant uses a regression target describing the change an adverb induced in an action embedding. Different from [165], we aim at learning the adverb-action representations and the video representations in a shared embedding space. Formulating the task as an alignment problem in a shared embedding space combined with compositional adverb-action representations significantly boosts the performance for video-adverb retrieval.

Learning with object attributes. Approaches for learning object-attribute pairs from images can be broadly categorized into classification [128, 150, 164, 168, 169] and retrieval approaches [33, 42, 99, 173, 244, 251, 254]. Our adverb-action compositions are most closely related to [244], which proposed a residual gating mechanism for learning compositional image-text embeddings. This mechanism proved particularly useful for retrieving images using both an image and a text query, the text describing a desired modification of the query image. We adapt a similar residual gating mechanism for learning compositional adverb-action embeddings by aligning the composition with action-focused video embeddings.

6.3 REGADA framework for video-adverb retrieval

In this section, we provide details about our proposed REGADA framework for video-adverb retrieval which is visualised in Fig. 6.1. We first describe the video-adverb retrieval task, and then provide details about our framework. Finally, we detail our training objective and the inference procedure for retrieval.

Task setting and dataset. The adverb-to-video retrieval task aims at retrieving matching videos from a pool of videos for a given adverb. Similarly, for the video-to-adverb retrieval task, given a video, the aim is to retrieve the adverb that best describes the action depicted

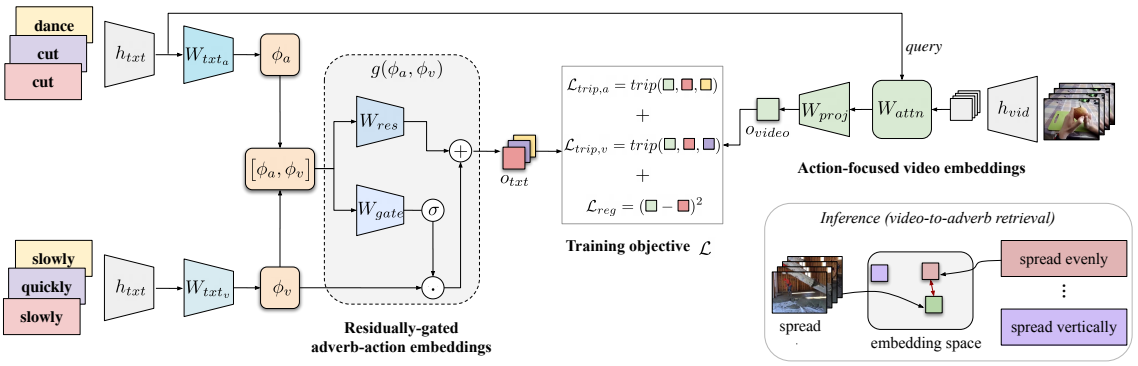


Figure 6.1: **Overview of our REGADA framework for video-adverb retrieval.** Our framework composes adverb-action embeddings with a gated residual between the adverbs ϕ_v and the concatenated action and adverb embeddings $[\phi_a, \phi_v]$. The training objective $\mathcal{L}$ aligns the learned text and video representations in a joint embedding space. For test time inference, outputs are obtained based on similarity in the embedding space.

in the video from a pool of pre-set adverbs. We denote a dataset with N samples, A action classes and V adverb classes by $\mathcal{D} = \{X_{[i]}, y_{[i]}\}_{i=1}^{N}$, consisting of video data $X_{[i]}$, and ground-truth action and adverb labels $y_{[i]} = \{a_{[i]}, v_{[i]}\}$ with one-hot encodings for the action $a_{[i]} \in \mathbb{R}^A$ and adverb $v_{[i]} \in \mathbb{R}^V$. We define the sets of possible actions and adverbs as $\mathcal{A}$ and $\mathcal{V}$. The set of all possible adverb-action combinations is $\mathcal{C} = \mathcal{V} \times \mathcal{A}$.

Our REGADA framework learns to align video and adverb-action representations in a joint embedding space. It generates compositional textual representations for adverb-action pairs using a text encoder. Additionally, the visual information is processed in a video encoder to obtain visual representations that contain information about the adverb associated with a given action. In the following, we describe how we obtain class label embeddings for the actions and adverbs, and how the video and text encoders process the video features and class label embeddings.

Residually-gated adverb-action embeddings. We obtain word embeddings for the action $a \in \mathcal{A}$ and for the adverb $v \in \mathcal{V}$ from a pre-trained language encoder h_{txt}, giving $\theta_v = h_{txt}(v)$, and $\theta_a = h_{txt}(a)$ with $\theta_a, \theta_v \in \mathbb{R}^{d_\theta}$. We then apply two linear maps $W_{txt_a}, W_{txt_v} : \mathbb{R}^{d_\theta} \rightarrow \mathbb{R}^{d_{dim}}$, such that $\phi_a = W_{txt_a}(\theta_a)$ and $\phi_v = W_{txt_v}(\theta_v)$. The action and adverb embeddings are then further processed jointly in our text encoder. Additionally, the action word embedding θ_a serves as a query vector in the video encoder's attention for generating an action-focused video embedding.

Our text encoder uses a residual gating mechanism which is based on [244]. Given ϕ_a and ϕ_{v_j} as inputs, the output of the text encoder is defined as:

$$o_{txt_j} = g(\phi_a, \phi_{v_j}) = \omega_g * \sigma(W_{gate}(\phi_a, \phi_{v_j})) \odot \phi_{v_j} + \omega_r * W^{res}(\phi_a, \phi_{v_j}), \tag{6.1}$$

where $j \in \{1, \cdots, V\}$, ω_g, ω_r are learnable scalar weights for balancing the gating mechanism and the residual, $\odot$ is an element-wise product, and σ the sigmoid function. W_{res} and W_{gate} are modelled using MLPs with N_r and N_g layers respectively. For those, the

input consisting of adverb and action embeddings, is first passed through a concatenation operator and batch normalisation [98] is applied. The subsequent layers consist of a linear map followed by dropout [220] with probability $drop_g$ and a Leaky ReLU [273]. The final layer is a linear projection to $\mathbb{R}^{d_{dim}}$.

We tackle video-adverb retrieval by aligning text and videos in a learned shared embedding space. Our residual gating mechanism models the composition as a transformation of the adverb embedding based on the action. The gating mechanism thereby allows to retain information from adverbs when actions do not provide useful semantic information.

Action-focused video embeddings. A pre-trained video classification network h_{vid} is used to extract a sequence of visual features $\boldsymbol{x}_{[i]} = \{x_1, ..., x_t, ..., x_T\}_i$, where $\boldsymbol{x}_{[i]} = h_{vid}(\mathcal{X}_{[i]})$ and $x_t \in \mathbb{R}^{d_x}$. We use T to denote the number of temporal segments in a video clip.

Given a sequence of video features $\boldsymbol{x}_{[i]}$ and its associated action word embedding $\theta_{a_{[i]}}$ (for easier readability, we omit the subscripts $_{[i]}$), we obtain action-focused video embeddings using a similar mechanism as the one proposed in [64]. The video embeddings are obtained using weak action-level ground-truth in the multi-head attention mechanism [241]. The action word embedding θ_a serves as the query in the attention to focus on parts of the video that are relevant to the given action, and ignore the temporal segments that may be relevant to other actions.

For the multi-head attention, we map the video features $\{x_t\}_{t\in[1,T]}$ to keys and values using linear maps $W_k : \mathbb{R}^{d_x} \rightarrow \mathbb{R}^{d_{head_x} H_x}$, $W_v : \mathbb{R}^{d_x} \rightarrow \mathbb{R}^{d_{head_x} H_x}$ with H_x heads and a dimension of d_{head_x} per head. We also map the action word embeddings θ_a to queries with $W_q : \mathbb{R}^{d_\theta} \rightarrow \mathbb{R}^{d_{head_x} H_x}$. For each attention head j, we have

$$p^j_{attn} = g^{DL}_{attn}\left(softmax\left(\frac{W^j_q(\theta_a)^T W^j_k(\boldsymbol{x})}{\sqrt{d_{head_x}}}\right)\right) W^j_v(\boldsymbol{x}), \tag{6.2}$$

where g^{DL}_{attn} denotes dropout with probability $drop_{attn}$.

We apply a linear mapping $W_{attn} : \mathbb{R}^{d_{head_x} H_x} \rightarrow \mathbb{R}^{d_{dim}}$ to aggregate the per-head attention giving the output video embedding $o_{attn} = W_{attn}([p^1_{attn}, \cdots, p^H_{attn}])$. The final output is obtained with an MLP, $W_{proj} : \mathbb{R}^{d_{dim}} \rightarrow \mathbb{R}^{d_{dim}}$,

$$o_{video} = W_{proj}(o_{attn}), \tag{6.3}$$

where each of the N_{proj} layers of W_{proj} consists of a linear layer $W^l_{proj} : \mathbb{R}^{d_{dim}} \rightarrow \mathbb{R}^{d_{dim}}$, layer normalisation [23] g^{LN}_{proj}, ReLU [172] g^{ReLU}_{proj}, and dropout g^{DL}_{proj} with probability $drop_{proj}$.

Training objectives. Our REGADA framework is trained with triplet losses based on [64] and with a direct regression loss between the video and text embeddings. We consider the triplet loss function $trip(a, p, n) = max(0, \|a - p\|_2 - \|a - n\|_2 + \mu)$, with the anchor embedding a, the embeddings for the positive and negative samples p and n, and the margin μ. The **action triplet loss** encourages the alignment of the video representation o_{video} and text embeddings with the matching action as opposed to a sampled negative action $\phi_{\bar{a}}$. For this, we use the video embedding o_{video} as the anchor, the text embedding

with ground truth action ϕ_a and adverb ϕ_v as the positive sample, and the text embedding of the same adverb but different action $\phi_{\bar{a}_i}$ as a negative:

$$\mathcal{L}_{trip,a} = \frac{1}{n}\sum_{i=1}^{n} trip(o_{video_i}, g(\phi_{a_i}, \phi_{v_i}), g(\phi_{\bar{a}_i}, \phi_{v_i})) \quad \text{for } \phi_{\bar{a}_i} \neq \phi_{a_i}. \tag{6.4}$$

We use an **adverb triplet loss** to push text embeddings containing the adverb antonym $\phi_{\bar{v}}$ away from the ground-truth text embedding:

$$\mathcal{L}_{trip,v} = \frac{1}{n}\sum_{i=1}^{n} trip(o_{video_i}, g(\phi_{a_i}, \phi_{v_i}), g(\phi_{a_i}, \phi_{\bar{v}_i})). \tag{6.5}$$

By restricting the negative samples for adverbs to their antonyms, the loss does not punish potential ambiguities of actions in videos (e.g. a drawer being opened slowly can at the same time be opened partially but not quickly). Our **regression loss** directly minimises the distance between the output video and text embeddings:

$$\mathcal{L}_{reg} = \frac{1}{n}\sum_{i=1}^{n}(o_{video_i} - g(\phi_{a_i}, \phi_{v_i}))^2. \tag{6.6}$$

The final loss is computed as the weighted sum of the above losses according to

$$\mathcal{L} = \lambda_a * \mathcal{L}_{trip,a} + \lambda_v * \mathcal{L}_{trip,v} + \lambda_{reg} * \mathcal{L}_{reg}, \tag{6.7}$$

with hyperparameters $\lambda_a, \lambda_v, \lambda_{reg} \in \mathbb{R}$.

Retrieving adverbs and videos (inference). Similar to [64], we evaluate our method on adverb-to-video and video-to-adverb retrieval given the ground-truth action a. For video-to-adverb retrieval, given a video $\boldsymbol{x}$ and action query a, we embed the video to obtain o_{video}, and we obtain embeddings for j adverb-action combinations o_{txt_j} for $j \in \{1, \cdots, V\}$. Using the cosine similarity metric we rank all the text embeddings o_{txt_j} by their similarity to the query video embedding o_{video} and we consider the highest-ranked pair as the retrieved adverb.

For adverb-to-video retrieval, given an adverb v and action a that are embedded to o_{txt}, we define the set of test videos containing action a as Γ. We rank all video embeddings o_{video_j} for videos in Γ using the similarity computed between each o_{video_j} and o_{txt} and select the video which is closest to o_{txt}.

6.4 Video-adverb retrieval benchmarks

In this section, we provide details about the datasets used in our experiments. In particular, we use five datasets for video-adverb retrieval. Furthermore, we propose two new dataset splits for the task of retrieving adverbs from videos for unseen adverb-action compositions.

Video-adverb retrieval datasets. HowTo100M Adverbs [64] consists of 5,824 video clips with annotations for 6 adverbs and 72 actions. In the following, we refer to HowTo100M

Dataset	# tr (s)	# t (s)	# tr (p)	# t (p)
VATEX	6603	3293	319	316
MSR-VTT	987	454	225	225
ActivityNet	1490	848	635	543

Table 6.1: Statistics of the proposed dataset splits for the retrieval of unseen adverb-action compositions on the MSR-VTT and ActivityNet datasets. (tr: train, t: test, s: video samples, p: adverb-action pairs)

Adverbs as **HowTo100M**. The recently proposed **Adverbs in Recipes** dataset has 10 adverbs, 48 actions and 7,003 videos. VATEX Adverbs [65] dataset has, with 34 adverbs and 135 actions, the largest variety of annotated adverbs and actions, consisting of 14,617 videos. We refer to VATEX Adverbs as **VATEX**. ActivityNet Adverbs [65] consists of 3,099 videos with 20 adverbs and 114 actions. We refer to it as **ActivityNet**. MSR-VTT Adverbs [65] is made up of 1,824 videos with 18 adverbs and 106 actions. In the following, we call this dataset **MSR-VTT**.

Unseen adverb-action compositions splits. We strive to explore the ability to recognise adverbs for novel adverb-action combinations. [65] proposed a dataset split for unseen compositions at test time for the VATEX dataset. Using the available videos in VATEX from [165], we replicate this split for the S3D video and text features used in this work, by omitting unavailable videos. We additionally propose new splits for unseen compositions on the ActivityNet and MSR-VTT datasets. We exclude HowTo100M Adverbs and Adverbs in Recipes, as both are subsets of HowTo100M which was used for pre-training the text and S3D video model. Hence, this would not comply with zero-shot learning principles.

To create splits for ActivityNet and MSR-VTT, we follow the protocol in [65]: We first split the set of possible adverb-action compositions into two non-overlapping sets, so that all adverbs and all actions are present in both sets, but individual compositions are only contained in one of the sets. We additionally constrain the compositions for each set so that for a given adverb-action composition, its antonym-action composition is assigned to the same set. We assign the videos from one of the sets to the training set and split the videos of the other half into two different sets, assigning half of the instances in each composition to the test set and the other to an unlabelled set (which is used to train [65] with pseudo-labelling). Tab. 6.1 shows details about the replicated split for VATEX, and for our proposed splits based on ActivityNet and MSR-VTT (full details are provided in the supplementary material).

6.5 Experiments

In this section, we provide details about the baselines, implementation details, and evaluation metrics used in this work. Video-adverb retrieval results on five benchmarks

are presented in Sec. 6.5.1, and we provide model ablation studies in Sec. 6.5.2. In Sec. 6.5.4, we investigate the transfer to unseen adverb-action compositions during inference.

Baselines. We report results for the **Prior** and **S3D pre-trained** baselines from [165]. **Prior** does not require any training but it uses the data distribution and adverb frequency for scoring. **S3D pre-trained** is also training-free and uses the similarity between frozen video and text representations from the S3D backbone jointly trained on video and text. **TIRG [244]** employs a similar residual gating mechanism as ReGAdA for image-text retrieval. To adapt it to the video domain, we use the same video encoder as our method. Different from ReGAdA, it models the composition as a transformation of the action embedding and uses a classification-based training objective. We also compare our framework to **Action Modifier** [64] and to the recently proposed **AC** frameworks [165]. AC tackles the task either as a classification (AC_{CLS}) or regression (AC_{REG}) problem.

Implementation details. We use the video and text features provided by [165] which were extracted using a frozen S3D model that was jointly pre-trained on video-text pairs from HowTo100M [160]. Here, $d_x = 1024$, T is the length of the video in seconds, and $d_\theta = 512$. ReGAdA uses an internal embedding dimension $d_{dim} = 400$. We use $N_g = 2$, except for HowTo100M and Adverbs in Recipes where $N_g = 3$ and $N_g = 4$ respectively. Additionally, we set $N_r = 2$ except for Adverbs in Recipes where we use $N_r = 3$. The dropout probability in the residual gating mechanism is $drop_g = 0.6$ for all datasets but Adverbs in Recipes and HowTo100M where we use $drop_g = 0.7$. The loss hyperparameters are chosen as $\lambda_a = 1$ for all datasets and $\lambda_v = 2.0$ for all datasets, except for $\lambda_v = 1.5$ on Adverbs in Recipes. Furthermore, we use a $\lambda_{reg} = 1.0$ for all dataset except for HowTo100M where $\lambda_{reg} = 1.5$. We train with a batch size of 512, and employ the Adam [110] optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay 10^{-5}. Our method is trained for 2000 epochs using a learning rate of 10^{-5} for all datasets with the exception of HowTo100M where we use $3 * 10^{-5}$. We follow [165], and train all baselines for 1000 epochs using a learning rate of 10^{-4}. We conduct all experiments on a single Nvidia 2080-Ti GPU.

Evaluation metrics. We follow [165], and report mean Average Precision (mAP) scores for adverb-to-video-retrieval, in particular **mAP M** ("adverb-to-video (all)" in [64]) and **mAP W**. mAP M is computed by ranking videos that contain the same ground-truth action according to their similarity to the adverb-action text embedding. For mAP W, the class scores are reweighed according to their support size in the test set. For video-to-adverb retrieval, we report binary antonym accuracy **Acc-A**. This is equivalent to ranking adverb-action embeddings according to their similarity to the embedded video and calculating the mAP by restricting the set of adverbs to the target adverb and its antonym ("video-to-adverb (antonym)" in [64]). Similar to [165], we report the best metrics independently. This means that models corresponding to each result may originate from different epochs.

	HowTo100M [64]			Adverbs in Recipes [165]			ActivityNet [65]			MSR-VTT [65]			VATEX [65]		
	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A
Priors	0.446	0.354	0.786	0.491	0.263	0.854	0.217	0.159	0.745	0.308	0.152	0.723	0.216	0.086	0.752
S3D pre-tr.	0.339	0.238	0.560	0.389	0.173	0.735	0.118	0.070	0.560	0.194	0.075	0.603	0.122	0.038	0.586
TIRG [244]	0.441	0.476	0.721	0.485	0.228	0.835	0.186	0.111	0.709	0.297	0.113	0.700	0.195	0.065	0.735
Act. M. [64]	0.406	0.372	0.796	0.509	0.251	0.857	0.184	0.125	0.753	0.233	0.127	0.731	0.139	0.059	0.751
AC_{CLS}† [165]	0.562	0.420	0.786	0.606	0.289	0.841	0.130	0.096	0.741	0.305	0.131	0.751	0.283	0.108	0.754
AC_{REG}† [165]	0.555	0.423	0.799	0.613	0.244	0.847	0.119	0.079	0.714	0.282	0.114	0.774	0.261	0.086	0.755
REGADA	**0.567**	**0.528**	**0.817**	**0.704**	**0.418**	**0.874**	**0.239**	**0.175**	**0.771**	**0.378**	**0.228**	**0.786**	**0.290**	**0.113**	**0.817**

Table 6.2: Results for adverb-to-video (mAP W/M) and video-to-adverb retrieval (Acc-A). Higher is better for all metrics. † refers to updated results provided by the authors.

6.5.1 Comparison with the state of the art

In Tab. 6.2, we present adverb-to-video retrieval and video-to-adverb retrieval results with our REGADA framework on five benchmark datasets. It can be observed that REGADA outperforms the baselines across all datasets. In particular, we see more significant improvements of our framework over the prior methods for the adverb-to-video retrieval metrics (mAP W and mAP M) compared to video-to-adverb retrieval (Acc-A). For instance, on the HowTo100M dataset REGADA outperforms AC_{CLS} for adverb-to-video retrieval with mAP M and mAP W scores of 0.528 and 0.567 compared to 0.420 and 0.562. For the video-to-adverb retrieval measure Acc-A, REGADA obtains a score of 0.817 compared to 0.786 with AC_{REG}.

The most recent and strongest competitor [165] optimises its systems using two different losses. The best results obtained from these two models are reported for each dataset and metric, showing no clear pattern as to which model variant is stronger. Our REGADA framework consistently outperforms both model variants [165] on all metrics and datasets. We hypothesise that our framework's strong performance can be attributed to its compositional embeddings which is a key element of REGADA.

6.5.2 Model ablations

This section analyses the impact of using different input text information, losses, and components in the text encoder on the overall video-adverb retrieval performance of REGADA.

Input to the text encoder. The gating mechanism in REGADA represents the composition as a residual on top of the adverb and allows the adverb information to be retained, leveraging the action as auxiliary information. We refer to the adverb as the *main* and the action as the *auxiliary* modality in REGADA. We investigate if a compositional adverb-action word embedding ϕ_{comp}, which directly embeds an adverb-action label pair (e.g. "*cut quickly*") with h_{text}, can be used as the main modality instead. Tab. 6.3 shows the impact of using different main and auxiliary modalities. REGADA obtains scores of 0.290 and 0.113 for mAP W and mAP M on VATEX compared to 0.245 and 0.078 when using ϕ_a as main modality and ϕ_v as auxiliary. This confirms that capturing information about the adverb is crucial for solving the task. Acc-A is less affected by the type of input

Text Input		HowTo100M [64]			Adverbs in Recipes [165]			ActivityNet [65]			MSR-VTT [65]			VATEX [65]		
main	*auxiliary*	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A
ϕ_a	ϕ_v	0.485	0.390	0.824	0.436	0.221	0.872	0.225	0.147	0.763	0.336	0.144	0.780	0.245	0.078	0.807
ϕ_{comp}	ϕ_v	0.498	0.454	0.827	0.518	0.322	0.877	0.220	0.150	0.751	0.350	0.144	0.771	0.255	0.084	0.808
ϕ_{comp}	ϕ_a	0.503	0.467	**0.830**	0.524	0.365	**0.881**	0.222	0.147	0.758	0.348	0.146	0.763	0.255	0.090	0.806
ϕ_v	ϕ_a	**0.567**	**0.528**	0.817	**0.704**	**0.418**	0.874	**0.239**	**0.175**	**0.771**	**0.378**	**0.228**	**0.786**	**0.290**	**0.113**	**0.817**

Table 6.3: Effect of using different types of input information for the text encoder in ReGAdA.

Loss			HowTo100M [64]			Adverbs in Recipes [165]			ActivityNet [65]			MSR-VTT [65]			VATEX [65]		
$\mathcal{L}_{trip,a}$	$\mathcal{L}_{trip,v}$	$\mathcal{L}_{reg}$	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A
✓	✗	✗	0.361	0.228	0.697	0.429	0.214	0.836	0.162	0.104	0.582	0.259	0.138	0.714	0.133	0.047	0.677
✗	✓	✗	0.340	0.236	0.740	0.430	0.213	0.846	0.128	0.079	0.664	0.260	0.127	0.737	0.166	0.062	0.743
✗	✗	✓	0.470	0.378	0.743	0.468	0.234	0.839	0.202	0.140	0.729	0.288	0.186	0.743	0.182	0.074	0.700
✓	✓	✗	0.367	0.246	0.755	0.468	0.239	0.851	0.157	0.098	0.674	0.273	0.116	0.737	0.174	0.062	0.756
✓	✓	✓	**0.567**	**0.528**	**0.817**	**0.704**	**0.418**	**0.874**	**0.239**	**0.175**	**0.771**	**0.378**	**0.228**	**0.786**	**0.290**	**0.113**	**0.817**

Table 6.4: Impact of using different losses to train ReGAdA. For losses that are not used, the corresponding scalar weight in $\mathcal{L}$ is set to zero.

Components			HowTo100M [64]			Adverbs in Recipes [165]			ActivityNet [65]			MSR-VTT [65]			VATEX [65]		
R	σ	SW	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A
✓	✓	✓	0.535	0.433	0.811	0.689	0.404	0.875	**0.256**	**0.190**	**0.771**	0.374	0.182	0.766	0.288	0.109	0.808
✓	✗	✗	0.512	0.496	0.811	0.501	0.269	0.862	0.234	0.171	0.770	0.360	0.194	0.780	0.260	0.098	0.804
✗	✓	✗	0.516	0.477	**0.817**	0.562	0.296	**0.877**	0.228	0.169	0.765	0.367	0.161	0.783	0.283	0.111	0.815
✓	✓	✗	**0.567**	**0.528**	**0.817**	**0.704**	**0.418**	0.874	0.239	0.175	**0.771**	**0.378**	**0.228**	**0.786**	**0.290**	**0.113**	**0.817**

Table 6.5: Impact of different components in the residually-gated text encoder. R: With residual branch W_{res}; σ: With sigmoid; SW: Sharing weights between W_{res} and W_{gate}.

information, ReGAdA obtains 0.817 compared to 0.806 when using ϕ_{comp} as main and ϕ_a as auxiliary modality. Overall, using ϕ_v as main and ϕ_a as auxiliary modality is most effective across datasets.

Losses. In Tab. 6.4, we show the impact of our three loss functions, $\mathcal{L}_{trip,a}$, $\mathcal{L}_{trip,v}$, and $\mathcal{L}_{reg}$. On VATEX, ReGAdA obtains a mAP W and mAP M of 0.290 and 0.113 compared to 0.182 and 0.074 when using only $\mathcal{L}_{reg}$. For Acc-A, ReGAdA obtains a score of 0.817 compared to 0.756 for $\mathcal{L}_{trip,a} + \mathcal{L}_{trip,v}$. The regression loss $\mathcal{L}_{reg}$ boosts the performance on all datasets significantly. Our novel loss combination gives the best video-adverb retrieval performance by better aligning adverb-action compositions and video representations. Previous work either only used triplet losses [64, 65] or used a fixed textual regression target [165].

Residual gating mechanism in the text encoder. Tab. 6.5 analyses the contributions of the components of the residual gating mechanism, such as the residual branch, the sigmoid, and weight sharing between the gated and residual branches. On VATEX, ReGAdA achieves the best results. Interestingly, sharing weights between the gated and residual branches yields only slightly weaker results, with a mAP-W score of 0.288 compared to 0.290 with ReGAdA. For mAP M and Acc-A, ReGAdA obtains 0.113 and 0.817 compared to 0.111 and 0.815 when not using the residual. While some configurations can achieve better results in selected metrics, ReGAdA yields consistent state-of-the-art results across all metrics, confirming our model design choices.

Figure 6.2: Example results for ReGaDa (Ours) on the VATEX dataset compared to those from AC_{REG}. The two left examples are success cases for our model. The third and fourth example show bidirectionally performed actions that are labelled with only one of the adverbs. The right-most example shows a wrongly labelled video. Full videos are available at: https://hummelth.github.io/ReGaDa

6.5.3 Qualitative Results

We show qualitative results for ReGaDa on the VATEX dataset in Fig. 6.2. In particular, success cases for ReGaDa which AC_{REG} retrieved a wrong adverb are shown below in the first and second columns. The third and fourth columns show videos with actions performed forwards/backwards, and upwards/downwards but labelled with only one of the adverbs. This makes both outputs plausible. The right-most column shows an example of a wrongly labelled video for which our model retrieves the correct adverb. This confirms ReGaDa's strong generalisation capabilities. In general, we observe that ReGaDa better captures directional movements or speed than AC_{REG}. It is also superior at disentangling the diverse visual effect of adverbs on different actions (e.g. crawl vs. bend backwards). This can potentially be attributed to the compositional nature of our learned adverb-action representations.

6.5.4 Generalisation to unseen adverb-action compositions

We additionally evaluate the ReGaDa framework on video-to-adverb retrieval for unseen adverb-action compositions, i.e. compositions that were not seen during training. We consider the existing VATEX benchmark and our proposed MSR-VTT and ActivityNet splits for this task (see Sec. 6.4). Following [65], we report binary antonym classification accuracy for video-to-adverb retrieval. We provide additional baseline results with the CLIP [197] model (details for this are provided in the supplementary material). In Tab. 6.6, we observe that ReGaDa significantly outperforms AC_{REG} on VATEX with an accuracy of 61.7 compared to 54.9. On ActivityNet, ReGaDa obtains a score of 58.4, outperforming [65] with a score of 57.0. This is impressive given that [65] was additionally trained on pseudo-labelled data. CLIP obtains an antonym accuracy of only 54.5 on VATEX, showing a limited fine-grained retrieval capability of CLIP. We provide a further analysis of exploiting different word embeddings for unseen compositions in the supplementary material. Overall, our model yields better results than any prior framework for both seen (c.f. Tab. 6.2) and unseen compositions.

Model	VATEX	ActivityNet	MSR-VTT
CLIP [197]	54.5	55.1	57.0
Act. Mod. [65]	53.8	57.0	56.0
AC_{CLS} [165]	54.3	55.1	53.7
AC_{REG} [165]	54.9	53.9	59.0
REGADA	**61.7**	**58.4**	**61.0**

Table 6.6: Retrieval of unseen adverb-action compositions on the VATEX, ActivityNet and MSR-VTT benchmarks. [65] uses pseudo-labelling.

6.6 Conclusion

In this work, we proposed a framework for video-adverb retrieval that uses a residual gating mechanism to generate compositional adverb-action representations from adverb and action word embeddings. Along with a novel training objective, our model achieves state-of-the-art results on five video-adverb retrieval benchmarks. Moreover, we introduce two additional dataset splits to benchmark the retrieval of unseen adverb-action compositions. Our proposed framework outperforms all prior works on this task, confirming that our text encoder results in better generalisation abilities.

7

Adapting Communicating MLLMs on the Fly in Referring Expression Tasks

Multimodal Large Language Models (MLLMs) exhibit varying comprehension levels in language and perception that complicate interacting with a diverse population of agents, similar to how miscommunication happens in humans, e.g., because intentions are not always known. In this work, we investigate whether MLLMs can adapt to the perceptual weaknesses of the communication partners in an online manner, i.e. change the way they describe their environment in a way that is understandable to their partner while communicating with them, via reinforcement learning. We experiment with two tasks: referring expression identification (REI) and referring expression segmentation (RES), where a speaker agent has to describe an object, and a listener has to identify it. To be successful, the speaker agent must discern the comprehension level of the listener and adapt accordingly, especially when the listener suffers from perceptual weaknesses such as color blindness or blurred vision. Unlike traditional offline alignment methods for LLMs, we fine-tune a Multimodal LLM (MLLM) online to adapt to other agents' conceptual understanding. Our experiments with four MLLMs on four datasets show that online adaptation is feasible in both REI and RES settings.

7.1 Introduction

Large Language Models (LLMs) and by extension Multimodal Large Language Models (MLLMs) have demonstrated remarkable capabilities across a variety of tasks [12, 16, 37, 193, 233]. When catering MLLMs with different architectures, e.g. vision backbones, language backbones, trained with different datasets etc, in our daily lives, we may notice the variability in their comprehension levels related to task-specific concepts, i.e. what resonates with some MLLMs might not be clear to others. Disparities may exist both in their natural language understanding, e.g., some might understand expert terminology while another might require descriptive explanations, and in the perceptual understanding of visual information, e.g., some might have disabilities such as blurred vision or color

blindness.

In this work, we focus on enabling MLLMs to adapt to perceptual misunderstandings of their communication partners, e.g., not perceiving colors correctly and therefore not responding to color attributes presented to them. Specifically, we fine-tune the MLLM online, i.e. on-the-fly, while it is interacting with another MLLM, based on its observed behavior. We model sequential interactions between pairs of agents during a vision-language referring expression tasks which is used as an environment for both adaptation and evaluation. Given one or two images, the speaker agent needs to describe the discriminating features of a target object, while the listener agent has to identify the correct object based on this description. To enhance overall task performance, the speaker has to learn which feature of the image allows the listener agent to discriminate the target object and adapt its communication based on the visual concepts understood by the listeners. We consider a referring expression identification (REI) task, where the listener has to identify one target image from a set of two images, and a referring expression segmentation (RES) task, where the listener has to segment the target object within a single image correctly. We present both settings in Fig. 7.1.

We employ several open-source MLLMs, namely LLaVA-7B, LLaVA-13B [135], Qwen [24], and PaliGemma [27] as the speaker and listener agents where the difference in MLLM capabilities and pre-training datasets simulate significant diversity. In addition, we introduce perceptual weaknesses to some listeners by providing them with blurred or grayscaled images to further increase listener variety. As the benchmark, we take inspiration from [55], but create a more realistic setting by modeling the interactions as free-form text, adding image transformations to simulate challenging adaptation scenarios, and scaling it to MLLMs. We evaluate the REI task on CLEVR [103], CUB [246], and ImageNet [57], while we use the RefCOCO [106] dataset to implement the RES task. We adapt the MLLMs on the fly using PPO [212], KTO [70], and NLPO [201] developed originally as preference learning methods for LLMs when fine-tuning the LoRA adapters [95]. Contrary to the typical use case of these algorithms for preference optimization [8, 181] where a carefully curated offline dataset of human preferences is collected, we test their efficacy during online interactions which is a more realistic and noisier setting.

Our contributions are as follows: 1) We introduce a flexible framework for evaluating four MLLMs and adapting them on the fly using four RL algorithms on natural-language-based communication tasks on four datasets to test their efficacy in online adaptation to a diverse set of communication partners. 2) We provide insights into the decision-making process of MLLMs finding that concepts related to color and shape are most important for performing well on these tasks. 3) Through extensive experimental results on two different communication tasks, four MLLMs, and four datasets, we show that adaptation is possible both the REI and RES task.

7.2 Related Work

A number of methods aim for parameter efficient adaption of large (language) models, which adapt a subset or an additional set of the parameters. LoRA [95] and its variants [129, 136, 137, 214, 262, 287] add a trainable residual low rank adaption for each matrix in the network, potentially quantizing it [58, 127, 281]. In contrast, sparse methods [18, 26] only adapt small subsets of the parameters. Adapter based methods [191] train adapter layers and yet another approach is to train a completely separate ladder side networks [154, 228]. As we aim to adapt large multimodal models online, we use LoRA [95] for adaptation.

For adapting an MLLM to obtain a desired functionality, such as the ability to adapt to a listener online, different RL methods [201, 216, 304] can be used. Proximal policy optimization (PPO) [212] is an on-policy actor critic algorithm, which is extended by NLPO [201]. It restricts the action space to a nucleus of most likely tokens. In contrast KTO [70] directly optimizes the LLM from binary preferences. On the other hand DPO [199] requires positive and negative pairs for the same context. All of the methods apart from DPO use a single reward per generation making them suitable for our task, thus, we compare their performance. Similar to our work [84, 134] perform (online) adaption based on model feedback in the context of generic model alignment, while we focus on personalization to individual conversational partners and their misunderstandings.

Personalizing generative language models has been studied for a long time, often viewed in the context of building an efficient conversational partner in dialogue systems [213, 218, 289]. In contrast, [147] reviews several theory of mind (TOM) based approaches to personalization, such as [229] which proposes a plug-and-play TOM based on an explicit simulator, that updates a copy of the model weights on the fly. Similarly, [200] internally models the behavior of the listener. In contrast, we only update a small amount of parameters using LoRA and do not need to simulate the listeners behavior. [248] adapts the speaker and listener differently, but studies the text-only task, whereas we consider a multi-modal image reference game. We follow an online approach, while [146, 292] personalizes chatbots by learning from large-scale user dialogue history.

Image identification tasks have been studied in visual dialogue settings in [11, 56, 176, 245]. Our work extends this, by incorporating impairments in the communication. [55] has studied conceptual image understanding through a reference game, but we extend their attribute constrained setting to free text generation.

7.3 Adapting the Speaker on the Fly in Referring Expression Tasks

We present a framework for referring expression communication tasks (Fig. 7.1) where a speaker agent describes images to a listener agent using visual concepts. The "speaker" is a single learner that participates in sequences of K episodes describing an image to a group of "listeners".

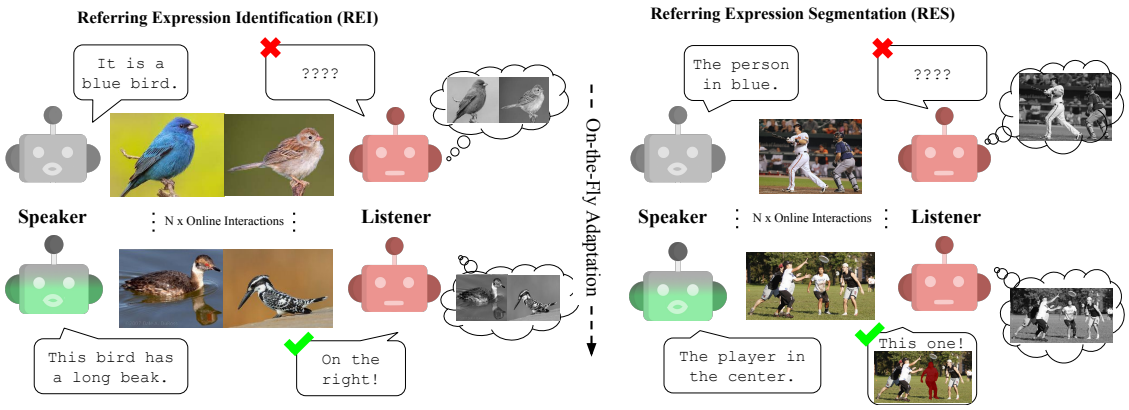


Figure 7.1: The speaker tries to identify a target object, but its pre-trained policy is not aware of misunderstandings of the listener agents, e.g., color blindness. Through interaction with the listener, the speaker learns on-the-fly to mention the shape instead of color because the listener is color-blind. The left interaction illustrates the REI task, while the right interaction shows the RES task.

Referring Expression Identification (REI) Task. In the REI task, each episode involves the speaker $\pi^{(s)}$ and listener $\pi^{(l)}$ being presented with a pair of images $[x_k^t, x_k^c]$. The speaker is assigned one image as the target x_k^t and the other as a confounding image x_k^c. The speaker then generates a description $m_k^{(s)}$ as a message to the listener for it to make its guess regarding the target's identity $m_k^{(l)}$, i.e. left or right image. The speaker will observe whether its description led to a correct or incorrect guess via a reward $r_k \in \{+1, -1\}$ communicated for every episode.

Referring Expression Segmentation (RES) Task. In the RES task, the speaker $\pi^{(s)}$ and listener $\pi^{(l)}$ are presented with a single image x_k in each episode. The speaker additionally receives the bounding box of a target object o_k^t for which the speaker generates a description $m_k^{(s)}$ with the intention to identify the object in the context of the image. Given the speaker's message, the listener generates a segmentation mask $m_k^{(l)}$ as a guess regarding the target object in the image. The intersection over union (IoU) metric between the predicted and the ground-truth segmentation masks serves as a reward of the episode for the speaker.

Based on this feedback from the reward alone, the speaker's goal is to change its policy $\pi^{(s)*}$, i.e., adapt its image description, to maximize the success rate of the listener agent to solve the referring expression task. To further increase the difficulty of each task, any listener may suffer from a perceptual weaknesses, i.e., color blindness or blurry vision, which is unknown to the speaker.

Since the listener operates as a black box from the perspective of the speaker, pinpointing the source of errors when they exhibit unexpected behavior can be challenging. When the listener makes an incorrect guess, identifying the source of the error becomes difficult, e.g., it could be a lack of comprehension in language, or the visual concepts used to describe the image.

When the listener fails to guess the correct object, the speaker should explore different descriptions to find a policy tailored for the listener. In this work, we examine, whether

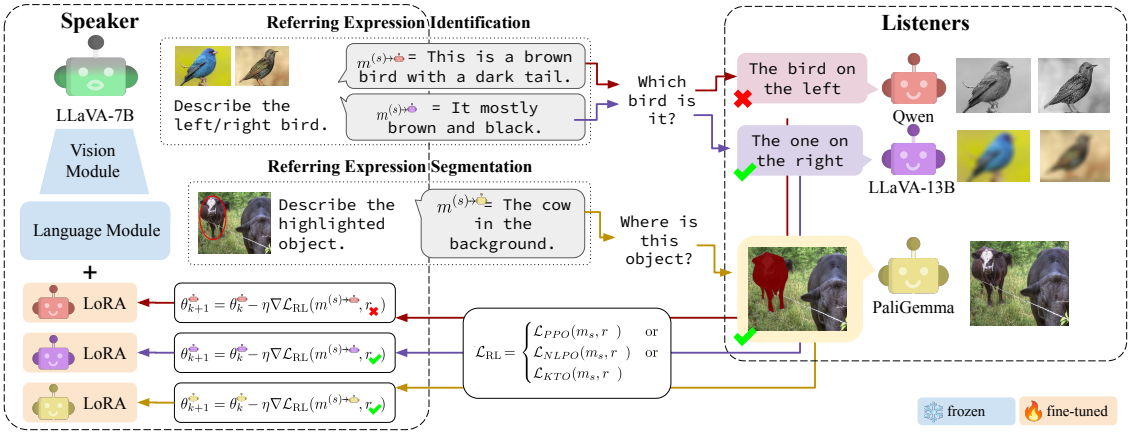


Figure 7.2: Speaker is asked to describe an object in the context of the REI or RES task. The description is passed to the Listeners which need to decide which object was described. Depending on the correctness of the decision, the Speaker receives a sparse reward and updates its LoRA weights to maximize the reward. For each type of Listener, we have a distinct set of LoRA weights.

LLM adaptation methods can successfully find policies that maximize task performance for a diverse set of listener agents solely from the reward signal in this multimodal, i.e. vision and language-based, framework.

7.3.1 Online MLLM Adaptation

To perform well on the referring expression tasks, the speaker agent needs to adapt to the listener in an online setting during ongoing interactions. After each episode the speaker can update its weights based on the reward provided by the listener's response. Through these rewards, the speaker increases the likelihood of generating descriptions that are adapted to the capabilities of the listener.

Reinforcement learning from human feedback (RLHF) [8, 53, 181, 221] is a popular technique to adapt LLMs to human preferences. Typically, a dataset of human preferences is collected, before a RLHF algorithm is applied either offline or through training a reward model to update the parameters of the LLM or MLLM for better human alignment. In this work, we explore how well RLHF algorithms extend to an online setting which is more challenging because the reward data is not carefully annotated and can be noisy, e.g., when the listener misunderstands the description, but still guesses correctly.

Proximal Policy Optimization (PPO) [212] is an on-policy actor-critic algorithm that treats language generation as Markov Decision Process (MDP) where at each state s_t in the sequence (current context), the next action a_t is chosen (token), until at the end of the sequence T a reward r is observed. As is typical in RL, the discounted expected reward of the policy is optimized $\mathbb{E}_\pi[\sum_{t=0}^{T} \gamma^t r(s_t, a_t)]$ with γ as the discount factor. PPO starts from the initially pre-trained MLLM $\pi_\theta = \pi_0$ and updates the policy using the following loss:

$$\mathcal{L}_{\text{PPO}}(\pi_{\theta_k}, \pi_{\theta_{k-1}}) = \mathbb{E}_{a_t, s_t \sim \pi_{\theta_k}} \left[\min \left(\phi^{\pi_{\theta_k}}_{\pi_{\theta_{k-1}}} A^{\pi_{\theta_{k-1}}}, \text{clip}(\phi^{\pi_{\theta_k}}_{\pi_{\theta_{k-1}}}, 1-\epsilon, 1+\epsilon) A^{\pi_{\theta_{k-1}}} \right) \right] \quad (7.1)$$

where $\phi^{\pi_{\theta_k}}_{\pi_{\theta_{k-1}}} = \frac{\pi_{\theta_k}(a_t|s_t)}{\pi_{\theta_{k-1}}(a_t|s_t)}$, ϵ is a hyperparameter and A^{π_θ} is the advantage function that estimates whether the current action is better than average.

As suggested by [259], a token-level penalty $\mathrm{KL}\left(\pi_q||\pi_p\right) = (\log \pi_p(a_t|s_t) - \log \pi_q(a_t|s_t))$ regularizes the reward function. This avoids large deviations from the pre-trained MLLM, i.e. the initial policy π_0. The updated reward is computed as:

$$\hat{r}(s_t, a_t) = r(s_t, a_t) - \beta \mathrm{KL}\left(\pi_\theta||\pi_0\right) \tag{7.2}$$

where the KL coefficient β is a hyperparameter.

Natural Language Policy Optimization (NLPO) [201] extends PPO by restricting the action-space with a reduced number of tokens. This is achieved by freezing a masked policy π_ψ every μ steps and sampling sentences during training from this masked policy. NLPO employs top-p sampling for π_ψ which limits the sampled tokens to the smallest subset of tokens with cumulative probability greater than the probability p. This additional constraints restricts the sampled sentences to be closer to the masked policy, a snapshot of a previous policy, preventing large deviations and divergence.

Kahneman-Tversky Optimization (KTO) [70] takes inspiration from prospect theory and proposes to directly optimize the LLM from binary preferences similar to DPO [199], instead of performing RLHF. In contrast to DPO, it does not require paired preference data. The loss function is defined as:

$$L^+_{\mathrm{KTO}}(\pi_\theta, \pi_0) = \mathbb{E}_{a_t, s_t \sim \pi_\theta}[\lambda^+(1 - \sigma(\beta(\log \phi^{\pi_\theta}_{\pi_0} - \mathbb{E}_{s' \sim \pi_\theta}\left[\mathrm{KL}(\pi_\theta \| \pi_0)\right])))] \quad \text{if } r = +1 \tag{7.3}$$

$$L^-_{\mathrm{KTO}}(\pi_\theta, \pi_0) = \mathbb{E}_{a_t, s_t \sim \pi_\theta}[\lambda^-(1 - \sigma(\beta(\mathbb{E}_{s' \sim \pi_\theta}\left[\mathrm{KL}(\pi_\theta \| \pi_0)\right] - \log \phi^{\pi_\theta}_{\pi_0})))] \quad \text{if } r = -1 \tag{7.4}$$

that depends on whether a generated sentence produced a +1 or -1 reward. $\lambda^{+/-}$ are hyperparameters for the two loss terms respectively. Since we do not have a static dataset, we sample sentences on-policy and shuffle the context, i.e. image input and prompt, within each batch for the KL term.

RL algorithms are known to be unstable [8, 53, 181] which is why KL terms have been introduced for fine-tuning LLMs. Nonetheless, a potential danger that can arise from this is that the policy of the speaker may diverge and start to generate unusual sentences which exploit the listener agent. These sentences may not describe the images correctly, or deviate from being grammatically correct, but enumerations of words instead. Careful selection of hyperparameters is generally important for success with any of these algorithms.

7.3.2 Efficient Adaptation of the Speaker Agent

Online adaptation of an MLLM does not only require a suitable optimization algorithm, but it should also be feasible in terms of update speed and flexibility as a common use-case may involve a speaker agent interacting with several listeners in parallel. As full-fine-tuning MLLMs is computationally expensive, we adapt these methods by using a parameter-efficient fine tuning method. Given the versatility of LoRA [95] for both the

visual domain and the text domain, and its simplicity, we employ it in our architecture. We add LoRA adapters on each linear layer in the LLM-module of the network. As a result, the total number of tuneable parameters are orders of magnitude smaller than the total number of parameters in the MLLM. One can initialize one set of LoRA adapters for each listeners and effortlessly swap out LoRA parameters when interacting with multiple listeners.

We employ LLaVA-7B as the speaker model for all experiments because it fits into the memory of a single GPU while training with LoRA adapters. Since the listener runs in inference mode, we also evaluate on LLaVA-13B, Qwen , and PaliGemma to increase listener diversity.

7.4 Experiments

We first introduce our experimental setting, i.e. our datasets, the agents, the training, and evaluation protocol. Then we present the weaknesses and strengths of current MLLMs when dealing with the visual-language referring expression tasks. Finally, we provide extensive experiments into adapting a speaker model to different listeners on four different datasets using three algorithms.

7.4.1 Experimental Setting

Datasets. We propose a framework for referring expression tasks on four datasets: CLEVR[103], CUB [246], ImageNet[57] for REI, and RefCOCO [106] for RES. CLEVR contains images with objects of varying attributes (size, color, material), requiring fine-grained reasoning to distinguish between different CLEVR scenes. CUB and ImageNet feature natural images with more conversationally relevant concepts. For REI on these datasets, we sample two images, randomly select one as the target, and ask the speaker to describe it in contrast to the other image. We shuffle their order when presenting the images to the listener to avoid trivial solutions, such as "the left image is the target image". Further, we ensure the images come from different classes for CUB and ImageNet. For RES, we employ RefCOCO which extends COCO [132] with human-annotated referring expressions and bounding box/segmentation mask annotations. This task requires contrasting a specific detail within an image's context, posing a different challenge from REI. To visually prompt the speaker on the target object, following [215] we use a red circle as big as the ground truth bounding box.

Agents. Our experiments consider pairs of agents: a speaker and a listener. Specifically, we use LLaVA-7B [135] as the speaker across all adaptation experiments, providing a good balance between its pre-trained capabilities to bootstrap from and a model size that allows us to fine-tune LoRA adapters on a single A100 40GB GPU. As listener agents, we employ LLaVA-7B, LLaVA-13B, Qwen (7B)[24] for REI, and PaliGemma (3B) [27] for RES, which is the only open model of reasonable size capable of producing segmentation masks as

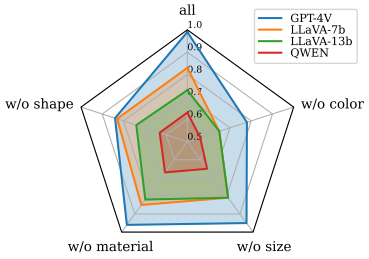


Figure 7.3: Performance for various agents on ground-truth descriptions with all attributes and with sets of three attributes for CLEVR.

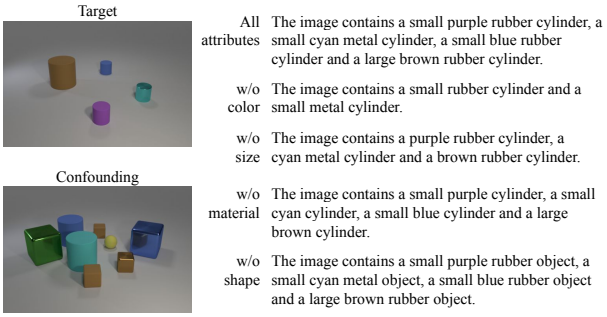


Figure 7.4: Example of ground-truth descriptions (right) on CLEVR for the target image (left) with all attributes, and with sets of three attributes.

output. Each listener model has distinct capabilities when it comes to image and language recognition, with Qwen being the weakest one. This diversity in listener agents simulates a population of agents, testing the speaker's ability to adapt its language effectively.

To introduce an additional challenge, we induce perceptual weaknesses in the listener agents: "color blindness" (grayscaled images) and "blurred vision" (Gaussian blur). These weaknesses require the speaker, which receives unaltered images, to adapt its language to account for concepts that are not recognizable by the listener agent.

Training and evaluation. We train the speaker (LLaVA-7B) with LoRA adapters on all linear layers of the LLM, keeping the vision module fixed. During online adaptation, we play three episodes before updating the parameters using PPO, NLPO, or KTO algorithms, resulting in a batch size of 3 which maximizes our memory usage. The speaker is trained for 1800 interactions (600 update steps) and evaluated on a held-out test set of 300 episodes per dataset. We use the average success rate as evaluation metric for REI and mean IoU for the RES task. Each experiment combines a specific speaker-listener pair either with or without perceptual weaknesses. We provide additional details about the MLLM prompts in Supp. F.2.

7.4.2 Evaluating Listeners with Ground-Truth Descriptions on CLEVR

CLEVR's detailed scene descriptions allow us to construct a ground-truth (GT) speaker agent for the REI task that produces image descriptions with perfect perception and reasoning abilities. This enables us to evaluate listeners given an ideal speaker. The produced descriptions mention all attributes that appear at least once in the target image, but do not exist in the confounding image. We also ablate the GT speaker by omitting one attribute type, measuring the importance of each attribute for REI. Examples of these image descriptions are shown in Fig. 7.4.

We evaluate our listener agents alongside GPT-4V, to obtain a reference for a state-of-the-art MLLM, and present the results in Fig. 7.3. We observe that when all attributes are present, GPT-4V performs best (0.99), followed by LLaVA-7B and LLaVA-13B (0.83 and

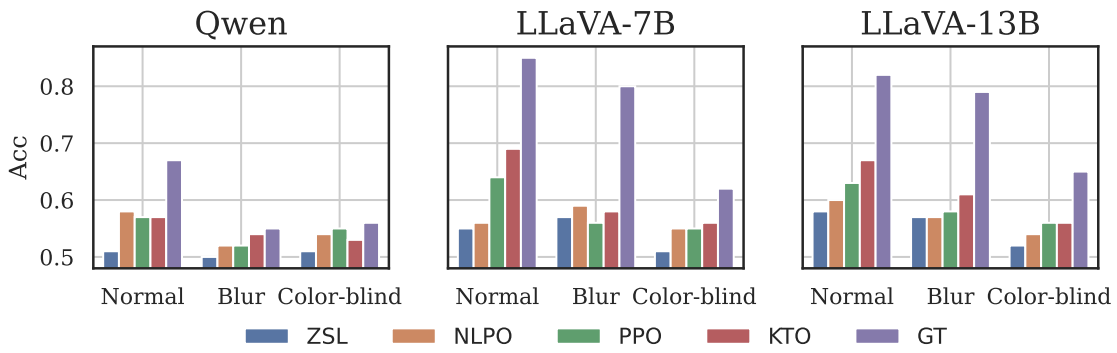


Figure 7.5: Comparing NLPO, PPO, KTO, GT on CLEVR. ZSL: no training was involved. Normal: no perceptual impairment, Blur: Blurry vision, Color blind: Vision with no color.

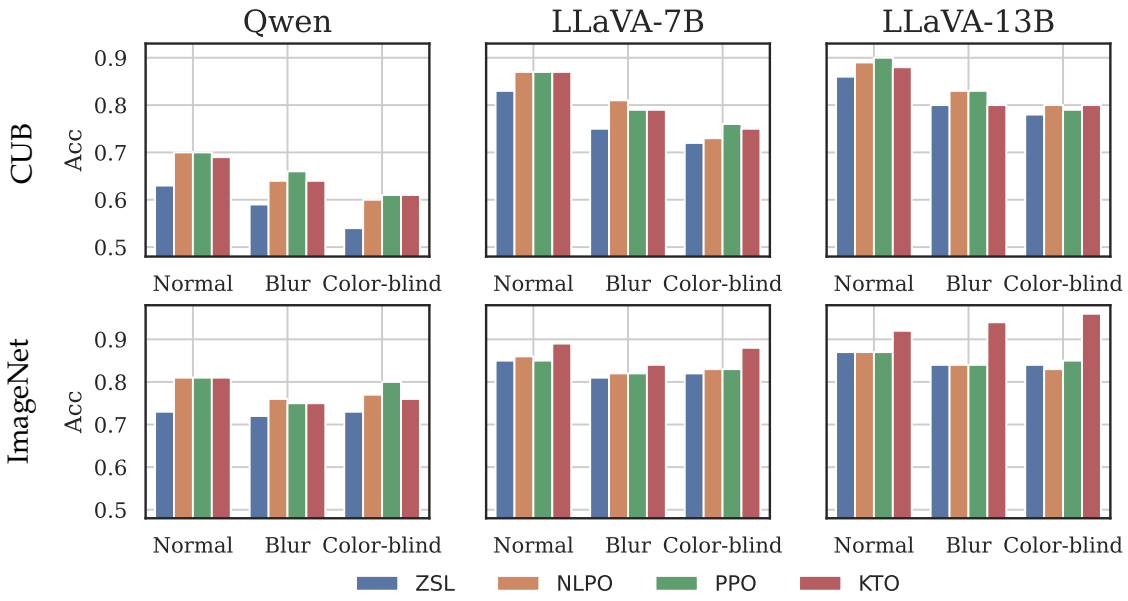


Figure 7.6: Results on the CUB (Top) and ImageNet (Bottom) datasets (REI task). ZSL means that no training was involved. Perceptual weakness refers to the visual impairment applied to the listener.

0.73), with QWEN being the weakest model (0.63) . Removing size and material attributes has little impact on performance, except for a slight increase in LLaVA-13B and QWEN's scores, indicating that size information is more confusing than helpful for these models, possibly because of perspective. In contrast, omitting shape information significantly affects GPT-4V's performance (from 0.99 to 0.84), while the other listeners are less affected, showing that GPT-4V is more sensitive to shape than other models.

Most notably, removing color information results in significant performance drops across all listeners, highlighting its importance for solving the REI task on CLEVR. These findings demonstrate that different MLLMs prioritize different attributes and have varying capabilities, as shown in Fig. 7.3. Even GPT-4V struggles to solve the task without color or shape information.

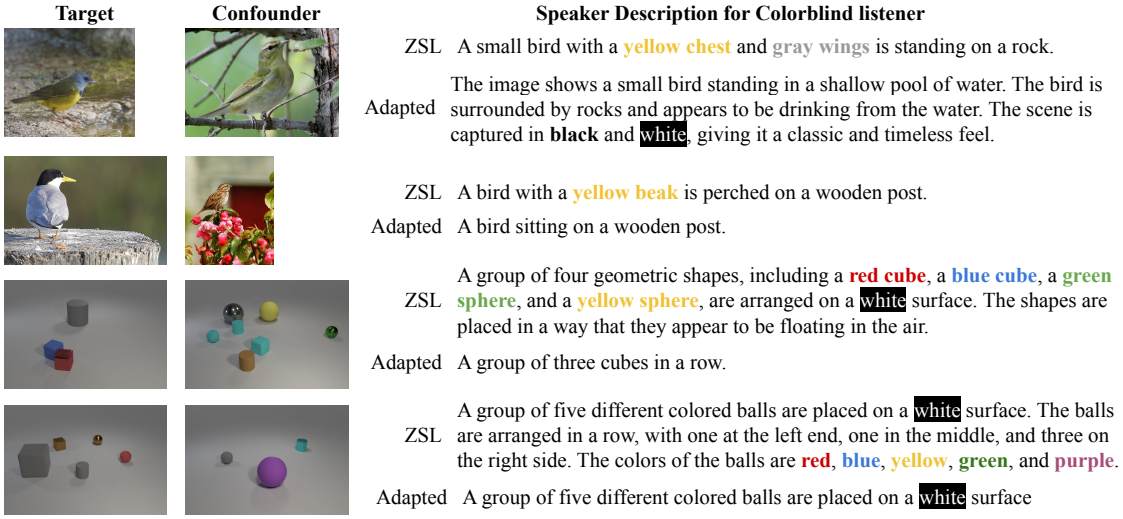


Figure 7.7: Qualitative results on CUB and CLEVR when the speaker interacts with a colorblind listener. We present the descriptions generated by the untrained agents (ZSL) and the descriptions obtained after training (Adapted). After adaption, the speaker avoids color attributes.

7.4.3 Comparing Listeners and Adaptation Methods on REI Task

REI on CLEVR. As shown in Fig. 7.5, when we do not adapt the speaker in the zero-shot learning (ZSL) setting, listener models achieve modest performance. The LLaVA-13B listener achieves the highest performance with an accuracy of 0.58. Introducing color blindness decreases performance for both LLaVA models, while blurred vision has little impact. Qwen performs weakest both with and without perceptual weaknesses, i.e., it struggles to understand the descriptions of LLaVA-7B.

KTO-based adaptation significantly improves performance for LLaVA-7B and LLaVA-13B (peaking at 0.69 and 0.67). Qwen also sees smaller improvements to 0.57. PPO-based adaptation yields smaller gains, while NLPO shows little improvement over zero-shot learning, except when Qwen is the listener. Testing these algorithms with perceptual weaknesses reveals reduced performance increases due to the harder task for the speaker. Blurred vision is generally easier to handle than color blindness, with KTO performing the best overall.

Compared to using GT descriptions for evaluating the listeners (0.67/0.82/0.85), there is a significant gap to the best adaptation results with KTO (0.57/0.67/0.69) even with normal vision. This suggests that the REI task is challenging enough for further research in online adaptation of MLLMs.

REI on natural images. Fig. 7.6 presents the adaptation results on CUB and ImageNet using natural images. We observe that all listeners perform well in ZSL, with LLaVA-13B achieving an accuracy of 0.86 (CUB) and 0.87 (ImageNet). The MLLMs are likely more familiar with such natural images making it easier for the speaker to pick out differences

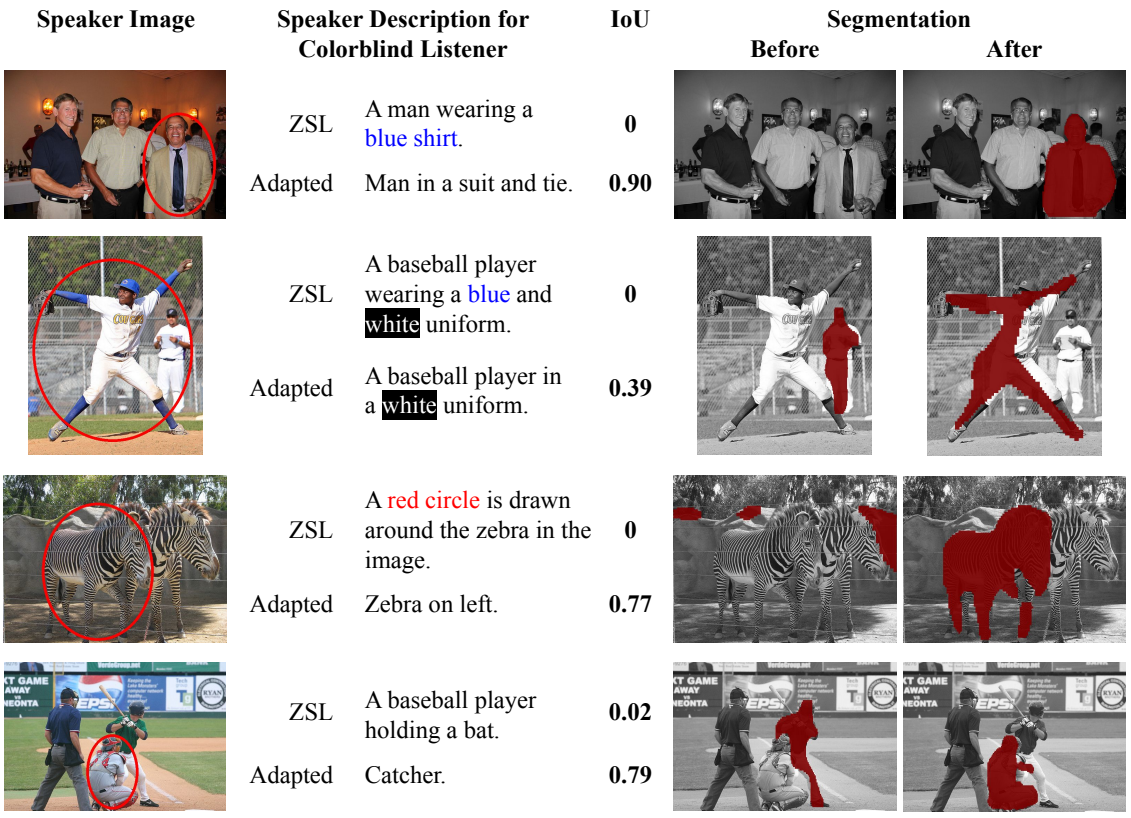


Figure 7.8: Qualitative results of the RES task on RefCOCO with a LLaVA-7B speaker and coloblind PaliGemma listener.

and the listener to recognize them. However, there is still a large gap to Qwen with 0.63/0.73 for CUB/ImageNet.

In general, adaptation methods provide a boost in performance for all listeners. While KTO-based adaptation excels on ImageNet, all three algorithms perform similarly well on CUB. Perceptual weaknesses have a larger impact on CUB, with removing color having the highest effect on performance. On ImageNet both weaknesses only slightly decrease the performance. This is consistent across listeners and algorithms.

In conclusion, online adaptation is possible for every tested agent and algorithm on the REI task. However, listener capabilities influence improvements, and different algorithms perform better on different datasets and listeners. Overall, KTO seems to work best when considering all experiments. At the same time, none of the existing algorithms are able to find a policy that achieves results close to the of the GT agent leaving room for improvement. Moreover, we find that adaptation on blurred or grayscale images can reach or surpass zero-shot learning performance on normal images, which is a desirable outcome in scenarios where we want to avoid a disadvantage for agents with perceptual weaknesses. This applies to a lesser degree on ImageNet, and was not generally true on CUB, where achieving this target could be an promising direction within the REI task framework.

7.4.4 Adapting to PaliGemma on the RES Task

On the referring expression segmentation task, we adapt the LLaVA-7B speaker to PaliGemma as listener on the RefCOCO dataset. In Fig. 7.9, we report the mean intersection over union (mIoU) for ZSL, PPO, NLPO, and KTO together with probing the PaliGemma listener with the ground truth (GT) referring expressions created by humans that come with the dataset.

We find that the RES task poses a particular challenge to some adaptation algorithms, because neither PPO or NLPO can significantly improve over the zero-shot descriptions in normal, blurred, and grayscaled images. Only KTO manages to obtain an improvement from 0.34 to 0.44 for normal images, from 0.28 to 0.41 in blurry images, and from 0.28 to 0.40 in grayscale images. At the same time, the GT descriptions still outperform the KTO adapted speaker reaching 0.63/0.56/0.61 mIoU in the three settings respectively. Thus, we conclude that there is still room for improvement for online adaptation to reach closer to human performance.

When inducing perceptual weaknesses on the PaliGemma listener, ZSL performance degrades, but to a lesser degree than for the REI task. This is expected because objects that are contrasted in RES are often easier to identify by their relation in the scene, e.g., where it is located spatially rather than by color or shapes. As a result, PaliGemma can deal with blur and grayscale relatively well in this context. Apart from KTO being the best adaptation algorithm for RES, we also find that KTO can adapt to perceptually weakened listeners to improve over ZSL performance of the normal listener.

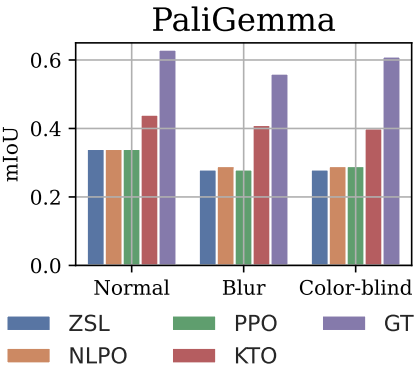


Figure 7.9: mIoU on RefCOCO for RES with LLaVA-7B speaker and PaliGemma listener.

7.4.5 Qualitative Analysis on Colorblind Listener

In Fig. 7.7, we show qualitative results on CUB, and CLEVR, by contrasting generated descriptions before and after adaptation on the REI task when interacting with a colorblind listener.

We observe that color attribute is mentioned predominantly before adaptation, and, apart from referring to "black" and "white", completely avoided after adaptation. On CUB for instance, the speaker mentions the "yellow chest" and "yellow beak" to discriminate the birds in the zero-shot setting, and learns to focus the description more on the surrounding scene and action performed by the bird to discriminate the two images after adaptation. On CLEVR, descriptions similarly contains many references to the color attributes in the initial descriptions, but they do not mention colors after adaptation. In contrast, the adapted descriptions focus on the overall count of the objects and are more concise than

the original ones. Moreover, zero-shot descriptions sometimes mix objects from both images, e.g., description in the third row mentions "red cube" and "blue cube" from the left image, and "green sphere" and "yellow sphere" from the right image. After adaptation this behaviour is suppressed and the speaker focuses more on the target image.

In Fig. 7.8, we show examples of the adaptation on RefCOCO for the RES task, again when the listener is colorblind. The first two rows exemplify how mentioning color can confuse the listener, e.g., in the second row, where the listener segments the incorrect baseball player because it cannot attribute the "blue" uniform to the correct one. After adaptation, not mentioning the colors and focusing on other aspects, such as the "suit and tie" in the first example, allows the listener to more accurately segment the target. Interestingly, there are a few examples where the visual prompting through the red circle [215] can cause incorrect descriptions mentioning the circle which is not visible to the listener. However, online adaptation can also correct for this failure case as seen in the third row, where the speaker correctly refers to the "zebra on left".

In conclusion, from these qualitative examples, we observe that the speaker learns to correctly identify the perceptual weakness of the listener, and adapts its description accordingly to be more effective in its communication.

7.5 Limitations

As it is widely known in the literature [8, 53, 181], RL algorithms tend to be unstable when the reward signal is noisy, or the actions space immense. During this study, we have observed that there is a divergence effect during online adaptation. Fig. 7.10 exemplifies this divergence effect on CLEVR dataset for LLaVA-13B which is representative of the observations on other datasets and with other listeners. For all our experiments, we report the performance after 1800 episodes. However, Fig. 7.10 shows the peak performance is sometimes achieved at different times during training due to the variance in online adaptation. One potential reason for this is the online nature of gathering training samples. The constantly changing policy during training affects the generated data, which in turn influences the future policy and exploration of possible descriptions. With the large actions space of MLLMs, it is challenging to keep these effects in check.

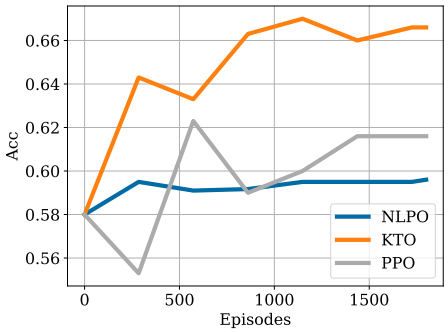


Figure 7.10: Divergence effect on CLEVR for LLaVA-13B. The performance fluctuates instead of monotonically improving.

7.6 Conclusion

In this work, we introduce a framework for two referring expression tasks (REI/RES) involving communicating MLLM agents. On these tasks, we study how MLLM agents can adapt to one another on-the-fly. Our online adaptation setting is significantly more challenging than aligning MLLMs on carefully collected offline datasets, while opening up new applications that require individual personalization. Every communication partner understands language and concepts required to solve the tasks at different levels and we introduce perceptual weaknesses to further control for agent variety. The referring expression tasks pose a challenge to currently available MLLMs, especially for images with fine-grained differences, and when precise segmentation is required. All the adaptation algorithms we have tested could improve task performance on REI with KTO working the best overall and being the only one achieving improvements on the RES task. These results show that, 1) it is possible to improve over the initial pre-trained policy by learning about the listener capabilities, and 2) we can perform this learning in an online setting. However, we also observe that current methods do not monotonically improve during the training process, and cannot find an "optimal" policy, since we have demonstrated that better ones exist with our GT agent experiments. With our task setting, we want to encourage further research on how to make online adaptation of MLLM effective and practically viable to extend to real-world scenarios for MLLM personalization.

8

Thesis Discussion and Conclusion

This thesis tackles the problem of multimodal learning in multiple settings, such as audio-visual video classification, video-adverb retrieval, and communication adaptation in the context of MLLMs. Moreover, as qualitative multimodal data is hard to obtain because the modalities must be paired, this work focuses on data efficient learning, emphasizing the generalized zero- and few-shot learning settings.

Chapters 2, 3, 4, and 5 presented a series of innovative works that paved the way for audio-visual generalized zero- and few-shot learning. These chapters formalized these settings correctly, introduced new benchmarks and baselines, and proposed new architectural novelties in the form of state-of-the-art methods. Moreover, Chapter 6 tackled the problem of video-adverb retrieval and proposed a new method to better solve this problem, along with additional zero-shot learning splits that will facilitate a more extensive comparison in this setting. Finally, Chapter 7 studied the task of communication adaptation on the fly between two MLLMs and provided insights into the performance of many MLLMs and adaptation algorithms.

The following sections will review each of the contributions individually and collectively, focusing on their strong aspects and underlining the current limitations and some directions that could be employed to mitigate these limitations in future works.

8.1 Discussion of results

8.1.1 Individual contributions

This thesis started by addressing the audio-visual generalized zero-shot learning task for video classification in Chapter 2. Before this work, there were only two prior works [152, 185] trying to solve this task. However, their setup was prone to data leakage, rendering the assumption of unseen classes void. Furthermore, they only used a single dataset, which was very small. Thus, this work correctly formalized the setting and eliminated the data leakage by providing training and evaluation protocols, dataset splits, and baselines. The benchmark introduced in this work comprises multiple datasets with a higher degree of

complexity than the previous one. Moreover, a novel architecture and new loss functions were introduced, leading to state-of-the-art performance.

However, the temporal dimension was not considered during the exploration in Chapter 2. As a result, Chapter 3 addressed the same problem of audio-visual generalized zero-shot learning by designing a system that can integrate the temporal information from the data, leading to better performance. To achieve this, Chapter 3 proposed a novel attention mechanism and loss functions.

The scope of the research done in this thesis was further expanded by moving into the area of audio-visual generalized few-shot learning for video classification in Chapter 4. As this area had not been explored before, this work defined the setting by providing training and evaluation protocols, a new benchmark, and baselines. Moreover, along with providing the setup for this setting, this work also proposed a state-of-the-art method, which used a novel fusion mechanism in combination with a diffusion model to generate synthetic audio-visual features for augmenting the training samples for the novel classes.

Chapter 5 revisited the problem from Chapter 2. Recently, large visual-language and audio-language models have attained solid performance in many tasks, and this work aimed at replacing the feature extractors in Chapter 2 with newer feature extractors for the audio and visual modalities, such as CLIP and CLAP. Both CLIP and CLAP have corresponding text encoders, allowing the use of both text encoders to encode the text. The newly introduced method in this work was better at leveraging the new features than previous methods. Furthermore, it was shown empirically that encoding the same text with two different text encoders corresponding to different modalities is beneficial. The proposed method, in combination with the improved text representation, led to state-of-the-art performance.

Chapter 6 moved to a different research question by tackling the task of video-adverb retrieval. A new method based on a residual gated mechanism was designed to better combine the adverb and action text embeddings, leading to enhanced text representation. Moreover, new losses were proposed that took advantage of the enhanced text embeddings more effectively. Finally, because all the possible combinations of adverb-action embeddings are unlikely to be captured in a dataset, this work also introduced new zero-shot dataset splits for a more comprehensive evaluation in this setting.

Finally, Chapter 7 studied the task of communication adaptation between two MLLMs. A new framework, along with benchmarks and baselines, was proposed for this task. Multiple types of MLLMs and adaptation algorithms were analyzed, and insights into their strengths and weaknesses were provided. Furthermore, these MLLMs were tested in settings that simulate visual impairments, such as colorblindness or blurred vision. To the best of our knowledge, this work studied this problem for the first time in the context of MLLMs by adapting them online, using very few interactions.

8.1.2 Collective contributions

Collectively, the whole research done in this thesis aims to advance the field of machine learning by focusing on multimodal data efficient learning. While all the works, except Chapter 7 whose goal was to test the capabilities of current MLLMs, proposed methods that obtain state-of-the-art performance, this was not the sole goal. As observed from all these chapters, a significant emphasis was placed on defining or correctly redefining different settings and introducing baselines and benchmarks to facilitate more research in these areas.

In Chapter 2, a significant focus was placed on correctly redefining the audio-visual generalized zero-shot learning setting by providing benchmark splits and training and evaluation protocols that would enable a correct evaluation in this setting. As this area was also relatively under-explored, a significant emphasis was put on implementing additional baselines. Furthermore, the same can be observed in Chapter 4, where one of the most important contributions was establishing the audio-visual generalized few-shot learning setting, providing benchmarks, and implementing many baselines. Thus, besides providing state-of-the-art methods, these two works aimed to develop new settings and facilitate future research.

Chapter 3 extended the setting of audio-visual generalized zero-shot learning from Chapter 2 by using the temporal information from the videos, offering the possibility for future works to tackle this problem by using the temporal context. Moreover, one goal in Chapter 5 was to adapt the extracted features from Chapter 2 to newer trends in deep learning by employing large visual and audio models for feature extraction and two text encoders for encoding the class names. As a result, these two works aimed to expand the setting introduced in Chapter 2 in two other important directions.

In Chapter 6, the task shifted to video-adverb retrieval. As this task was already quite well established, many benchmarks and baselines were already provided. However, there was still a lack of benchmarks in zero-shot learning, with only a few provided. One main goal was to extend the number of zero-shot benchmarks by adding new ones for a more extensive comparison.

Chapter 7 focused on providing a framework for evaluating the communication adaptation capabilities of MLLMs. New benchmarks were introduced based on the referring expression identification (REI) and referring expression segmentation (RES) tasks. In line with the previous chapters, the goal of this work was also to tackle this problem by using as little training data as possible.

Moreover, all these works provided insights into various fusion mechanisms for combining information from multiple modalities. The most important observation was that there are better ways to fuse information from multiple modalities than full attention. Furthermore, it was shown that this is not a trivial problem, and different fusion mechanisms may work well for some tasks but not others.

Thus, collectively, the research done in this thesis introduced multiple novel settings or

corrected previously introduced settings by providing multiple benchmarks and baselines. Furthermore, it provided insights into ways of fusing information from multiple modalities. Hopefully, this thesis will encourage more researchers to provide better solutions to these settings.

8.2 Conclusion and Future Directions

Research in unimodal learning has been explored for a long time, leading to remarkable results in vision and language domains. However, only recently has there been an increased focus on multimodal learning with the breakthrough of the transformer architecture and models like CLIP, which showed that multimodal learning is already achieving satisfactory performance in some tasks. However, many of these multimodal models usually focus on the visual-language task, where the visual modality is represented by images, neglecting most of the other modalities, such as video and audio. This thesis focuses mainly on the video modality in different settings, such as audio-visual learning and video-adverb retrieval, while also tackling the popular setting of visual-language learning in the context of communication adaptation in MLLMs. Moreover, all these tasks are studied in the data efficient learning scenario, focusing strongly on generalized zero- and few-shot learning. As many of the tasks presented in these chapters are significantly underexplored, this thesis also provides benchmarks and baselines and introduces new models to achieve state-of-the-art performance, setting the stage for future exploration.

As with every line of research, the works presented in this thesis also have limitations. Chapter 2 and 5 have the limitation of using temporally averaged features. While Chapter 3 and 4 try to extend the models to take advantage of the temporal information, some limitations remain. All these works use frozen pre-trained feature extractors. This decision was taken due to significant limitations in computational resources. Training the feature extractors together with the fusion networks would also significantly boost performance. One line of research could consist of making this training more efficient by only training specific parts or adding small adapters in the feature extractors using parameter-efficient fine-tuning methods, such as [95, 154, 285]. This would significantly reduce the computational burden of training these feature extractors.

Another limitation of Chapters 2, 3 and 4 is that the text embeddings of classes are based on the class names (e.g., archery, dog barking). While Chapter 5 extends this by introducing prompts represented by complete sentences such that CLIP and CLAP can be used, future improvements can still be made. Devising better text representations of classes, such as using descriptions of the concepts captured in the class names (e.g., what archery means, instead of just using the word archery), would potentially enhance the text representation and it would facilitate a better match between the audio-visual features and the text representation of the correct class.

Chapter 6 uses the ground-truth action as a query for obtaining the video representation

similar to previous works, which can be considered a limitation. This simplifies the real-world scenario, where the model should be able to identify both the action and the adverb that describes the video. Further research could explore designing models that can perform both tasks well.

A final limitation is found in Chapter 7, which studies the ability to adapt the communication between two MLLMs. As mentioned in Chapter 7, it was observed that there are some instabilities in training these systems, and training them longer does not always lead to the best performance. This may happen due to the online setting, where new interactions are generated as the speaker MLLM is adapted to the listener MLLM. This is challenging, as the policy of the speaker MLLM can drift very quickly, leading to inadequate or incomprehensible outputs. One promising avenue for exploration would involve a memory buffer, which could store older interactions. Then, one could use both the interactions from the buffer and the online interactions to adapt the speaker MLLM. This could alleviate the problem by making the system more stable.

Bibliography

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. "Youtube-8m: A large-scale video classification benchmark". In: *arXiv preprint arXiv:1609.08675*. 2016 (cit. on pp. 43, 123, 131).

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. "Gpt-4 technical report". In: *arXiv preprint arXiv:2303.08774*. 2023 (cit. on p. 54).

[3] Jonas Adler and Sebastian Lunz. "Banach wasserstein gan". In: *NeurIPS*. 2018 (cit. on p. 43).

[4] Triantafyllos Afouras, Yuki M Asano, Francois Fagan, Andrea Vedaldi, and Florian Metze. "Self-supervised object detection from audio-visual correspondence". In: *CVPR*. 2022 (cit. on pp. 14, 28, 42, 55).

[5] Triantafyllos Afouras, Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. "Deep audio-visual speech recognition". In: *IEEE TPAMI*. 2018 (cit. on pp. 14, 28, 42, 55).

[6] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. "ASR is all you need: Cross-modal distillation for lip reading". In: *ICASSP*. 2020 (cit. on pp. 14, 28, 42, 55).

[7] Triantafyllos Afouras, Andrew Owens, Joon Son Chung, and Andrew Zisserman. "Self-Supervised Learning of Audio-Visual Objects from Video". In: *ECCV*. 2020 (cit. on pp. 14, 28, 42, 55).

[8] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. "Back to Basics: Revisiting REINFORCE Style Optimization for Learning from Human Feedback in LLMs". In: *arXiv preprint arXiv:2402.14740*. 2024 (cit. on pp. 79, 82, 83, 90).

[9] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. "Label-embedding for image classification". In: *IEEE TPAMI*. 2015 (cit. on pp. 12, 14, 20, 21, 28, 55, 122).

[10] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. "Evaluation of output embeddings for fine-grained image classification". In: *CVPR*. 2015 (cit. on pp. 12, 14, 20, 21, 28, 55, 122).

[11] Stephan Alaniz, Diego Marcos, and Zeynep Akata. "Learning Decision Trees Recurrently Through Communication". In: *CVPR*. 2021 (cit. on p. 80).

[12] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. "Flamingo: a Visual Language Model for Few-Shot Learning". In: *NeurIPS*. 2022 (cit. on p. 78).

[13] Jean-Baptiste Alayrac, Adria Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. "Self-supervised multimodal versatile networks". In: *NeurIPS* (2020) (cit. on p. 67).

[14] Taha Alhersh, Heiner Stuckenschmidt, Atiq Ur Rehman, and Samir Brahim Belhaouari. "Learning human activity from visual data using deep learning". In: *IEEE Access*. 2021 (cit. on p. 66).

[15] Humam Alwassel, Dhruv Mahajan, Lorenzo Torresani, Bernard Ghanem, and Du Tran. "Self-supervised learning by cross-modal audio-video clustering". In: *NeurIPS*. 2020 (cit. on pp. 14, 28, 42, 55).

[16] Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. "Gemini: A Family of Highly Capable Multimodal Models". In: *arXiv preprint arXiv:2312.11805*. 2023 (cit. on p. 78).

[17] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. "Localizing moments in video with natural language". In: *ICCV*. 2017 (cit. on p. 67).

[18] Alan Ansell, E. Ponti, Anna Korhonen, and Ivan Vulic. "Composable Sparse Fine-Tuning for Cross-Lingual Transfer". In: *ACL*. 2021 (cit. on p. 80).

[19] Relja Arandjelovic and Andrew Zisserman. "Objects that sound". In: *ECCV*. 2018 (cit. on pp. 14, 28, 42, 55).

[20] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks". In: *ICML*. 2017 (cit. on p. 51).

[21] Yuki M. Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi. "Labelling unlabelled videos from scratch with multi-modal self-supervision". In: *NeurIPS*. 2020 (cit. on pp. 13, 14, 19, 20, 22, 34, 54, 55, 121–124).

[22] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. "Soundnet: Learning sound representations from unlabeled video". In: *NeurIPS*. 2016 (cit. on pp. 14, 28, 42, 55).

[23] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. "Layer normalization". In: *arXiv preprint arXiv:1607.06450*. 2016 (cit. on pp. 16, 31, 70).

[24] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. "Qwen Technical Report". In: *arXiv preprint arXiv:2309.16609*. 2023 (cit. on pp. 79, 84).

[25] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. "Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval". In: *ICCV*. 2021 (cit. on p. 67).

[26] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. "BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models". In: *ACL*. 2022 (cit. on p. 80).

[27] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. "PaliGemma: A versatile 3B VLM for transfer". In: *arXiv preprint arXiv:2407.07726*. 2024 (cit. on pp. 79, 84).

[28] Mina Bishay, Georgios Zoumpourlis, and Ioannis Patras. "Tarn: Temporal attentive relation network for few-shot and zero-shot action recognition". In: *BMVC*. 2019 (cit. on pp. 12, 14, 15, 42).

[29] Andreas Blattmann, Robin Rombach, Kaan Oktay, Jonas Müller, and Björn Ommer. "Semi-Parametric Neural Image Synthesis". In: *NeurIPS*. 2022 (cit. on p. 43).

[30] Yang Bo, Yangdi Lu, and Wenbo He. "Few-shot learning of video action recognition only based on video contents". In: *WACV*. 2020 (cit. on pp. 45, 49, 134).

[31] Wim Boes and Hugo Van hamme. "Audiovisual transformer architectures for large-scale classification and synchronization of weakly labeled audio events". In: *ACM MM*. 2019 (cit. on pp. 29, 42).

[32] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. "Enriching Word Vectors with Subword Information". In: *TACL*. 2017 (cit. on p. 140).

[33] Damian Borth, Rongrong Ji, Tao Chen, Thomas Breuel, and Shih-Fu Chang. "Large-scale visual sentiment ontology and detectors using adjective noun pairs". In: *ACM MM*. 2013 (cit. on p. 68).

[34] Biagio Brattoli, Joseph Tighe, Fedor Zhdanov, Pietro Perona, and Krzysztof Chalupka. "Rethinking zero-shot video classification: End-to-end training for realistic applications". In: *CVPR*. 2020 (cit. on pp. 12–15, 27).

[35] Andrew Brown, Vicky Kalogeiton, and Andrew Zisserman. "Face, body, voice: Video person-clustering with multiple modalities". In: *ICCV*. 2021 (cit. on p. 14).

[36] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: *NeurIPS*. 2020 (cit. on p. 140).

[37] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott M. Lundberg, Harsha Nori, Hamid Palangi, Marco Túlio Ribeiro, and Yi Zhang. "Sparks of Artificial General Intelligence: Early experiments with GPT-4". In: *arXiv preprint arXiv:2303.12712*. 2023 (cit. on p. 78).

[38] Kaidi Cao, Jingwei Ji, Zhangjie Cao, Chien-Yi Chang, and Juan Carlos Niebles. "Few-shot video classification via temporal alignment". In: *CVPR*. 2020 (cit. on pp. 41, 42).

[39] Joao Carreira, Skanda Koppula, Daniel Zoran, Adria Recasens, Catalin Ionescu, Olivier Henaff, Evan Shelhamer, Relja Arandjelovic, Matt Botvinick, Oriol Vinyals, et al. "Hierarchical perceiver". In: *arXiv preprint arXiv:2202.10890*. 2022 (cit. on pp. 45, 49, 134).

[40] Joao Carreira and Andrew Zisserman. "Quo vadis, action recognition? a new model and the kinetics dataset". In: *CVPR*. 2017 (cit. on p. 54).

[41] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. "An empirical study and analysis of generalized zero-shot learning for object recognition in the wild". In: *ECCV*. 2016 (cit. on pp. 20–22, 34, 49, 50, 60, 133).

[42] Chao-Yeh Chen and Kristen Grauman. "Inferring analogous attributes". In: *CVPR*. 2014 (cit. on p. 68).

[43] Honglie Chen, Weidi Xie, Triantafyllos Afouras, Arsha Nagrani, Andrea Vedaldi, and Andrew Zisserman. "Audio-Visual Synchronisation in the wild". In: *BMVC*. 2021 (cit. on pp. 14, 55).

[44] Honglie Chen, Weidi Xie, Triantafyllos Afouras, Arsha Nagrani, Andrea Vedaldi, and Andrew Zisserman. "Localizing Visual Sounds the Hard Way". In: *CVPR*. 2021 (cit. on pp. 14, 28, 42, 55).

[45] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. "Vggsound: A large-scale audio-visual dataset". In: *ICASSP*. 2020 (cit. on pp. 13, 19, 20, 44).

[46] Hui Chen, Guiguang Ding, Zijia Lin, Sicheng Zhao, and Jungong Han. "Cross-modal image-text retrieval with semantic consistency". In: *ACM MM*. 2019 (cit. on p. 67).

[47] Ke Chen, Xingjian Du, Bilei Zhu, Zejun Ma, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. "HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection". In: *ICASSP*. 2022 (cit. on p. 54).

[48] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. "Fine-grained video-text retrieval with hierarchical graph reasoning". In: *CVPR*. 2020 (cit. on p. 68).

[49] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. "A closer look at few-shot classification". In: *ICLR*. 2019 (cit. on p. 42).

[50] Yanbei Chen, Yongqin Xian, A. Sophia Koepke, Ying Shan, and Zeynep Akata. "Distilling Audio-Visual Knowledge by Compositional Contrastive Learning". In: *CVPR*. 2021 (cit. on pp. 14, 28, 42, 55).

[51] Ying Cheng, Ruize Wang, Zhihao Pan, Rui Feng, and Yuejie Zhang. "Look, listen, and attend: Co-attention network for self-supervised audio-visual representation learning". In: *ACM MM*. 2020 (cit. on pp. 14, 55).

[52] Jeong Choi, Jongpil Lee, Jiyoung Park, and Juhan Nam. "Zero-shot learning for audio-based music classification and tagging". In: *ISMIR*. 2019 (cit. on p. 15).

[53] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. "Deep Reinforcement Learning from Human Preferences". In: *NeurIPS*. 2017 (cit. on pp. 82, 83, 90).

[54] Joon Son Chung and Andrew Zisserman. "Out of time: automated lip sync in the wild". In: *ACCV*. 2016 (cit. on pp. 14, 55).

[55] Rodolfo Corona, Stephan Alaniz, and Zeynep Akata. "Modeling Conceptual Understanding in Image Reference Games". In: *NeurIPS*. 2019 (cit. on pp. 79, 80).

[56] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M. F. Moura, Devi Parikh, and Dhruv Batra. "Visual Dialog". In: *CVPR*. 2016 (cit. on p. 80).

[57] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *CVPR*. 2009 (cit. on pp. 79, 84).

[58] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. "QLoRA: Efficient Finetuning of Quantized LLMs". In: *NeurIPS*. 2024 (cit. on p. 80).

[59] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *NAACL*. 2019 (cit. on pp. 15, 29, 54).

[60] Prafulla Dhariwal and Alexander Nichol. "Diffusion models beat gans on image synthesis". In: *NeurIPS*. 2021 (cit. on p. 43).

[61] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. "Decoupling zero-shot semantic segmentation". In: *CVPR*. 2022 (cit. on pp. 55, 65).

[62] Jianfeng Dong, Xirong Li, and Cees GM Snoek. "Predicting visual features from text for image and video caption retrieval". In: *IEEE Transactions on Multimedia*. 2018 (cit. on p. 67).

[63] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *ICLR*. 2021 (cit. on pp. 29, 54).

[64] Hazel Doughty, Ivan Laptev, Walterio Mayol-Cuevas, and Dima Damen. "Action Modifiers: Learning from Adverbs in Instructional Videos". In: *CVPR*. 2020 (cit. on pp. 66–68, 70, 71, 73–75, 139, 140, 142).

[65] Hazel Doughty and Cees GM Snoek. "How do you do it? Fine-grained action understanding with pseudo-adverbs". In: *CVPR*. 2022 (cit. on pp. 66–68, 72, 74–77, 139, 140, 142).

[66] Matthijs Douze, Arthur Szlam, Bharath Hariharan, and Hervé Jégou. "Low-shot learning with large-scale diffusion". In: *CVPR*. 2018 (cit. on p. 42).

[67] Joshua P Ebeneze, Yongjun Wu, Hai Wei, Sriram Sethuraman, and Zongyi Liu. "Detection of audio-video synchronization errors via event detection". In: *ICASSP*. 2021 (cit. on pp. 14, 55).

[68] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. "Clap learning audio concepts from natural language supervision". In: *ICASSP*. 2023 (cit. on p. 55).

[69] Patrick Esser, Robin Rombach, Andreas Blattmann, and Bjorn Ommer. "Imagebart: Bidirectional context with multinomial diffusion for autoregressive image synthesis". In: *NeurIPS*. 2021 (cit. on p. 43).

[70] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. "Kto: Model alignment as prospect theoretic optimization". In: *ICML*. 2024 (cit. on pp. 79, 80, 83).

[71] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. "Describing objects by their attributes". In: *CVPR*. 2009 (cit. on p. 28).

[72] Haytham M Fayek and Anurag Kumar. "Large Scale Audiovisual Learning of Sounds with Weakly Labeled Data". In: *IJCAI*. 2020 (cit. on pp. 28, 35, 42, 45, 49, 126, 134).

[73] Andrea Frome, Greg Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. "Devise: A deep visual-semantic embedding model". In: *NeurIPS*. 2013 (cit. on pp. 14, 20, 21, 28, 55, 122).

[74] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. "Multi-modal transformer for video retrieval". In: *ECCV*. 2020 (cit. on pp. 29, 42, 68).

[75] Chuang Gan, Deng Huang, Peihao Chen, Joshua B Tenenbaum, and Antonio Torralba. "Foley music: Learning to generate music from videos". In: *ECCV*. 2020 (cit. on pp. 14, 28, 42, 55).

[76] Ruohan Gao and Kristen Grauman. "Co-separating sounds of visual objects". In: *ICCV*. 2019 (cit. on pp. 14, 28, 42, 55).

[77] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. "A neural algorithm of artistic style". In: *arXiv preprint arXiv:1508.06576*. 2015 (cit. on p. 43).

[78] Yuying Ge, Yixiao Ge, Xihui Liu, Dian Li, Ying Shan, Xiaohu Qie, and Ping Luo. "Bridging video-text retrieval with multiple choice questions". In: *CVPR*. 2022 (cit. on p. 68).

[79] Shir Goldstein and Yael Moses. "Guitar Music Transcription from Silent Video." In: *BMVC*. 2018 (cit. on pp. 14, 28, 42, 55).

[80] Yuan Gong, Yu-An Chung, and James Glass. "Ast: Audio spectrogram transformer". In: *INTERSPEECH*. 2021 (cit. on p. 54).

[81] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial networks". In: *Communications of the ACM*. 2020 (cit. on p. 43).

[82] Shreyank N Gowda. "Synthetic Sample Selection for Generalized Zero-Shot Learning". In: *CVPR Workshop*. 2023 (cit. on p. 55).

[83] Shreyank N Gowda, Laura Sevilla-Lara, Kiyoon Kim, Frank Keller, and Marcus Rohrbach. "A New Split for Evaluating True Zero-Shot Action Recognition". In: *DAGM GCPR*. 2021 (cit. on pp. 14, 15).

[84] Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares-López, Alexandre Ramé, Thomas Mesnard, Yao Zhao, Bilal Piot, Johan Ferret, and Mathieu Blondel. "Direct Language Model Alignment from Online AI Feedback". In: *arXiv preprint arXiv:2402.04792*. 2024 (cit. on p. 80).

[85] Akshita Gupta, Sanath Narayan, Salman Khan, Fahad Shahbaz Khan, Ling Shao, and Joost van de Weijer. "Generative multi-label zero-shot learning". In: *IEEE TPAMI*. 2023 (cit. on p. 55).

[86] Lukas Haas, Silas Alberti, and Michal Skreta. "Learning Generalized Zero-Shot Learners for Open-Domain Image Geolocalization". In: *arXiv preprint arXiv:2302.00275*. 2023 (cit. on p. 55).

[87] Meera Hahn, Andrew Silva, and James M Rehg. "Action2vec: A crossmodal embedding approach to action learning". In: *arXiv preprint arXiv:1901.00484*. 2019 (cit. on pp. 12, 14, 15, 68).

[88] Yun Hao, Yukun Su, Guosheng Lin, Hanjing Su, and Qingyao Wu. "Contrastive Generative Network with Recursive-Loop for 3D point cloud generalized zero-shot classification". In: *Pattern Recognition*. 2023 (cit. on p. 55).

[89] Bharath Hariharan and Ross Girshick. "Low-shot visual recognition by shrinking and hallucinating features". In: *ICCV*. 2017 (cit. on pp. 42, 44).

[90] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. "ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding". In: *CVPR*. 2015 (cit. on pp. 13, 19, 44, 67, 139).

[91] Dan Hendrycks and Kevin Gimpel. "Gaussian error linear units (gelus)". In: *arXiv preprint arXiv:1606.08415*. 2016 (cit. on pp. 16, 32).

[92] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. "CNN architectures for large-scale audio classification". In: *ICASSP*. 2017 (cit. on pp. 34, 43, 54, 56, 123, 131).

[93] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models". In: *NeurIPS*. 2020 (cit. on p. 47).

[94] Jie Hong, Zeeshan Hayder, Junlin Han, Pengfei Fang, Mehrtash Harandi, and Lars Petersson. "Hyperbolic Audio-visual Zero-shot Learning". In: *ICCV*. 2023 (cit. on pp. 53, 54, 56, 59–61).

[95] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. "LoRA: Low-Rank Adaptation of Large Language Models". In: *ICLR*. 2022 (cit. on pp. 79, 80, 83, 95).

[96] Thomas Hummel, Otniel-Bogdan Mercea, A. Sophia Koepke, and Zeynep Akata. "Video-adverb retrieval with compositional adverb-action embeddings". In: *BMVC*. 2023 (cit. on pp. 11, 149).

[97] Vladimir Iashin and Esa Rahtu. "A better use of audio-visual cues: Dense video captioning with bi-modal transformer". In: *BMVC*. 2020 (cit. on pp. 29, 42).

[98] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *ICML*. 2015 (cit. on pp. 16, 30, 57, 70).

[99] Phillip Isola, Joseph J Lim, and Edward H Adelson. "Discovering states and transformations in image collections". In: *CVPR*. 2015 (cit. on p. 68).

[100] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. "Image-to-image translation with conditional adversarial networks". In: *CVPR*. 2017 (cit. on p. 43).

[101] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. "Perceiver: General perception with iterative attention". In: *ICML*. 2021 (cit. on pp. 35, 45, 49, 126, 127, 130, 134).

[102] Amir Jamaludin, Joon Son Chung, and Andrew Zisserman. "You said that?: Synthesising talking faces from audio". In: *IJCV*. 2019 (cit. on pp. 14, 28).

[103] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning". In: *CVPR*. 2017 (cit. on pp. 79, 84).

[104] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. "Decoupling Representation and Classifier for Long-Tailed Recognition". In: *ICLR*. 2020 (cit. on p. 42).

[105] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. "Large-Scale Video Classification with Convolutional Neural Networks". In: *CVPR*. 2014 (cit. on pp. 19, 34, 43, 123, 131).

[106] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. "Referitgame: Referring to objects in photographs of natural scenes". In: *EMNLP*. 2014 (cit. on pp. 79, 84).

[107] Alec Kerrigan, Kevin Duarte, Yogesh Rawat, and Mubarak Shah. "Reformulating zero-shot action recognition for multi-label actions". In: *NeurIPS*. 2021 (cit. on p. 27).

[108] Naji Khosravan, Shervin Ardeshir, and Rohit Puri. "On Attention Modules for Audio-Visual Synchronization." In: *CVPR Workshop*. 2019 (cit. on pp. 14, 55).

[109] Seongwoong Kim and Dong-Wan Choi. "Better Generalized Few-Shot Learning Even Without Base Data". In: *AAAI*. 2023 (cit. on p. 42).

[110] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *ICLR*. 2015 (cit. on pp. 21, 34, 49, 59, 73).

[111] Elyor Kodirov, Tao Xiang, and Shaogang Gong. "Semantic autoencoder for zero-shot learning". In: *CVPR*. 2017 (cit. on p. 14).

[112] A Sophia Koepke, Olivia Wiles, Yael Moses, and Andrew Zisserman. "Sight to sound: An end-to-end approach for visual piano transcription". In: *ICASSP*. 2020 (cit. on pp. 14, 28, 42, 55).

[113] A. Sophia Koepke, Olivia Wiles, and Andrew Zisserman. "Visual pitch estimation". In: *SMC*. 2019 (cit. on pp. 14, 28, 42, 55).

[114] Bruno Korbar, Du Tran, and Lorenzo Torresani. "Cooperative learning of audio and video models from self-supervised synchronization". In: *NeurIPS*. 2018 (cit. on pp. 14, 28, 42, 55).

[115] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. "Dense-captioning events in videos". In: *ICCV*. 2017 (cit. on p. 67).

[116] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *NeurIPS*. 2012 (cit. on p. 1).

[117] Anurag Kumar, Maksim Khadkevich, and Christian Fügen. "Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes". In: *ICASSP*. 2018 (cit. on p. 54).

[118] Sai Kumar Dwivedi, Vikram Gupta, Rahul Mitra, Shuaib Ahmed, and Arjun Jain. "Protogan: Towards few shot learning for action recognition". In: *ICCVW*. 2019 (cit. on pp. 42, 43, 45, 49, 52, 134).

[119] David Kurzendörfer, Otniel-Bogdan Mercea, A. Sophia Koepke, and Zeynep Akata. "Audio-Visual Generalized Zero-Shot Learning using Pre-Trained Large Multi-Modal Models". In: *CVPR Workshop*. 2024 (cit. on pp. 11, 149).

[120] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. "Attribute-based classification for zero-shot visual object categorization". In: *IEEE TPAMI*. 2013 (cit. on p. 55).

[121] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. "Less is more: Clipbert for video-and-language learning via sparse sampling". In: *CVPR*. 2021 (cit. on p. 68).

[122] Ang Li, Allan Jabri, Armand Joulin, and Laurens Van Der Maaten. "Learning visual n-grams from web data". In: *ICCV*. 2017 (cit. on p. 55).

[123] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. "Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training". In: *AAAI*. 2020 (cit. on p. 29).

[124] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. "Visualbert: A simple and performant baseline for vision and language". In: *arXiv preprint arXiv:1908.03557*. 2019 (cit. on p. 29).

[125] Xiang Li, Congcong Wen, Yuan Hu, and Nan Zhou. "RS-CLIP: Zero shot remote sensing scene classification via contrastive vision-language supervision". In: *Int. J. Appl. Earth Obs. Geoinf.* 2023 (cit. on p. 55).

[126] Xinzhe Li, Qianru Sun, Yaoyao Liu, Qin Zhou, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. "Learning to self-train for semi-supervised few-shot classification". In: *NeurIPS*. 2019 (cit. on p. 42).

[127] Yixiao Li, Yifan Yu, Chen Liang, Pengcheng He, Nikos Karampatziakis, Weizhu Chen, and Tuo Zhao. "LoftQ: LoRA-Fine-Tuning-Aware Quantization for Large Language Models". In: *ICLR*. 2024 (cit. on p. 80).

[128] Yong-Lu Li, Yue Xu, Xiaohan Mao, and Cewu Lu. "Symmetry and group in attribute-object compositions". In: *CVPR*. 2020 (cit. on p. 68).

[129] Vladislav Lialin, Namrata Shivagunde, Sherin Muckatira, and Anna Rumshisky. "ReLoRA: High-Rank Training Through Low-Rank Updates". In: *ICLR*. 2024 (cit. on p. 80).

[130] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. "Open-vocabulary semantic segmentation with mask-adapted clip". In: *CVPR*. 2023 (cit. on pp. 55, 65).

[131] Chung-Ching Lin, Kevin Lin, Lijuan Wang, Zicheng Liu, and Linjie Li. "Cross-modal Representation Learning for Zero-shot Action Recognition". In: *CVPR*. 2022 (cit. on p. 27).

[132] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. "Microsoft coco: Common objects in context". In: *ECCV*. 2014 (cit. on p. 84).

[133] Yan-Bo Lin and Yu-Chiang Frank Wang. "Audiovisual Transformer with Instance Attention for Audio-Visual Event Localization". In: *ACCV*. 2020 (cit. on pp. 29, 42).

[134] Aiwei Liu, Haoping Bai, Zhiyun Lu, Xiang Kong, Simon Wang, Jiulong Shan, Mengsi Cao, and Lijie Wen. "Direct Large Language Model Alignment Through Self-Rewarding Contrastive Prompt Distillation". In: *arXiv preprint arXiv:2402.11907*. 2024 (cit. on p. 80).

[135] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. "Visual Instruction Tuning". In: *NeurIPS*. 2023 (cit. on pp. 79, 84).

[136] Jing Liu, Toshiaki Koike-Akino, Pu Wang, Matthew Brand, Ye Wang, and Kieran Parsons. "LoDA: Low-Dimensional Adaptation of Large Language Models". In: *NeurIPS Workshop*. 2023 (cit. on p. 80).

[137] Shih yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. "DoRA: Weight-Decomposed Low-Rank Adaptation". In: *ICML*. 2024 (cit. on p. 80).

[138] Song Liu, Haoqi Fan, Shengsheng Qian, Yiru Chen, Wenkui Ding, and Zhongyuan Wang. "HiT: Hierarchical Transformer with Momentum Contrast for Video-Text Retrieval". In: *ICCV*. 2021 (cit. on p. 29).

[139] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. "Learning to propagate labels: Transductive propagation network for few-shot learning". In: *ICLR*. 2019 (cit. on p. 42).

[140] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. "Use what you have: Video retrieval using representations from collaborative experts". In: *BMVC*. 2019 (cit. on p. 68).

[141] Yang Liu, Jishun Guo, Deng Cai, and Xiaofei He. "Attribute attention for semantic disambiguation in zero-shot learning". In: *CVPR*. 2019 (cit. on p. 28).

[142] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks". In: *NeurIPS*. 2019 (cit. on p. 29).

[143] Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li. "Segclip: Patch aggregation with learnable centers for open-vocabulary semantic segmentation". In: *ICML*. 2023 (cit. on pp. 55, 65).

[144] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. "CLIP4Clip: An empirical study of CLIP for end to end video clip retrieval and captioning". In: *Neurocomputing*. 2022 (cit. on p. 68).

[145] Pingchuan Ma, Stavros Petridis, and Maja Pantic. "End-to-end audio-visual speech recognition with conformers". In: *ICASSP*. 2021 (cit. on p. 55).

[146] Zhengyi Ma, Zhicheng Dou, Yutao Zhu, Hanxun Zhong, and Ji rong Wen. "One Chatbot Per Person: Creating Personalized Chatbots based on Implicit User Profiles". In: *SIGIR*. 2021 (cit. on p. 80).

[147] Ziqiao Ma, Jacob Sansom, Run Peng, and Joyce Chai. "Towards A Holistic Landscape of Situated Theory of Mind in Large Language Models". In: *EMNLP*. 2023 (cit. on p. 80).

[148] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *JMLR*. 2008 (cit. on pp. 22, 38).

[149] Sagnik Majumder, Changan Chen, Ziad Al-Halah, and Kristen Grauman. "Few-Shot Audio-Visual Learning of Environment Acoustics". In: *NeurIPS*. 2022 (cit. on p. 41).

[150] Massimiliano Mancini, Muhammad Ferjad Naeem, Yongqin Xian, and Zeynep Akata. "Open World Compositional Zero-Shot Learning". In: *CVPR*. 2021 (cit. on pp. 68, 140).

[151] Prasanna Mayilvahanan, Thaddäus Wiedemer, Evgenia Rusak, Matthias Bethge, and Wieland Brendel. "Does CLIP's Generalization Performance Mainly Stem from High Train-Test Similarity?" In: *ICLR*. 2024 (cit. on p. 65).

[152] Pratik Mazumder, Pravendra Singh, Kranti Kumar Parida, and Vinay P Namboodiri. "Avgzslnet: Audio-visual generalized zero-shot learning by reconstructing label features from multi-modal embeddings". In: *WACV*. 2021 (cit. on pp. 6, 13, 15, 16, 18, 20, 21, 27–29, 35, 53, 54, 56, 60, 61, 92, 122–125).

[153] Xinhao Mei, Chutong Meng, Haohe Liu, Qiuqiang Kong, Tom Ko, Chengqi Zhao, Mark D Plumbley, Yuexian Zou, and Wenwu Wang. "Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research". In: *TASLP*. 2023 (cit. on pp. 7, 54–56, 59, 65).

[154] Otniel-Bogdan Mercea, Alexey Gritsenko, Cordelia Schmid, and Anurag Arnab. "Time- Memory- and Parameter-Efficient Visual Adaptation". In: *CVPR*. 2024 (cit. on pp. 80, 95, 150).

[155] Otniel-Bogdan Mercea, Thomas Hummel, A. Sophia Koepke, and Zeynep Akata. "Temporal and cross-modal attention for audio-visual zero-shot learning". In: *ECCV*. 2022 (cit. on pp. 10, 41, 44, 45, 49, 50, 53, 54, 56, 58–60, 133, 134, 140, 149).

[156] Otniel-Bogdan Mercea, Thomas Hummel, A. Sophia Koepke, and Zeynep Akata. "Text-to-feature diffusion for audio-visual few-shot learning". In: *DAGM GCPR*. 2023 (cit. on pp. 11, 55, 149).

[157] Otniel-Bogdan Mercea, Lukas Riesch, A. Sophia Koepke, and Zeynep Akata. "Audio-visual Generalised Zero-shot Learning with Cross-modal Attention and Language". In: *CVPR*. 2022 (cit. on pp. 9, 27–29, 34, 35, 41, 43–45, 49, 50, 53, 54, 56, 59–61, 126, 127, 129, 130, 133, 134, 140, 149).

[158] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. "End-to-end learning of visual representations from uncurated instructional videos". In: *CVPR*. 2020 (cit. on pp. 66, 67).

[159] Antoine Miech, Ivan Laptev, and Josef Sivic. "Learning a text-video embedding from incomplete and heterogeneous data". In: *arXiv preprint arXiv:1804.02516*. 2018 (cit. on p. 68).

[160] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. "Howto100m: Learning a text-video embedding by watching hundred million narrated video clips". In: *ICCV*. 2019 (cit. on pp. 67, 73).

[161] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space". In: *ICLR*. 2013 (cit. on pp. 15, 28, 47, 52, 56, 140).

[162] Shaobo Min, Hantao Yao, Hongtao Xie, Chaoqun Wang, Zheng-Jun Zha, and Yongdong Zhang. "Domain-aware visual bias eliminating for generalized zero-shot learning". In: *CVPR*. 2020 (cit. on pp. 22, 133).

[163] Mehdi Mirza and Simon Osindero. "Conditional generative adversarial nets". In: *arXiv preprint arXiv:1411.1784*. 2014 (cit. on p. 43).

[164] Ishan Misra, Abhinav Gupta, and Martial Hebert. "From red wine to red tomato: Composition with context". In: *CVPR*. 2017 (cit. on p. 68).

[165] Davide Moltisanti, Frank Keller, Hakan Bilen, and Laura Sevilla-Lara. "Learning Action Changes by Measuring Verb-Adverb Textual Relationships". In: *CVPR*. 2023 (cit. on pp. 66–68, 72–75, 77, 139–143).

[166] Liliane Momeni, Triantafyllos Afouras, Themos Stafylakis, Samuel Albanie, and Andrew Zisserman. "Seeing wake words: Audio-visual keyword spotting". In: *BMVC*. 2020 (cit. on pp. 14, 55).

[167] Liliane Momeni, Mathilde Caron, Arsha Nagrani, Andrew Zisserman, and Cordelia Schmid. "Verbs in Action: Improving verb understanding in video-language models". In: *ICCV*. 2023 (cit. on p. 68).

[168] Muhammad Ferjad Naeem, Yongqin Xian, Federico Tombari, and Zeynep Akata. "Learning Graph Embeddings for Compositional Zero-Shot Learning". In: *CVPR*. 2021 (cit. on pp. 68, 140).

[169] Tushar Nagarajan and Kristen Grauman. "Attributes as operators: factorizing unseen attribute-object compositions". In: *ECCV*. 2018 (cit. on p. 68).

[170] Arsha Nagrani, Joon Son Chung, Samuel Albanie, and Andrew Zisserman. "Disentangled speech embeddings using cross-modal self-supervision". In: *ICASSP*. 2020 (cit. on pp. 14, 55).

[171] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. "Attention bottlenecks for multimodal fusion". In: *NeurIPS*. 2021 (cit. on pp. 14, 40, 41, 45, 49, 55, 134).

[172] Vinod Nair and Geoffrey E Hinton. "Rectified linear units improve restricted boltzmann machines". In: *ICML*. 2010 (cit. on pp. 16, 30, 57, 70).

[173] Zhixiong Nan, Yang Liu, Nanning Zheng, and Song-Chun Zhu. "Recognizing unseen attribute-object pair with generative model". In: *AAAI*. 2019 (cit. on p. 68).

[174] Medhini Narasimhan, Shiry Ginosar, Andrew Owens, Alexei A Efros, and Trevor Darrell. "Strumming to the beat: Audio-conditioned contrastive video textures". In: *WACV*. 2022 (cit. on pp. 14, 28, 42, 55).

[175] Sanath Narayan, Akshita Gupta, Fahad Shahbaz Khan, Cees GM Snoek, and Ling Shao. "Latent embedding feedback and discriminative features for zero-shot classification". In: *ECCV*. 2020 (cit. on pp. 14, 28, 42, 43).

[176] Jinjie Ni, Tom Young, Vlad Pandelea, Fuzhao Xue, Vinay Vishnumurthy Adiga, and E. Cambria. "Recent advances in deep learning based dialogue systems: a systematic survey". In: *Artificial Intelligence Review*. 2021 (cit. on p. 80).

[177] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. "Zero-shot learning by convex combination of semantic embeddings". In: *ICLR*. 2013 (cit. on p. 55).

[178] Zachary Novack, Julian McAuley, Zachary Chase Lipton, and Saurabh Garg. "Chils: Zero-shot image classification with hierarchical label sets". In: *ICML*. 2023 (cit. on p. 55).

[179] Andreea-Maria Oncescu, Joao F Henriques, Yang Liu, Andrew Zisserman, and Samuel Albanie. "Queryd: A video dataset with high-quality text and audio narrations". In: *ICASSP*. 2021 (cit. on p. 67).

[180] Mayu Otani, Yuta Nakashima, Esa Rahtu, Janne Heikkilä, and Naokazu Yokoya. "Learning joint representations of videos and sentences with web image search". In: *ECCV*. 2016 (cit. on p. 67).

[181] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. "Training language models to follow instructions with human feedback". In: *NeurIPS*. 2022 (cit. on pp. 79, 82, 83, 90).

[182] Andrew Owens and Alexei A Efros. "Audio-visual scene analysis with self-supervised multisensory features". In: *ECCV*. 2018 (cit. on pp. 14, 28, 42, 55).

[183] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. "Ambient sound provides supervision for visual learning". In: *ECCV*. 2016 (cit. on pp. 14, 28, 42, 55).

[184] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. "Learning sight from sound: Ambient sound provides supervision for visual learning". In: *IJCV*. 2018 (cit. on pp. 14, 28, 42, 55).

[185] Kranti Parida, Neeraj Matiyali, Tanaya Guha, and Gaurav Sharma. "Coordinated joint multimodal embeddings for generalized audio-visual zero-shot classification and retrieval of videos". In: *WACV*. 2020 (cit. on pp. 6, 13, 15, 16, 20, 21, 27–29, 35, 53, 54, 56, 60, 61, 92, 122).

[186] Jae Sung Park, Sheng Shen, Ali Farhadi, Trevor Darrell, Yejin Choi, and Anna Rohrbach. "Exposing the limits of video-text models through contrast sets". In: *ACL*. 2022 (cit. on p. 68).

[187] Mandela Patrick, Yuki M Asano, Ruth Fong, João F Henriques, Geoffrey Zweig, and Andrea Vedaldi. "Multi-modal self-supervision from generalized data transformations". In: *NeurIPS*. 2020 (cit. on pp. 14, 28, 40, 42, 55).

[188] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander G Hauptmann, Joao F. Henriques, and Andrea Vedaldi. "Support-set bottlenecks for video-text representation learning". In: *ICLR*. 2021 (cit. on pp. 66, 67).

[189] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "GloVe: Global Vectors for Word Representation". In: *EMNLP*. 2014 (cit. on pp. 15, 140).

[190] Toby Perrett, Alessandro Masullo, Tilo Burghardt, Majid Mirmehdi, and Dima Damen. "Temporal-relational crosstransformers for few-shot action recognition". In: *CVPR*. 2021 (cit. on p. 42).

[191] Jonas Pfeiffer, Ivan Vulic, Iryna Gurevych, and Sebastian Ruder. "MAD-X: An Adapter-based Framework for Multi-task Cross-lingual Transfer". In: *EMNLP*. 2020 (cit. on p. 80).

[192] Hieu Pham, Zihang Dai, Golnaz Ghiasi, Kenji Kawaguchi, Hanxiao Liu, Adams Wei Yu, Jiahui Yu, Yi-Ting Chen, Minh-Thang Luong, Yonghui Wu, et al. "Combined scaling for zero-shot transfer learning". In: *Neurocomputing*. 2023 (cit. on p. 55).

[193] A. J. Piergiovanni, Isaac Noble, Dahun Kim, Michael S. Ryoo, Victor Gomes, and Anelia Angelova. "Mirasol3B: A Multimodal Autoregressive model for time-aligned and contextual modalities". In: *arXiv preprint arXiv:2311.05698*. 2023 (cit. on p. 78).

[194] KR Prajwal, Liliane Momeni, Triantafyllos Afouras, and Andrew Zisserman. "Visual Keyword Spotting with Attention". In: *BMVC*. 2021 (cit. on p. 14).

[195] Hang Qi, Matthew Brown, and David G Lowe. "Low-shot learning with imprinted weights". In: *CVPR*. 2018 (cit. on p. 42).

[196] Rui Qian, Di Hu, Heinrich Dinkel, Mengyue Wu, Ning Xu, and Weiyao Lin. "Multiple sound sources localization from coarse to fine". In: *ECCV*. 2020 (cit. on pp. 14, 55).

[197] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. "Learning transferable visual models from natural language supervision". In: *ICML*. 2021 (cit. on pp. 1, 7, 54–56, 59, 65, 76, 77, 135, 140–142).

[198] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. "Language Models are Unsupervised Multitask Learners". In: *OpenAI blog*. 2019 (cit. on p. 29).

[199] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. "Direct preference optimization: Your language model is secretly a reward model". In: *NeurIPS*. 2024 (cit. on pp. 80, 83).

[200] Roberta Raileanu, Emily L. Denton, Arthur Szlam, and Rob Fergus. "Modeling Others using Oneself in Multi-Agent Reinforcement Learning". In: *ICML*. 2018 (cit. on p. 80).

[201] Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. "Is Reinforcement Learning (Not) for Natural Language Processing: Benchmarks, Baselines, and Building Blocks for Natural Language Policy Optimization". In: *ICLR*. 2023 (cit. on pp. 79, 80, 83).

[202] Sachin Ravi and Hugo Larochelle. "Optimization as a model for few-shot learning". In: *ICLR*. 2017 (cit. on p. 42).

[203] Adrià Recasens, Jason Lin, João Carreira, Drew Jaegle, Luyu Wang, Jean baptiste Alayrac, Pauline Luc, Antoine Miech, Lucas Smaira, Ross Hemsley, and Andrew Zisserman. "Zorro: the masked multimodal transformer". In: *arXiv preprint arXiv:2301.09595*. 2023 (cit. on pp. 45, 49, 134).

[204] Katrin Renz, Kashyap Chitta, Otniel-Bogdan Mercea, A. Sophia Koepke, Zeynep Akata, and Andreas Geiger. "PlanT: Explainable Planning Transformers via Object-Level Representations". In: *CoRL*. 2022 (cit. on p. 150).

[205] Alina Roitberg, Manuel Martinez, Monica Haurilet, and Rainer Stiefelhagen. "Towards a fair evaluation of zero-shot action recognition using external data". In: *ECCV Workshop*. 2018 (cit. on pp. 14, 15).

[206] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-resolution image synthesis with latent diffusion models". In: *CVPR*. 2022 (cit. on p. 43).

[207] Bernardino Romera-Paredes and Philip Torr. "An embarrassingly simple approach to zero-shot learning". In: *ICML*. 2015 (cit. on pp. 12, 14, 55).

[208] Andrew Rouditchenko, Angie Boggust, David Harwath, Dhiraj Joshi, Samuel Thomas, Kartik Audhkhasi, Rogerio Feris, Brian Kingsbury, Michael Picheny, Antonio Torralba, et al. "AVLnet: Learning Audio-Visual Language Representations from Instructional Videos". In: *Interspeech*. 2020 (cit. on p. 67).

[209] Aniket Roy, Anshul Shah, Ketul Shah, Anirban Roy, and Rama Chellappa. "DiffAlign: Few-shot learning using diffusion based synthesis and alignment". In: *arXiv preprint arXiv:2212.05404*. 2022 (cit. on p. 42).

[210] Divya Saxena and Jiannong Cao. "Generative adversarial networks (GANs) challenges, solutions, and future directions". In: *ACM Computing Surveys (CSUR)*. 2021 (cit. on p. 43).

[211] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. "Generalized zero-and few-shot learning via aligned variational autoencoders". In: *CVPR*. 2019 (cit. on pp. 12, 14).

[212] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347*. 2017 (cit. on pp. 79, 80, 82).

[213] Iulian Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. "Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models". In: *AAAI*. 2015 (cit. on p. 80).

[214] Ying Sheng, Shiyi Cao, Dacheng Li, Coleman Hooper, Nicholas Lee, Shuo Yang, Christopher Chou, Banghua Zhu, Lianmin Zheng, Kurt Keutzer, Joseph E. Gonzalez, and Ion Stoica. "S-LoRA: Serving Thousands of Concurrent LoRA Adapters". In: *arXiv preprint arXiv:2311.03285*. 2023 (cit. on p. 80).

[215] Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. "What does clip know about a red circle? visual prompt engineering for vlms". In: *ICCV*. 2023 (cit. on pp. 84, 90).

[216] Charles Burton Snell, Ilya Kostrikov, Yi Su, Mengjiao Yang, and Sergey Levine. "Offline RL for Natural Language Generation with Implicit Language Q Learning". In: *arXiv preprint arXiv:2206.11871*. 2022 (cit. on p. 80).

[217] Jake Snell, Kevin Swersky, and Richard Zemel. "Prototypical networks for few-shot learning". In: *NeurIPS*. 2017 (cit. on p. 42).

[218] Haoyu Song, Weinan Zhang, Jingwen Hu, and Ting Liu. "Generating Persona Consistent Dialogues by Exploiting Natural Language Inference". In: *AAAI*. 2019 (cit. on p. 80).

[219] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. "UCF101: A dataset of 101 human actions classes from videos in the wild". In: *arXiv preprint arXiv:1212.0402*. 2012 (cit. on pp. 13, 19, 44).

[220] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting". In: *JMLR*. 2014 (cit. on pp. 16, 30, 57, 70).

[221] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. "Learning to summarize from human feedback". In: *NeurIPS*. 2020 (cit. on p. 82).

[222] Kun Su, Xiulong Liu, and Eli Shlizerman. "Multi-Instrumentalist Net: Unsupervised Generation of Music from Body Movements". In: *arXiv preprint arXiv:2012.03478*. 2020 (cit. on pp. 14, 28, 42, 55).

[223] Kun Su, Xiulong Liu, and Eli Shlizerman. "How Does it Sound?" In: *NeurIPS*. 2021 (cit. on pp. 14, 55).

[224] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. "Vl-bert: Pre-training of generic visual-linguistic representations". In: *arXiv preprint arXiv:1908.08530*. 2019 (cit. on p. 29).

[225] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. "Learning video representations using contrastive bidirectional transformer". In: *arXiv preprint arXiv:1906.05743*. 2019 (cit. on p. 29).

[226] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. "Videobert: A joint model for video and language representation learning". In: *ICCV*. 2019 (cit. on p. 29).

[227] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. "Learning to compare: Relation network for few-shot learning". In: *CVPR*. 2018 (cit. on p. 42).

[228] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. "Lst: Ladder side-tuning for parameter and memory efficient transfer learning". In: *NeurIPS*. 2022 (cit. on p. 80).

[229] Ece Takmaz, Nicolo' Brandizzi, Mario Giulianelli, Sandro Pezzelle, and Raquel Fern'andez. "Speaking the Language of Your Listener: Audience-Aware Adaptation via Plug-and-Play Theory of Mind". In: *ACL*. 2023 (cit. on p. 80).

[230] Hao Tan and Mohit Bansal. "Lxmert: Learning cross-modality encoder representations from transformers". In: *EMNLP*. 2019 (cit. on p. 29).

[231] Senem Tanberk, Zeynep Hilal Kilimci, Dilek Bilgin Tükel, Mitat Uysal, and Selim Akyokuş. "A hybrid deep model using deep learning and dense optical flow approaches for human activity recognition". In: *IEEE Access*. 2020 (cit. on p. 66).

[232] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. "Fourier features let networks learn high frequency functions in low dimensional domains". In: *NeurIPS*. 2020 (cit. on pp. 30, 46).

[233] Chameleon Team. "Chameleon: Mixed-Modal Early-Fusion Foundation Models". In: *arXiv preprint arXiv:2405.09818*. 2024 (cit. on p. 78).

[234] Yapeng Tian, Dingzeyu Li, and Chenliang Xu. "Unified multisensory perception: weakly-supervised audio-visual video parsing". In: *ECCV*. 2020 (cit. on p. 15).

[235] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. "Audio-visual event localization in unconstrained videos". In: *ECCV*. 2018 (cit. on pp. 14, 28, 42, 55).

[236] Atousa Torabi, Niket Tandon, and Leonid Sigal. "Learning language-visual embedding for movie understanding with natural-language". In: *arXiv preprint arXiv:1609.08124*. 2016 (cit. on p. 67).

[237] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. "Learning spatiotemporal features with 3d convolutional networks". In: *ICCV*. 2015 (cit. on pp. 34, 43, 54, 56, 123, 131).

[238] Efthymios Tzinis, Scott Wisdom, Aren Jansen, Shawn Hershey, Tal Remez, Daniel PW Ellis, and John R Hershey. "Into the wild with audioscope: Unsupervised audio-visual separation of on-screen sounds". In: *ICLR*. 2021 (cit. on pp. 14, 55).

[239] Nicolas Usunier, David Buffoni, and Patrick Gallinari. "Ranking with ordered weighted pairwise classification". In: *ICML*. 2009 (cit. on p. 21).

[240] Arash Vahdat, Karsten Kreis, and Jan Kautz. "Score-based generative modeling in latent space". In: *NeurIPS*. 2021 (cit. on p. 43).

[241] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *NeurIPS*. 2017 (cit. on pp. 1, 16, 29, 31, 46, 54, 70).

[242] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. "Generalized zero-shot learning via synthesized examples". In: *CVPR*. 2018 (cit. on pp. 12, 14, 55).

[243] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. "Matching networks for one shot learning". In: *NeurIPS*. 2016 (cit. on p. 42).

[244] Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, and James Hays. "Composing text and image for image retrieval-an empirical odyssey". In: *CVPR*. 2019 (cit. on pp. 67–69, 73, 74, 142).

[245] Harm de Vries, Florian Strub, A. P. Sarath Chandar, Olivier Pietquin, H. Larochelle, and Aaron C. Courville. "GuessWhat?! Visual Object Discovery through Multimodal Dialogue". In: *CVPR*. 2016 (cit. on p. 80).

[246] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. "The caltech-ucsd birds-200-2011 dataset". In: *California Institute of Technology*. 2011 (cit. on pp. 28, 79, 84).

[247] Hualiang Wang, Yi Li, Huifeng Yao, and Xiaomeng Li. "Clipn for zero-shot ood detection: Teaching clip to say no". In: *ICCV*. 2023 (cit. on p. 55).

[248] Jian Wang, Chak Tou Leong, Jiashuo Wang, Dongding Lin, Wenjie Li, and Xiao-Yong Wei. "Instruct Once, Chat Consistently in Multiple Rounds: An Efficient Tuning Framework for Dialogue". In: *arXiv preprint arXiv:2402.06967*. 2024 (cit. on p. 80).

[249] Qian Wang and Ke Chen. "Zero-shot visual recognition via bidirectional latent embedding". In: *IJCV*. 2017 (cit. on pp. 14, 15).

[250] Xiaohan Wang, Linchao Zhu, and Yi Yang. "T2vlad: global-local sequence alignment for text-video retrieval". In: *CVPR*. 2021 (cit. on pp. 29, 42).

[251] Xiaoyang Wang and Qiang Ji. "A unified probabilistic approach modeling relationships between attributes and objects". In: *ICCV*. 2013 (cit. on p. 68).

[252] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. "Vatex: A large-scale, high-quality multilingual dataset for video-and-language research". In: *ICCV*. 2019 (cit. on pp. 67, 139).

[253] Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. "Simpleshot: Revisiting nearest-neighbor classification for few-shot learning". In: *arXiv preprint arXiv:1911.04623*. 2019 (cit. on p. 42).

[254] Yang Wang and Greg Mori. "A discriminative latent model of object classes and attributes". In: *ECCV*. 2010 (cit. on p. 68).

[255] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. "Low-shot learning from imaginary data". In: *CVPR*. 2018 (cit. on p. 42).

[256] Olivia Wiles, A. Sophia Koepke, and Andrew Zisserman. "X2face: A network for controlling face generation using images, audio, and pose codes". In: *ECCV*. 2018 (cit. on pp. 14, 28).

[257] Michael Wray and Dima Damen. "Learning visual actions using multiple verb-only labels". In: *BMVC*. 2019 (cit. on p. 68).

[258] Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen. "Fine-grained action retrieval through multiple parts-of-speech embeddings". In: *ICCV*. 2019 (cit. on pp. 67, 68).

[259] Jeff Wu, Long Ouyang, Daniel M. Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul F. Christiano. "Recursively Summarizing Books with Human Feedback". In: *arXiv preprint arXiv:2109.10862*. 2021 (cit. on p. 83).

[260] Meiliu Wu and Qunying Huang. "IM2City: image geo-localization via multi-modal learning". In: *ACM SIGSPATIAL Workshop*. 2022 (cit. on p. 55).

[261] Wenhao Wu, Haipeng Luo, Bo Fang, Jingdong Wang, and Wanli Ouyang. "Cap4Video: What Can Auxiliary Captions Do for Text-Video Retrieval?" In: *CVPR*. 2023 (cit. on p. 68).

[262] Yichao Wu, Yafei Xiang, Shuning Huo, Yulu Gong, and Penghao Liang. "LoRA-SP: Streamlined Partial Parameter Adaptation for Resource-Efficient Fine-Tuning of Large Language Models". In: *International Conference on Algorithms, Microchips and Network Applications*. 2024 (cit. on p. 80).

[263] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. "Latent embeddings for zero-shot classification". In: *CVPR*. 2016 (cit. on pp. 55, 140).

[264] Yongqin Xian, Bruno Korbar, Matthijs Douze, Lorenzo Torresani, Bernt Schiele, and Zeynep Akata. "Generalized Few-Shot Video Classification with Video Retrieval and Feature Generation". In: *IEEE TPAMI*. 2021 (cit. on pp. 41–45, 49, 51, 134).

[265] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. "Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly". In: *IEEE TPAMI*. 2018 (cit. on pp. 14, 20, 28, 34, 59).

[266] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. "Feature generating networks for zero-shot learning". In: *CVPR*. 2018 (cit. on pp. 14, 28, 55).

[267] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. "f-vaegan-d2: A feature generating framework for any-shot learning". In: *CVPR*. 2019 (cit. on pp. 14, 20, 21, 28, 42, 43).

[268] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. "Audiovisual slowfast networks for video recognition". In: *arXiv preprint arXiv:2001.08740*. 2020 (cit. on pp. 14, 40, 41, 55).

[269] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. "Sun database: Large-scale scene recognition from abbey to zoo". In: *CVPR*. 2010 (cit. on p. 28).

[270] Guo-Sen Xie, Li Liu, Xiaobo Jin, Fan Zhu, Zheng Zhang, Jie Qin, Yazhou Yao, and Ling Shao. "Attentive region embedding network for zero-shot learning". In: *CVPR*. 2019 (cit. on p. 28).

[271] Huang Xie, Okko Räsänen, and Tuomas Virtanen. "Zero-Shot Audio Classification with Factored Linear and Nonlinear Acoustic-Semantic Projections". In: *ICASSP*. 2021 (cit. on p. 15).

[272] Huang Xie and Tuomas Virtanen. "Zero-Shot Audio Classification via Semantic Embeddings". In: *TASLP*. 2021 (cit. on p. 15).

[273] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. "Empirical evaluation of rectified activations in convolutional network". In: *arXiv preprint arXiv:1505.00853*. 2015 (cit. on p. 70).

[274] Haoming Xu, Runhao Zeng, Qingyao Wu, Mingkui Tan, and Chuang Gan. "Cross-modal relation-aware networks for audio-visual event localization". In: *ACM MM*. 2020 (cit. on pp. 14, 55).

[275] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. "Msr-vtt: A large video description dataset for bridging video and language". In: *CVPR*. 2016 (cit. on pp. 67, 139).

[276] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. "A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model". In: *ECCV*. 2022 (cit. on pp. 55, 65).

[277] Ran Xu, Caiming Xiong, Wei Chen, and Jason Corso. "Jointly modeling deep video and compositional text to bridge vision and language in a unified framework". In: *AAAI*. 2015 (cit. on pp. 67, 68).

[278] Wenjia Xu, Yongqin Xian, Jiuniu Wang, Bernt Schiele, and Zeynep Akata. "Attribute prototype network for zero-shot learning". In: *NeurIPS*. 2020 (cit. on pp. 12, 14, 20, 21, 28, 122).

[279] Wenjia Xu, Yongqin Xian, Jiuniu Wang, Bernt Schiele, and Zeynep Akata. "VGSE: Visually-Grounded Semantic Embeddings for Zero-Shot Learning". In: *CVPR*. 2022 (cit. on p. 14).

[280] Xun Xu, Timothy M Hospedales, and Shaogang Gong. "Multi-task zero-shot action recognition with prioritised data augmentation". In: *ECCV*. 2016 (cit. on pp. 12, 15).

[281] Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhensu Chen, Xiaopeng Zhang, and Qi Tian. "QA-LoRA: Quantization-Aware Low-Rank Adaptation of Large Language Models". In: *arXiv preprint arXiv:2309.14717*. 2023 (cit. on p. 80).

[282] Hanyu Xuan, Zhenyu Zhang, Shuo Chen, Jian Yang, and Yan Yan. "Cross-modal attention network for temporal inconsistent audio-visual event localization". In: *AAAI*. 2020 (cit. on p. 15).

[283] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. "Few-shot learning via embedding adaptation with set-to-set functions". In: *CVPR*. 2020 (cit. on p. 42).

[284] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. "Coca: Contrastive captioners are image-text foundation models". In: *Transactions on Machine Learning Research (TMLR)*. 2022 (cit. on p. 55).

[285] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models". In: *arXiv preprint arXiv:2106.10199*. 2021 (cit. on p. 95).

[286] Hongguang Zhang, Li Zhang, Xiaojuan Qi, Hongdong Li, Philip HS Torr, and Piotr Koniusz. "Few-shot action recognition with permutation-invariant attention". In: *ECCV*. 2020 (cit. on p. 41).

[287] Longteng Zhang, Lin Zhang, Shaohuai Shi, Xiaowen Chu, and Bo Li. "LoRA-FA: Memory-efficient Low-rank Adaptation for Large Language Models Fine-tuning". In: *arXiv preprint arXiv:2308.03303*. 2023 (cit. on p. 80).

[288] Yi-Kai Zhang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. "Audio-Visual Generalized Few-Shot Learning with Prototype-Based Co-Adaptation". In: *Interspeech*. 2022 (cit. on p. 41).

[289] Yizhe Zhang, Xiang Gao, Sungjin Lee, Chris Brockett, Michel Galley, Jianfeng Gao, and William B. Dolan. "Consistent Dialogue Generation with Self-supervised Feature Learning". In: *arXiv preprint arXiv:1903.05759*. 2019 (cit. on p. 80).

[290] Hang Zhao, Chuang Gan, Wei-Chiu Ma, and Antonio Torralba. "The sound of motions". In: *ICCV*. 2019 (cit. on pp. 14, 55).

[291] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. "The sound of pixels". In: *ECCV*. 2018 (cit. on pp. 14, 55).

[292] Hanxun Zhong, Zhicheng Dou, Yutao Zhu, Hongjin Qian, and Ji rong Wen. "Less is More: Learning to Refine Dialogue History for Personalized Dialogue Generation". In: *arXiv preprint arXiv:2204.08128*. 2022 (cit. on p. 80).

[293] Hang Zhou, Ziwei Liu, Xudong Xu, Ping Luo, and Xiaogang Wang. "Vision-infused deep audio inpainting". In: *ICCV*. 2019 (cit. on pp. 14, 28, 42, 55).

[294] Luowei Zhou, Chenliang Xu, and Jason Corso. "Towards automatic learning of procedures from web instructional videos". In: *AAAI*. 2018 (cit. on p. 67).

[295] Ziqin Zhou, Yinjie Lei, Bowen Zhang, Lingqiao Liu, and Yifan Liu. "Zegclip: Towards adapting clip for zero-shot semantic segmentation". In: *CVPR*. 2023 (cit. on pp. 55, 65).

[296] Linchao Zhu and Yi Yang. "Compound memory networks for few-shot video classification". In: *ECCV*. 2018 (cit. on pp. 41, 42).

[297] Linchao Zhu and Yi Yang. "Actbert: Learning global-local video-text representations". In: *CVPR*. 2020 (cit. on p. 67).

[298] Lingyu Zhu and Esa Rahtu. "V-slowfast network for efficient visual sound separation". In: *WACV*. 2022 (cit. on pp. 14, 55).

[299] Xizhou Zhu, Jinguo Zhu, Hao Li, Xiaoshi Wu, Hongsheng Li, Xiaohua Wang, and Jifeng Dai. "Uni-perceiver: Pre-training unified architecture for generic perception for zero-shot and few-shot tasks". In: *CVPR*. 2022 (cit. on p. 55).

[300] Yi Zhu, Yang Long, Yu Guan, Shawn Newsam, and Ling Shao. "Towards universal representation for unseen action recognition". In: *CVPR*. 2018 (cit. on p. 15).

[301] Yizhe Zhu, Mohamed Elhoseiny, Bingchen Liu, Xi Peng, and Ahmed Elgammal. "A generative adversarial approach for zero-shot learning from noisy texts". In: *CVPR*. 2018 (cit. on p. 14).

[302] Yizhe Zhu, Jianwen Xie, Bingchen Liu, and Ahmed Elgammal. "Learning feature-to-feature translator by alternating back-propagation for generative zero-shot learning". In: *ICCV*. 2019 (cit. on pp. 14, 28).

[303] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. "Cross-task weakly supervised learning from instructional videos". In: *CVPR*. 2019 (cit. on p. 68).

[304] Daniel M. Ziegler, Nisan Stiennon, Jeff Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. "Fine-Tuning Language Models from Human Preferences". In: *arXiv preprint arXiv:1909.08593*. 2019 (cit. on p. 80).

A

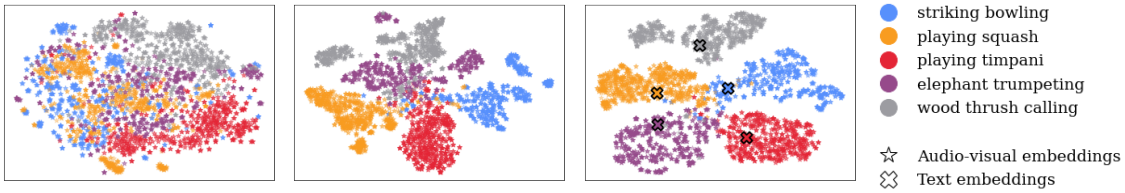
Audio-visual Generalised Zero-shot Learning with Cross-modal Attention and Language

In this supplementary material, we include additional qualitative results (Sec. A.1) and quantitative results (Sec. A.2) for our proposed audio-visual (G)ZSL framework.

A.1 Additional Qualitative Results

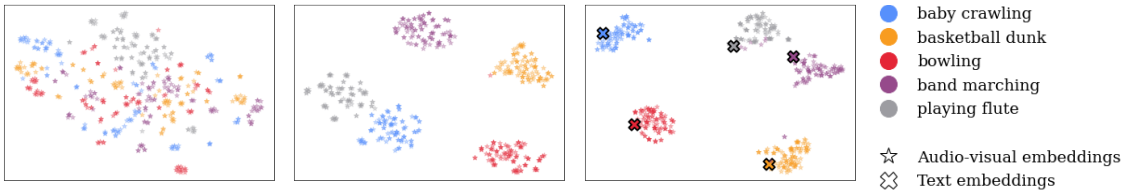
We provide additional qualitative results for our proposed AVCA model for the tasks of audio-visual GZSL and ZSL. We present t-SNE visualisations for the learnt audio-visual embeddings on the VGGSound-GZSL and UCF-GZSL datasets in Fig. A.1 and Fig. A.2.

In Fig. A.1a, we can observe that the input audio features do not demonstrate a clear separation between the visualised classes for the VGGSound-GZSL dataset. The visual features exhibit a better clustering as can been seen in Fig. A.1b. However, the visual features also include classes, such as *elephant trumpeting* and *wood thrush calling*, that are not clustered cleanly. Our AVCA model outputs multi-modal features that improve the clustering for both, seen and unseen classes (Fig. A.1c). The learnt features for the two



(a) Input audio embeddings (b) Input visual embeddings (c) Learnt audio-visual embeddings

Figure A.1: t-SNE visualisation for three seen (*striking bowling*, *playing squash*, *playing timpani*) and two unseen (*elephant trumpeting*, *wood thrush calling*) test classes from the VGGSound-GZSL dataset, showing (a) audio and (b) visual features extracted with SeLaVi [21], and (c) learnt audio-visual embeddings of our model. Textual class label embeddings are visualised with a cross.



(a) Input audio embeddings (b) Input visual embeddings (c) Learnt audio-visual embeddings

Figure A.2: t-SNE visualisation for three seen (*baby crawling*, *basketball dunk*, *bowling*) and two unseen (*band marching*, *playing flute*) test classes from the UCF-GZSL dataset, showing (a) audio and (b) visual features extracted with SeLaVi [21], and (c) learnt audio-visual embeddings of our model. Textual class label embeddings are visualised with a cross.

Method type	Model	VGGSound-GZSL^{cls}				UCF-GZSL^{cls}				ActivityNet-GZSL^{cls}			
		S	U	HM	ZSL	S	U	HM	ZSL	S	U	HM	ZSL
ZSL	ALE [9]	26.13	1.72	3.23	4.97	45.42	29.09	35.47	32.30	0.89	6.16	1.55	6.16
	SJE [10]	16.94	2.72	4.69	3.22	19.39	32.47	24.28	32.47	37.92	1.22	2.35	4.35
	DEVISE [73]	29.96	1.94	3.64	4.72	29.58	34.80	31.98	35.48	0.17	5.84	0.33	5.84
	APN [278]	6.46	6.13	6.29	6.50	13.54	28.44	18.35	29.69	3.79	3.39	3.58	3.97
Audio-visual ZSL	CJME [185]	10.86	2.22	3.68	3.72	33.89	24.82	28.65	29.01	10.75	5.55	7.32	6.29
	AVGZSLNet [152]	15.02	3.19	5.26	4.81	74.79	24.15	36.51	31.51	13.70	5.96	8.30	6.39
	AVCA	12.63	6.19	**8.31**	**6.91**	63.15	30.72	**41.34**	**37.72**	16.77	7.04	**9.92**	**7.58**

Table A.1: Evaluating AVCA and state-of-the-art (G)ZSL methods for audio-visual GZSL and ZSL on the VGGSound, UCF, and ActivityNet (G)ZSL^{cls} benchmarks using features extracted from audio/video classification networks. We report the mean class accuracy on the seen (S) and unseen (U) test classes, and their harmonic mean (HM) for GZSL performance. The ZSL performance is evaluated on the test subset of samples from unseen classes.

unseen classes *elephant trumpeting* and *wood thrush calling* are clustered and well-separated as opposed to the input features. This is impressive, since both classes were not included in the training set.

Similarly, for the UCF-GZSL dataset, we can observe in Fig. A.2a that the input audio features are not grouped according to classes. In contrast, the visual input embeddings mostly exhibit a clear clustering of different classes. However, the classes *baby crawling* and *playing flute* are not well-separated as can be seen in Fig. A.2b. This improves through learning, since the learnt audio-visual features in Fig. A.2c show a clear divide between those two classes. In addition to that, the output embeddings for the unseen classes *band marching* and *playing flute* are overwhelmingly clustered well, too.

To summarise, our model learns to cluster both seen and unseen classes for different datasets by transferring information from the training data to unseen classes at test time.

Dataset	# classes				# videos
	all	tr	v(U)	ts(U)	ts(U)
VGGSound-GZSL^{cls}	271	138	69	64	3200
UCF-GZSL^{cls}	48	30	12	6	845
ActivityNet-GZSL^{cls}	198	99	51	48	4052

Table A.2: Statistics for our VGGSound, UCF, and ActivityNet (G)ZSL^{cls} datasets, showing the number (#) of classes in our splits (tr: train, v: validation, ts: test; S: seen, U: unseen). ^{cls} indicates the dataset splits that allow to use VGGish features pre-trained on YouTube-8M. The full details about the dataset splits can be found at https://github.com/ExplainableML/AVCA-GZSL.

A.2 Additional Quantitative Results

In this section, we provide additional quantitative results obtained with our AVCA. We present results for training and evaluating our AVCA model with a different set of input features in Sec. A.2.1. In particular, we use features extracted from networks that were pretrained for audio and video classification. We perform an additional ablation study that gradually transforms AVCA into AVGZSLNet [152] in Sec. A.2.2. Complete results that include the U and S performance for Tab. 2.3 in the main paper are provided in Sec. A.2.3. Finally, we give details about the number of parameters and GFLOPS required for training our AVCA model in Sec. A.2.4

A.2.1 Using features extracted audio/video classification networks

We additionally trained and tested our model and the baseline models using features extracted from audio and video classification networks (instead of the SeLaVi [21] features used in the main paper). In particular, the visual features were extracted with C3D [237], pretrained for video classification on Sports1M [105]. The audio features were extracted with VGGish [92], pretrained for audio classification on Youtube-8M [1]. We averaged the extracted features across time, resulting in a 4096-dimensional visual feature and a 128-dimensional audio feature for each video.

However, to use the audio features extracted from a network that was pretrained on Youtube-8M, we removed the test unseen classes from the VGGSound-GZSL, UCF-GZSL, and ActivityNet-GZSL datasets that had an overlap with Youtube-8M. This resulted in slightly different dataset splits (VGGSound-GZSL^{cls}, UCF-GZSL^{cls}, and ActivityNet-GZSL^{cls}) detailed in Tab. A.2.

We provide results for training and evaluating our AVCA and the baselines using audio and video classification features in Tab. A.1. AVCA outperforms all the baselines on all three datasets. On VGGSound-GZSL^{cls}, ACVA obtains a HM of 8.31% and ZSL of 6.91% compared to a HM of 6.29% for APN and a ZSL performance of 6.50% for APN. On UCF-GZSL^{cls}, AVCA obtains a HM of 41.34% and a ZSL of 37.72% compared to a HM of 36.51% for AVGZSLNet and a ZSL performance of 35.48% for DEVISE. On

Model	VGGSound-GZSL		UCF-GZSL		ActivityNet-GZSL	
	HM	ZSL	HM	ZSL	HM	ZSL
AVGZSLNet [152]	5.83	5.28	18.05	13.65	6.44	5.40
W/o x-att	6.02	4.81	26.82	18.37	6.50	5.64
W x-att with l_c loss	4.88	4.55	19.38	12.95	11.58	8.40
AVCA	**6.31**	**6.00**	**27.15**	**20.01**	**12.13**	**9.13**

Table A.3: Ablation that gradually transforms our AVCA model into AVGZSLNet [152]. W/o x-att optimises each branch in isolation and their output predictions are averaged. x-att denotes cross-attention. l_c loss is the loss function used to train AVGZSLNet.

Model	VGGSound-GZSL			UCF-GZSL			ActivityNet-GZSL		
	S	U	HM	S	U	HM	S	U	HM
Visual branch	7.02	3.68	4.83	50.18	13.21	20.92	11.80	5.53	7.53
Audio branch	7.74	2.55	3.84	12.99	10.78	11.78	4.56	3.87	4.19
AVCA	14.90	4.00	**6.31**	51.53	18.43	**27.15**	24.86	8.02	**12.13**

Table A.4: Influence of *training* AVCA with different modalities for GZSL on the VGGSound-GZSL, UCF-GZSL and ActivityNet-GZSL datasets measuring the GZSL performance on seen (S) and unseen (U) test classes and their harmonic mean (HM). Using both modalities yields the strongest GZSL performances.

ActivityNet-GZSL^{cls}, AVCA outperforms AVGZSLNet with a HM of 9.92% compared to 8.30% and a ZSL of 7.58% compared to 6.39% for AVGZSLNet. These results show that AVCA outperforms the other competitors also when using audio and video classification features, proving again that our cross-attention mechanism and training objective provide a boost in performance.

A.2.2 Ablating AVCA in relation to AVGZSLNet

We additionally perform an ablation study that gradually transforms the AVCA model into AVGZSLNet [152] in Tab. A.3. We show how our model components influence the (G)ZSL performance, resulting in our AVCA model that outperforms AVGZSLNet on all three datasets. For this ablation, we use the SeLaVi [21] features and the same setup as in the main paper. W/o x-att corresponds to AVGZSLNet trained with our loss function (without our cross-attention). It can be observed that W/o x-att provides improvements on UCF-GZSL, with a HM of 26.82% compared to 18.05% and a ZSL performance of 18.37% compared to 13.65%. W x-att with l_c loss corresponds to AVGZSLNet with cross-attention and with the loss function proposed for AVGZSLNet. In this case, it can be observed that the cross-attention improves the results over AVGZSLNet with a HM of 11.58% compared to 6.44% and ZSL performance of 8.40% compared to 5.40% on ActivityNet-GZSL. These improvements can also be observed on the other datasets, showing that our novel loss and our cross-attention mechanism improve the performance over AVGZSLNet.

A.2.3 Extended results for training AVCA with different modalities

In this section, we extend the ablation study that uses different modalities for training (Tab. 2.3 in the main paper) by adding the performance on the seen (S) and unseen (U) test classes for all the datasets in Tab. A.4.

On all three datasets it can be observed that there is an increase in both seen and unseen performance when using AVCA compared to using the Visual branch or the Audio branch. On VGGSound-GZSL, we can observe that the S performance for AVCA is 14.90% compared to 7.74% for the Visual branch. The U performance on VGGSound-GZSL is also stronger for AVCA than for the Visual branch, with a score of 4.00% compared to 3.68%. On the UCF-GZSL dataset, the S performance increases only slightly, from 50.18% for the Visual branch to 51.53% for AVCA. However, there is a significant increase in the U performance, from 13.21% for the Visual branch to 18.43% for AVCA. Finally, on ActivityNet-GZSL, AVCA yields a S score of 24.86% compared to 11.80% for the Visual branch. The U performance increases from 5.53% for the Visual branch to 8.02% for AVCA. These results show that the S/U performance increases significantly when using AVCA compared to the Visual branch or the Audio branch, leading to better HM/ZSL performances.

A.2.4 Number of parameters in AVCA.

AVCA contains 1.69M parameters in total, which is comparable to the 1.32M parameters used in AVGZSLNet [152]. ALE/SJE/DEVISE are significantly smaller with only 307.2k parameters. AVCA has a computational complexity of 2.36 GFLOPS, while AVGZSLNet has a computational complexity of 1.38 GFLOPS. Again, the fewest GFLOPS are required for ALE/SJE/DEVISE which have a computational complexity of 0.32 GFLOPS. These statistics show that AVCA is comparable to AVGZSLNet while providing significantly better results on all three datasets.

B

Temporal and cross-modal attention for audio-visual zero-shot learning

In the supplementary material, we provide additional details about baselines (Sec. B.1), and present further model ablations (Sec. B.2). Additionally, we study t-SNE visualisations for TCAF and [157] (Sec. B.3), and provide a comparison of the computational complexity of TCAF and some of the baselines (Sec. B.4).

B.1 Additional details about baselines

In the following, we detail our adaptations of Attention Fusion [72] and of the Perceiver [101] to the (G)ZSL setting (which we briefly summarised in Sec. 3.4.2 of our manuscript).

B.1.1 Attention Fusion

In order to use Attention Fusion [72] in the (G)ZSL setting, we take the same temporal audio and visual features as inputs as TCAF. Following TCAF, we embed the input features into the same feature dimension using A_{enc} and V_{enc}. Instead of directly mapping to the number of classes, as the authors originally proposed, A_{enc} and V_{enc} map the features to $\mathbb{R}^{d_{dim}}$. The embedded features are then temporally averaged to obtain a single $d_{dim}-$dimensional feature vector for each modality. The attention weight α, which is used for fusing both modalities, is computed using the channel-wise concatenation of the audio and visual embeddings through a linear layer $f_{attn} : \mathbb{R}^{2*d_{dim}} \rightarrow \mathbb{R}^{d_{dim}}$, followed by a sigmoid function. Both modalities are then fused to create the output token o_c through $o_c = \alpha \odot \phi_{a,avg} + (1 - \alpha) \odot \phi_{v,avg}$, where $\phi_{a,avg}$ and $\phi_{v,avg}$ are the temporally averaged audio and visual features. o_c is then projected using the same projection function O_{proj}, decoder D_o, and text embedding projections as in TCAF. We train Attention Fusion using the same learning rate and loss functions as TCAF.

B.1.2 Perceiver

The Perceiver [101] takes the same audio and visual features as input as TCAF. For consistency between frameworks, we again embedded the input features to the same feature dimension using A_{enc} and V_{enc}, and equip both TCAF and the Perceiver with the same temporal and modality information by adding positional embeddings as described in the main paper. Our goal was to directly compare our cross-attention mechanism with the Perceiver attention. Therefore, we adapted the cross-attention, self-attention and dense layer blocks of the Perceiver to use the same internal dimensions as TCAF. We also added a dropout layer at the end of dense layer blocks to match the dense blocks in TCAF. For the randomly initialised latent array, we use 64 latent tokens with dimension $\mathbb{R}^{d_{dim}}$ for all datasets. Increasing the number of latent tokens did not provide a boost in performance, but significantly increased the computational costs. One of the latent tokens is used as the output classification token c_o. We use one cross-attention block and one self-attention block per layer without weight sharing and use the same number of layers as TCAF. This results in just a slightly higher number of parameters for the Perceiver than for our TCAF. The output token c_o is projected using the projection function O_{proj} and the decoder D_o. The computations for the text embeddings are analogous to TCAF. We train the Perceiver using the same learning rate and loss functions as our model.

B.2 Additional model ablations

In this section, we first study the impact of using temporal embeddings (Sec. B.2.1) and of the number and design of the cross-attention layers in TCAF (Sec. B.2.2). Next, we evaluate the impact on performance when adding noise to the audio modality (Sec. B.2.3). Finally, we present results of transforming TCAF to [157] (Sec. B.2.4).

B.2.1 Influence of using temporal information

In the following, we investigate the influence of using temporal information when learning multi-modal video representation for (G)ZSL with TCAF. Since the operations in our audio-visual transformer layers (cf. Sec. 3.3.2 in the manuscript) are invariant to permutation, the feature tokens are additionally equipped with temporal information through the addition of positional embeddings pos_t. Without temporal embeddings, the model is unable to put data from one time step in temporal relation to information from the other time steps. Temporal embeddings therefore allow the model to understand the concept of time.

Tab. B.1 shows results for training and evaluating TCAF with (+) and without (−) temporal embeddings (pos_t). The highest harmonic mean is achieved when using temporal embeddings. For instance for ActivityNet-GZSL^{cls}, our model that does not use temporal embeddings ($-pos_t$) obtains only a HM of 8.69% and a ZSL score of 5.53%, compared to a HM of 12.20% and a ZSL score of 7.96% when using temporal embeddings.

Positional embeddings	VGGSound-GZSL^{cls}				UCF-GZSL^{cls}				ActivityNet-GZSL^{cls}			
	S	U	HM	ZSL	S	U	HM	ZSL	S	U	HM	ZSL
$-pos_t$	**15.78**	4.66	7.19	4.97	27.35	26.02	26.67	28.06	21.80	5.43	8.69	5.53
$+pos_t$ (TCAF)	12.63	**6.72**	**8.77**	**7.41**	**67.14**	**40.83**	**50.78**	**44.64**	**30.12**	**7.65**	**12.20**	**7.96**

Table B.1: Influence of temporal information provided through positional embeddings (pos_t) on the (G)ZSL performance on the VGGSound-GZSL^{cls}, UCF-GZSL^{cls}, and ActivityNet-GZSL^{cls} datasets.

Layer configurations	VGGSound-GZSL^{cls}				UCF-GZSL^{cls}				ActivityNet-GZSL^{cls}			
	S	U	HM	ZSL	S	U	HM	ZSL	S	U	HM	ZSL
1 layer w/o FF	19.70	4.47	7.29	4.66	63.30	26.45	37.31	27.85	15.10	4.59	7.04	4.63
1 layer	**17.95**	4.78	7.55	5.13	40.07	29.40	33.92	29.74	28.22	4.85	8.27	4.89
1/2*(all layers) w/o FF	11.33	4.25	6.18	4.59	38.72	23.17	28.99	23.28	8.13	3.35	4.75	3.40
1/2*(all layers)	12.08	4.69	6.75	5.12	**77.19**	30.18	43.40	34.18	28.65	6.04	9.98	6.25
1/2*(all layers) $+A_{self}$	14.62	4.56	6.96	4.97	53.05	34.83	42.05	35.84	**31.38**	5.93	9.97	6.51
all layers w/o FF	14.41	4.28	6.60	4.59	32.57	25.77	28.78	28.86	7.44	3.27	4.54	3.33
all layers	12.63	**6.72**	**8.77**	**7.41**	67.14	**40.83**	**50.78**	**44.64**	30.12	**7.65**	**12.20**	**7.96**

Table B.2: Varying the number of cross-attention layers in TCAF and the use of feed forward (FF) functions in the cross-attention layers.

Similar observations can be made for VGGSound-GZSL^{cls} and UCF-GZSL^{cls}, showing the importance of temporal information for learning strong video representations.

B.2.2 Impact of using different amounts of cross-attention layers and of varying the cross-attention layer design

In Tab. B.2, we present ablations on the number of cross-attention layers used in our model. Furthermore, we investigate the relevance of using feed forward functions (FF) in our cross-attention layers.

For TCAF, we used 8 cross-attention layers on VGGSound-GZSL^{cls} (all layers). On the UCF-GZSL^{cls} and ActivityNet-GZSL^{cls} datasets, we used 6 layers (all layers). We observe that using more layers is beneficial for GZSL and ZSL performance across all datasets. Moreover, we observe that, in general, eliminating the feed forward functions leads to a decrease in performance. Finally, using only half of the layers jointly with self-attention (1/2*(all layers) $+A_{self}$) leads to worse overall HM performance than using half of the layers without self-attention (1/2*(all layers)). This is in line with the experiments in the main paper, where adding the self-attention leads to worse results.

This ablation shows that using only cross-attention is beneficial even when using a different number of layers. Furthermore, using more cross-attention layers that are equipped with feed forward functions brings a boost in performance.

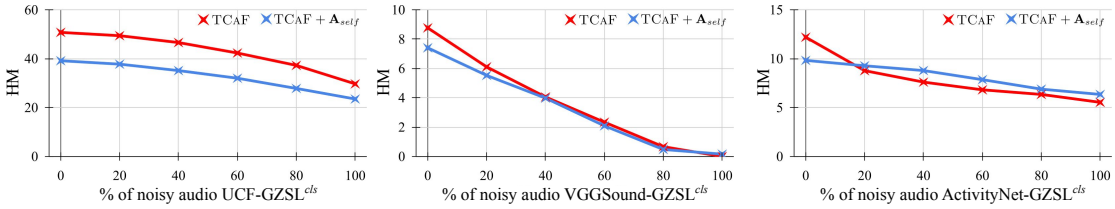


Figure B.1: Robustness of TCAF and TCAF +A_{self} to noise added to different proportions of the audio stream on UCF-GZSL^{cls}, VGGSound-GZSL^{cls} and ActivityNet-GZSL^{cls}.

B.2.3 Impact of noise in audio stream on GZSL performance

In this section, we study how the GZSL performance (HM) of TCAF decreases when noise is added to increasing temporal portions of the audio signal on all three datasets. We study both TCAF and TCAF +A_{self} in Fig. B.1. It can be observed that an increase in the proportion of noise leads to a decrease in the GZSL performance for both models on all three datasets. Furthermore, it can be observed that TCAF is significantly more robust to perturbations on UCF-GZSL^{cls} and slightly more robust on VGGSound-GZSL^{cls}. On the other hand, we can observe that on ActivityNet-GZSL^{cls} the trend is reversed, with TCAF +A_{self} being slightly more robust. Overall, it can be argued that TCAF is more robust across all three datasets than TCAF +A_{self}.

B.2.4 Transforming TCAF into [157]

Our TCAF builds on the AVCA [157] framework for audio-visual GZSL. To highlight the benefits of TCAF compared to AVCA, we show results for transforming TCAF into AVCA [157] in Tab. B.3.

TCAF exploits temporal information and obtains a HM performance of 8.77% on VGGSound-GZSL^{cls} compared to a HM of 7.65% (TCAF avg input) when using temporally averaged inputs. Moreover, TCAF uses an enhanced cross-modal attention to effectively gather multi-modal information. On the other hand, the attention mechanism of [157] uses temporally averaged feature inputs, which leads to a HM of 6.82% on VGGSound-GZSL^{cls} ([157]). Additionally, TCAF uses a single output branch and a classification token to aggregate the multi-modal information. In contrast, [157] uses two branches and no classification token which leads to a HM of 6.27% (w/o class. token) on VGGSound-GZSL^{cls}. Finally, our training objective avoids triplet losses, i.e. there is no overhead to train with positive and negative pairs. Using triplet losses similar to those used in [157] leads to a lower performance (TCAF + $l_{triplet}$) than TCAF. The same trend can be observed for the other datasets, proving that our architectural choices are more suitable for the audio-visual (G)ZSL task.

Model	VGGSound-GZSL^{cls}				UCF-GZSL^{cls}				ActivityNet-GZSL^{cls}			
	S	U	HM	ZSL	S	U	HM	ZSL	S	U	HM	ZSL
[157]	12.63	6.19	8.31	6.91	63.15	30.72	41.34	37.72	16.77	7.04	9.92	7.58
TCAF +att from [157]	10.08	5.16	6.82	5.41	39.47	28.85	33.33	29.79	5.58	2.37	3.33	2.43
TCAF avg input	11.69	5.69	7.65	6.16	12.00	20.46	15.13	20.59	16.43	3.26	5.44	3.42
w/o class. token	**18.40**	3.78	6.27	4.25	31.70	32.57	32.13	33.26	11.87	3.80	5.75	3.90
TCAF +$l_{triplet}$	14.51	4.78	7.19	5.06	**71.61**	35.91	47.83	40.00	18.74	6.58	9.74	6.63
TCAF	12.63	**6.72**	**8.77**	**7.41**	67.14	**40.83**	**50.78**	**44.64**	**30.12**	**7.65**	**12.20**	**7.96**

Table B.3: Transforming TCAF into [157]

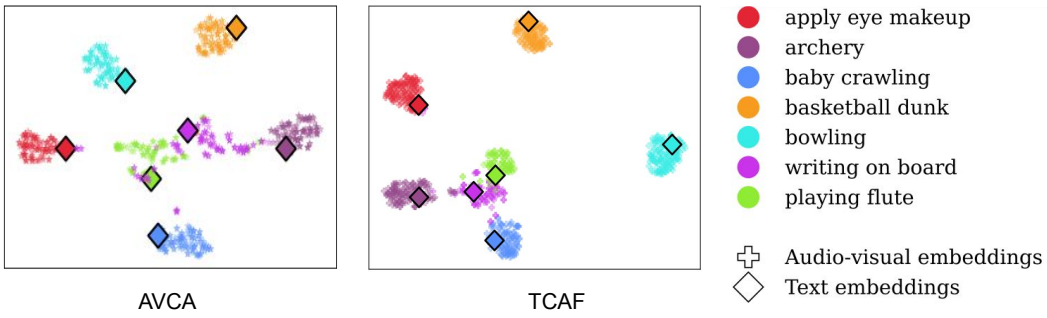


Figure B.2: t-SNE visualisations for five seen (*apply eye makeup, archery, baby crawling, basketball dunk, bowling*) and two unseen (*playing flute, writing on board*) test classes from the UCF-GZSL dataset, showing the difference between TCAF and [157]. Textual class label embeddings are visualised with a square.

B.3 t-SNE comparison between TCAF and [157]

We show t-SNE visualisations that highlight the difference between TCAF and [157] in Fig. B.2. It can be observed that in the case of [157], the classes overlap more than in the case of TCAF. In particular, this can be observed for the unseen classes. Moreover, for [157], the clusters are less concentrated than for TCAF.

B.4 Computational complexity

The computational complexity increases with the length of the temporal sequence. Using the average duration of the data in UCF-GZSL^{cls} and a single forward pass for a batch of 256 samples, TCAF requires 51.8 GFLOPS vs 174.1 for [101] and 4.4 for [157]. The Perceiver [101] uses a transformer architecture along with the temporal dimension, while [157] does not use the temporal dimension. Thus, it can be observed that TCAF is more resource-efficient than the most similar baseline. TCAF was trained on a single NVIDIA 2080Ti GPU.

C

Text-to-feature diffusion for audio-visual few-shot learning

In Sec. C.1, we describe the procedure used to extract the audio and visual features that are used as inputs to our AV-Diff framework. In Sec. C.2, we provide additional experimental results for (G)FSL with 20 shots, along with reporting the GFSL performance on base and novel classes across all shots and datasets. Finally, we provide additional ablations on the hybrid attention and diffusion model.

C.1 Feature extraction

We train AV-Diff on already pre-extracted temporal features for the audio and visual modalities. We used C3D [237] which was pretrained on Sports1M [105] and VGGish [92] pre-trained on Youtube-8M [1] to extract audio and visual features respectively. Each audio feature is represented by a 128-dimensional vector corresponding to one second of audio data. To extract the visual features, we first resampled the videos to 25fps and then extracted a 4096-dimensional vector for 16 consecutive video frames.

C.2 Additional experimental results

We present (G)FSL results for 20 shots on the UCF-FSL, VGGSound-FSL and ActivityNet-FSL datasets in Sec. C.2.1. In Sec. C.2.2, we discuss the 1-,5-,10- and 20-shot (G)FSL performance on base and novel classes across all three datasets (which complements Sec. 4.5.2 of the main paper). Finally, Sec. C.2.3 shows additional ablations on the hybrid attention and diffusion model.

C.2.1 (G)FSL in the 20-shot setting

In Tab. C.1 (bottom), we provide additional (G)FSL results for the 20-shot setting with AV-Diff and related methods. Similar to our observations in the main paper with 1, 5,

and 10 shots, AV-DIFF achieves state-of-the-art performance for 20 shots, outperforming all related methods in the FSL and GFSL (HM) settings.

Similar to the conclusions for ActivityNet-FSL in the main paper, it can be observed that the ranking of baselines changes dramatically on ActivityNet-FSL, while AV-DIFF still remains the best, showing that our model is also more robust on 20 shots.

The HM and FSL performances on 20 shots for AV-DIFF and for the related methods are higher compared to the lower shots. The increase in performance for AV-DIFF from 10 to 20 shots is similar to the one from 5 to 10 shots. However, the most significant boost in performance happens between the 1-shot and 5-shot settings, showing that the gain in performance decreases as more training samples for novel classes are added. Similar trends can also be observed for the related methods.

C.2.2 Performance on base and novel classes

In the main paper, we only presented the GFSL results in terms of the harmonic mean of the performance on the B (base) and N (novel) classes (Tab. 4.2 in the main paper). The harmonic mean is crucial as it evaluates how robust a system is, and it also provides higher scores to systems which are very balanced and which are less biased towards either B or N. In this section, we are going to analyse the performance of the components that are used to calculate the HM, namely the B and N performance, to have a better idea of the models' strengths and weaknesses. It can be seen in Tab. C.1 that in the majority of cases, AV-DIFF obtains state-of-the-art performance on B and N, but there are still some exceptions, as presented below.

In the 1-shot setting, it can be observed that MBT outperforms AV-DIFF on N in VGGSound-FSL and B in UCF-FSL, with scores of 21.34% and 79.89% compared to 21.25% and 77.94% for AV-DIFF. However, MBT is very biased towards one of the metrics. On VGGSound-FSL, the bias is towards N, and MBT obtains a very low score on B, only 11.21%, compared to 19.44% for AV-DIFF. The same applies on UCF-FSL, where MBT is very biased towards B. For B on VGGSound-FSL, AV-DIFF obtains a performance of 19.44% compared to 28.55% SLDG. While AV-DIFF scores similarly on both metrics in VGGSound-FSL, SLDG obtains a B score which is more than twice that of N, showing how unbalanced and biased SLDG is. An interesting observation that can be made in the 1-shot setting is that on VGGSound-FSL, AV-DIFF is not able to attain state-of-the-art performance in B or N, but it still performs overall much better than the systems that outperform AV-DIFF in these two metrics.

In the 5-shot setting, AV-DIFF is outperformed on B in both VGGSound-FSL and UCF-FSL by the Perceiver, with scores of 31.46% and 83.56% compared to 30.88% and 74.11% for AV-DIFF. Moreover, on VGGSound-FSL, AV-DIFF is also outperformed on N by MBT with scores of 31.79% vs 31.50% for AV-DIFF. However, both MBT and Perceiver have a bigger bias towards one of the metrics, leading to a lower HM on VGGSound-FSL. On UCF-FSL, it can be clearly observed that Perceiver is biased towards B, obtaining a

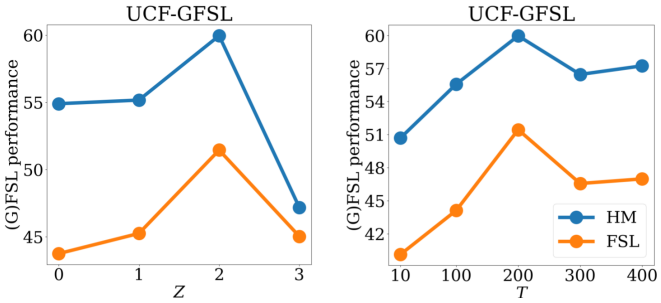


Figure C.1: (G)FSL performance (5-shot) for different numbers of self- (Z) and full attention layers (*left*), and different amounts of noise addition time steps T on UCF-FSL (*right*).

score which is more than twice that of N. For AV-DIFF this is not the case, as scores for both B and N are much more balanced.

The same observations can be made in the 10- and 20-shot settings where sometimes AV-DIFF is outperformed in one of the B or N, but still achieves a higher HM overall. While most of the baselines that outperform AV-DIFF in one of the metrics are usually very biased towards that metric, this is not always the case. For example, in the 20-shot setting on UCF-FSL, Att. Fusion slightly outperforms AV-DIFF on N with a score of 61.02% compared to 59.94% for AV-DIFF. However, on B, AV-DIFF significantly outperforms Att. Fusion with a score of 86.51% compared to 79.39% for Att. Fusion. While in this case Att. Fusion is very well balanced, it is still worse overall than AV-DIFF, as it only slightly outperforms AV-DIFF in N but it is significantly outperformed in B.

Interestingly, for different methods, the N score is sometimes higher than B. This is likely due to the use of calibrated stacking [41]. A similar behaviour has been observed by several other works, such as [155, 157, 162]

Overall, AV-DIFF is not necessarily the best in both B and N every single time. However, across all shots and datasets, AV-DIFF achieves state-of-the-art GFSL performance in terms of the HM. This shows that AV-DIFF is the most balanced and robust among all the methods, as it can consistently score very high on both B and N.

C.2.3 Ablation on hybrid attention and diffusion.

In Fig. C.1 (left), we analyse the impact of the number of self-attention layers Z and full-attention layers used. For values of $Z < 2$ the performance increases consistently and reaches a peak performance at $Z = 2$ for both metrics on UCF-FSL. It appears that changing the attention in late layers of the network is beneficial. Finally, we ablate over the timesteps T for adding noise to the original feature in the diffusion model in Fig. C.1 (right). The (G)FSL performance maximizes for $T = 200$ on UCF-FSL which corresponds to the number of timesteps used in AV-DIFF.

1-shot	VGGSound-FSL				UCF-FSL				ActivityNet-FSL			
	B	N	HM	FSL	B	N	HM	FSL	B	N	HM	FSL
Att. F. [72]	15.16	15.77	15.46	16.37	38.91	35.98	37.39	36.88	3.48	5.78	4.35	5.82
Perc. [101]	18.46	17.51	17.97	18.51	74.57	31.33	44.12	33.73	30.32	12.14	17.34	12.53
MBT [171]	11.21	21.34	14.70	21.96	79.89	26.37	39.65	27.99	17.07	12.24	14.26	12.63
TCaF [155]	20.93	18.34	19.54	20.01	66.18	33.64	44.61	35.90	23.85	12.62	16.50	13.01
Proto [118]	8.85	13.65	10.74	14.08	60.12	27.72	37.95	28.08	2.02	4.40	2.77	4.40
SLDG [30]	28.55	11.94	16.83	17.57	73.15	27.45	39.92	28.91	23.22	9.58	13.57	10.30
TSL [264]	17.09	20.72	18.73	22.44	68.18	33.04	44.51	35.17	8.96	10.18	9.53	10.77
HiP [39]	23.39	16.39	19.27	18.64	16.20	33.26	21.79	34.88	25.02	9.53	13.80	10.31
Zorro [203]	17.49	20.51	18.88	21.79	67.85	32.94	44.35	34.52	19.67	11.55	14.56	11.94
AVCA [157]	4.53	10.28	6.29	10.29	82.86	29.59	43.61	31.24	14.15	11.73	12.83	12.22
AV-Diff	19.44	21.25	**20.31**	**22.95**	77.94	38.45	**51.50**	**39.89**	32.77	12.86	**18.47**	**13.80**

5-shot	VGGSound-FSL				UCF-FSL				ActivityNet-FSL			
	B	N	HM	FSL	B	N	HM	FSL	B	N	HM	FSL
Att. F. [72]	28.64	27.82	28.22	31.57	63.27	43.69	51.68	47.18	5.00	8.05	6.17	8.13
Perc. [101]	31.46	28.52	29.92	33.58	83.56	34.27	48.60	40.47	35.66	20.15	25.75	21.50
MBT [171]	23.86	31.79	27.26	34.95	80.61	32.72	46.55	34.53	25.36	21.48	23.26	22.38
TCaF [155]	24.34	28.11	26.09	32.22	73.76	33.73	46.29	37.39	24.45	21.35	22.79	21.81
Proto [118]	25.27	25.08	25.17	28.87	63.69	31.79	42.42	33.63	1.61	7.81	2.67	7.81
SLDG [30]	29.74	15.98	20.79	25.17	65.44	25.28	36.47	28.56	29.40	17.95	22.29	19.16
TSL [264]	15.02	27.75	19.49	29.50	68.80	40.62	51.08	42.42	9.93	12.27	10.97	12.77
HiP [39]	30.01	24.18	26.82	30.67	33.65	39.74	36.44	42.23	21.98	15.39	18.10	16.25
Zorro [203]	29.06	30.07	29.56	35.17	69.13	41.49	51.86	42.59	25.72	21.03	23.14	21.94
AVCA [157]	13.24	20.15	15.98	20.50	84.80	34.64	49.19	36.70	19.18	21.09	20.09	21.65
AV-Diff	30.88	31.50	**31.19**	**36.56**	74.11	50.35	**59.96**	**51.45**	35.84	21.61	**26.96**	**23.00**

10-shot	VGGSound-FSL				UCF-FSL				ActivityNet-FSL			
	B	N	HM	FSL	B	N	HM	FSL	B	N	HM	FSL
Att. F. [72]	26.87	35.89	30.73	39.02	73.53	47.77	57.91	52.19	12.58	9.27	10.67	10.78
Perc. [101]	32.64	34.73	33.65	40.73	71.88	44.97	55.33	47.86	37.06	25.03	29.88	26.46
MBT [171]	26.76	34.43	30.12	38.93	84.07	35.62	50.04	39.73	29.06	24.98	26.86	26.03
TCaF [155]	26.62	31.73	28.95	36.43	84.28	39.93	54.19	47.61	27.86	22.32	24.78	23.33
Proto [118]	30.48	29.26	29.85	34.80	70.28	40.03	51.01	40.68	2.63	8.81	4.05	8.81
SLDG [30]	28.32	20.99	24.11	29.48	49.35	26.29	34.31	26.96	34.69	23.20	27.81	25.35
TSL [264]	17.96	28.15	21.93	31.29	74.31	51.63	60.93	55.63	9.31	11.76	10.39	12.18
HiP [39]	28.43	30.12	29.25	35.13	75.54	38.14	50.69	43.29	24.32	16.10	19.37	17.06
Zorro [203]	28.48	36.68	32.06	40.66	82.88	45.67	58.89	49.06	30.11	25.05	27.35	26.33
AVCA [157]	13.39	27.83	18.08	28.27	71.96	38.93	50.53	39.17	26.36	25.68	26.02	26.76
AV-Diff	32.15	36.05	**33.99**	**41.39**	84.62	51.69	**64.18**	**57.39**	37.91	26.02	**30.86**	**27.81**

20-shot	VGGSound-FSL				UCF-FSL				ActivityNet-FSL			
	B	N	HM	FSL	B	N	HM	FSL	B	N	HM	FSL
Att. F. [72]	31.43	37.88	34.35	44.08	79.39	61.02	69.00	63.20	15.51	11.41	13.15	13.22
Perc. [101]	33.11	37.66	35.24	43.77	77.81	48.29	59.59	52.66	32.30	31.06	31.67	32.21
MBT [171]	28.41	37.95	32.49	43.19	81.73	42.35	55.80	44.58	36.21	28.60	31.96	30.76
TCaF [155]	32.48	29.41	30.87	38.89	75.71	47.38	58.29	51.99	35.87	27.61	31.20	29.88
Proto [118]	31.44	32.66	32.04	38.42	61.07	49.32	54.57	50.48	25.05	8.17	12.32	14.65
SLDG [30]	33.20	19.53	24.59	33.30	81.08	39.52	53.14	43.95	32.60	30.80	31.68	32.44
TSL [264]	18.21	29.32	22.47	32.07	76.82	49.44	60.16	52.02	9.68	15.01	11.77	15.78
HiP [39]	32.03	29.83	30.89	38.46	71.59	43.43	54.06	48.07	33.78	17.59	23.13	20.67
Zorro [203]	29.84	39.46	33.98	43.63	87.82	48.46	62.45	57.10	34.15	28.55	31.10	30.31
AVCA [157]	15.30	32.20	20.75	32.64	60.00	44.93	51.39	44.93	24.47	29.88	26.91	30.76
AV-Diff	33.17	39.46	**36.04**	**44.79**	86.51	59.94	**70.81**	**65.72**	39.25	31.06	**34.68**	**32.89**

Table C.1: **Novel (N) and base (B) performance for audio-visual (G)FSL**: 1-shot, 5-shot, 10-shot, and 20-shot performance of AV-Diff and compared methods on the VGGSound-FSL, UCF-FSL and ActivityNet-FSL datasets. The harmonic mean (HM) of the mean class accuracies for base and novel classes are reported for GFSL. The FSL performance considers only the test subset of novel classes.

D

Audio-Visual Generalized Zero-Shot Learning using Pre-Trained Large Multi-Modal Models

D.1 Additional Details about Textual Feature Extraction

D.1.1 CLIP Feature Extraction

To boost zero-shot classification performance, [197] calculate normalized CLIP text embeddings for an ensemble of text prompts to retrieve final textual embeddings. Then the mean is taken and the result is normalized again. Normalizing the individual CLIP text representations is necessary in order to obtain a meaningful averaged vector. The second normalization facilitates the calculation of cosine similarity scores. Note that image embeddings are normalized as well.

For UCF-GZSL^{cls} and ActivityNet-GZSL^{cls}, we use an ensemble of 48 different prompt templates for each class. UCF-GZSL^{cls} and ActivityNet-GZSL^{cls} have a similar context since both are action recognition datasets. Hence, we use the same text prompts for these two datasets (see listing D.1). These templates are taken from the CLIP repository[1].

```
CLIP_prompt_templates = [
    'a photo of a person {}.',
    'a video of a person {}.',
    'a example of a person {}.',
    'a demonstration of a person {}.',
    'a photo of the person {}.',
    'a video of the person {}.',
    'a example of the person {}.',
    'a demonstration of the person {}.',
    'a photo of a person using {}.',
    'a video of a person using {}.',
```

[1] https://github.com/openai/CLIP/blob/main/data/prompts.md#ucf101

```
'a example of a person using {}.',
'a demonstration of a person using {}.',
'a photo of the person using {}.',
'a video of the person using {}.',
'a example of the person using {}.',
'a demonstration of the person using {}.',
'a photo of a person doing {}.',
'a video of a person doing {}.',
'a example of a person doing {}.',
'a demonstration of a person doing {}.',
'a photo of the person doing {}.',
'a video of the person doing {}.',
'a example of the person doing {}.',
'a demonstration of the person doing {}.',
'a photo of a person during {}.',
'a video of a person during {}.',
'a example of a person during {}.',
'a demonstration of a person during {}.',
'a photo of the person during {}.',
'a video of the person during {}.',
'a example of the person during {}.',
'a demonstration of the person during {}.',
'a photo of a person performing {}.',
'a video of a person performing {}.',
'a example of a person performing {}.',
'a demonstration of a person performing {}.',
'a photo of the person performing {}.',
'a video of the person performing {}.',
'a example of the person performing {}.',
'a demonstration of the person performing {}.',
'a photo of a person practicing {}.',
'a video of a person practicing {}.',
'a example of a person practicing {}.',
'a demonstration of a person practicing {}.',
'a photo of the person practicing {}.',
'a video of the person practicing {}.',
'a example of the person practicing {}.',
'a demonstration of the person practicing {}.',
```

```
]
```

Listing D.1: Text prompt templates that were used to create CLIP label embeddings for UCF-GZSL^{cls} and ActivityNet-GZSL^{cls}.

VGGSound-GZSL^{cls} contains videos of a variety of categories and hence more general prompts are required. The prompts that we used to create CLIP text embeddings for VGGSound-GZSL^{cls} can be seen in listing D.2.

```
VGGSound_CLIP_prompt_templates = [
    'a photo of {}.',
    'a video of {}.',
    'a example of {}.',
    'a demonstration of {}.',
    'a photo of the person {}.',
    'a video of the {}.',
    'a example of the {}.',
    'a demonstration of the {}.'
]
```

Listing D.2: Text prompt templates that were used to create CLIP text embeddings for VGGSound-GZSL^{cls}.

D.1.2 CLAP Feature Extraction

We use the same procedure as in D.1.1 to extract textual CLAP embeddings. For UCF-GZSL^{cls} and ActivityNet-GZSL^{cls} we use the prompts as in listing D.3. For VGGSound-GZSL^{cls}, we use the prompts given in listing D.4.

```
CLAP_prompt_templates = [
    'a person {} can be heard.',
    'a example of a person {} can be heard.',
    'a demonstration of a person {} can be heard.',
    'the person {} can be heard.',
    'a example of the person {} can be heard.',
    'a demonstration of the person {} can be heard.',
    'a person using {} can be heard.',
    'a example of a person using {} can be heard.',
    'a demonstration of a person using {} can be heard.',
    'a example of the person using {} can be heard.',
    'a demonstration of the person using {} can be heard.',
    'a person doing {} can be heard.',
    'a example of a person doing {} can be heard.',
    'a demonstration of a person doing {} can be heard.',
```

```
    'a example of the person doing {} can be heard.',
    'a demonstration of the person doing {} can be heard.',
    'a example of a person during {} can be heard.',
    'a demonstration of a person during {} can be heard.',
    'a example of the person during {} can be heard.',
    'a demonstration of the person during {} can be heard.',
    'a person performing {} can be heard.',
    'a example of a person performing {} can be heard.',
    'a demonstration of a person performing {} can be heard.',
    'a example of the person performing {} can be heard.',
    'a demonstration of the person performing {} can be heard.',
    'a person practicing {} can be heard.',
    'a example of a person practicing {} can be heard.',
    'a demonstration of a person practicing {} can be heard.',
    'a example of the person practicing {} can be heard.',
    'a demonstration of the person practicing {} can be heard.'
]
```

Listing D.3: Text prompt templates that were used to create CLAP label embeddings for UCF-GZSL^{cls} and ActivityNet-GZSL^{cls}.

```
VGGSound_CLAP_prompt_templates = [
    'a {} can be heard.',
    'a example of a {} can be heard.',
    'a demonstration of a {} can be heard.',
    'the {} can be heard.',
    'a example of the {} can be heard.',
    'a demonstration of the {} can be heard.',
    '{} can be heard.',
    'a example of {} can be heard.',
    'a demonstration of {} can be heard.'
]
```

Listing D.4: Text prompt templates that were used to create CLAP text embeddings for VGGSound-GZSL^{cls}.

E

Video-adverb retrieval with compositional adverb-action embeddings

E.1 Dataset splits for unseen adverb-action compositions

In this section, we provide further details about our proposed dataset splits for unseen adverb-action compositions based on the ActivityNet Adverbs [64, 90] and MSR-VTT Adverbs [64, 275] datasets. In Tab. E.1, we include information about the number of unlabelled samples (i.e. videos) and the number of unlabelled pairs (i.e. adverb-action compositions) in the dataset splits. The unlabelled samples are not used by ReGAdA, but we designed the splits so that we can fairly evaluate previous work [65] that uses unlabelled samples for training. The number of unlabelled samples and unlabelled pairs usually ranges from 30% to 50% of the total number of training samples and training pairs. This is significant, as methods like [65] use more training data than ReGAdA while performing significantly worse as observed in Tab. 6.6 in the main paper. We refer to the ActivityNet Adverbs and MSR-VTT Adverbs datasets as ActivityNet and MSR-VTT respectively.

In addition to the ActivityNet Adverbs and MSR-VTT Adverbs datasets, we use the VATEX Adverbs dataset [64, 252], and in particular the corresponding splits for unseen adverb-action compositions introduced in [65]. However, we use the same pre-extracted features as the current state-of-the-art work [165]. As some of the videos used in the split in [65] are not available anymore, it is not possible to extract S3D features for those. Hence, this resulted in fewer samples in the dataset, the number of training samples being reduced from 6921 to 6603, unlabelled samples from 3469 to 3317, and test samples from 3457 to 3293. In the following, we refer to the VATEX Adverbs dataset as VATEX.

Dataset	# train samples	# unlabelled samples	# test samples	# pairs train	# pairs unlabelled	# pairs test
VATEX	6603	3317	3293	319	168	316
MSR-VTT	987	306	454	225	114	225
ActivityNet	1490	634	848	635	537	543

Table E.1: Statistics of our dataset splits for the retrieval of unseen adverb-action compositions on the MSR-VTT Adverbs and ActivityNet Adverbs datasets. Statistics are also provided for the VATEX Adverbs dataset for features from [165].

E.2 Exploring the use of different word embeddings for unseen adverb-action compositions

Our ReGAdA framework composes adverb and action text embeddings in a shared embedding space. Specifically, we used a text model that was jointly trained with the S3D video model. In this section, we show results for different choices of word embeddings. Existing and widely-adopted word embeddings like GloVe [189], word2vec [161], and fastText [32] rely on unsupervised learning techniques to generate vector representations of words based on their co-occurrence statistics in a large corpus of text. Specifically, word2vec and GloVe focus on co-occurrences of words, whereas fastText uses co-occurrences of n-gram characters, which can be useful when dealing with rare words.

Model	VATEX	ActivityNet	MSR-VTT
CLIP [197]	54.5	55.1	57.0
Act. Mod. [65]	53.8	57.0	56.0
AC_{CLS} [165]	54.3	55.1	53.7
AC_{REG} [165]	54.9	53.9	59.0
ReGAdA	61.7	**58.4**	**61.0**
ReGAdA w2v	60.5	53.1	60.0
ReGAdA fastText	60.8	53.5	57.3
ReGAdA GloVe	58.0	54.0	57.7
ReGAdA GPT-3	**63.3**	53.5	60.3

Table E.2: Effect of using different types of word embeddings in our ReGAdA framework on the performance for retrieving unseen action-adverb compositions on the VATEX, ActivityNet and MSR-VTT benchmarks. [65] uses pseudo-labelling.

Prior works on video-adverb retrieval leveraged GloVe embeddings of class labels [64, 65], while approaches in zero-shot learning commonly use word2vec or fastText embeddings as side information [150, 155, 157, 168, 263]. However, recent advances in language modelling have shown impressive progress on a variety of natural language processing tasks. For instance, large language models incorporate contextual information at the sentence level and beyond, which could result in more informative and accurate embeddings. To investigate their usefulness for our retrieval task, we extract word embeddings with GPT-3 [36] using the OpenAI API for the `text-embedding-ada-002` model. While word2vec, fastText, and GloVe provide 300-dimensional embeddings, GPT-3 embeddings have a much larger dimension of 1536. All text embeddings are projected to 400-dimensional vectors before being input into the text encoder. For CLIP [197], we extract visual CLIP features for each second of the video and

CLIP text embeddings from the action-adverb labels (e.g. *cut slowly*). We then use the cosine similarity between temporally-averaged frame features and text embeddings for retrieval.

Tab. E.2 shows that the choice of the text embedding results in significant performance changes, measured by the binary antonym classification accuracy. ReGAdA uses text embeddings jointly trained with the S3D video model like the other baselines (referred to as S3D embeddings in the following), and it is able to outperform all the baselines, as shown in the main paper. However, from Tab. E.2 it can be observed that ReGAdA with S3D embeddings is outperformed by ReGAdA with GPT-3 embeddings on VATEX, leading to a performance of 63.3 compared to 61.7 for S3D embeddings. GPT-3 embeddings contain more contextual and fine-grained semantic information but suffer from a significant reduction in dimensions in the projection. We find that higher-dimensional text embeddings perform worse when training data is scarce (e.g. 53.5/60.3 for GPT-3 vs. 58.4/61.0 for S3D on ActivityNet/MSR-VTT), likely caused by a lack of training data to learn the down-projection. Overall, word2vec, fastText, and GloVe embeddings yield slightly worse results than S3D embeddings across datasets.

E.3 Training without antonyms

In Tab. E.3, we present the video-to-adverb and adverb-to-video retrieval performance when training without antonyms. This task was introduced in [165]. For the results in the main paper, ReGAdA is trained with antonyms as negative examples in its triplet loss. As it might not always be feasible to require adverb-action samples that are additionally annotated with an adverb-antonym, this scenario inspects the generalisation capabilities of ReGAdA to dataset settings with fewer constraints.

When training without adverb-antonyms, ReGAdA randomly samples an adverb as a negative sample which is not identical to the positive adverb sample. As there is no access to information about the adverb-antonym during evaluation, the Acc-A metric cannot be used in this context.

In Tab. E.3 we can observe that ReGAdA outperforms all prior methods for this task across all datasets and metrics. For example, on VATEX ReGAdA obtains a mAP W score of 0.292 compared to 0.283 for AC_{CLS}. Moreover, ReGAdA obtains a mAP M score of 0.136 which significantly outperforms AC_{CLS} with a score of 0.108.

E.4 Comparing ReGAdA with CLIP

In this section, we present additional video-adverb retrieval results with CLIP [197] in addition to the retrieval results for unseen compositions (see Tab. E.2).

Similar to the experiment on unseen compositions (see Sec. E.2), we use the cosine similarity between temporally-averaged CLIP frame features and text embeddings for the

	HowTo100M [64]		Adverbs in Recipes [165]		ActivityNet [65]		MSR-VTT [65]		VATEX [65]	
	mAP W	mAP M	mAP W	mAP M	mAP W	mAP M	mAP W	mAP M	mAP W	mAP M
Priors	0.446	0.354	0.491	0.263	0.217	0.159	0.308	0.152	0.216	0.086
S3D pre-trained	0.339	0.238	0.389	0.173	0.118	0.071	0.194	0.075	0.122	0.038
TIRG [244]	0.441	0.476	0.485	0.228	0.186	0.111	0.297	0.113	0.195	0.065
Act Mod [64]	0.408	0.352	0.508	0.249	0.187	0.127	0.233	0.134	0.144	0.060
AC_{CLS}† [165]	0.562	0.420	0.606	0.289	0.130	0.096	0.305	0.131	0.283	0.108
AC_{REG}† [165]	0.573	0.481	0.667	0.319	0.143	0.093	0.287	0.121	0.282	0.100
ReGAdA	**0.580**	**0.536**	**0.668**	**0.466**	**0.282**	**0.211**	**0.401**	**0.252**	**0.292**	**0.136**

Table E.3: Results *without* antonyms during training for adverb-to-video retrieval (mAP W/M). Higher is better for all metrics. † refers to updated results provided by the authors of [165].

retrieval with CLIP. Additionally, we examine the impact of replacing the S3D video/text embeddings of ReGAdA with CLIP embeddings ($\text{ReGAdA}_{\text{CLIP}}$).

In Tab. E.4, we can observe that CLIP performs marginally better than the S3D pre-trained baseline. Using CLIP features in ReGAdA improves adverb retrieval (Acc-A) slightly on ActivityNet and VATEX. However, $\text{ReGAdA}_{\text{CLIP}}$ is worse than ReGAdA for video retrieval, likely caused by inferior visual features when extracting those only from a few video frames.

	ActivityNet			MSR-VTT			VATEX		
	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A	mAP W	mAP M	Acc-A
S3D pre-tr.	0.118	0.070	0.560	0.194	0.075	0.603	0.122	0.038	0.586
CLIP [197]	0.120	0.067	0.611	0.206	0.084	0.677	0.129	0.039	0.644
$\text{ReGAdA}_{\text{CLIP}}$	0.201	0.151	**0.781**	0.352	0.142	0.784	0.247	0.098	**0.837**
ReGAdA	**0.239**	**0.175**	0.771	**0.378**	**0.228**	**0.786**	**0.290**	**0.113**	0.817

Table E.4: Comparing ReGAdA with CLIP as a baseline, and when replacing ReGAdA's S3D video/text embeddings with CLIP embeddings ($\text{ReGAdA}_{\text{CLIP}}$).

E.5 Seed experiments

In Tab. E.5, we provide experimental results that test the robustness of our model with regard to the seeds used, as done in [165]. To compute these numbers, we use four seeds and compute the mean and the standard deviation over these runs. It can be observed that ReGAdA achieves a higher mean than the other baselines. Furthermore, the standard deviation with our model is relatively low.

	Adverbs in Recipes [165]		
	mAP W	mAP M	Acc-A
Act Mod	0.394 ± 0.023	0.140 ± 0.026	0.843 ± 0.013
MLP+Act Mod	0.407 ± 0.044	0.151 ± 0.033	0.842 ± 0.012
AC_{CLS} †	0.605 ± 0.001	0.287 ± 0.001	0.841 ± 0.000
AC_{REG} †	0.611 ± 0.002	0.239 ± 0.007	0.845 ± 0.001
ReGADA	**0.699 ± 0.004**	**0.419 ± 0.012**	**0.876 ± 0.001**

Table E.5: Performance of our ReGADA framework on the Adverbs in Recipes dataset when using multiple random seeds. † refers to updated results provided by the authors of [165].

F

Adapting Communicating MLLMs on the Fly in Referring Expression Tasks

F.1 Broader Impact

In this work we study the capabilities of a speaker to adapt to a listener. We considered MLLMs adapting to other MLLMs, but one could apply these methods also for adapting MLLMs to humans. If such techniques were used to adapt MLLMs to humans, people with malicious intend could purposefully teach the MLLMs to produce harmful or otherwise undesirable content. Online adaptation could effectively overwrite previously learned safety measures of the alignment phase. A possible solution could involve intertwining or following online adaptation with alignment training. Additional research is require to measure both opportunities and risks in this scenario.

In the setting where we adapt an MLLM agent to another MLLM agent, malicious actors could try to exploit systems employing MLLMs by programmatically learning to maximize a desired action of the target MLLM. These "hacks" or "jailbreaks" are a security concern for everyone deploying MLLM, especially if they are deployed adapting to the users. As a result, research on defense mechanisms is just as important as developing more advanced ways to enable personalization.

On the other hand, we believe that allowing MLLMs to adapt to the specific needs of a users can enable new use cases and improve inclusion across diverse population groups. More effective communication towards users with disabilities could lower the barrier of entry and learning curve to bring MLLM technology and their advancement to a broad audience.

F.2 MLLM Prompting Details

The referring expression identification (REI) task starts with the speaker generating a description for the target image. The prompt given to the speaker is:

```
Write a description for the left/right image, such that it can be
```

`differentiated from the right/left image, but do not talk about the right/left image. Do not name which image you are describing.`

Subsequently, with the help of the speaker's response, the listener generates a sentence containing its guess. For LLaVA listener agents, we use the query template:

`Does this sentence: '{`$m^{(s)}$`}' describe the left image or the right image? Do not explain your reasoning.`

where $\{m^{(s)}\}$ is replaced with the description written by the speaker. On the other hand, Qwen gets the prompt:

`Which image does the sentence '{`$m^{(s)}$`}' describe? A. Picture 1 B. Picture 2.`

After receiving the listener's answer, the reward is computed by looking for keywords, i.e. "left, A, 1" and "right, B, 2", and comparing it with the ground truth label.

For the referring expression segmentation (RES) task, the prompt given to the speaker is:

`Write a short description for the highlighted object.`

The PaliGemma listener is then prompted with:

`segment '{`$m^{(s)}$`}'`

where "segment" is a PaliGemma specific keyword to induce its segmentation capabilites. The model proceeds to output tokens that can be translated to a segmentation mask. We calculate the intersection over union (IoU) between the predicted segmentation mask and the ground truth segmentation mask as a reward for the speaker. Since KTO requires a binary reward, we binarize the IoU values with a threshold of 0.5.

Since LLaVA models can only take a single picture as input, we concatenate the images horizontally and add a white bar between them before feeding them to LLaVA-7B and LLaVA-13B. As a result of this step, LLaVA refers to the images as left or right. No such processing is necessary with Qwen, as it can handle multiple images in a single query. Qwen automatically labels them as picture 1 and picture 2.

F.3 Ground-Truth Descriptions with Perceptually Weakened Listeners

We present the evaluation of the GT speaker against listeners with perceptual weakness in Fig. F.1. We observed that both blurry and grayscale images cause a significant drop in performance, with the latter having the greatest impact.

When all attributes are mentioned, blurring decreases the scores of LLaVA-7B and Qwen from 0.83 and 0.63 to 0.74 and 0.53. LLaVA-13B maintains its accuracy of 0.73. When

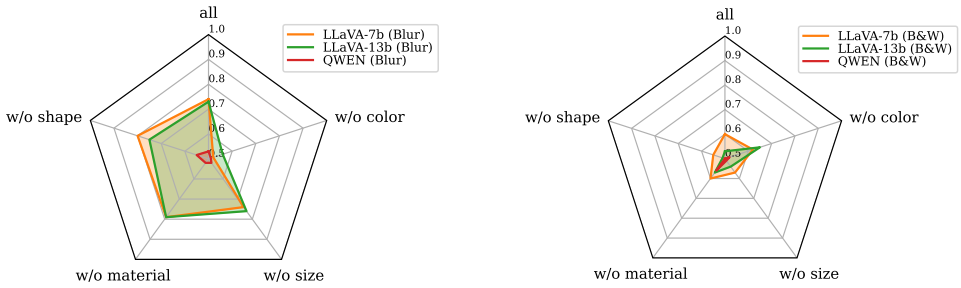


Figure F.1: Performance for ground-truth descriptions with blurred vision (left) and color blindness (right).

the speaker additionally does not mention any color attributes in the description, the accuracy of all listeners drop to near-random performance, with LLaVA-13B performing best at 0.56 accuracy. This result indicates that colors are vital for agents with blurry vision. Removing shapes from the descriptions increases the scores by a small margin in all cases, which suggests this information could be confusing in the presence of blur. Additionally, LLaVA models gain a few percent accuracy when materials are not mentioned in the description. Finally, we would like to highlight that Qwen achieves at most 0.55 score in this setup, which is very close to random guessing.

With grayscale images, LLaVA-7B achieves the lowest score of 0.55 when shape information is lacking in the descriptions, and has the highest accuracy of 0.62 with colors removed. The worst and best cases for LLaVA-13B are again without shape (0.51) and without color (0.65), which have a larger difference compared to the smaller version of LLaVA. Those results show color information starts to confuse the models as it is useless, and mentioning shape is more important in this case. Similar to blurry images, Qwen has a very low performance, with a maximum score of 0.56. These observations support our previous findings that shape and color are the most important attributes for performing well on the REI task with CLEVR images.

F.3.1 Additional Qualitative Results on REI

In Fig. F.2 we show qualitative results for the REI task on CLEVR, CUB and ImageNet by contrasting generated descriptions before and after adaptation. In CLEVR the original description is much longer and even if the speaker is able to mention all the objects in the image, the associated shapes and color are oftentimes incorrect. On the other hand, after adaptation, the descriptions are much shorter, mentioning a subset but distriminative part of the scene. The adapted policy frequently mentioning shapes (“blocks”, “balls”) and colors (“yellow and silver”) provides additional evidence that these attributes are important and easier to recognize for MLLMs in this context.

The ZSL descriptions generated for CUB images are generic and long, often applying to both images. The speaker tends to confuse the confounding image into the description, for instance, when talking about the bird “facing the camera” and the “black and yellow”

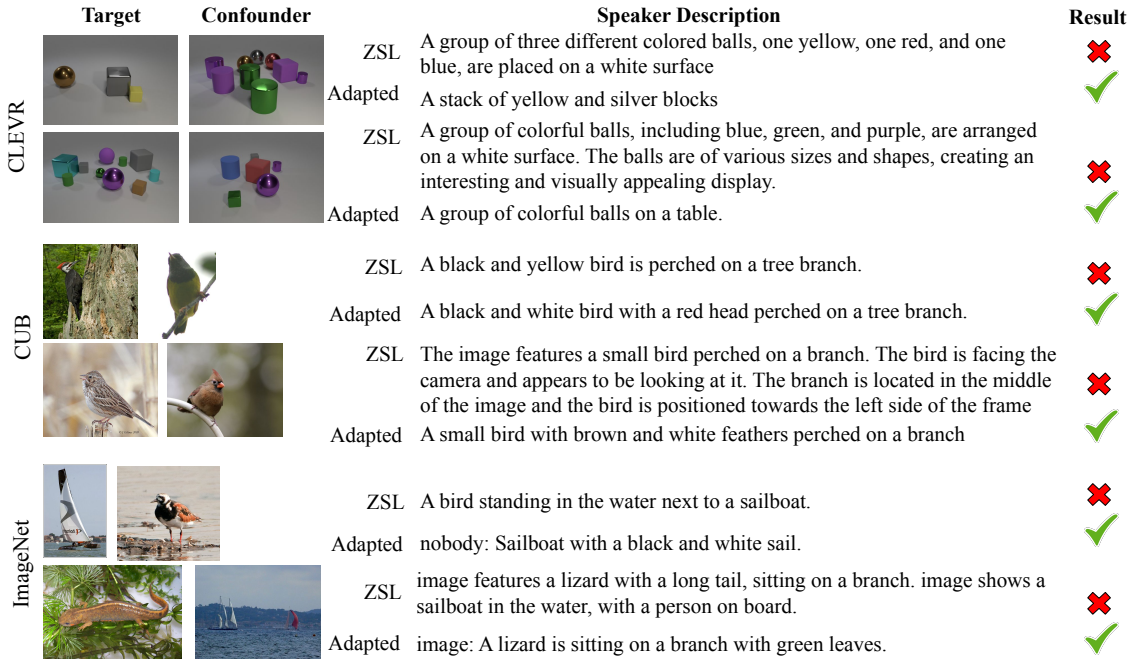


Figure F.2: Qualitative results for CLEVR, CUB and ImageNet datasets. We present the descriptions generated by the untrained agents (ZSL) and the descriptions obtained after training (Adapted).

bird mixing the colors of both birds. In contrast, the trained agent just mentions the essential distinguishable aspects of the target images ("brown and white feathers" and "black and white bird with a red head"). Lastly, on ImageNet, one failure case of the untrained speaker is that it describes both images without clearly identifying the target. After training, it learns to focus on describing the content of the target image by itself. In conclusion, from these qualitative examples, we observe that the model learns to be more concise, focusing on the correct image and primarily mentions the relevant attributes, which more frequently include color and shape.

F.4 Computational Resources

For every experiment, we use 2x A100 40GB GPUs, where one GPU is used for the listener and the other for the speaker. Since the speaker is trained, it requires more computational resources than the listener. It is possible to fit a 13B parameter model into the memory of a single GPU in inference mode for the listener. However, training MLLM only allows models up to 7B parameters on a single GPU, even when using a parameter-efficient fine-tuning method such as LoRA. The training time depends on the lengths of sentences LLaVA generates as the speaker. Longer token sequences take more time to produce as well as to backpropagate through the model. While the length of generations usually diminishes as the speaker adapts to the listener, we also observe the generated descriptions vary in lengths for the different dataset. Overall, a single experiment of playing 1800 REI

episodes and performing 600 update steps (batch size 3) takes around 5-6 hours training time.

F.5 Hyperparameters

For all experiments, we perform a grid search over a subset of hyperparameters and report the results of the best set of hyperparameters. Generally, there was no single set of hyperparameters that performed well across all experiment. The hyperparameters that we considered for grid search are: the learning rate lr, the rank r of the LoRA and the α parameters in LoRA. Depending on the algorithms, datasets and models, the lr was searched in the interval [1e-7, 1e-8, 1-9], the r was searched in the interval [32, 64, 128] and the α was searched in the interval [64, 128, 256, 512, 1024, 2048]. The remaining hyperparameters were kept fixed without performing a grid search. Specifically, for β in KTO we used 0.1, and for PPO and NLPO we used 0.2. ϵ in PPO was set to 1, top-p sampling in NLPO was set to 0.9. λ^- and λ^+ were set to 1.0.

G

Publications and contributions

G.1 Publications

This thesis is based on the following publications. An overview of the contributions can be found in Sec. 1.7. Asterisks (*) indicate shared first author publications. Bold names correspond to the name of the author of this thesis.

1. [157] **O.-B. Mercea**, L. Riesch, A. S. Koepke, Z. Akata, "Audio-visual Generalised Zero-shot Learning with Cross-modal Attention and Language". In: *IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR)*. 2022.

2. [155] **O.-B. Mercea** *, T. Hummel *, A. S. Koepke, Z. Akata, "Temporal and cross-modal attention for audio-visual zero-shot learning". In: *The European Conference on Computer Vision (ECCV)*. 2022.

3. [156] **O.-B. Mercea**, T. Hummel, A. S. Koepke, Z. Akata, "Text-to-feature diffusion for audio-visual few-shot learning". In: *DAGM German Conference on Pattern Recognition (DAGM GCPR)*. 2023.

4. [96] T. Hummel, **O.-B. Mercea**, A. S. Koepke, Z. Akata, "Video-adverb retrieval with compositional adverb-action embeddings". In: *The British Machine Vision Conference (BMVC)*, ***Oral***. 2023.

5. [119] D. Kurzendörfer * , **O.-B. Mercea** *, A. S. Koepke, Z. Akata, "Audio-Visual Generalized Zero-Shot Learning using Pre-Trained Large Multi-Modal Models". In: *CVPR Workshop on Learning with Limited Labelled Data for Image and Video Understanding*. 2024.

6. S. Alaniz, **O.-B. Mercea**, Y. Durmazkeser, Z. Akata, "Adapting Communicating MLLMs on the Fly in Referring Expression Tasks". *Submitted* in 2024.

The following publications were done during the course of the PhD, but they are not part of this thesis as the tasks studied in these works significantly diverge from the tasks presented in this thesis.

1. [204] K. Renz, K. Chitta, **O.-B. Mercea**, A. S. Koepke, Z. Akata, A. Geiger, "PlanT: Explainable Planning Transformers via Object-Level Representations". In: *Conference on Robot Learning (CoRL)*. 2022.

2. [154] **O.-B. Mercea**, A. Gritsenko, C. Schmid, A. Arnab, "Time-, Memory- and Parameter-Efficient Visual Adaptation". In: *IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR), **Highlight***. 2024.

G.2 Contributions

This section presents the contributions of the authors for the publications included in this thesis, as mentioned in Sec. G.1

Chapter 2: Audio-visual Generalised Zero-shot Learning with Cross-modal Attention and Language.

This work was done in collaboration with Lukas Riesch, A. Sophia Koepke, and Zeynep Aktata. Otniel-Bogdan Mercea was the first author and contributed by developing the state-of-the-art system, implementing some of the baselines and benchmarks, and running many experiments. Lukas Riesch contributed to the initial codebase for the project, did some of the experiments, and helped prepare the benchmarks and some of the baselines. A. Sophia Koepke and Zeynep Akata had a supervisory role by offering weekly meetings throughout the project to discuss progress and future milestones. All authors contributed to writing the paper.

Chapter 3: Temporal and cross-modal attention for audio-visual zero-shot learning.

This work was done with Thomas Hummel, A. Sophia Koepke, and Zeynep Akata. Otniel-Bogdan Mercea and Thomas Hummel were both shared first authors and contributed equally. Otniel-Bogdan Mercea contributed more to the task formulation and the losses, while Thomas Hummel contributed more to the model's design and attention. Finally, A. Sophia Koepke and Zeynep Akata had a supervisory role by offering weekly meetings throughout the project to discuss progress and future milestones. All authors contributed to writing the paper.

Chapter 4: Text-to-feature diffusion for audio-visual few-shot learning.

This work was done with Thomas Hummel, A. Sophia Koepke, and Zeynep Akata. Otniel-Bogdan Mercea was the first author and contributed by providing the benchmarks, implementing the baselines, developing the state-of-the-art system, and running most experiments. Thomas Hummel contributed ideas, helped run some of the experiments, and implemented some of the ablations. A. Sophia Koepke and Zeynep Akata had a supervisory role by offering weekly meetings throughout the project to discuss progress and future milestones. All authors contributed to writing the paper.

Chapter 5: Audio-Visual Generalized Zero-Shot Learning using Pre-Trained Large Multi-Modal Models.

This work was done in collaboration with David Kurzendörfer, A. Sophia Koepke, and Zeynep Akata. David Kurzendörfer and Otniel-Bogdan Mercea were both joint first authors and contributed equally. Otniel-Bogdan Mercea contributed by providing the initial codebase for this project along with already implemented baselines and provided insights into this task. Otniel-Bogdan Mercea also developed the project's initial idea, wrote significant parts of the paper, and helped David Kurzendörfer whenever he encountered an issue. David Kurzendörfer developed the state-of-the-art system and adapted the prior benchmarks introduced by Otniel-Bogdan Mercea to this new setting. Moreover, David Kurzendörfer ran all the experiments presented in this paper. A. Sophia Koepke and Zeynep Akata had a supervisory role by offering weekly meetings throughout the project to discuss progress and future milestones. All authors contributed to writing the paper.

Chapter 6: Video-adverb retrieval with compositional adverb-action embeddings.

This work was done with Thomas Hummel, A. Sophia Koepke, and Zeynep Akata. Thomas Hummel was the first author, and his contributions were related to developing the state-of-the-art system, running most of the experiments, proposing the project's original idea, and implementing many ablations. Otniel-Bogdan Mercea was the second author, and he contributed to implementing some of the ablations, running some of the experiments, and creating the additional zero-shot dataset splits. A. Sophia Koepke and Zeynep Akata had a supervisory role by offering weekly meetings throughout the project to discuss progress and future milestones. All authors contributed to writing the paper.

Chapter 7: Adapting Communicating MLLMs on the Fly in Referring Expression Tasks. This work was done in collaboration with Stephan Alaniz, Yavuz Durmazkeser, and Zeynep Akata. Stephan Alaniz was the first author, and he helped supervise all the stages of the project, implemented some adaptation algorithms and baselines, and contributed significantly to the RES task and to writing the paper. Otniel-Bogdan Mercea was the second author and contributed to running most of the experiments on the REI task and developed some of the techniques required to make the communication adaptation work. Yavuz Durmazkeser contributed to implementing some of the baselines, ablations, adaptation algorithms and run some experiments in both REI and RES tasks. Zeynep Akata had a supervisory role by offering weekly meetings throughout the project to discuss progress and future milestones. All the authors contributed to writing the paper.