FunnyNet-W: Multimodal Learning of Funny Moments in Videos in the Wild

Zhi-Song Liu, Robin Courant, Vicky Kalogeiton

▶ **To cite this version:**

Zhi-Song Liu, Robin Courant, Vicky Kalogeiton. FunnyNet-W: Multimodal Learning of Funny Moments in Videos in the Wild. International Journal of Computer Vision, 2024, 132, pp.2885 - 2906. 10.1007/s11263-024-02000-2 . hal-04823203

HAL Id: hal-04823203

https://cnrs.hal.science/hal-04823203v1

Submitted on 6 Dec 2024

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

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

Distributed under a Creative Commons Attribution 4.0 International License

International Journal of Computer Vision (2024) 132:2885–2906
https://doi.org/10.1007/s11263-024-02000-2

FunnyNet-W: Multimodal Learning of Funny Moments in Videos in the Wild

Zhi-Song Liu[1] · **Robin Courant**[2] · **Vicky Kalogeiton**[2]

Received: 31 August 2023 / Accepted: 9 January 2024 / Published online: 23 February 2024
© The Author(s) 2024

Abstract

Automatically understanding funny moments (i.e., the moments that make people laugh) when watching comedy is challenging, as they relate to various features, such as body language, dialogues and culture. In this paper, we propose FunnyNet-W, a model that relies on cross- and self-attention for visual, audio and text data to predict funny moments in videos. Unlike most methods that rely on ground truth data in the form of subtitles, in this work we exploit modalities that come naturally with videos: (a) video frames as they contain visual information indispensable for scene understanding, (b) audio as it contains higher-level cues associated with funny moments, such as intonation, pitch and pauses and (c) text automatically extracted with a speech-to-text model as it can provide rich information when processed by a Large Language Model. To acquire labels for training, we propose an unsupervised approach that spots and labels funny audio moments. We provide experiments on five datasets: the sitcoms TBBT, MHD, MUStARD, Friends, and the TED talk UR-Funny. Extensive experiments and analysis show that FunnyNet-W successfully exploits visual, auditory and textual cues to identify funny moments, while our findings reveal FunnyNet-W's ability to predict funny moments in the wild. FunnyNet-W sets the new state of the art for funny moment detection with multimodal cues on all datasets with and without using ground truth information.

Keywords Multimodal learning · Vision+language · Video understanding · Humor detection

1 Introduction

We understand the world by using our senses, especially in multimedia areas. All signals can stimulate one's feelings and reactions. Funniness is universal and timeless: in 1900 BC Sumerians wrote the first joke and it is still funny nowadays. However, whereas humans can easily understand funny moments, even from different cultures and eras, machines do not. Even though the number of interactions between humans and machines is growing fast, identifying funniness is still a brake on making these interactions spontaneous. Actually, understanding funny moments is a complex concept since they can be purely visual, purely auditory, or they can mix both cues: there is no recipe for the perfect joke.

Recently, there have been attempts to understand the nature of jokes, humour, and funny moments (Annamoradnejad & Zoghi, 2020; Weller & Seppi, 2020). However, most of these works have relied solely on textual cues, with only a few incorporating videos (Patro et al., 2021; Kayatani et al., 2021). The limitation of these approaches lies in their dependence on external transcripts in the form of manual subtitles, which are not naturally available with raw video data. In contrast, advancements in the field of speech-to-text have made it easier to extract accurate transcripts from raw audio waveforms that naturally accompany videos. This enables processing natural language to better understand the overall context. Furthermore, including audio as a modality in the funny moment detection pipeline is essential, as raw audio carries essential and complementary cues, including tones, pauses, pitch, pronunciation, and background noises (Zadeh et al., 2018; Castro et al., 2019). When speaking, the way people convey their message is as important as the actual content being delivered. Similarly, visual content plays a crucial role. For example, the same phrase spoken by the same person can elicit different emotional responses depending on the context (see Fig. 1). Facial expressions, body gestures,

Communicated by Jürgen Gall.

✉ Zhi-Song Liu
zhisong.liu@lut.fi

[1] Computer Vision and Pattern Recognition Laboratory, Lappeenranta-Lahti University of Technology, Lappeenranta, Finland

[2] LIX, Ecole Polytechnique, IP Paris, Palaiseau, France

Mia: Three tomatoes are walking down the street -- a poppa tomato, a momma tomato, and a little baby tomato. Baby tomato starts lagging behind. Poppa tomato gets angry, goes over to the baby tomato, and squishes him... and says, ... "Catch up."

Fig. 1 What is funny? Audio cues along with visual frames and textual data are a rich source of information for identifying funny moments in videos. Video scene from Pulp Fiction, 1994, source video https://www.youtube.com/watch?v=4L5LjjYVsHQ

and scene context contribute to a better understanding of the intended meaning, thereby influencing the perceived funniness.

Therefore, in this paper, we introduce FunnyNet-W, a multimodal model for predicting funny moments in videos. It comprises three encoders: (a) visual encoder, which captures the global contextual information of a scene; (b) textual encoder, which represents the overall understanding of a scene; and (c) audio encoder, which captures voice and language effects; and the Cross Attention Fusion (CAF) module, i.e., a new module that learns cross-modality correlations hierarchically so that features from different modalities can be combined to form a unified feature for prediction. Thus, FunnyNet-W is trained to learn to embed all cross-attention features in the same space via self-supervised contrastive learning (Chen et al., 2020), in addition to classifying clips as funny or not funny. To obtain labeled data, we exploit the laughter that naturally exists in sitcom TV shows. We define as 'funny-moment' any n-second clip followed by laughter; and 'not-funny' the clips not followed by laughter. To extract laughter, we propose an unsupervised labeling approach that clusters audio segments into laughter, music, voice and empty, based on their waveform difference.[1] Moreover, we enrich the Friends dataset with laughter annotations.

Our extensive experimentation and analysis show that combining audio, visual and textual cues (that all come naturally with videos) is suitable for funny-moment detection. Moreover, we compare FunnyNet-W to the state of the art on five datasets including sitcoms (TBBT, MHD, MUStARD, and Friends) and TED talks (UR-Funny), and show that it outperforms all other methods for all metrics and input configurations. Note that even by using only automatically generated text from audio, FunnyNet-W outperforms all other methods that rely on ground-truth text in the form of subtitles. Furthermore, we examine the difference between our proposed FunnyNet-W and automatic chatbots based on Large-Language-Models (LLMs). Our findings show that without specific prompt engineering under the few-shot setting, chatbots cannot understand the funniness of texts. Instead, our proposed FunnyNet-W significantly outperforms chatbots in prediction accuracy, highlighting the importance of specific multimodal training for this task. We also apply FunnyNet-W to data from other domains, i.e., movies, stand-up comedies, and audiobooks. For quantitative evaluation, we apply FunnyNet-W on a sitcom without canned laughter manually annotated. It shows that FunnyNet-W predicts funny moments without fine-tuning, revealing its flexibility for funny-moment detection in the wild.

Our contributions are summarized as follows: (1) We introduce FunnyNet-W, a model for funny moment detection that uses audio, visual, and textual modalities that come automatically with videos. FunnyNet-W combines features from the three modalities using the proposed CAF module relying on cross and self-attention; (2) Extensive experiments and analysis highlight that FunnyNet-W successfully exploits audio, visual and textual cues; (3) FunnyNet-W achieves the new state of the art on five datasets. We also demonstrate its generalizability by comparing it to automatic LLM chatbots and its flexibility by showcasing in-the-wild applications. The code is available online on the project page: https://www.lix.polytechnique.fr/vista/projects/2024_ijcv_liu/.

A preliminary version of this work has been published in ACCV 2022 (Liu et al., 2022). We significantly extend it in the following ways:

- **Motivation.** We propose FunnyNet-W, a multimodal model for funny moment detection in videos. FunnyNet-W follows the same motivation from FunnyNet, i.e. leverage modalities that come with videos for free. Given that most funny moments are inherently associated with language, in addition to the audiovisual features of FunnyNet, FunnyNet-W leverages speech-to-text features. For this, we automatically generate text from speech by leveraging Automatic Speech Recognition methods, and then pair it with the rich representation capability of Large Language Models (LLMs), thus enabling to better understand the specificities of language. This is motivated thoroughly in the Introduction, in Sect. 3.4 and experimentally evaluated in the new Sects. 5.2.1, 5.2.2, 6.1, and 6.2.
- **Architecture.** FunnyNet uses audio, visual and face encoders to process the multimodal signals. The face encoder, however, is cumbersome and requires an external face detection model. For this reason, in FunnyNet-W we do not use a face encoder. Instead, FunnyNet-W uses an LLM text encoder to process textual data that are automatically transcribed. Moreover, in FunnyNet-W, we use a more modern visual encoder. The differences between the two models are described in Sect. 3.4 and experimentally compared in Sect. 5.1 (Table 1) and Sect. 5.2.1.

[1] Note that we use the laughter solely as an indicator for data labeling, but the laughter is not the included in the audio segments of FunnyNet-W. Once FunnyNet-W is trained, it can detect funny moments in any video, with or without laughter.

Table 1 Comparison to the state of the art on five datasets

Method/metrics	Wild	TBBT		MHD		MUStARD		UR-Funny		Friends	
		F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc
Random	–	46.3	50.0	56.1	50.9	48.3	48.7	50.2	50.2	51.0	51.0
All positive	–	60.3	43.2	75.6	60.8	66.7	50.0	75.4	50.7	66.7	50.0
All negative	–	0.0	56.8	0.0	39.2	0.0	50.0	0.0	49.3	0.0	50.0
MUStARD 2019 (V+A+T^{gt}) (Castro et al., 2019)	–	–	–	–	–	71.7	71.8	–	–	–	–
MSAM 2021 (V+T^{gt}) (Patro et al., 2021)	–	–	-	81.3	72.4	–	–	- -	–	–	-
MISA 2020 (V+A+T^{gt}) (Hazarika et al., 2020)	–	–	–	–	–	–	66.2	–	69.8	-	–
HKT 2021 (V+A+T^{gt}) (Hasan et al., 2021)	–	–	–	–	–	–	79.4	–	77.4	–	- -
LaughM† 2021 (T^{gt}) (Kayatani et al., 2021)	–	64.2	70.5	**86.5**	76.3	68.6	68.7	71.9	67.6	74.7	59.8
FunnyNet (V+A+T^{gt}) (Liu et al., 2022)	–	73.8	75.8	83.4	78.6	79.5	79.9	84.1	79.9	88.2	85.8
FunnyNet (V+F+A+T^{gt}) (Liu et al., 2022)	–	75.9	78.3	85.2	79.6	83.2	82.0	84.4	80.2	88.8	86.4
FunnyNet-W (V+A+T^{gt})	–	**78.5**	**80.0**	84.6	**80.1**	**85.9**	**84.1**	**84.5**	**80.2**	**89.3**	**86.7**
FunnyNet: V+F+A (Liu et al., 2022)	✓	69.6	74.0	**84.0**	**79.3**	**81.4**	**81.0**	83.7	78.0	86.8	84.8
FunnyNet-W: V+A+T^{a}	✓	**78.2**	**79.1**	83.6	78.9	80.1	**81.0**	**84.2**	**80.3**	**88.2**	**85.6**

Bold values indicate the higher performance compared to others
Modalities used per method A: audio, V: visual frames, T^{gt}: ground truth text (subtitles or transcript), T^{a}: automatically generated text (text extracted from speech), F: face. The column 'Wild' signifies the methods that can run *in the wild*, i.e., automatically without requiring ground truth information either for training or for testing. Note, most methods require ground truth labels (mostly in the form of textual subtitles or transcripts) both for training and testing. This is in contrast to FunnyNet-W which can automatically process videos in the wild. †Reproduced results: we use the exact model as in (Kayatani et al., 2021), pre-train it on Friends and fine-tune it on the other datasets

– **Experiments and analysis.** We provide more insights and content to explain the performance of FunnyNet-W. Specifically, we experimentally demonstrate and discuss the benefits of the new encoders, each modality and their fusion module, of the length of the input time window, of the losses used as opposed to alternative ones (Sect. 5.2). Furthermore, we provide a thorough qualitative and intuitive analysis of each modality and their fusion, as well as failure cases (Sect. 6).
– **In the wild applications.** In addition to experimenting on other domains as in FunnyNet (Sect. 7.1), we perform two in-the-wild applications: first, we compare FunnyNet-W against chatbots based on LLMs and show that relying solely on language with or without prompt engineering is insufficient for detecting funnyness (Sect. 7.2); and second, we replace real speech by synthetic speech and showcase the importance of real vocals for funny moment detection (Sect. 7.3).

2 Related Work

Sarcasm and Humor Detection

Sarcasm and humor share similar styles (irony, exaggeration and twist) but also differ from each other in terms of representation. Sarcasm usually relates to dialogues; hence, most methods detect sarcasm by processing language using human efforts. For instance, Davidov et al. (2010) collects a speech dataset from social media using the hashtag and manual labeling, while others (Rockwell, 2000; Tepperman et al., 2006) study the acoustic patterns related to sarcasm, like slower speaking rates or higher volumes of voice. In contrast, a humorous moment is defined as the moment before laughter (Castro et al., 2019; Hasan et al., 2019). Hence, such methods (Bertero & Fung, 2016; Castro et al., 2019; Hasan et al., 2019; Kayatani et al., 2021; Hasan et al., 2021) process audios to extract laughter for labeling. Nevertheless, for prediction, most such approaches focus solely on language models (Annamoradnejad & Zoghi, 2020; Weller & Seppi, 2020) or on multiple cues including text (Hasan et al., 2019, 2021). For instance, LaughMachine (Kayatani et al., 2021) proposes vision and language attention mechanisms, while MSAM (Patro et al., 2021) combines self-attention blocks and LSTMs to encode vision and text. (Hazarika et al., 2020) use first an advanced BERT (Devlin et al., 2019) model to process long-term textual correlation and then vision for the prediction. Following this, Rahman et al. (2020) propose a Multimodal Adaptation Gate to efficiently leverage textual cues to explore better representation for sentiment analysis. OxfordTVG-HIC (Li et al., 2023) proposes a dataset with 2.9 M image-text pair for humor detection. A few methods also explore audio. For instance, MUStARD (Castro et al., 2019) and URFUNNY (Hasan et al., 2019) process text, audio and frames using LSTM to explore long-term correlations, while HKT (Hasan et al., 2021) classifies language (context and punchline) and non-verbal cues (audio and frame) to

learn cross-attention correlations for humor prediction. They combine audio with other information (video and texts) in a simple feature fusion process without investigating the inter-correlations in depth. Specifically, they stack multimodal features to learn the global weighting parameters without considering the biases in different domains. In contrast, we believe that funny scenes can be triggered by mutual signals from multimodalities; hence, in this work, we explore the cross-domain agreement of cues with contrastive training. Moreover, FunnyNet-W eliminates the need for external textual annotation by relying solely on raw audiovisual cues, and extracts textual cues directly from the audio that naturally accompanies videos.

Sound Event Detection and Laughter detection. Sound event detection aims to identify and timestamp sound events within audio recordings. Most attempts either rely on annotated data (Mesaros et al., 2016) or use source separation techniques (Défossez et al., 2019; Rouard et al., 2023). The choice of input representation is crucial, and most methods use Mel spectrograms (Mesaros et al., 2017; Wang et al., 2021; Niizumi et al., 2021, 2023; Saeed et al., 2021) instead of audio waveforms. This choice is motivated by their computational efficiency, interpretability, and effortless integration into conventional vision models. In our work, we focus on a specific acoustic event: laughter. We leverage these detected laughter as pseudo-labels to train FunnyNet-W.

Laughter detection. The literature in this domain remains relatively scarce. Some methods rely on physiological sensors (Barral et al., 2017; Shimasaki & Ueoka, 2017), while others (Ryokai et al., 2018; Gillick et al., 2021) follow the conventional supervised learning paradigm to train deep neural laughter detectors. Nevertheless, the latter approach requires annotated datasets, a challenging endeavour in the context of this specialized domain. For instance, the authors of Gillick et al. (2021) experiment with the Switchboard dataset (Holliman et al., 1992), which contains manually annotated laughter timestamps from phone conversation, and also manually annotate laughter timestamps from 1000 clips of AudioSet dataset (Gemmeke et al., 2017). In contrast, our laughter detector is unsupervised, robust and straightforward, by leveraging the specific attributes of multichannel audio data. Our method sidesteps the need for complex annotations, presenting a promising alternative within the laughter detection landscape.

Multimodal tasks. Over the past decade, the number of tasks that require multiple modalities has increased either due to their intrinsic multimodal nature or due to the potential performance enhancements of adding extra modalities. Here, we review some approaches that are directly related to our work in terms of multimodality and modality fusion.

Audio+Video. For instance, Gabbay et al. (2018); Afouras et al. (2020) recognize the facial movements to separate the speaker's voice in the audio. Senocak et al. (2018); Tian et al. (2018) temporally align the audio and video using attention to locate the speaker. The former (Senocak et al., 2018) proposes a triplet network to process the query, positive and negative samples to encourage the query to be close with positive samples and far from negative samples. The latter (Tian et al., 2018) collects an Audio-Visual Event (AVE) dataset to better handle audio-visual alignment. Several methods extend this to other applications, such as audiovisual generation (Zhou et al., 2020) that generates an audio-driven talking face from a single source image and pose video.

Video+Language. Several tasks involve combining language and visual–in particular video– modalities. One notable category encompasses video-to-text tasks, including video captioning (Wang et al., 2020; Lin et al., 2022), which entails generating natural language captions for video sequences. A more challenging, yet very similar task is video question answering (Liang et al., 2020; Yang et al., 2023; Zhu et al., 2022), where the goal is to comprehend the content well enough to respond to queries effectively. In contrast, Singer et al. Singer et al. (2022) propose an approach focusing on text-to-video generation. Finally, video-text retrieval (Dong et al., 2021; Bain et al., 2021; Fang et al., 2021) aims to facilitate bidirectional exploration of both video and textual content.

Audio+Language. Numerous research directions focus only on audiovisual modalities. A major audiovisual task lies in speech emotion recognition (Yoon et al., 2018; Priyasad et al., 2020), which aims to connect audio and text to categorize emotions. A speech emotion recognition pipeline consists of modality fusion followed by classification. Parallel to the well-established image-text retrieval task, the domain of audio-text retrieval has also received substantial attention (Lou et al., 2022; Koepke et al., 2022; Xin et al., 2023; Mei et al., 2022). This task employs similar techniques based on measuring feature similarity between the modalities. Another complex audiovisual challenge is audio captioning, where the objective is to generate textual descriptions from acoustic inputs. Most approaches rely on the classical encoder-decoder architecture (Koizumi et al., 2020; Shen et al., 2023; Kim et al., 2023).

Video+Language+Audio. Some works have extended previous tasks by combining the three modalities: audio, video and text. For example, certain approaches incorporate the acoustic modality into the conventional video captioning pipeline (Iashin & Rahtu, 2020; Liu et al., 2023). Additionally, Deng et al. (2018) introduce an acoustic modality to enhance emotion recognition. Rouditchenko et al. (2021) learns a shared audio-visual embedding space directly from raw video inputs via self-supervision. Han et al. (2023) use CLIP (Radford et al., 2021) to align audio-visual signals to produce audio descriptions. Hong et al. (2023) propose a hyperbolic loss to align audio-visual features in a tree-shaped space. All these works show improvements in compari-

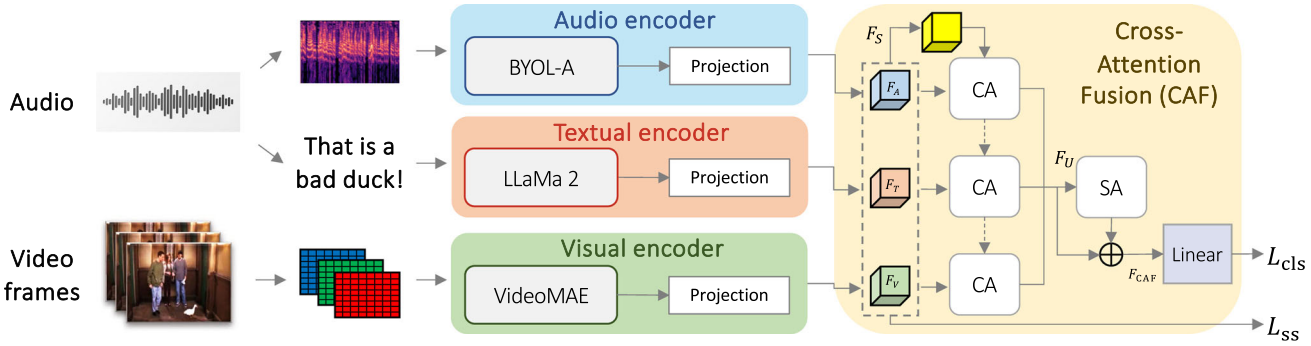

Fig. 2 Architecture of FunnyNet-W. Given audio-visual clips, FunnyNet-W predicts funny moments in videos. It consists of the audio (*blue*), textual (*red*), and visual (*green*) encoders, whose outputs pass through the Cross Attention Fusion (*CAF*), which consists of cross-attention (*CA*) and self-attention (*SA*) for feature fusion. It is trained to embed all modalities in the same space via self-supervision (L_{ss}) and to classify clips as funny or not-funny (L_{cls}) (Color figure online)

son to unimodal baselines. In contrast to previous methods that depend on distinct annotated sources and ground-truth modalities (for instance subtitles for text or ground-truth annotations), our proposed FunnyNet-W extracts multiple additional modalities– audio and text– from a single modality source, namely video, using non-perfect extraction techniques, such as speech-to-text models.

Modality alignment. Recently many works (Radford et al., 2021; Guzhov et al., 2022; Girdhar et al., 2023) have shown promising efforts for acquiring shared multimodal embeddings by leveraging large-scale datasets. Notably, the first breakthrough of text and image embedding was achieved with CLIP (Radford et al., 2021). Comparable milestones have been reached in diverse modalities; for instance, Morgado et al. (2021) proposes to learn a powerful audio-visual representation from videos. Other works extend the original CLIP language-image representation with new modalities such as audio in Guzhov et al. (2022), or video in Lin et al. (2022), Xue et al. (2023). Recently, the ImageBind (Girdhar et al., 2023) unifies six distinct modalities into a shared embedding space. The key to success consists in aligning features from the different modalities.

Attention mechanisms (Vaswani et al., 2017) is natural for connecting multimodal signals. For instance, Wei et al. (2020) employ cross-attention to model inter- and intra-modality relationships, Tan et al. (2021) leverage contrastive cross-attention, Jaegle et al. (2021) and Lee et al. (2020) use iterative cross-attention. In addition, (Nagrani et al., 2021) introduced attention bottlenecks with randomly initialized bottleneck tokens for modality fusion. In contrast, our fusion mechanism builds on this idea but differs by (i) employing modality projections as bottlenecks and (ii) integrating an extra self-attention block to capture fused token correlations. These works illustrate the natural strength of attention mechanisms in aligning multiple modalities within a unified space. Numerous multimodal tasks benefit from this capability of modality alignment. Applications such as summarization (Narasimhan et al., 2021), retrieval (Gabeur et al., 2020; Bain et al., 2021), audiovisual classification (Nagrani et al., 2021), predicting goals (Epstein & Vondrick, 2021), human replacement (Dufour et al., 2022). Tan et al. (2021) and Wei et al. (2020) iteratively apply self and cross-attention to explore correlations among modalities. Instead, FunnyNet-W both fuses all modalities and in parallel learns the cross-correlation among different modalities; this avoids any biases that may be caused by one dominant modality.

3 Method

Here, we present FunnyNet-W, its training process and losses (Sects. 3.1–3.2). For training labels, we propose an unsupervised laughter detector (Sect. 3.3).

Overview. FunnyNet-W consists of (i) three encoders: the visual encoder with videos as input, the audio encoder with audio as input, and the text encoder with subtitles as input. To parse the subtitles, we use an automatic speech recognition (ASR) system (Radford et al., 2022); (ii) the proposed Cross-Attention Fusion (CAF) module, which explores cross- and intra-modality correlations by using cross- and self-attentions in the encoders' outputs. Then, the fused feature is fed to a binary classifier. The overall architecture is illustrated in Fig. 2. FunnyNet-W is trained to embed all modalities in the same space via self-supervised contrastive loss and to classify clips as funny or not. For training, we exploit laughter that naturally exists in TV Shows: we define it as 'funny-moment' for any audiovisual snippet followed by laughter; and 'not-funny' for any audiovisual snippet not followed by laughter.

3.1 FunnyNet-W Architecture

FunnyNet-W utilizes raw inputs from videos, including the audio waveform and frames.

Audio Encoder. First, the audio waveform is transformed into a Mel spectrogram.[2] This spectrogram, denoted as $\mathbf{X}_{\text{audio}}$, is then passed through an audio encoder to generate a 1D feature vector. Finally, a projection head is applied to obtain a N-dimensional vector $\mathbf{F}_{\text{A}} \in \mathbb{R}^N$.

Text Encoder. The corresponding transcripts, denoted as X_{text}, are extracted from the audio waveform using an automatic speech recognition model (Radford et al., 2022). These transcripts are then encoded into a feature vector using the text encoder. Subsequently, a projection is performed to obtain a N-dimensional vector $\mathbf{F}_{\text{T}} \in \mathbb{R}^N$.

Visual Encoder. The visual encoder employs an architecture based on the transformer to process video frames. The input frames, denoted as $\mathbf{X}_{\text{visual}}$, are divided into patches from several consecutive frames. Unlike conventional approaches that use a 'classification token' to obtain a general representation, we compute the representation by averaging the pooled features from all patches. This process results in a feature vector, which is then projected to a N-dimensional vector $\mathbf{F}_{\text{V}} \in \mathbb{R}^N$ using a projection head. The video context complements the audio in providing richer content (Hasan et al., 2019). Additionally, in the absence of sound and transcripts, visual cues can also elicit laughter.

Projection Head. This module consists of two linear layers separated by a GeLU activation function. Dropout and normalization layers are applied after the linear layers. It takes the features outputted from each encoder and projects them into a shared N-dimensional multimodal feature space.

Cross-Attention Fusion (CAF). It learns the cross-domain correlations among vision, audio and text (yellow box Fig. 2). It consists of (a) three cross-attention (CA) and (b) one self-attention (SA) modules, described below:

– **Cross-attention** is used in cross-domain knowledge transfer to learn across-cue correlations by attending the features from one domain to another (Mohla et al., 2020; Nam et al., 2017; Wei et al., 2020). In CAF, it models the relationship among vision, audio, and textual features. We stack all features as $\mathbf{F}_{\text{S}} \in \mathbb{R}^{3\times 512}$, and then feed $\mathbf{F}_{\text{S}}$ into three cross-attention modules to attend to vision, text, and audio, respectively (Fig. 2). Next, the scaled attention per modality is computed as $\sigma\left(\frac{\mathbf{Q_S K_i^T}}{\sqrt{d}}\right)\mathbf{V_i}$, where $i=\{V, T, A\}$ for {vision, text, audio}, and σ the softmax. The query $\mathbf{Q}$ comes from the stacked features: $\mathbf{Q}_{\text{S}}=\mathbf{F}_{\text{S}}\mathbf{W}^{\text{Q}_\text{S}}$, while the key $\mathbf{K}$ and value $\mathbf{V}$ come from a single modality as $\mathbf{K_i}=\mathbf{F_i}\mathbf{W}^{\text{K}_i}$, and $\mathbf{V_i}=\mathbf{F_i}\mathbf{W}^{\text{V}_i}$. Next, we obtain three cross-attentions and sum them to a unified feature $\mathbf{F}_{\text{U}}$ as:

$$\text{F}_{\text{U}} = \sum_{i\in\{V,F,A\}} \sigma\left(\frac{\mathbf{Q_S K_i^T}}{\sqrt{d}}\right)\mathbf{V_i} \quad . \tag{1}$$

– **Self-attention** computes the intra-correlation of the F_{U} features, which are further summed with a residual F_{U} as:

$$\text{F}_{\text{CAF}} = \mathbf{F}_{\text{U}} + \sigma\left(\frac{\mathbf{Q_U K_U^T}}{\sqrt{d}}\right)\mathbf{V_U} \quad , \tag{2}$$

where $\mathbf{Q_U}=\mathbf{F_U}\mathbf{W}^{\text{Q}_\text{U}}$, $\mathbf{K_U}=\mathbf{F_U}\mathbf{W}^{\text{K}_\text{U}}$, $\mathbf{V_U}=\mathbf{F_U}\mathbf{W}^{\text{V}_\text{U}}$. Finally, we average F_{CAF} tokens and feed it to a classification layer.

Discussion. CAF differs to existing methods (Mohla et al., 2020; Wei et al., 2020) in the computation of the cross attention. Using stacked features F_{S} to attend to each modality Q_{S} brings three benefits: (a) it is order-agnostic: for any modality pair we compute cross-attention once, instead of twice by interchanging queries and keys/values; this results in reduced computation; (b) each modality serves as a query to search for tokens in other modalities; this brings rich feature fusion; and (c) it generalizes to any number of modalities, resulting in scalability[3].

3.2 Training Process and Loss Functions

Subtitle extraction. To extract transcripts from the raw waveform, we use the WhisperX system (Bain et al., 2023). WhisperX enforces alignment of the automatic speech recognition model Whisper (Radford et al., 2022) with an external voice activity detection model to produce accurate word-level timestamps. This approach results in a Time-Accurate Speech Transcription, very similar to manually transcribed subtitles.

Positive and Negative Samples. To create samples, we exploit the laughter that naturally exists in episodes. We define as 'funny' any n-sec clip followed by laughter;

[2] Mel spectrogram is a 2D acoustic time-frequency representation of sound.

[3] Note that although the CAF module scales linearly with the number of modalities, the total training time complexity is increased quadratically with the number of modalities ($\mathcal{O}(d^2)$), because all modality pairs are taken into consideration when computing the loss (see Eq. 4). However, not all loss pairs are necessary if one modality plays a dominant role, then we can skip some loss pairs. Our experiments in Table 5 also show the importance of different modalities. For instance, the increase in F1 from the single text-only T model to the $T + A$ model is 7%, while the increase of F1 from the $T + A$ model to the $T + A + V$ model is 3.7%. This indicates that it may not be necessary to use more modalities if the information they offer is overlapping and not complementary.

'not-funny' any n-sec clip not followed by laughter. More formally, given a laughter at timestep (t_s, t_e), we extract a n-sec clip at $(t_s - n, t_s)$ and we split it into audio and video. For each video, we sample n frames (1 FPS). For the audio, we resample it at 16,000 Hz and transform it to Mel spectrogram. Thus, each sample corresponds to n sec and consists of a Mel spectrogram for the audio and a n-frame long video. In practice, we use 8-sec clips as the average time between two canned laughters, and it also leads to better performances (ablations of n-sec clips and n-frames per clip in supplementary). Note that we clip the audio based on the starting time of the laughter so the positive samples do not include any laughter.

Self-Supervised Contrastive Loss. To capture 'mutual' audiovisual information, we solve a self-supervised synchronization task (Chung & Zisserman, 2016; Korbar, 2018; Owens & Efros, 2018): we encourage visual features to be correlated with true audios and uncorrelated with audios from other videos. Given the i-th pair of visual v^i and true audio features a^i and N other audios from the same batch: $a_1, ..., a_N$ we minimize the loss (Chen et al., 2020; Chung et al., 2019; Oord et al., 2018):

$$L_{\text{cotrs}} = -\log \frac{\exp(\text{S}(v^i, a^i)/\tau)}{\sum_{j=1}^{N} \exp(\text{S}(v^i, a^j)/\tau)} \quad , \tag{3}$$

where S the cosine similarity and τ the temperature factor. Equation (3) accounts for audio and visual features. Here, we compute the contrastive loss between all three modalities, i.e., visual-audio, text-audio, and visual-text. Thus, our self-supervised loss is:

$$L_{\text{ss}} = -\frac{1}{3}\left(L_{\text{cotrs}}^{v^i,a^i} + L_{\text{cotrs}}^{v^i,t^i} + L_{\text{cotrs}}^{t^i,a^i}\right). \tag{4}$$

Final Loss. FunnyNet is trained with a Softmax loss Y_{cls} to predict if the input is funny or not, and the L_{ss} to learn 'mutual' information across modalities. Thus, the final loss is:

$$L = \lambda_{\text{ss}} L_{\text{ss}} + \lambda_{\text{cls}} L_{\text{cls}} \quad , \tag{5}$$

where λ_{ss}, λ_{cls} the weighting parameters that control the importance of each loss.

3.3 Unsupervised Laughter Detection

To detect funny moments, we design an unsupervised laughter detector consisting of 3 steps (Fig. 3).

1. **Remove Voices.** Background audios include sounds, music, laughter; instead, voice (speech) is part of the foreground audio. We remove voices from audios by exploiting multichannel audio specificities. Given raw waveform audios, when the audio is stereo (two channels), the voices are centered and are common in both channels (Huber & Runstein, 2012); hence, by subtracting the channels, we remove the voice and keep the background audio. In surround tracks (six channels), we remove the voice channel (Huber & Runstein, 2012) and keep the background ones.
2. **Background Audios.** The waveforms from (i) are mostly empty with sparse peaks corresponding to audio: laughter and music. To split them into background and empty segments, we use an energy-based peak detector[4] that detects peaks based on the waveform energy. Then, we keep background segments and convert them to log-scaled Mel spectrograms.
3. **Cluster Audio Segments.** For each laughter and music segment, we extract features using a self supervised pretrained encoder. Then, we cluster all audio segments using K-means to distinguish the laughter from the music ones.

3.4 Differences to the ACCV 2022 version (Liu et al., 2022)

FunnyNet (Liu et al., 2022) and FunnyNet-W use multimodal input signals from videos and identify whether a video input is funny or not. Here, we describe the three main methodological differences between the two models.

FunnyNet (Liu et al., 2022) uses three encoders: (a) visual encoder for video frames (Timesformer (Bertasius et al., 2021)) (b) audio encoder for voice and background audio (BYOL (Niizumi et al., 2021)) and (c) face encoder for facial expressions (ResNet (Schroff et al., 2015)). However, the face encoder is cumbersome as it requires an external model for face detection, thus leading to higher runtime and decreasing its applicability. For this reason, in FunnyNet-W we remove the face encoder, which leads to slight performance drops but increased gains in applicability and scalability. Moreover, FunnyNet-W uses the more modern visual encoder VideoMAE.

Furthermore, for FunnyNet-W, we made the following three-fold observation. First, most funny moments are inevitably related to language. Second, recent advances in Automatic Speech Recognition (ASR) (Radford et al., 2022; Bain et al., 2023) have rendered it possible to exploit the vocal part of the audio (i.e., the part where people are speaking) and automatically transcribe existing dialogues. Third, the recent explosion of Large Language Models (LLMs) offers remarkable capabilities in processing text and dialogues across a wide range of tasks. Combining these would mean transcribing dialogues via ASR *for free*, then using an LLM encoder

[4] https://github.com/amsehili/auditok.

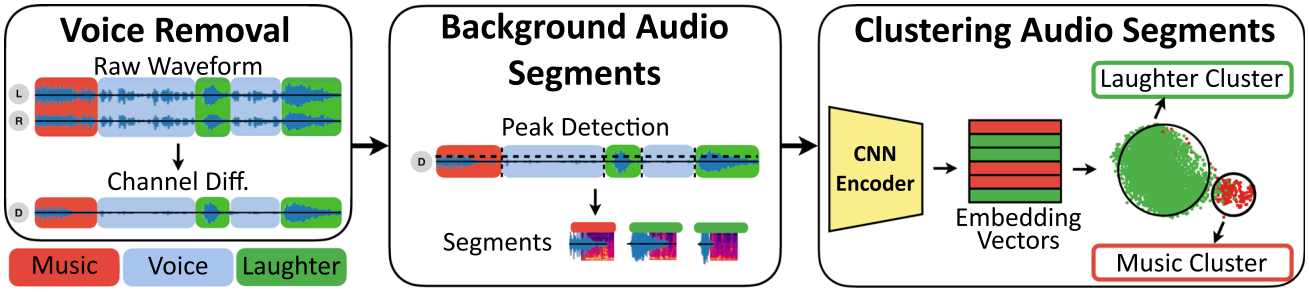

Fig. 3 Proposed laughter detector. It takes raw waveforms as input and consists of (i) removing voices by subtracting channels (here, the audio is stereo with 2 channels), (ii) detecting peaks, and (iii) clustering audios to music and laughter

to detect funniness. However, FunnyNet does not rely on textual data. We consider this a wasted opportunity, as the textual data via LLMs can boost the representation capability of the model and, in turn, its performance. To this end, FunnyNet-W differs from FunnyNet in two aspects: it relies on ASR to transcribe dialogues *for free* and it uses an LLM text encoder for processing the text (Llama-2 (Touvron et al., 2023)).

4 Datasets and Metrics

Datasets. We use five datasets.

- **The Big Bang Theory (TBBT)** dataset (Kayatani et al., 2021) contains 228 episodes of *TBBT* TV show: (183,23,22) for (train,val,test). All episodes come with video, audio and subtitles, labelled as humor (or non) if followed (or not) by laughter.
- **Multimodal Humor Dataset (MHD)** (Patro et al., 2021) contains episodes from the TV show *TBBT*, with 110 episodes split (84,6,20) for (train,val,test) (disjoint splits to TBBT). It contains multiple modalities; the subtitles are tagged as humor (or not).
- **MUStARD** (Castro et al., 2019) contains 690 segments from four TV shows with video-audio-transcript labelled as sarcastic or not.
- **UR-Funny** (Hasan et al., 2019) contains 1866 TED-talk segments with video-audio-transcript labelled as funny or not.
- **Friends** (Brown et al., 2021; Kalogeiton & Zisserman, 2020) contains all 25 episodes (∼23 min) from the third season of *Friends* (∼10 h). We split them into 15 training (1–15), 5 validation (16–20) and 5 test episodes (21–25). Each episode comes with video, audio, face, body, voice tracks and features with speaker identifiers. In this work, we enrich this dataset by providing manually annotated laughter time codes. These annotated laughter time codes consist of time-stamps of the start and the end of all canned or not laughter. This results in 3.5k time-codes, with an average duration of 3 sec (0.3−16.5 sec), 138 average number of laughter per episode (109 to 182). The annotations are available: https://www.lix.polytechnique.fr/vista/projects/2024_ijcv_liu/.

Metrics. To evaluate **FunnyNet-W**, we use classification accuracy (Acc) and F1 score (F1).
For **laughter detector**, we use sample-scale at the detection level and frame-scale at the temporal level to compute precision (Pre), recall (Rec) and F1.

5 Experiments

In this section, we provide experiments for FunnyNet-W. First, we compare to the state of the art (Sect. 5.1), then we provide an ablation of each component of FunnyNet-W (Sect. 5.2), and finally, we ablate our unsupervised laughter detector (Sect. 5.3).
Implementation Details. We train FunnyNet using Adam optimizer (Loshchilov & Hutter, 2019) with a learning rate of 1×10^{-4}, batch size of 32 and Pytorch (Paszke et al., 2019). The input audio is first downsampled by fixed sampling frequency (16,000 Hz) and then transformed to log-scaled Mel spectrogram by mel-spaced frequency bins F = 64. At training, we use data augmentation: for frames, we randomly apply rotation and horizontal/vertical flipping, and randomly set the sampling rate to 8 frames; for audios, we apply random forward/backward time shifts and random Gaussian noises. For subtitles, we tokenize them as $max\_length = 64$ inputs and send them to the language models.
Setting. In our experiments, we train FunnyNet-W on Friends. For MUStARD and UR-Funny, we fine-tune FunnyNet-W on their respective train sets. For TBBT and MHD, we fine-tune it only with a subset of the training set from TBBT (32 random episodes) These datasets come with data samples of uneven lengths. If the sample length is larger than 8 s (our best setting), we crop the last 8-second sequence to fit our model; otherwise, we pad zeros to the end. For UR-FUNNY, we exclude from training the data samples with no

sounds. For audio, textual, and visual encoders, we used the corresponding pre-trained models for feature extraction.

To understand the funny moments in the wild, we consider that subtitles do not come naturally with other modalities. Different from text-driven funny detections (Castro et al., 2019; Hasan et al., 2021, 2019), we instead use off-the-shelf audio-to-text models, like WhisperX (Bain et al., 2023), to automatically generate texts from audios for funny detection. Hence, in addition to audiovisual data, we experiment with both real and automatically-generated texts (in the form of subtitles) for funny moment detection.

5.1 Comparison to the State of the Art

Here, we evaluate FunnyNet on five datasets: TBBT, MHD, MUStARD, UR-Funny and Friends and compare it to the state of the art: MUStARD (Castro et al., 2019), MSAM (Patro et al., 2021), MISA (Hazarika et al., 2020), HKT (Hasan et al., 2021) and LaughM (Kayatani et al., 2021). Table 1 reports the results (including random, positive and negative baselines) for both metrics. We indicate the modalities each method uses as A: audio, V: video, T^{gt}: ground-truth text, T^{a}: automatically-generated text (speech-to-text), and F: face. Furthermore, we also indicate in the 'Wild' column the methods that can run automatically without requiring ground-truth information (either for training or testing). Note that most methods require ground truth labels (mostly in the form of textual subtitles or transcripts) either for training or testing (T^{gt}). This is in contrast to FunnyNet-W, which can automatically process videos in the wild by exploiting speech-to-tex models (T^{a}).

The first part of Table 1 (no wild) demonstrates that overall the proposed FunnyNet-W (V+A+T^{gt}) outperforms all methods on all five datasets. For TBBT it outperforms the LaughM by a notable margin of +10% for F1 and Acc, and FunnyNet by +3% in both metrics. For MHD, it outperforms MSAM by 3% in F1 and 7% in Acc, LaughM and FunnyNet by 3% and 1%, respectively, in Acc. Furthermore, FunnyNet-W outperforms MUStARD, MISA, HKT, LaughM and FunnyNet by 3–12% in F1 and 2–15% in Acc for MUStARD and 1–15% in F1 and 1–10% in Acc for UF-Funny. For Friends, we observe similar patterns, where we outperform LaughM by 15% in F1 and 26% in Acc and FunnyNet by approximately +1% in both F1 and Acc. These results confirm the effectiveness of FunnyNet compared to other methods.

The major advantage and motivation of FunnyNet-W and its predecessor FunnyNet (Liu et al., 2022) is the fact that they can run *in the wild*, i.e. without requiring ground truth data either at training or test time. To this end, the second part of Table 1 reports results when experimenting in the wild. We observe that for TBBT, remarkably FunnyNet-W outperforms its predecessor FunnyNet by 5–10% in F1 and Acc, while for MHD it is inferior by 1% or similar for MUStARD. For UR-Funny and Friends, FunnyNet-W outperforms FunnyNet consistently by 1–3% in all metrics. When we compare FunnyNet-W-T^{a} to the first part of the table, we observe that it still produces on par or superior results to all other methods. This clearly shows the superiority of FunnyNet-W even when compared to methods that have access to manually annotated ground-truth data.

Our remarks are: First, FunnyNet-W outperforms most methods in both metrics in both settings, when using ground truth text (T^{gt}) or when being *in the wild* (T^{a}). Second, the performance in the out-of-domain UR-Funny is significantly high. Third, for TBBT and MHD our results are much less optimized than the ones from LaughM or MSAM, as we do not have access to the exact same test videos as either work, so inevitably there are some time shifts or wrong labels[5] and we use much fewer training data (32 vs 183 episodes in LaughM vs 84 episodes in MHD). These highlight that FunnyNet-W is an effective model for funny moment detection.

Note that in the remainder of this work, unless stated otherwise, using the automatically-generated text (in the form of subtitles) is the **default** setting of FunnyNet-W. For simplicity, we denote the T^{a} by T.

5.2 Ablation of FunnyNet-W

In this section, we provide ablations of FunnyNet-W. Specifically we ablate the encoders (Sect. 5.2.1), the modalities (Sect. 5.2.2), the cross attention fusion module (Sect. 5.2.3), the length of input videos (Sect. 5.2.4) and the losses (Sect. 5.2.5).

5.2.1 Ablation of Encoders

Visual encoder. Table 2 ablates two video encoders on Friends, i.e. Timesformer (Bertasius et al., 2021) and VideoMAE (Tong et al., 2022) for two scenarios: one using automatically generated text (T^{a}) and when using ground-truth text (T^{gt}). Given the same video sequence, we use the best settings for them (8 frames for Timesformer and 16 frames for VideoMAE). We observe that using VideoMAE outperforms Timesformer by about 1–3% in F1 score and 2–3% in Acc. This is expected because VideoMAE is a larger model, and it also uses a masked autoencoder for unsupervised learning; hence it can generalize better than Timesformer. When comparing the results between using ground truth and automatically generated texts, we observe that the improvements of using VideoMAE are consistent, and the differences are very small (1–2% in both F1 and Acc).

Text encoder. Table 3 ablates three different text encoders in FunnyNet-W on Friends: Bert (Devlin et al., 2019), GPT2

[5] The label time shift is 0.3–1 s on TBBT and 0.3–2 s on v2.

(Solaiman et al., 2019) and LlaMa-2 (Touvron et al., 2023) (7B model) for two scenarios (one using automatically generated text (T^a) and when using ground-truth text (T^{gt})). Given the other ablation study, we choose VideoMAE and BYOL-A as the best visual and audio encoders, respectively. We observe that using LlaMa-2 gives the best improvements in both F1 and Acc. Interestingly, using GPT2 results in inferior performance than using Bert. This finding is consistent with what we observe in LLM models. For LlaMa-2, we note that the differences between using ground truth and automatically generated texts are minor, about 0.8–1% in both F1 and accuracy.

Audio encoder. Table 4 ablates four audio encoders on Friends: Beats (Chen et al., 2023), CAV-MAE (Gong et al., 2023), BYOL-A-v2 (Niizumi et al., 2023) and BYOL-A (Niizumi et al., 2021). Given the previous ablation studies, we choose VideoMAE and LlaMa-2 as the best visual and text encoders and operate directly with automatically generated text (T^a). The results show that CAV-MAE, BYOL-A-v2 and BYOL-A perform on par (approximately 1% difference in F1 and Accuracy). In our experiments, we use BYOL-A as it results in the best F1 and Accuracy but it also requires fewer parameters than the other models.

Subtitles sources. Tables 2 and 3 report results for two scenarios: one using automatically generated text (T^a) and when using ground-truth text (T^{gt}). Consistently, we observe that using the ground truth text outperforms using the automatically-generated one. This is expected, as T^a includes imperfect transcripts. We note, however, that the difference in both F1 and Accuracy are minor (1–3% for both metrics). This highlights that substituting ground-truth with an automatic speech-to-text model is a good trade-off between good performance and the ability to run in the wild, i.e., without requiring manual ground truth labels.

Table 2 Ablation of visual encoders on Friends

Modality			F1	Acc
A	V	T^a		
BYOL-A	Timesformer	Bert	84.2	80.9
	VideoMAE		85.3	82.3
A	V	T^{gt}		
BYOL-A	Timesformer	Bert	84.9	80.8
	VideoMAE		87.2	83.8

A: audio, V: visual frames, T^{gt}: ground truth text, T^a: automatically generated text (text extracted from speech)

Table 3 Ablation of text encoders on Friends

Modality			F1	Acc
A	V	T^a		
BYOL-A	VideoMAE	Bert	85.3	82.3
		GPT2	85.2	82.3
		LlaMa-2	88.2	85.6
A	V	T^{gt}	F1	Acc
BYOL-A	VideoMAE	Bert	87.2	83.8
		GPT2	88.1	85.6
		LlaMa-2	89.3	86.8

A: audio, V: visual frames, T^{gt}: ground truth text, T^a: automatically generated text (text extracted from speech)

Table 4 Ablation of audio encoders on Friends

Modality			F1	Acc
A	V	T^a		
BEATS	VideoMAE	LlaMa-2	78.2	65.1
CAV-MAE			87.3	83.8
BYOL-A-v2			87.6	84.7
BYOL-A			88.2	85.6

A: audio, V: visual frames, T^{gt}: ground truth text, T^a: automatically generated text (text extracted from speech)

5.2.2 Ablation of Modalities

Table 5 ablates all modalities of FunnyNet-W on the Friends test set. Using text alone (third row) produces better results than when using the visual or audio modality alone (first and second rows). This highlights the efficiency of large dataset pre-training and the representation power of Large Language Models (since we use LlaMa 2 as the textual encoder). Using audio alone (second row) leads to the second-best performance compared to using single modalities, underlying that audio is more suitable than visual cues for our task, as it encompasses the way of speaking (tone, pauses). Combining modalities outperforms using single ones: combining visual and audio (fourth row) or visual and text (sixth row) increases the F1 by approximately 1.3–10% and the Acc by 0.2–15%. This is expected as audio or text bring complementary information to the visual modality (Morgado et al., 2021; Radford et al., 2021) and their combination helps discriminate funny

Table 5 Ablation of modalities of FunnyNet-W on Friends test set

Modality			F1	Acc
V	A	T		
✓	–	–	73.2	64.1
–	✓	–	73.7	66.6
–	–	✓	77.8	68.1
✓	✓	–	84.3	79.3
–	✓	✓	84.5	80.3
✓	–	✓	74.9	64.3
✓	✓	✓	**88.2**	**85.6**

Bold values indicate the higher performance compared to others

Table 6 Ablation of CAF of FunnyNet-W on Friends test set. (A: audio, V: visual frames, T: text)

CAF		A+V		A+T		V+T		A+V+T	
Self	Cross	F1	Acc	F1	Acc	F1	Acc	F1	Acc
–	–	80.1	76.5	81.0	76.9	73.5	63.8	82.4	77.8
✓	–	81.1	77.3	81.4	77.5	74.4	64.4	85.7	81.8
–	✓	83.6	78.7	82.3	78.7	74.6	64.2	85.4	81.4
✓	✓	**84.3**	**79.3**	**84.5**	**80.3**	**74.9**	**64.3**	**88.2**	**85.6**
MMCA (Wei et al., 2020)		83.1	78.3	83.4	79.8	73.6	63.8	87.0	84.5
CoMMA (Tan et al., 2021)		83.5	78.5	83.9	**80.3**	74.2	64.1	87.6	85.1

Bold values indicate the higher performance compared to others

moments. Combining audio and text (fifth row) leads to larger boosts than audio+visual or text+visual (fourth and sixth rows), as audio and text contain complementary information regarding character dialogues, expression in voices and background music. Overall, using all modalities achieves the best performance.

5.2.3 Ablation of Cross-Attention Fusion (CAF)

Table 6 reports results with various cross- and self-attention fusions in CAF. We observe that including either self- or cross-attention (second, third rows) brings improvements over not having any (first row), indicating that they enhance the feature representation. The fourth row shows that using them both for feature fusion leads to the best performance. For completeness, we also compare CAF against the state of the art: MMCA (Wei et al., 2020) and CoMMA (Tan et al., 2021). All CAF, MMCA and CoMMA use self and cross-attentions jointly for feature extraction. Their main difference is that both MMCA and CoMMA first use self-attention to individually process each modality, then concatenate all modalities together and process them using cross-attention to output the final feature representation. Instead, CAF uses cross-attention to gradually fuse one modality with the rest of the modalities to fully explore cross-modal correlations. The results (fourth, fifth, and last rows) show that CAF outperforms MMCA (Wei et al., 2020) and CoMMA (Tan et al., 2021) by 0.1-−0.4 in F1 score and 0.03-−0.2 in accuracy. This reveals the importance of the gradual modality fusion, and hence the superiority of CAF.

5.2.4 Impact of Time

In this section, we examine the impact the length of the time window has on the final results, as well as the number of sampled frames within the time window.

Influence of Time Window. Following (Bertasius et al., 2021), our proposed FunnyNet-W is trained on fixed-length inputs of multiple modalities that last 8 s. Here, we examine the impact that the length of time window has on FunnyNet-W and illustrate results on four datasets (as well as their average in a dashed red line) in Fig. 4. For this, we use input time windows of varying lengths (from 2 to 16 s) in either the visual encoder of FunnyNet-W (referred to as FunnyNet-W V, top in Fig. 4) or the audio encoder of FunnyNet-W (referred to as FunnyNet-W A, bottom in Fig. 4).

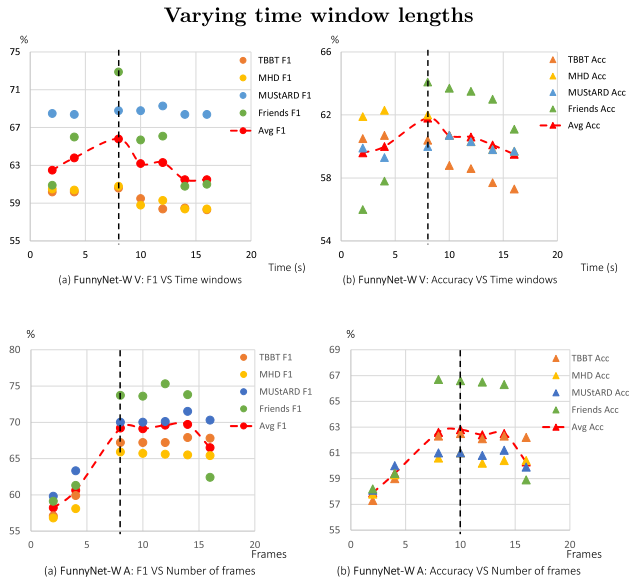

Fig. 4 Comparison of various time window lengths used as input of the (top) visual encoder of FunnyNet-W (referred to as FunnyNet-W V) and (bottom) audio encoder of FunnyNet-W (referred to as FunnyNet-W A). We illustrate *(left,a)* the F1 score and *(right,b)* the accuracy on different datasets. The average results are plotted in red lines (Color figure online)

When ablating the input length of the visual input (top in Fig. 4), we observe that using approximately 8 s achieves the best performance compared to all other settings. Specifically, for F1 (a, left), we observe that for all datasets, the best result is achieved when using 8 s, whereas the second and third results are achieved when using 10 and 12 s length of the input. For Accuracy (b, right), the performance follows the same trend: the best accuracy is reached for 8-second inputs, while the 10 and 12-second inputs reach the second and third-best accuracies. Interestingly, for both F1 and Accuracy, for the average amongst all datasets (red dashed

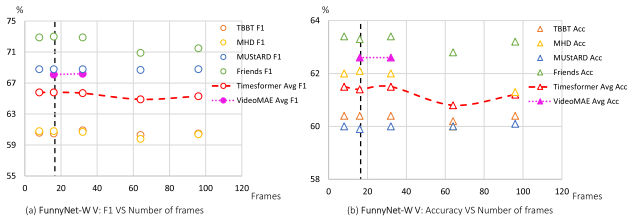

Fig. 5 Comparison of different lengths of time windows for the visual encoder of FunnyNet-W (referred to as FunnyNet-W V). We illustrate *(a)* the F1 score and *(b)* the accuracy on different datasets. The average results are plotted in red points and lines for Timesformer and magenta for VideoMAE (Color figure online)

lines), we observe that both metrics degrade when using longer visual input windows (e.g. more than 15 s). This is probably because longer inputs contain too much visual or audio information across both positive and negative samples, which confuses the model and leads to more incorrect predictions.

When ablating the input length of the audio input (bottom in Fig. 4), we observe that similar to the previous conclusions, the time window of 8 s leads to the best performance both in F1 and Accuracy. Nevertheless, using a longer time window improves the prediction accuracy, in contrast to the visual ablation. Specifically, the best time window setting is between 8 and 12 s. For any time windows outside this range, the performance is getting worse.

In our experiments, we use a time window of 8 s as a good trade-off between the performance of the visual and audio encoders.

Influence of Sampled Frames. Given the input time window of 8 s, we test the scenario where we sample different numbers of frames within a fixed 8-second time window. In particular, we examine the impact when sampling from 8 to 100 frames. The results are shown in Fig. 5, where we illustrate the (left, a) F1 score and (right, b) accuracy over the number of sampled frames. Our results suggest that the number of frames has no or only a trivial impact on the final performance. This is expected, since sampling more frames in a fixed time window mainly produces redundancy without introducing new relevant information. Furthermore, in this ablation, we also compare the results obtained when using Timesformer (red points and dashed line) and VideoMAE (magenta points and dashed line) in the visual encoder. We observe that using VideoMAE outperforms Timesformer in all settings, hence the final FunnyNet-W uses VideoMAE for the visual encoder.

5.2.5 Ablation of Losses

FunnyNet-W uses the classification L_{cls} and the self-supervised contrastive losses L_{ss}. Here, we examine their impact by training FunnyNet-W with and without L_{ss}. Table 7 reports the results on Friends, where we observe

Table 7 Ablation of losses used to train FunnyNet-W

L_{cls}	L_{ss}	L_{koleo}	L_{clip}	F1	Acc
✓	–	–	–	70.9	68.0
✓	✓	–	–	**88.2**	**85.6**
✓	✓	✓	–	86.7	84.6
✓	✓	–	✓	87.3	84.6
✓	✓	✓	✓	85.4	82.2

Bold values indicate the higher performance compared to others

that adding L_{ss} improves over +10 in all metrics (first two rows). This reveals that using the auxiliary self-supervised task of syncing audiovisual data helps to identify the funny moments in videos.

Recently, Koleo (Sablayrolles et al., 2019) (L_{koleo}) and CLIP (Radford et al., 2021) (L_{clip}) have been proposed for improving unsupervised feature clustering. To examine the impact of these two losses, we train FunnyNet-W with different loss combinations and show the results in Table 7. We observe that including Koleo and/or CLIP losses (third-fifth rows) results in a small drop in both F1 and accuracy compared to the proposed loss configuration (second row). Regarding the Koleo loss, this drop is probably because Koleo encourages a uniform span of the features within a batch which maximizes the variances of features and affects the binary decisions on the boundaries. Regarding the CLIP loss, the drop can be explained by the fact that CLIP is widely used for multi-class feature projection, which may complicate the funny or not-funny classification

Model complexity. We also compare in Table 8 the complexity of FunnyNet and FunnyNet-W to the other state-of-the-art models. Note that both models use pre-trained visual, audio and text encoders. For completion, we also report the metrics when including the complexity of the visual, audio and text backbone encoders. We observe that the gain in performances and the unsupervised aspect of FunnyNet-W impacts its complexity. Indeed, FunnyNet-W is a huge model, with an increase of approximately 52 GFLOPS, 16M of parameters and 11ms on runtime, in comparison to the second-heaviest model (Hazarika et al., 2020). Additionally, when comparing FunnyNet to FunnyNet-W, the latter replaces the face encoder with a text encoder and uses larger visual and text encoders, VideoMAE and LlaMa2, respectively. These lead to higher complexity on GFLOPS and parameters. However, the overall inference time is reduced because it does not require online per-frame face detection, masking, and feature extraction.

5.3 Analysis of Unsupervised Laughter Detector

Comparison to the state of the art. We compare our laughter detector with the state of the art: LD (Ryokai et al.,

Table 8 Comparison to the state of the art of FLOPs count (FLOPs), number of parameters (Params) and inference runtime average (Runtime)

Model	FLOPs (10^9)	Params (10^6)	Runtime (ms)
MISA 2020 (V+A+T) (Hazarika et al., 2020)	138.8	111.2	33.64
HKT 2021 (V+A+T) (Hasan et al., 2021)	7.6	16.8	25.91
LaughM 2021 (T) (Kayatani et al., 2021)	2.5	112.4	11.15
FunnyNet^b	4.4	39.5	45.25
FunnyNet (V+F+A)	190.9	126.9	45.25
FunnyNet-W^b	3.1	29.6	30.21
FunnyNet-W (V+F+T)	72,190	70228	30.21

We report two versions of model complexity, FunnyNet (V+F+A) and FunnyNet-W (V+F+T) with including pre-trained encoders, and FunnyNet^b and FunnyNet-W^b without including pre-trained encoders

Table 9 Evaluation of laughter detection on friends

	Temporal				Det IoU = 0.3			Det IoU = 0.7		
	Acc	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
LD (Ryokai et al., 2018)	43.6	35.7	**99.0**	52.3	25.7	22.1	23.4	4.0	3.7	3.8
RLD (Gillick et al., 2021)	74.5	58.9	62.0	59.7	66.2	53.7	59.1	18.5	15.0	16.5
Ours Wav2CLIP (Wu et al., 2021)	77.6	64.5	63.7	63.7	91.3	61.2	73.1	49.7	33.5	39.9
Ours CAV-MAE (Gong et al., 2023)	84.3	75.4	74.2	74.6	92.3	80.0	85.6	52.2	45.5	48.4
Ours BYOL-A (Niizumi et al., 2021)	86.0	76.9	79.4	77.8	94.6	82.3	**87.8**	54.1	47.1	50.3
Ours BYOL-A-v2 (Niizumi et al., 2023)	82.1	67.7	**82.6**	74.1	92.5	**83.5**	87.6	51.9	47.0	49.2
Ours BEATs (Chen et al., 2023)	**86.4**	**78.4**	78.3	**78.1**	**95.2**	81.6	87.7	**55.1**	**47.3**	**50.8**

Bold values indicate the higher performance compared to others
We compare five versions of our laughter detector, denoted as 'Ours', employing different feature encoders, along with two external audio laughter detectors. The last row corresponds to the actual configuration used in FunnyNet-W

2018) laughter detector used in Castro et al. (2019) and RLD (Gillick et al., 2021). The results on the Friends dataset are presented in Table 9. Overall, our detector demonstrates superior performance compared to both supervised methods. Notably, our detector combined with BEATs features consistently demonstrates superior performance, excelling for instance in temporal precision (78.4%), and detection precision for both thresholds (95.2% for 0.3 and 55.1% for 0.7). Our method combined with BYOL-A and BYOL-A-v2 features also showcases a balanced performance, maintaining high temporal accuracy (86.0% and 82.1% respectively). In comparison, LD exhibits high temporal recall (99.0%) but lower temporal precision (35.7%) highlighting a bias in its predictions. While RLD achieves a better balance between temporal precision and recall (58.9% 62.0% respectively) it is still far from our results.

Furthermore, we evaluate our detector using five audio feature extractors: Wav2CLIP (Wu et al., 2021), CAV-MAE (Gong et al., 2023), two versions of BYOL-A (Niizumi et al., 2021, 2023), and BEATs (Chen et al., 2023). Among these, the BEATs encoder exhibits the most suitable audio representation capacity for our detector, providing the best results (last row). During the analysis of the laughter detection, we make three important observations: (i) The majority of false positives are unfiltered sounds that are not easily separable using K-means clustering. (ii) The majority of false negatives correspond to intra-diegetic laughter, which is typically less loud and therefore more challenging to detect. (iii) The peak detector fails in scenarios where music overlaps with laughter, such as in party settings.

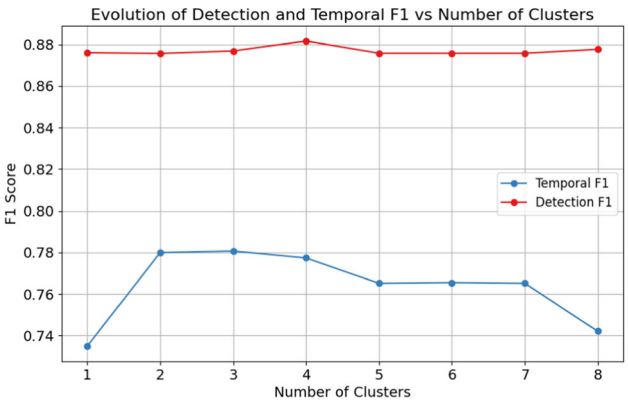

Fig. 6 Evolution of the temporal (*blue*) and detection (*red*) F1 scores according to the number of clusters chosen for the K-means algorithm at the end of the laughter detection pipeline (Color figure online)

Influence of Clustering on the Detection Performance. Here, we examine how the choice of the cluster count param-

Fig. 7 Visualization of (**a**–**d**) funny, (**e**,**f**) non-funny predictions on the Friends test set. We show the audio, visual and text inputs, the learned average weights of cross-attentions from CAF (*pie chart*), and the subtitles (for better understanding) (Color figure online)

eter K in the K-means algorithm influences the performance of our laughter detector. In practice, laughter chunks significantly outnumber music chunks. Consequently, in the third stage of Fig. 3, we exclude the smallest cluster- identified as the music cluster through empirical assessment- and retain the clusters comprising the laughter chunks.

Figure 6 shows the performance of the detection pipeline both at the detection level (red lines) and at the temporal level (blue lines) as a function of different numbers of clusters (x-axis). Overall, we make the following three observations: (1) For 1 cluster, we note that using one cluster is equivalent to no clustering. (2) Between 2 and 4 clusters, we note that F1 scores are higher than for 1 cluster. Here, there are enough degrees of freedom for the K-means algorithm to correctly detect the centroid of the music cluster. (3)For more than 7 clusters, we note that F1 scores tend to converge to the same value as for 1 cluster. Here, there are too many degrees of freedom for the K-means algorithm, and therefore it detects multiple centroids for the music cluster. Thus, the higher the number of clusters, the smaller the music sub-cluster we have, with the extreme case of having one cluster per sample, thus having the same effect as no clustering.

Moreover, Fig. 6 shows that the detection F1 score (red line) is less sensitive to the number of clusters than the tem-

poral F1 score (blue line). This can be explained by the fact that music chunks are generally longer than laughter chunks. Thus, by removing longer false positive chunks, we improve temporal metrics, whereas the impact is less important at the sample scale for detection metrics.

6 Analysis of FunnyNet-W

6.1 Modality Impact

To visualize the impact of modalities, we compute the average attention values on the three CA modules (CA boxes in Fig. 2) and then, show the average weights for each modality in the pie chart of each example in Fig. 7. For this, we show (a-d) four positive and (e,f) two negative samples on Friends with frames, subtitles and audio spectrogram (left) and pitch (right). We observe that the contribution of each modality varies; the commonality though is that audio contributes more than half, followed by text and finally visual features. Specifically, in cases where there is a strong audio signal, the contribution of audio increases significantly. This is illustrated when the character yells ('Chandler' in positive example (a), or pauses the speech ('Chandler' in positive example (b), or the speech rate speeds up ('Phoebe' in positive example (c) or speech volumes change suddenly ('Chandler' and 'Joey' in positive example (d). In contrast, in negative (e) and (f), the tone, volume, pitch or rhythm do not change greatly, so the text starts to play a bigger role in determining them as non-funny scenes. Furthermore, we observe that in the (c) and (e) examples, the visual feature plays very little role in the final prediction probably because the scenes do not capture the whole character's bodies and their movement, so the visual model can offer only little information.

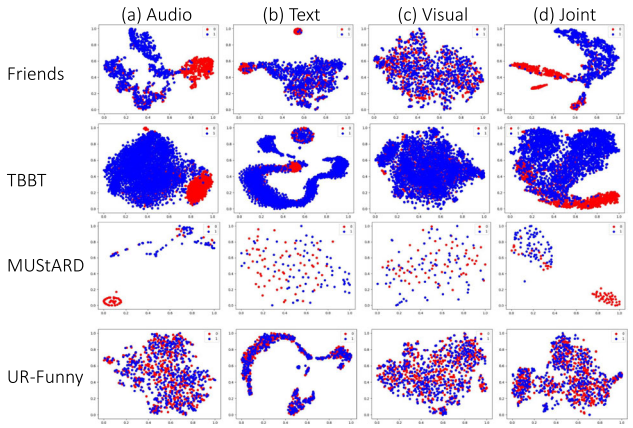

Fig. 8 t-SNE visualization of embeddings on Friends for (**a**) audio, (**b**) text, (**c**) visual, (**d**) all modalities. We show positive (*blue*) and negative samples (**red**)

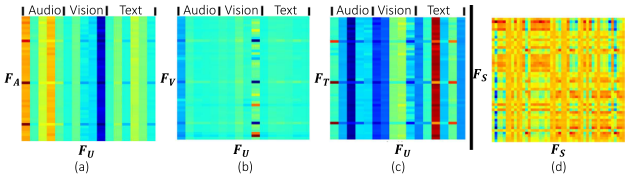

Fig. 9 CAF attention maps on the test set of Friends. **a**–**c**) Cross Attention between the unified feature F_U (coming from all modalities) and audio, vision and text; **d** Self Attention on F_U

6.2 Feature Visualization

Figure 8 shows the t-SNE (Hinton & Roweis, 2002) visualization of features: (a, b, c) display the unimodal distributions of audio, text, and visual features respectively, while (d) corresponds to all modalities for four datasets. Blue colour corresponds to funny samples and red to not-funny ones. All single features, and in particular the visual and textual ones, are scattered in the 2D space without clear boundaries between positives and negatives. Interestingly, for Friends, TBBT and MUSTaRD the audio features alone exhibit a notable ability to discriminate positive and negative samples; this is probably because of the punchlines used in these shows that typically occur at the end of sentences. For these three datasets, we observe that the joint embedding of all modalities results in the best separation between positives and negatives. Interestingly, for UR-Funny, a dataset without ending punchlines, all combinations of modalities (either single or joint) fail to distinguish funny from not-funny moments. This is probably due to the domain shift between samples from this dataset (TED-talk segments) and the samples used at training (sitcoms).

6.3 Impact of CAF Module

To examine the effect of CAF, we visualize in Fig. 9 the learned attention maps: red indicates higher and blue lower attention. (a,b,c) display the cross-attention between the unified F_U and (a) audio, (b) visual, (c) text features. Since F_U is stacked from audio, vision, and text, we observe that each modality highly attends to itself (especially text). We also observe that the audio encoder also attends to the text encoder, indicating that there is mutual information shared between text and audio. Finally, (d) displays the self-attention map between F_U, where we observe that F_U attends to all tokens with different weights. The small color differences on the diagonal and anti-diagonal areas suggest that the joint features have approximately uniform representations for the final classification.

Fig. 10 Failure cases on Friends split into three main groups: **a**, **b** strong emotional responses expressed by single wording, **c** subtle sarcastic comments with straight face and no follow-up indications, and **d** inside jokes depending on long-term understanding

6.4 Failure Analysis

By examining the results, we observe three main groups of failure cases. First, when characters have strong emotional responses expressed only by single words (such as 'haha', 'no!') is not always funny. However, all modalities incorrectly, yet confidently predict them as funny. Figure 10a and b show this. In Fig. 10a, 'Rachel' laughs sarcastically, which is not funny (subtitle 'ha ha'). FunnyNet-W incorrectly predicts it as positive. In Fig. 10b, Monica screams loudly and falls on Rachel (subtitle 'Ah'). The sudden high pitch of the scream gives the wrong signal to the model and wrongly predicts it as positive. Second, when the funny moment is expressed only with subtle indications, typically sarcasm without a follow-up signal (indicative facial expression or grimace, surprise, pause in dialogue, phrase, joke). In such cases, FunnyNet-W may fail to discriminate these subtle cues that come usually with human-level understanding. Figure 10c indicates such an example, where Ross gives a sarcastic response to Joey without changing facial expression or tone; in this case, FunnyNet-W incorrectly predicts the scene as negative. Third, in most cases, all modalities fail to understand inside jokes that depend on long-term dependencies. For instance, Fig. 10d is the case where the context is so long (the previous awkward moment between Ross and Rachel) that the model wrongly predicts the scene as a not-funny moment. All audio, visual or text fail to give discriminative signals to indicate the funniness.

7 Funny Scene Detection in the Wild

7.1 Applications From Other Domains

In this section, we show applications of FunnyNet and FunnyNet-W in videos from other domains.

1. Sitcoms without Canned Laughter. In FunnyNet (Liu et al., 2022), we collect 9 episodes of the first season (∼180 min) of *Modern Family* (Lloyd and Levitan, 2009)[6] without canned laughter. We manually annotate as positive every punchline that could lead to laughter, resulting in 453 positives (we will make them available). Figure 11a shows a correctly predicted funny moment between two characters who vary their speech rhythm and tones.

2. Movies with Diverse Funny Styles. Fig. 11b depicts such an example from the *Dumb and Dumber* film (Farrelly (1994)). Our model correctly detects funny moments followed by silence or a speaker's change of tone.

3. Stand-Up Comedies. They contain several punchlines that make audiences laugh. We experiment on the Jerry Seinfeld *23h to Kill* stand-up comedy. Figure 11c shows that FunnyNet detects funny moments correctly and confidently, as Jerry is highly expressive (expressions, gestures).

4. Audio-Only. As audio is the most discriminative cue, we examine its impact on out-of-domain audios: narrating jokes and reading books. Our model detects funny punchlines from jokes, mostly when they are accompanied by a change of pitch or pause; for the audiobook, it successfully detects funny moments when the reader's voice imitates a character.

[6] https://www.youtube.com/playlist?list=PL8v3aNB88WMM0iwOUeLpgFf3pHH9uxz7_.

(a) Funny moment in sitcom (b) Funny moment in movie (c) Funny moment in stand-up

Fig. 11 Funny moments in the wild. Three examples from sitcoms, movies, and stand-up with diverse contents, e.g. the sitcom does not have canned laughter, the movie contains dramatic acting performance, and the stand-up is a one-man show without interactions

7.2 FunnyNet-W against LLM Chatbot

Recently, several large language models (LLMs) (OpenAI, 2023; Touvron et al., 2023) have been fine-tuned and are used as chatbots (OpenAI, 2021; Köpf et al., 2023). With their expansive knowledge and context-aware responses, they have significantly advanced language understanding and generation, which enable them to perform a wide range of language-related tasks. In this context, we compare the proposed FunnyNet-W against a chatbot to assess its performance relative to these general models. Specifically, we use the LlaMa-2 (Touvron et al., 2023) chatbot on the Friends dataset.

Prompting. We evaluate the language LlaMa-2 chatbot in two setups.
First, with or without prompt training:

- **zero-shot setting**, where we prompt the chatbot with a transcript sample and ask it to determine whether it is funny or not. We do that iteratively for all test samples of the Friends test set. The prompt we use is "*Is the following sentence funny or not?* ≪subtitles≫", where ≪subtitles≫ corresponds to each test sample. However, given the popular nature of the 'Friends' sitcom, the chatbot may have already seen samples or even the whole transcript of the TV show during training. We hypothesize that this impacts its performance positively, as the chatbot not only has knowledge of the dialogues that follow, but also knows the comments of the community for each pun or joke.
- **few-shot setting**, where we prompt the chatbot with some training samples followed by the testing sample within the token context limit. The prompt we use is twenty training samples (ten positives and ten negatives): "*This sentence is funny:* ≪subtitles≫. *This sentence is not funny* ≪subtitles≫.", followed by the testing sample: "*Is the following sentence funny or not?* ≪subtitles≫". In this case, the chatbot uses the training samples to better distinguish the specific TV show type of humour.

Table 10 Chatbot vs FunnyNet-W

Prompt engineering	Prompt training	F1	Accuracy
Generic	–	14.5	41.8
	✓	44.3	46.5
Specific	–	64.1	53.2
	✓	71.1	55.9
FunnyNet-W (T)		77.8	68.1
FunnyNet-W (A+V+T)		88.2	85.6

Second, by performing a simple prompt engineering (i.e. part of the prompt that gives context to the chatbot):

- **general system prompt**, we prompt the chatbot with the general system prompt (referred to as 'Generic'): "*You are a helpful, respectful and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic, dangerous, or illegal content. [...]*". This system prompt makes the chatbot act as a general chatbot without any prior on the task.
- **specific system prompt**, we prompt the chatbot with the task-specific system prompt (referred to as 'Specific'): "*I will give some sentences, and you need to say if it's funny or not, reply only by yes or no.*". This kind of system prompt helps the chatbot limit its range and focus on the task only. Moreover, it forces the chatbot to answer, whereas the general system prompt leads sometimes to hesitating answers.

Experimental results. Table 10 reports the results when prompting the LlaMa-2 chatbot. We observe that without prompt training, the chatbot's performance drops both with and without prompt engineering. Additionally, we observe the importance of prompt engineering: when using the specific prompt (with or without training) the performances are higher than 50% in both metrics, whereas the generic prompt (no prompt engineering) results in very low performances. This is in line with the current bibliography on LLMs, where

Models		LlaMa-2		FunnyNet-W
		w/o PT	w PT	
Funny (positive)	They are putting together this panel to talk about fossils they just found in Peru and the Discovery channel is gonna film it. Oh my god, who's gonna watch that?	No	Yes	Yes
	I didn't wear this suit for a year because you hated it. You're not my girlfriend anymore, Now that you're on your own, you're free to look as stupid as you'd like.	No	Yes	Yes
Not funny (Negative)	I hope it won't be too weird. will it? Rache? No, not at all. I'm actually gonna bring someone myself	No	Yes	No
	Let me walk you home and stop by every newsstand and burn every copy of The Times and The Post.	Yes	Yes	No

Fig. 12 Examples of using Chatbot, with or without prompt training (w PT or w/o PT), and FunnyNet-W for funny prediction. FunnyNet-W can give correct predictions (green) for both positive and negative examples, while LlaMa-2 fails to give good results and the prompt training brings small improvements (Color figure online)

prompt engineering is crucial for higher accuracy. We believe that more prompt engineering will increase the performance; yet, this is outside the scope of this work. Overall, we observe that FunnyNet-W outperforms all examined cases with the chatbot. This highlights the need for specific model training for funny moment detection. Interestingly, we note that the performance of FunnyNet-W using text only is close to the one of the LlaMa-2 chatbot, thus showcasing the impressive representation power of LLM chatbots.

Figure 12 illustrates four examples: two positive (first and second rows) and two negative samples (third and fourth rows). For the positive samples, we observe that LlaMa-2 correctly understand its funniness (green) with prompt training, whereas when there is no prompt the results are incorrect. For two negative samples, most predictions from the chatbot are incorrect, most likely because it picks up the words with strong emotional expressions, like "weird" and "burn", resulting in false positives. In all examples, FunnyNet-W correctly predicts the results because it does not solely rely on text, but also audio and visual features.

Overall, our findings are twofold: (1) using only subtitles is insufficient to understand the funniness in video scenes, and (2) since we only do minor prompt engineering (generic and specific), the results of LLMs cannot outperform the proposed FunnyNet-W. Potentially, by improving the prompts, we can further improve the performance of LLMs.

7.3 Impact of Audio

In the context of funny moment detection, audio is more relevant than text (Liu et al., 2022) because it contains more information, including vocals, pauses, pitch variations, speech rate variations, rhythm and timing, accent and pronunciation, emotional tone, music and background noise. To highlight the importance of audio, in this section, we test FunnyNet-W by replacing the ground truth audio with automatic machine sounds.

For this, we generate corresponding synthetic audios from the ground truth subtitles of the Friends dataset with a text-to-audio model.[7] Note that the synthetic audio only mimics the vocals between characters without any background sounds and, more importantly, without including all additional voice cues that help identify the emotional state of the character and the dialogue. Then, we train FunnyNet-W with the synthetic voices and test it on both the real and synthetic voices and respectively, we test FunnyNet-W (trained on real voices) on both real and synthetic voices. Table 11 reports the results. When training with synthetic voice (first and second column), we observe that testing on real voices (second row) outperforms testing on synthetic ones (first row) by a large margin, i.e. approximately 10–15% for both metrics. Similarly, when training with real voice (third and fourth columns), we observe that there is a significant difference in performance (or approximately 20% in both metrics) between testing on synthetic and real data. These results show that simply replacing real voice with synthetic ones omits other important information, such as background audio, and music; hence, the model makes more correct predictions when the test set contains additional auditory information (real) rather than a simple voice (synthetic). When we test on synthetic voices (first row), we observe that training either with synthetic or real voice produces similar results. This is because the test set contains synthetic data, and therefore learning the specificities of voice is not necessary for good performance. However, when we test on real voices (second row), we observe that training with real voices (columns 3–4) outperforms training with synthetic ones (columns 1–2) by a large margin (e.g. for Acc 79.6% for synthetic vs. 85.6% for real). This clearly shows that the real voice includes important additional cues (pause, intonation, etc.) that help FunnyNet-W discriminate funniness.

Table 11 Ablation of synthetic and real voice when training and testing FunnyNet-W on Friends

Model		Training voice			
		Synthetic		Real	
		F1	Accuracy	F1	Accuracy
Test voice	Synthetic	65.5	67.7	68.8	66.8
	Real	83.1	79.6	88.2	85.6

To further analyze the effect of voice, we perform here a qualitative comparison using Spleeter.[8] Specifically, Fig. 13 illustrates the spectrum heatmaps between (a) real vocal, (b) real accompaniment (non-vocal parts, such as background music, sounds, talks, audio), (c) synthetic vocal, and (d) the differences between real and synthetic audio. We visualize the heatmaps of examples (two rows), where in both cases FunnyNet-W correctly predicts the funniness when using real

[7] https://github.com/pndurette/gTTS.

[8] https://github.com/deezer/spleeter.

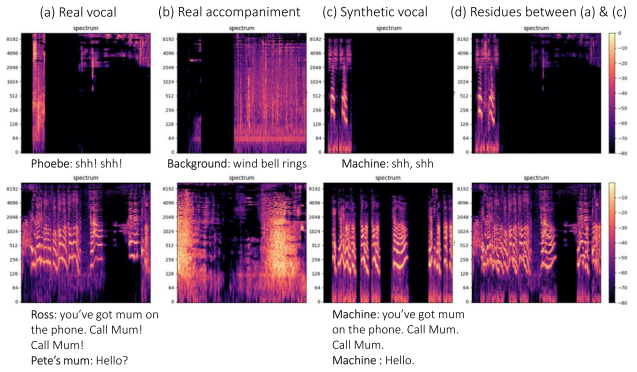

Fig. 13 Visualization of real and synthetic audio on Friends. We show real vocals (**a**), real accompaniment (**b**), synthetic vocals (**c**) and the residues between real and synthetic vocals (**d**)

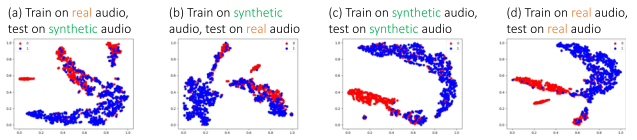

Fig. 14 T-SNE visualization of real and synthetic audio on Friends. We show positive (*blue*) and negative (*red*) samples to indicate the feature distributions (Color figure online)

audio and incorrectly when using synthetic audio. The first row shows the funny moment when Phoebe tries to shush the wind bell while the wind bell keeps ringing. This contrast between vocals and non-vocal sounds (i.e. in this case bell ringing) is missing from the synthetic vocals. Row 2 shows that when Ross screams excitedly ('Call Mum!'), his voice triggers the smartphone to dial Pete's mum. This strong vocal expression does not appear on the synthetic vocals. Both these examples indicate that audio plays a key role in funny moment detection because it contains not only background sounds but also expressions and feelings from the characters leading to better scene understanding.

Furthermore, we also use T-SNE to visualize the data clustering in Fig. 14. This visualization shows that when the train and test data come from the same domain (either real or synthetic, c and d figures), the positive and negative distributions (blue and red points) are clearly separable. This is in contrast to the (a,b) figures, where the two synthetic and real domains are mixed (i.e., training on one domain and testing on another); in this case, the two distributions overlap more, as expected due to domain shift (Kalogeiton et al., 2016; Torralba & Efros, 2011).

8 Ethical Discussion

Practical Impact. There are various potential applications for FunnyNet-W. First, it may be useful to collect a large dataset of funny moments (similar to (Li et al., 2023)), so for example, cognitive researchers could study funniness mechanisms at a large scale. Next, it may be useful to enable artists to edit films more easily, without relying on a live audience. Finally, it may be useful to enhance human–machine interactions. For instance, adding a sense of humor to conversational agents would make the relation more natural and spontaneous.

However, FunnyNet-W is part of artificial intelligence systems that tend to analyze complex human specificities and behaviors (e.g., conversational agents). Given the nature of these systems, their usage and deployment should be done with caution. For instance, in the particular case of FunnyNet-W, it could enhance identity fraud methods, by better mimicking the sense of humor of victims.

Societal Impact. FunnyNet-W is trained mainly with Western cultural materials, especially from the USA, which do not necessarily represent uniform demographics. In particular, we mainly tackle funniness in American sitcoms, which covers a very specific type of humor. Therefore, without fine-tuning, FunnyNet-W might have difficulties in generalizing to funny moments from other cultures, as humor is highly thematic, and themes vary from one culture to another. Moreover, the audio modality might also be highly impacted by cultural bias, as expressiveness is strongly related to culture, e.g., actor performances change a lot from one country to another, leading to misinterpretations. In addition to the cultural barrier, FunnyNet-W includes language bias. Indeed, the audio as well as the textual modality are trained with the English language. This can be a limiting factor for generalization and transferability across languages, as jokes or puns often rely on language specificities. We also note that the textual modality is limited by alphabets that vary among languages.

Environmental Impact. All experiments are done on NVIDIA RTX4090 and A100 GPUs, with each of them requiring 215W in power supply. For this project, we use approximately 800 GPU hours. Training a FunnyNet-W model with all three modalities requires around 6 GPU hours on NVIDIA RTX4090, which amounts to 1.29 kWh and 300.75g of CO2 emitted.

9 Conclusions

We introduced FunnyNet, an audiovisual model for funny moment detection. In contrast to works that rely on text, FunnyNet exploits audio that comes naturally with videos and contains high-level cues (pauses, tones, etc). Our findings show audio is the dominant cue for signaling funny situations, while video offers complementary information. Extensive analysis and visualizations also support our finding that audio is better than text (in the form of subtitles) when it comes to scenes with no or simple dialogue but with hilarious acting or funny background sounds. Our results show the effectiveness of each component of FunnyNet, which outper-

forms the state of the art on the TBBT, MUStARD, MHD, UR-Funny and Friends. Future work includes analyzing the contribution of audio cues (pitch, tone, etc).

Acknowledgements This work is supported by a DIM RFSI grant, a Hi!Paris collaborative project grant, the ANR projects WhyBehind-Scenes ANR-22-CE23-0007 and APATE ANR-22-CE39-0016, and the HPC resources of IDRIS under the allocation 2022-AD011013951 made by GENCI.

Funding Open Access funding provided by LUT University (previously Lappeenranta University of Technology (LUT)).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Afouras, T., Chung, J.S., & Zisserman, A. (2020). The conversation: Deep audio-visual speech enhancement. In *INTERSPEECH*.

Annamoradnejad, I., & Zoghi, G. (2020). Colbert: Using bert sentence embedding for humor detection. arXiv:2004.12765.

Bain, M., Huh, J., Han, T., & Zisserman, A. (2023). Whisperx: Time-accurate speech transcription of long-form audio. arXiv:2303.00747.

Bain, M., Nagrani, A., Varol, G., & Zisserman, A. (2021). Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*.

Barral, O., Kosunen, I., & Jacucci, G. (2017). No need to laugh out loud: Predicting humor appraisal of comic strips based on physiological signals in a realistic environment. *ACM Transactions on Computer-Human Interaction, 24*(6), 1–29.

Bertasius, G., Wang, H., & Torresani, L. (2021). Is space-time attention all you need for video understanding? In *ICML*.

Bertero, D., & Fung, P. (2016). Deep learning of audio and language features for humor prediction. In *LREC*.

Brown, A., Kalogeiton, V., & Zisserman, A. (2021). Face, body, voice: Video person-clustering with multiple modalities. In *ICCV*.

Castro, S., Hazarika, D., Pérez-Rosas, V., Zimmermann, R., Mihalcea, R., & Poria, S. (2019). Towards multimodal sarcasm detection (an *Obviously* perfect paper). In *ACL*.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *ICML*.

Chen, S., Wu, Y., Wang, C., Liu, S., Tompkins, D., Chen, Z., & Wei, F. (2023). Beats: Audio pre-training with acoustic tokenizers. In *ICML*.

Chung, J.S., & Zisserman, A. (2016). Out of time: Automated lip sync in the wild. In *ACCV*.

Chung, S.W., Chung, J.S., & Kang, H.G. (2019). Perfect match: Improved cross-modal embeddings for audio-visual synchronisation. In *ICASSP*.

Davidov, D., Tsur, O., & Rappoport, A. (2010). Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *ACL*.

Défossez, A., Usunier, N., Bottou, L., & Bach, F. (2019). Music source separation in the waveform domain. arXiv:1911.13254.

Deng, D., Zhou, Y., Pi, J., & Shi, B.E. (2018). Multimodal utterance-level affect analysis using visual, audio and text features. arXiv:1805.00625.

Devlin, J., Chang, M.W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Dong, J., Li, X., Xu, C., Yang, X., Yang, G., Wang, X., & Wang, M. (2021). Dual encoding for video retrieval by text. In *IEEE TPAMI*.

Dufour, N., Picard, D., & Kalogeiton, V. (2022). Scam! Transferring humans between images with semantic cross attention modulation. In *ECCV*,

Epstein, D., & Vondrick, C. (2021). Learning goals from failure. In *CVPR*.

Fang, H., Xiong, P., Xu, L., & Chen, Y. (2021). Clip2video: Mastering video-text retrieval via image clip. arXiv:2106.11097.

Farrelly, P. (Director), Dumb and Dumber (Film), Katja Motion Picture Corporation, 1994. https://www.imdb.com/title/tt0109686.

Gabbay, A., Ephrat, A., Halperin, T., & Peleg, S. (2018). Seeing through noise: Visually driven speaker separation and enhancement. In *ICASSP*.

Gabeur, V., Sun, C., Alahari, K., & Schmid, C. (2020). Multi-modal transformer for video retrieval. In *ECCV*.

Gemmeke, J.F., Ellis, D.P., Freedman, D., Jansen, A., Lawrence, W., Moore, R.C., Plakal, M., & Ritter, M. (2017). Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*.

Gillick, J., Deng, W., Ryokai, K., & Bamman, D. (2021). Robust laughter detection in noisy environments. In *INTERSPEECH*.

Girdhar, R., El-Nouby, A., Liu, Z., Singh, M., Alwala, K.V., Joulin, A., & Misra, I. (2023). Imagebind: One embedding space to bind them all. arXiv:2305.05665.

Gong, Y., Rouditchenko, A., Liu, A.H., Harwath, D., Karlinsky, L., Kuehne, H., & Glass, J.R. (2023). Contrastive audio-visual masked autoencoder. In *ICLR*.

Guzhov, A., Raue, F., Hees, J., & Dengel, A. (2022). Audioclip: Extending clip to image, text and audio. In *ICASSP*.

Han, T., Bain, M., Nagrani, A., Varol, G., Xie, W., & Zisserman, A. (2023). Autoad II: The sequel—Who, when, and what in movie audio description. In *ICCV*.

Hasan, M.K., Lee, S., Rahman, W., Zadeh, A., Mihalcea, R., Morency, L.P., & Hoque, E. (2021). Humor knowledge enriched transformer for understanding multimodal humor. In *AAAI*.

Hasan, M.K., Rahman, W., Bagher Zadeh, A., Zhong, J., Tanveer, M.I., Morency, L.P., & Hoque, M.E. (2019). UR-FUNNY: A multimodal language dataset for understanding humor. In *EMNLP-IJCNLP*.

Hazarika, D., Zimmermann, R., & Poria, S. (2020). Misa: Modality-invariant and-specific representations for multimodal sentiment analysis. In *ACM International Conference on Multimedia*.

Hinton, G., & Roweis, S. (2002). Stochastic neighbor embedding. In *NeurIPS*.

Holliman, E., Godfrey, J., & McDaniel, J. (1992). Switchboard: Telephone speech corpus for research and development. In *ICASSP*.

Hong, J., Hayder, Z., Han, J., Fang, P., Harandi, M., & Petersson, L. (2023). Hyperbolic audio-visual zero-shot learning. In *ICCV*.

Huber, D. M., & Runstein, R. (2012). *Modern recording techniques*. Milton Park: Routledge.

Iashin, V., & Rahtu, E. (2020). Multi-modal dense video captioning. In *CVPR-workshops*.

Jaegle, A., Gimeno, F., Brock, A., Zisserman, A., Vinyals, O., & Carreira, J. (2021). Perceiver: General perception with iterative attention. In *ICML*.

Kalogeiton, V., & Zisserman, A. (2020). Constrained video face clustering using 1nn relations. In *BMVC*.

Kalogeiton, V., Ferrari, V., & Schmid, C. (2016). Analysing domain shift factors between videos and images for object detection. In *IEEE TPAMI*.

Kayatani, Y., Yang, Z., Otani, M., Garcia, N., Chu, C., Nakashima, Y., & Takemura, H. (2021). The laughing machine: Predicting humor in video. In *WACV*.

Kim, M., Sung-Bin, K., & Oh, T.H. (2023). Prefix tuning for automated audio captioning. In *ICASSP*.

Koepke, A. S., Oncescu, A. M., Henriques, J., Akata, Z., & Albanie, S. (2022). Audio retrieval with natural language queries: A benchmark study. *IEEE Transactions on Multimedia*. https://doi.org/10.1109/TMM.2022.3149712

Koizumi, Y., Masumura, R., Nishida, K., Yasuda, M., & Saito, S. (2020). A transformer-based audio captioning model with keyword estimation. arXiv:2007.00222.

Köpf, A., Kilcher, Y., von Rütte, D., Anagnostidis, S., Tam, Z.R., Stevens, K., Barhoum, A., Duc, N.M., Stanley, O., Nagyfi, R., ES, S., Suri, S., Glushkov, D., Dantuluri, A., Maguire, A., Schuhmann, C., Nguyen, H., & Mattick, A. (2023). Openassistant conversations—Democratizing large language model alignment. arXiv:2304.07327.

Korbar, B. (2018). Co-training of audio and video representations from self-supervised temporal synchronization. In *CoRR*.

Lee, J.T., Jain, M., Park, H., & Yun, S. (2020). Cross-attentional audio-visual fusion for weakly-supervised action localization. In *ICLR*.

Li, R., Sun, S., Elhoseiny, M., & Torr, P. (2023). Oxfordtvg-hic: Can machine make humorous captions from images? arXiv:2307.11636.

Liang, Z., Jiang, W., Hu, H., & Zhu, J. (2020). Learning to contrast the counterfactual samples for robust visual question answering. In *EMNLP*.

Lin, Z., Geng, S., Zhang, R., Gao, P., de Melo, G., Wang, X., Dai, J., Qiao, Y., & Li, H. (2022). Frozen clip models are efficient video learners. In *ECCV*.

Lin, K., Li, L., Lin, C.C., Ahmed, F., Gan, Z., Liu, Z., Lu, Y., & Wang, L. (2022). Swinbert: End-to-end transformers with sparse attention for video captioning. In *CVPR*.

Liu, Z.S., Courant, R., & Kalogeiton, V. (2022). Funnynet: Audiovisual learning of funny moments in videos. In *ACCV*.

Liu, X., Huang, Q., Mei, X., Liu, H., Kong, Q., Sun, J., Li, S., Ko, T., Zhang, Y., Tang, L.H., et al. (2023). Visually-aware audio captioning with adaptive audio-visual attention. In *INTERSPEECH*.

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. In *ICLR*.

Lou, S., Xu, X., Wu, M., & Yu, K. (2022). Audio-text retrieval in context. In *ICASSP*.

Mei, X., Liu, X., Liu, H., Sun, J., Plumbley, M.D., & Wang, W. (2022). Language-based audio retrieval with pre-trained models. In *DCASE*.

Mesaros, A., & Heittola, T., Virtanen, T. (2016). Tut database for acoustic scene classification and sound event detection. In *European Signal Processing Conference*.

Mesaros, A., Heittola, T., Benetos, E., Foster, P., Lagrange, M., Virtanen, T., & Plumbley, M. D. (2017). Detection and classification of acoustic scenes and events: Outcome of the DCASE 2016 challenge. *IEEE/ACM Transactions on Audio: Speech, and Language Processing, 26*(2), 379–393.

Mohla, S., Pande, S., Banerjee, B., & Chaudhuri, S. (2020). Fusatnet: Dual attention based spectrospatial multimodal fusion network for hyperspectral and lidar classification. In *CVPR*.

Morgado, P., Vasconcelos, N., & Misra, I. (2021). Audio-visual instance discrimination with cross-modal agreement. In *CVPR*.

Nagrani, A., Yang, S., Arnab, A., Jansen, A., Schmid, C., & Sun, C. (2021). Attention bottlenecks for multimodal fusion. In *NeurIPS*.

Nam, H., Ha, J.W., & Kim, J. (2017). Dual attention networks for multimodal reasoning and matching. In *CVPR*.

Narasimhan, M., Rohrbach, A., & Darrell, T. (2021). Clip-it! language-guided video summarization. In *NeurIPS*.

Niizumi, D., Takeuchi, D., Ohishi, Y., Harada, N., & Kashino, K. (2021). Byol for audio: Self-supervised learning for general-purpose audio representation. In *International Joint Conference on Neural Networks*.

Niizumi, D., Takeuchi, D., Ohishi, Y., Harada, N., & Kashino, K. (2023). Byol for audio: Exploring pre-trained general-purpose audio representations. *IEEE/ACM Transactions on Audio: Speech, and Language Processing, 31*, 137–151.

Oord, A.V.D., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. arXiv:1807.03748.

OpenAI (2021). ChatGPT: Conversational ai powered by GPT-3.5. OpenAI Blog.

OpenAI (2023). Gpt-4 technical report. arXiv:2303.08774.

Owens, A., & Efros, A.A. (2018). Audio-visual scene analysis with self-supervised multisensory features. In *ECCV*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*.

Patro, B.N., Lunayach, M., Srivastava, D., Sarvesh, S., Singh, H., & Namboodiri, V.P. (2021). Multimodal humor dataset: Predicting laughter tracks for sitcoms. In *WACV*.

Priyasad, D., Fernando, T., Denman, S., Sridharan, S., & Fookes, C. (2020). Attention driven fusion for multi-modal emotion recognition. In *ICASSP*.

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *ICML*.

Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). Robust speech recognition via large-scale weak supervision. arXiv:2212.04356.

Rahman, W., Hasan, M.K., Lee, S., Bagher Zadeh, A., Mao, C., Morency, L.P., & Hoque, E. (2020). Integrating multimodal information in large pretrained transformers. In *ACL*.

Rockwell, P. (2000). Lower, slower, louder: Vocal cues of sarcasm. *Journal of Psycholinguistic Research, 29*, 483–495.

Rouard, S., Massa, F., & Défossez, A. (2023). Hybrid transformers for music source separation. In *ICASSP*

Rouditchenko, A., Boggust, A., et al. (2021). AVLnet: Learning Audio-Visual Language Representations from Instructional Videos. In *INTERSPEECH*.

Ryokai, K., Durán López, E., Howell, N., Gillick, J., & Bamman, D. (2018). Capturing, representing, and interacting with laughter. In *Conference on human factors in computing systems*.

Sablayrolles, A., Douze, M., Schmid, C., & Jégou, H. (2019). Spreading vectors for similarity search. In *ICLR*.

Saeed, A., Grangier, D., & Zeghidour, N. (2021). Contrastive learning of general-purpose audio representations. In *ICASSP*.

Schroff, F., Kalenichenko, D., & Philbin, J. (2015/. Facenet: A unified embedding for face recognition and clustering. In *CVPR*.

Senocak, A., Oh, T.H., Kim, J., Yang, M.H., & Kweon, I.S. (2018). Learning to localize sound source in visual scenes. In *CVPR*.

Shen, X., Li, D., Zhou, J., Qin, Z., He, B., Han, X., Li, A., Dai, Y., Kong, L., Wang, M., et al. (2023). Fine-grained audible video description. In *CVPR*.

Shimasaki, A., & Ueoka, R. (2017). Laugh log: E-textile bellyband interface for laugh logging. In *Conference extended abstracts on human factors in computing systems*.

Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al. (2022). Make-a-video: Text-to-video generation without text-video data. arXiv:2209.14792

Solaiman, I., Brundage, M., Clark, J., Askell, A., Herbert-Voss, A., Wu, J., Radford, A., & Wang, J. (2019) Release strategies and the social impacts of language models. CoRR.

Tan, R., Plummer, B.A., Saenko, K., Jin, H., & Russell, B. (2021). Look at what i'm doing: Self-supervised spatial grounding of narrations in instructional videos. In *NeurIPS*.

Tepperman, J., Traum, D., & Narayanan, S.S. (2006). 'yeah right': Sarcasm recognition for spoken dialogue systems. In *INTERSPEECH*.

Tian, Y., Shi, J., Li, B., Duan, Z., & Xu, C. (2018). Audio-visual event localization in unconstrained videos. In *ECCV*.

Tong, Z., Song, Y., Wang, J., & Wang, L. (2022). Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. NeurIPS.

Torralba, A., & Efros, A.A. (2011). Unbiased look at dataset bias. In *CVPR*.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023). Llama: Open and efficient foundation language models. arXiv:2302.13971.

Touvron, H., Martin, L., Stone, K., Albert, P., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. arXiv:2307.09288.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *NeurIPS*.

Wang, L., Luc, P., Recasens, A., Alayrac, J.B., & Oord, A.V.D. (2021). Multimodal self-supervised learning of general audio representations. arXiv:2104.12807.

Wang, T., Zheng, H., Yu, M., Tian, Q., & Hu, H. (2020). Event-centric hierarchical representation for dense video captioning. *IEEE Transactions on Circuits and Systems for Video Technology, 31*(5), 1890–1900.

Wei, X., Zhang, T., Li, Y., Zhang, Y., & Wu, F. (2020). Multi-modality cross attention network for image and sentence matching. In *CVPR*.

Weller, O., & Seppi, K. (2020). The rjokes dataset: A large scale humor collection. In *LREC*.

Wu, H.H., Seetharaman, P., Kumar, K., & Bello, J.P. (2021) Wav2clip: Learning robust audio representations from clip. arXiv:2110.11499.

Xin, Y., Yang, D., & Zou, Y. (2023). Improving text-audio retrieval by text-aware attention pooling and prior matrix revised loss. In *ICASSP*.

Xue, H., Sun, Y., Liu, B., Fu, J., Song, R., Li, H., & Luo, J. (2023). Clip-vip: Adapting pre-trained image-text model to video-language representation alignment. *ICLR*.

Yang, A., Nagrani, A., Seo, P.H., Miech, A., Pont-Tuset, J., Laptev, I., Sivic, J., & Schmid, C. (2023). Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *CVPR*.

Yoon, S., Byun, S., & Jung, K. (2018). Multimodal speech emotion recognition using audio and text. In *IEEE spoken language technology workshop*.

Zadeh, A., Liang, P.P., Mazumder, N., Poria, S., Cambria, E., & Morency, L.P. (2018). Memory fusion network for multi-view sequential learning. In *AAAI*.

Zhou, H., Xu, X., Lin, D., Wang, X., & Liu, Z. (2020). Sep-stereo: Visually guided stereophonic audio generation by associating source separation. In *ECCV*.

Zhu, W., Pang, B., Thapliyal, A.V., Wang, W.Y., & Soricut, R. (2022). End-to-end dense video captioning as sequence generation. In *ACL*.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.