AutoAD-Zero: A Training-Free Framework for Zero-Shot Audio Description

Junyu Xie[1], Tengda Han[1], Max Bain[1], Arsha Nagrani[1], Gül Varol[1,2], Weidi Xie[1,3], and Andrew Zisserman[1]

[1] Visual Geometry Group, University of Oxford
[2] LIGM, École des Ponts ParisTech
[3] School of Artificial Intelligence, Shanghai Jiao Tong University
https://www.robots.ox.ac.uk/vgg/research/autoad-zero/

Abstract. Our objective is to generate Audio Descriptions (ADs) for both movies and TV series in a training-free manner. We use the power of off-the-shelf Visual-Language Models (VLMs) and Large Language Models (LLMs), and develop visual and text prompting strategies for this task. Our contributions are three-fold: (i) We demonstrate that a VLM can successfully name and refer to characters if directly prompted with character information through visual indications without requiring any fine-tuning; (ii) A two-stage process is developed to generate ADs, with the first stage asking the VLM to comprehensively describe the video, followed by a second stage utilising a LLM to summarise dense textual information into one succinct AD sentence; (iii) A new dataset for TV audio description is formulated. Our approach, named AutoAD-Zero, demonstrates outstanding performance (even competitive with some models fine-tuned on ground truth ADs) in AD generation for both movies and TV series, achieving state-of-the-art CRITIC scores.

Keywords: Visual-Language Models · Audio Description

1 Introduction

An Audio Description (AD) soundtrack provides a description of the visual content of a video for the visually impaired. It covers aspects of the story that cannot be inferred from the audio soundtrack, particularly "who" is in the scene and "what" they are doing. Furthermore, in order not to overlap with the dialogue, the AD is usually placed in the gaps between characters speaking. Given the increasing power of Visual-Language Models (VLMs), there has naturally been a growing research interest in automating the production of ADs for movies. Most recent approaches have either fine-tuned components of an open-source pre-trained large-scale VLM, such as the visual-text bridge, for the AD task [12–14,28,35,38], or have used very powerful propriety models such as GPT-4/GPT-4v in a zero-shot way, without fine-tuning [7,24,46]. Both approaches have limitations: there is insufficient data to fine-tune these large models for the AD task – to enable the model to distill out the essential information for AD and ignore the rest, and

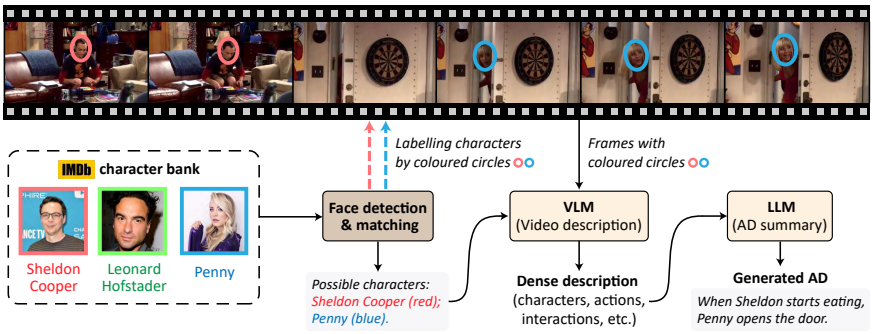

Fig. 1: A training-free framework for zero-shot AD generation. AutoAD-Zero features a two-stage process, where a VLM initially generates a comprehensive video description from multiple aspects, followed by an LLM-based AD summary in the second stage. To incorporate character information into this framework, character faces in the input video are matched with those in an external character bank and labelled with coloured circles. The corresponding character names and colour codes are then provided as text prompts to the VLM.

current zero-shot techniques fail due to a lack of knowledge of who the characters are, a core requirement for AD.

In this paper, we show that it is possible to adopt pre-trained models with novel visual and text prompting. Somewhat surprisingly, this zero-shot, training-free method reaches competitive performance with all previous AD approaches, and even gets state-of-the-art in some metrics. To achieve this we explore two ideas, as demonstrated in Fig. 1: first, use a *visual prompt* on the video frames so that we can refer to characters by name when text prompting the VLM. We achieve this by "circling" the characters in the frame and color coding for each identity. This "circle" prompting [33] enables the VLM both to identify the characters in the clip, and also to refer to them by name when describing their actions, interactions, and locations. The second idea is *not* to predict the AD directly, but instead to take advantage of the strength of Large Language Models (LLMs) at providing summaries. We introduce a two-stage approach to generating the AD: in the first stage, a detailed description of "who is doing what" is generated by prompting the VLM with questions about the visually tagged characters; in the second stage, an LLM is used to summarise the detailed description into the style and desired length for the AD. This two-stage approach has multiple advantages – it forces the AD to be more visually grounded; it enables the information in the AD to be prioritised for the temporal interval available; it enables the style of AD to be tailored for particular groups, such as being age appropriate [31] or a simpler form of the language (for non-native speakers).

The great advantage of our training-free approach, is that the method can be directly plugged into new VLMs as they are released (which is very often at the

moment) to benefit from improvements in their descriptive power. It can also be plugged into propriety models because no training is required.

We also investigate generating AD automatically for TV series, such as *"Friends"* and *"The Big Bang Theory"*. This domain differs from movies in a number of aspects: a TV series can often have a cast of characters that is persistent over multiple episodes; and due to shorter episodes, the series often have denser dialogues than films. In older series, the cinematography can also differ, with more close-ups and mid-shots in TV than in movies, whereas there may be more long shots in movies. To this end, we introduce a new dataset of TV series (TV-AD) with ground truth aligned AD annotations.

In summary, we make the following contributions: (i) We devise a visual-textual prompting mechanism to incorporate character information into the VLM in a training-free manner; (ii) We propose a two-stage AD generation process, where a comprehensive character-aware video description is first generated by a VLM, followed by an LLM-based summary stage to obtain the final AD; (iii) We formulate a new TV-AD dataset and investigate the nature of TV ADs in comparison to their movie counterparts; (iv) The proposed AutoAD-Zero model is evaluated on multiple movie/TV AD datasets, including MAD-Eval [13], CMD-AD [13], and TV-AD, significantly outperforming existing training-free AD generation methods. When compared to models explicitly fine-tuned on ground truth AD annotations, AutoAD-Zero demonstrates competitive performance and achieves state-of-the-art results on the CRITIC scores.

2 Related Work

Previous AD and video captioning models. With the power of large vision and language models, rapid progress has been made in video captioning. The dense video captioning task focuses on short video clips [25, 27, 32], and other variants summarise longer video segments that span a few minutes [26, 47]. The task of Audio Description (AD) generation is closer to dense video captioning with some differences, where the model aims to describe visual elements in the movie or TV densely over time, in a story-telling manner. Unlike video captioning, AD generation requires a highly summarised output with strict formatting, coherent context [13, 38, 45], and character information [12, 28]. Notably in [12], a separate character recognition module is trained to feed character identities to the VLM, which is shown to be critical compared with an alternate method that applies GPT-4 model for AD generation [46]. Character recognition in movies or TV typically can be achieved by face matching with portrait exemplars [15] or joint audio-visual matching that includes voice [20]. The emotion of the characters is also explored for movie descriptions [35].

On the data acquisition side, due to the copyright constraints of movies and TV series, early works share short video clips [29] or pre-computed visual features [34], limiting the scale and power of AD generation models. [14] proposes to align YouTube videos from [3] with AD annotations to acquire longer movie

clips with pixels. A recent work [10] collects datasets with amateur-made short movies (13 mins on average) that are publicly available and easier to share.

Prompting for LLMs and VLMs. Prompting refers to designing instructions that guide the pre-trained models to generate desired outputs. One stream of works learns vector prompts with targeted training data, on both LLMs [22, 23] and VLMs [18, 48, 49]. Another stream designs text-only prompts which is more compatible with black-box pre-trained models prohibiting any model modifications. For example, it is found that adding a text prompt "let's think step by step" improves the performance of GPT3 on various tasks [19]. Using text prompts with chain-of-thought examples can activate the reasoning in LLM's answers [39].

Our work is closer to "visual prompting", that provides supplementary *visual* information to the VLMs without any fine-tuning [4, 17]. In [33], the authors explore visual prompting for large-scale VLMs like CLIP, discovering that drawing a red circle around an object can effectively direct the model's attention, leading to strong performance in zero-shot referring expressions comprehension, and keypoint localization tasks. In [5], the authors introduce a multimodal model that decodes arbitrary visual prompts, i.e. visual markers overlayed on the RGB image, allowing intuitive interaction with images through natural cues such as "red bounding box" or "pointed arrow". In [43], the authors introduce Set-of-Mark (SoM), a visual prompting method that enhances the visual grounding capabilities of large multimodal models like GPT-4v. By using interactive segmentation models to partition images and overlaying them with marks, SoM enables models to answer questions requiring visual grounding.

Training-free VLM adaptation. With the strong performance and huge training cost of large-scale pretrained VLMs, many works attempt to adapt VLMs to specific tasks in a training-free manner. Yan *et al*. [44] showcases GPT-4v on a wide range of multi-modal applications, such as recognizing celebrities and famous landmarks, and understanding code snippets from screenshots. State-of-the-art VLMs are also adapted to various medical applications [40, 42], but these works find the VLMs still lack expertise in medical knowledge.

Our work is closely related to MM-Narrator [46] and LLM-AD [7], where MM-Narrator leverages GPT-4/GPT-4v with multi-modal in-context learning to generate the audio descriptions for long movies, in an auto-regressive manner with memories. A concurrent work, LLM-AD, applies visual prompting to GPT-4v by overlaying character names onto movie frames. In contrast, we introduce a visual-textual prompting approach that provides character names as a part of the text prompts with colour-coded circles as fine-grained visual indicators, leading to superior performance.

3 Prompting VLM with Character Information

To incorporate character information into ADs, prior works [12, 14, 38] explicitly tune the VLM to adapt to character information. We instead explore a training-free route that involves two steps, as detailed in Fig. 2: (i) character recognition

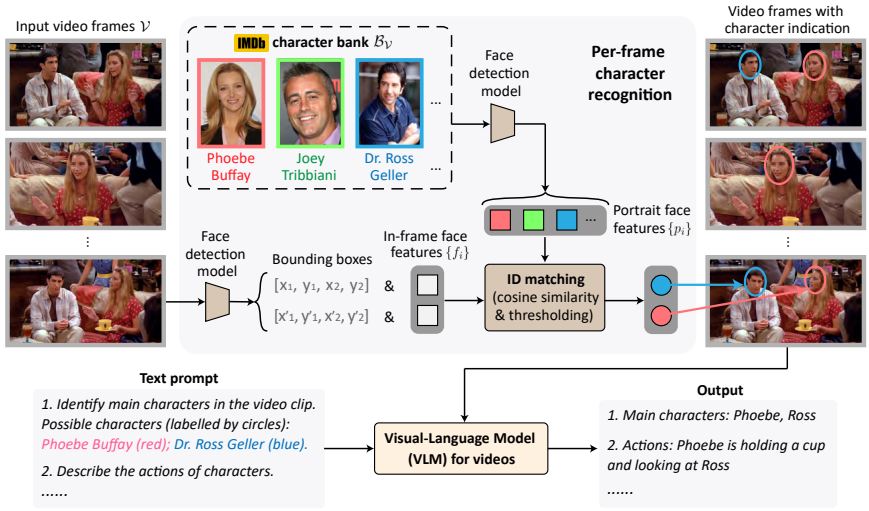

Fig. 2: Character recognition and VLM prompting. An off-the-shelf face detection model is employed to obtain bounding boxes and face features in video frames. These "in-frame face features" are then matched with "portrait face features" extracted from character profile images, which determines the identities in the video. To prompt the VLM with character information, character faces are labelled by coloured circles, with corresponding names and colour codes provided in the text prompt.

based on an off-the-shelf face detection model, then (ii) prompt VLM with visual character indications.

3.1 Character Recognition by Face Detection

Given a video clip $\mathcal{V}$ from a movie or a TV series episode, we query the corresponding cast list from the database. The video-specific character bank can be then obtained as $\mathcal{B}_\mathcal{V} = \{[\texttt{char}_i, \mathcal{P}_i]\}_{i=1}^{C}$ where there are C characters, $\texttt{char}_i$ indicates the character name, and $\mathcal{P}_i$ denotes the profile portrait image of the corresponding actor. An exemplary character bank shown in Fig. 2 is:

$$\{[\textit{"Phoebe Buffay"}, \mathcal{P}_1], [\textit{"Joey Tribbiani"}, \mathcal{P}_2], [\textit{"Dr. Ross Geller"}, \mathcal{P}_3], ...\}$$

We then formulate a pipeline to detect faces in the video frame and match them with those in the character bank. This process is *independently* conducted for each frame.

Face Feature Extraction. Given a video frame $\mathcal{I}$ in the clip $\mathcal{V}$, an off-the-shelf face detection model Φ is employed to identify faces:

$$\{\texttt{Bbox}_j, f_j\}_{j=1}^{N} = \Phi(\mathcal{I}) \tag{1}$$

where N faces are detected in a frame, with the feature embedding of each face f_j extracted, together with a predicted bounding box coordinate $\texttt{Bbox}_j$. We refer to the set of face embeddings as *in-frame face features* $\{f_i\}_{i=1}^{N}$.

Similarly, a face embedding is extracted from each portrait image $\mathcal{P}_i$ in the character bank, together forming a set of *portrait face features* $\{p_i\}_{i=1}^{C}$.

Face ID Matching. To match the faces within video frames with those in portrait images, we evaluate the cosine similarities $\mathcal{A}_{ij}$ between two sets of features and identify the best-matched portrait face for each in-frame face feature. Formally,

$$\texttt{ID}_j = \max_i(\mathcal{A}_{ij}), \quad \text{where } \mathcal{A}_{ij} = \frac{p_i \cdot f_j}{|p_i||f_j|} \tag{2}$$

where $\texttt{ID}_j$ denotes the predicted face ID in the character bank. As a result, the predicted bounding boxes within the video frame could be matched with a character name, forming a triplet set $\{[\texttt{char}_{\texttt{ID}_j}, \texttt{Bbox}_j, \mathcal{A}_{\texttt{ID}_j j}]\}_{j=1}^{N}$, where $\mathcal{A}_{\texttt{ID}_j j}$ is the matching score.

We filter out triplets based on the matching scores and the following rules:

- If $\mathcal{A}_{\texttt{ID}_j j} < \epsilon$, where ϵ denotes a threshold, this indicates that the face matching is not confident, possibly owing to poor video quality, motion blur, non-important characters' faces, *etc.* We therefore remove those faces (triplets) from the set.
- If the predicted IDs for multiple faces are the same within a frame, we keep the one with the highest matching score $\mathcal{A}_{\texttt{ID}_j j}$.

3.2 Visual Prompting VLM by Circling Character Faces

In the training-free design, one key challenge lies on the strategy to prompt VLM for understanding the association between visible characters and their names. Recent work [33] has revealed that drawing a red circle (ellipse) could effectively direct the CLIP's attention to a local region while maintaining global information understanding. Inspired by this, and given the fact that CLIP is popularly adopted as the visual encoder for recent VLMs, we overlay coloured circles around character faces in the frame as direct visual indications.

As shown in Fig. 2, we assign a unique colour code for each character, and transform the corresponding bounding box coordinates into circles that are overlaid on the raw video frames. These frames with coloured circles are then directly taken as input to a VLM, while the character names and corresponding colours are listed in the input text prompts, e.g. *"Phoebe Buffay (red)"*. By leveraging coloured circles as a bridge, it can be observed that the VLM is capable of associating character information between text and video and using it to accomplish other tasks, such as action description.

4 Two-Stage Training-Free AD Generation

In this section, we introduce a training-free method to generate AD sentences from video frame inputs. This approach consists of two stages: (i) comprehensive character-aware video description by a VLM, and (ii) AD summary by an LLM based on the outputs of the first stage.

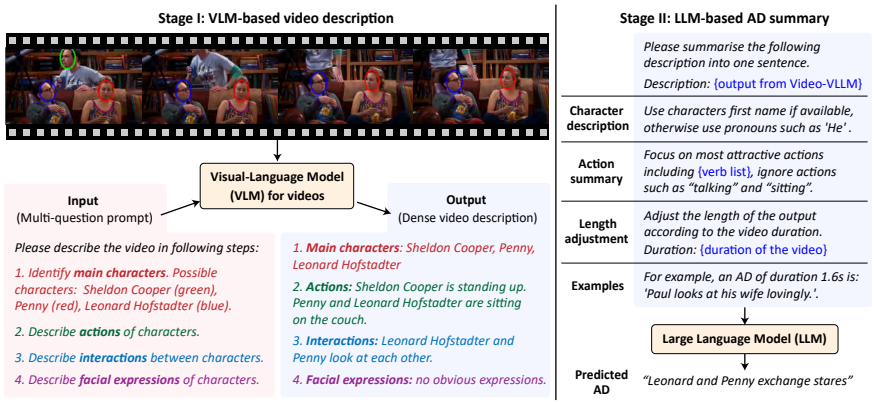

Fig. 3: Two-stage training-free AD generation. The first stage adopts a VLM to produce a comprehensive video description, covering aspects including main characters, actions, interactions, and facial expressions. The second stage uses an LLM to summarise the video into a single AD sentence, extracting the most relevant character and action information, and adjusting the content and style according to specific rules.

4.1 Stage I: VLM-Based Video Description

The VLM processes video frames along with a list of text-form questions (e.g. who are the characters? and what are they doing?) to generate a comprehensive video description. Compared to directly outputting a one-sentence AD, this process adopts the Chain-of-Thought (CoT) approach, guiding the model to systemically understand the video.

Fig. 3 (left) provides a simplified version of the input text prompt, where we begin by asking the model to **identify the main characters** ("who") in the video (point 1). Following the process described in Sec. 3.2, we label individual characters with colored circles and refer to them in the text by specifying the colors, such as *"Sheldon Cooper (red)"*. We found that by first asking the model for character names, it effectively remembers and naturally uses these names for describing actions.

We then proceed with questions about "what", that consists of three major directions, asking the model to:

- Describe the **actions** of characters (point 2). This includes characters' states (e.g. sitting or standing), gestures, movements (e.g. walking), and interactions with objects (e.g. pouring a drink).
- Describe the **interactions** between characters (point 3). Despite potentially overlapping with point 2, this point emphasises the importance of focusing globally on multiple characters and understanding their interactions, including physical contacts (e.g., hugging) and distant communications (e.g. looking at others).
- Describe the **facial expressions** of characters (point 4). This instruction encourages the model to focus on facial details that may convey the charac-

ters' emotions. Furthermore, directing the model to zoom in on facial regions enhances the accuracy of character recognition.

Additional directions on describing the environment and character appearances are also investigated, but no significant improvements are observed (see ablations in Sec. 6.5).

These points together form a multi-question prompt that is fed into VLM along with the raw video frames. We ensure that the model answers are in the same order as the questions (the template is shown in Fig. 3), resulting in a dense video description. One can also ask the VLM model to summarise the description into a one-sentence AD at the end. However, we found it challenging to enforce its adherence to the AD format while simultaneously generating high-quality descriptions.

4.2 Stage II: LLM-Based AD Summary

As demonstrated in Fig. 3 (right), the second stage aims to summarise the video description obtained from the first stage into a single AD sentence, while adjusting the outputs according to specific rules. Note that, we use an LLM for this task that takes text-form inputs only (no video inputs).

Specifically, the text prompt starts with *"Please summarise the following description into one sentence."*, followed by the provision of first-stage outputs. The remaining instructions contain four main components:

- **Character description**, which formalises the use of character names (i.e. only the first name) and pronouns, rather than descriptions such as "a man".
- **Action summary**, which asks the LLM to extract the most important characters and their actions. To guide this process, we extract top-k ($k = 15$) most frequent verbs in *training splits* of AD datasets to form a verb list. Additionally, actions related to static states (e.g. sitting) and talking-related actions are considered non-informative for ADs, as they can be inferred from previous contexts or dialogues.
- **Length adjustment**, which provides the duration of the time windows for ADs as hints that indicate how much detail should be included in the output.
- **Examples**, which offer references for the LLM regarding AD styles and lengths.

After combining all these factors into the text prompt, the LLM outputs a single sentence as the summarised AD.

5 TV-AD Dataset

In this section, we introduce a new dataset, namely TV-AD, that contains ground truth audio descriptions for TV series. Sec. 5.1 specifies the dataset details and provides a comparison with a movie AD dataset. Sec. 5.2 elaborates on how the dataset is formulated by aligning AD annotations with TV episodes.

5.1 Dataset Details

Tab. 1 provides statistics of the TV-AD dataset, which features episodes across multiple TV series including *"The Big Bang Theory", "Friends", "Frasier", "Seinfeld"*, etc. (The full list is available in the arxiv version [41].) We further divide the dataset into training (TV-AD-Train, $\sim$ 31k ADs) and evaluation splits (TV-AD-Eval, $\sim$ 3k ADs), ensuring that the TV series do not overlap between the two splits. The evaluation split contains AD annotations for TV videos that are publicly available (from TVQA [21]) and will be publicly released.

Split	TV series	Seasons	Episodes	ADs	Total duration
Train	11	18	326	31030	214h
Eval	2	5	100	2983	37h
Total	13	23	426	34013	251h

Table 1: Statistics of the TV-AD dataset. Train and Eval splits consist of different TV series.

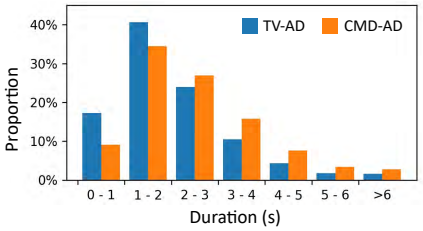

Fig. 4: Comparison of AD duration between TV-AD and CMD-AD.

Comparison with CMD-AD. CMD-AD [14] (where "CMD" stands for Condensed Movie Dataset [3]) is a movie AD dataset that contains $\sim$ 100k ADs from $\sim$ 1.4k movies. The duration of ADs between TV-AD and CMD-AD are compared in Fig. 4. Movie ADs (mean duration = 2.51s) are shown to be generally longer, while TV ADs (mean duration = 2.07s) consist of a large proportion within $1-2$s. This could be due to the shorter time of each TV episode compared to movies, as well as fast-paced plots and dialogues in TV series, resulting in more compressed audio descriptions.

5.2 Dataset Formulation: Aligning ADs with TV Series

The AudioVault website[4] (as also used by [13,14,34]) provides human-annotated audio descriptions for thousands of movies and TV series in the form of audio files, with the spoken ADs merged with the original TV soundtracks. To formulate the TV-AD dataset, we (i) temporally align the AudioVault soundtracks with TV episodes; (ii) extract human-annotated ADs from AudioVault soundtracks and transcribe them into text forms.

As noted by [14], the alignments between movies and AudioVault soundtracks are non-trivial owing to (i) non-identical soundtracks; (ii) different recording speeds (e.g. NTSC 29.97 fps *vs.* RAL 25 fps). Furthermore, in TV series, ADs are typically shorter than those in movies, and there is more frequent mixing between spoken ADs and sound effects (e.g. laughing). Both factors pose greater challenges for accurate AD extraction from the soundtracks.

To solve these challenges, we propose an improved pipeline that starts with AudioVault and TV soundtracks and consists of two major steps, namely soundtrack pre-processing and AD filtering, as demonstrated in Fig. 5.

[4] `https://audiovault.net/`

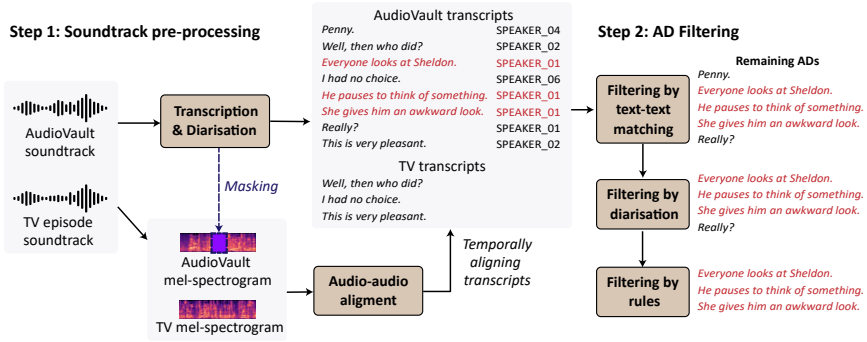

Fig. 5: Dataset formulation. The aim is to convert AD annotations from AudioVault into text form and align them with TV episodes. The main pipeline consists of two steps: (i) The soundtrack pre-processing step, which aligns the AudioVault and TV timestamps via audio-audio matching and transcribes both sound sources into transcripts; (ii) The AD filtering step, which retrieves text-form ADs from the transcripts and performs further cleaning.

Step 1: Soundtrack Preprocessing. In this step, we convert both AudioVault and TV soundtracks into text-form transcripts and temporally align them. In detail, we first use WhisperX [2] to transcribe both soundtracks into text forms along with the corresponding timestamps. Additionally, we apply a diarisation module to identify speakers for the AudioVault transcripts. Formally, the resultant transcripts are denoted as $\mathcal{S}_{\text{AV}} = \{(s_1, t_1, c_1), \dots (s_n, t_n, c_n)\}$ and $\mathcal{S}_{\text{TV}} = \{(s'_1, t'_1), \dots (s'_m, t'_m)\}$, where s_i represents transcribed sentences, along with the time stamps t_i, and c_i indicates the speaker IDs estimated by the diarisation module.

These transcribed sentences are then temporally matched in terms of the timestamps by an audio-audio alignment. This process follows the same design as proposed in [14]. (Note, we found audio-audio alignment alone is already precise enough, therefore skipping the first text-text alignment step in their original proposal). Specifically, the soundtracks are transformed into mel-spectrograms ($\mathcal{M}_{\text{AV}}$ and $\mathcal{M}_{\text{TV}}$) that capture the low-level audio representations. Based on the diarisation results, we mask out the regions in the AudioVault mel-spectrogram that overlap with AD timestamps. We then evaluate the correlation map $\mathcal{M}_{\text{AV}}$ and $\mathcal{M}_{\text{TV}}$, and fit it with a linear function f using RANSAC. This function can be used to convert timestamps from AudioVault to TV soundtracks, i.e. $f{:}\{t_{\text{AV}} \rightarrow t_{\text{TV}}\}$, therefore aligning two modalities.

Step 2: AD Filtering. As previously noted, simple diarisation on AudioVault soundtracks for TV series can not accurately determine AD speakers with non-AD speakers, and therefore cannot be directly applied to extract ADs. We instead formulate a multi-step filtering mechanism that identifies ADs from ASR transcripts that contain both AD and non-AD speakers.

First, we compare the pairs of AudioVault sentences s_t (either AD or non-AD speakers) and TV sentences (just non-AD speakers) $s'_{t\pm\Delta t}$, with Δt denoting the maximum difference between their timestamp centers. If the word error rate (WER) between a pair of sentences is below a threshold ϵ', i.e. $\text{WER}(s_t, s'_{t\pm\Delta t}) < \epsilon'$, we consider the two sentences are matched (therefore correspond to a non-AD speaker), and remove the sentence s_t from the AudioVault transcripts. As exemplified in Fig. 5, this filtering by text-text-matching effectively removes a majority of non-AD contents, though still affected by transcription and alignment errors.

Second, we perform diarisation-based filtering that first determines the AD speaker by majority voting, followed by the removal of all sentences from non-AD speakers. After filtering, a few non-AD sentences may remain, albeit marginally, due to potential diarisation errors.

Finally, we conduct rule-based heuristic filtering, which discards sentences containing certain symbols (e.g. *"?"* and *"!"*), pronouns (e.g. *"I"* and *"We"*), and phrases (e.g. *"Let's go."*), that never appear in ADs. Further details are discussed in the arxiv version [41].

6 Experiments

6.1 Datasets

We evaluate the model performance using several datasets, focusing on character recognition (as an intermediate task), movie AD generation, and TV AD generation.

Character Recognition Dataset. For character recognition evaluation, we follow [12] and adopt a 4-movie subset of **MovieNet** [16], where ground-truth character names are provided in each keyframe (shot) within the movie.

Movie AD Datasets. We report the performance on two movie AD datasets, namely **CMD-AD** [14] and **MAD-Eval** [13]. The former is generated by aligning AD data with video clips from CMD (Condensed Movie Dataset) [3], resulting in 101k ground truth ADs within $\sim$ 1.4k movies. In this work, we adopt its evaluation set (CMD-AD-Eval), which consists of $\sim$ 7k ADs for 100 movies. On the other hand, MAD-Eval is a 10-movie subset from LSMDC [30], featuring $\sim$ 6.5k ground truth AD annotations.

TV AD Dataset. We adopt the new **TV-AD** dataset (introduced in Sec. 5) and measure the performance on its evaluation split, which includes around 3k ADs in 100 TV episodes.

6.2 Evaluation Metrics

To measure the **character recognition performance** in the intermediate stage, we compare the predicted and ground truth character name lists and compute the IoU, precision, and recall of the prediction.

For AD evaluation, we monitor the performance on three metrics, namely CIDEr, CRITIC, and LLM-AD-Eval.

Methods	IoU	Precision	Recall
AutoAD-II [12]	70.8	75.8	85.6
AutoAD-Zero **(Ours)**	**75.8**	**79.3**	**85.8**

Table 2: Character recognition results reported on 4 MovieNet movies. Note, AutoAD-III [14] shares the same character recognition module as AutoAD-II [12].

CIDEr [37] is a common metric that evaluates the quality of text descriptions. It measures the relevance and uniqueness of words in candidate descriptions based on a Term Frequency-Inverse Document Frequency (TF-IDF) weighting scheme. However, this word-matching-based metric has its limitations, as ADs with the same semantics can be presented in various ways. Since many recent works have adopted CIDEr scores, we report the performance for comparison purposes.
CRITIC [14] measures the accuracy of character identification in the predicted ADs. Specifically, a co-referencing model is employed to replace ambiguous pronouns such as *"He"* or *"She"* in ADs with official names from the character banks. Then, two name sets from each pair of predicted and GT ADs are compared, and the IoU is calculated to give a CRITIC score.
LLM-AD-Eval [14] utilises LLMs to judge the quality of generated ADs by scoring them between 1 (lowest) and 5 (highest). The evaluation focuses on the matching between generated and ground truth ADs in terms of human actions, objects, and interactions. We directly adopt the LLM prompt provided in the original paper and use open-source models LLaMA2-7B [36] and LLaMA3-8B [1] for the evaluation.

6.3 Implementation Details

Character Recognition Details. To formulate the character bank for each movie or TV episode, we extract top-10 characters from the IMDb website[5] along with their corresponding actor profile images. For face detection and recognition, we adopt the InsightFace package[6], which is developed based on ArcFace [9] and RetinaFace [8]. The threshold for face matching (detailed in 3.1) is set to $\epsilon = 0.2$.
AD Prediction Details. For stage I, we use VideoLLaMA2-7B [6] as the Video-LLM model, which takes in 8 video frames. At stage II, we adopt LLaMA3-8B [1] for AD summary. The exact text prompts are available in the arxiv version [41].

6.4 Character Recognition Results

In Tab. 2, we compare the character recognition module with the one adopted in AutoAD-II [12], which trains a transformer-based model with CLIP feature inputs to predict the probability of each character. In contrast, our method first employs an off-the-shelf face detection model with a face-matching mechanism

[5] `https://www.imdb.com`
[6] `https://github.com/deepinsight/insightface`

Models	Training -free	CMD-AD			MAD-Eval	TV-AD		
		CIDEr	CRITIC	LLM-AD-eval	CIDEr	CIDEr	CRITIC	LLM-AD-eval
AutoAD-I [13]	✗	–	–	–	14.3	–	–	–
AutoAD-II [12]	✗	13.5	8.2	–	19.2	–	–	–
AutoAD-III [14]	✗	**25.0**	32.7	**2.89** | **2.01**	24.0	**26.1***	15.9*	2.78* | 1.99*
Uni-AD [38]	✗	–	–	–	**28.2**	–	–	–
MM-Narrator (GPT-4) [46]	✓	–	–	–	13.9	–	–	–
MM-Narrator (GPT-4v) [46]	✓	–	–	–	9.8	–	–	–
LLM-AD [7]	✓	–	–	–	20.5	–	–	–
AutoAD-Zero **(Ours)**	✓	17.7	**43.7**	2.83 | 1.96	22.4	22.6	**27.6**	**2.94** | **2.00**

Table 3: Comparison with other methods on CMD-AD, MAD-Eval, and TV-AD. "*" denotes the results of the AutoAD-III model that we reproduced by training it on the TV-AD training split. The LLM-AD-eval scores are evaluated using LLaMA2-7B (left) and LLaMA3-8B (right).

to obtain raw character predictions. These predictions are then encoded into coloured circles to visually prompt the Video-LLM, as detailed in Fig. 2.

Following [12], we report the character recognition performance on 4 MAD-Eval movies with keyframe character annotations from MovieNet. Note that our reported performance is measured on the final Video-LLM outputs. As demonstrated in Tab. 2, AutoAD-Zero achieves higher IoU, precision, and recall in character recognition, indicating that our predicted character names are more accurate and reliable compared to those predicted by AutoAD-II.

6.5 Audio Description Predictions

In this section, we compare our training-free method with prior models and provide qualitative examples. We also present a detailed ablation analysis in the arxiv version [41], which verifies the key designs in our two-stage framework.

Quantitative Comparison. Overall, AD generation methods are majorly divided into two classes: (i) models explicitly trained on ground truth movie or TV AD annotations, including AutoAD-I to III [12–14] and Uni-AD [38]; and (ii) *training-free* approaches that perform *zero-shot* AD generation *without* fine-tuning on AD datasets, including MM-Narrator [46], LLM-AD [7], and AutoAD-Zero (our method). The latter methods are more flexible and extendable in terms of performance, as they could be directly integrated with more advanced models. However, it is challenging for training-free models to outperform those fine-tuned on human-annotated ADs that directly benefit from domain-specific knowledge.

We focus on evaluating model performance on CMD-AD and TV-AD. Since several prior works report their CIDEr scores on MAD-Eval, we also include it in our comparison. As demonstrated in Tab. 3, among the training-free methods, AutoAD-Zero surpasses MM-Narrator (both GPT-4 and GPT-4V versions) on the MAD-Eval evaluation. When compared to models fine-tuned on AD datasets, AutoAD-Zero delivers competitive results on CIDEr and LLM-AD-eval metrics and achieves state-of-the-art performance on CRITIC score, indicating our superior character identification accuracy.

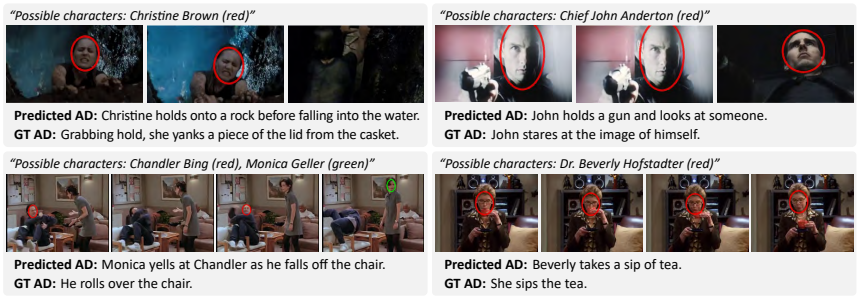

Fig. 6: Qualitative visualisations for CMD-AD (top) and TV-AD (bottom). The faces are labelled by coloured circles, with corresponding names and colour codes listed above. The examples are from *"Drag Me to Hell"* (top left), *"Minority Report"* (top right), *"Friends"* (bottom left), and *"The Big Bang Theory"* (bottom right).

Qualitative Visualisation. Fig. 6 provides several examples on CMD-AD and TV-AD. By prompting VLM with character information (i.e. coloured circles in images and names in text prompts), AutoAD-Zero is capable of associating characters with their actions. In the bottom left example, even though faces are sparsely labelled (as characters may turn away from the camera), the VLM successfully links the character across the frames (tracking) and identifies individual actions (e.g. *Monica - yelling*, *Chandler - falling off the chair*).

7 Discussion and Extensions

In this paper, we propose AutoAD-Zero, a zero-shot AD generation framework featuring visual prompting of character information and a two-stage description-summarisation process. The two-stage design offers notable advantages: it brings the extensibility to use more advanced VLMs and LLMs in the future, and the flexibility to tailor the second-stage LLM to fit specific styles.

However, the visual and text prompt design is still based on a heuristic trial-and-error approach which could be optimised. Also, errors accumulate between stages – if the first-stage VLM fails to describe an action, the second-stage LLM cannot summarise it. Moreover, recent VLMs and LLMs may have already encountered popular characters and scenes during training, potentially undermining the zero-shot nature of the task. In the future, data leakage could be alleviated by adopting more recent or amateur film data [11] for evaluation.

Potential extensions of this work include: (i) Enhancing the VLM with additional knowledge from "experts" or "specialists" (e.g. location from a scene recognition model); (ii) Expanding the second stage by providing the LLM with more context, including previous ADs, dialogues, and plot summaries.

Acknowledgments. This research is supported by the UK EPSRC Programme Grant Visual AI (EP/T028572/1), a Clarendon Scholarship, a Royal Society Research Professorship RP\R1\191132, and ANR-21-CE23-0003-01 CorVis.

References

1. AI@Meta: Llama 3 model card (2024), https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md
2. Bain, M., Huh, J., Han, T., Zisserman, A.: WhisperX: Time-accurate speech transcription of long-form audio. In: INTERSPEECH (2023)
3. Bain, M., Nagrani, A., Brown, A., Zisserman, A.: Condensed movies: Story based retrieval with contextual embeddings. In: ACCV (2020)
4. Bar, A., Gandelsman, Y., Darrell, T., Globerson, A., Efros, A.: Visual prompting via image inpainting. NeurIPS (2022)
5. Cai, M., Liu, H., Mustikovela, S.K., Meyer, G.P., Chai, Y., Park, D., Lee, Y.J.: Making large multimodal models understand arbitrary visual prompts. In: CVPR (2024)
6. Cheng, Z., Leng, S., Zhang, H., Xin, Y., Li, X., Chen, G., Zhu, Y., Zhang, W., Luo, Z., Zhao, D., Bing, L.: Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. arXiv preprint arXiv:2406.07476 (2024)
7. Chu, P., Wang, J., Abrantes, A.: LLM-AD: Large language model based audio description system. arXiv preprint arXiv:2405.00983 (2024)
8. Deng, J., Guo, J., Ververas, E., Kotsia, I., Zafeiriou, S.: Retinaface: Single-shot multi-level face localisation in the wild. In: CVPR (2020)
9. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: ArcFace: Additive angular margin loss for deep face recognition. In: CVPR (2019)
10. Ghermi, R., Wang, X., Kalogeiton, V., Laptev, I.: Short Film Dataset (SFD): A benchmark for story-level video understanding. arXiv preprint arXiv:2406.10221 (2024)
11. Ghermi, R., Wang, X., Kalogeiton, V., Laptev, I.: Short film dataset (sfd): A benchmark for story-level video understanding. arXiv preprint arXiv:2406.10221 (2024)
12. Han, T., Bain, M., Nagrani, A., Varol, G., Xie, W., Zisserman, A.: AutoAD II: The sequel – who, when, and what in movie audio description. In: ICCV (2023)
13. Han, T., Bain, M., Nagrani, A., Varol, G., Xie, W., Zisserman, A.: AutoAD: Movie Description in Context. In: CVPR (2023)
14. Han, T., Bain, M., Nagrani, A., Varol, G., Xie, W., Zisserman, A.: AutoAD III: The prequel – back to the pixels. In: CVPR (2024)
15. Huang, Q., Liu, W., Lin, D.: Person search in videos with one portrait through visual and temporal links. In: ECCV (2018)
16. Huang, Q., Xiong, Y., Rao, A., Wang, J., Lin, D.: Movienet: A holistic dataset for movie understanding. In: ECCV (2020)
17. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: ECCV (2022)
18. Ju, C., Han, T., Zheng, K., Zhang, Y., Xie, W.: Prompting visual-language models for efficient video understanding. In: ECCV (2022)
19. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. NeurIPS (2022)
20. Korbar, B., Huh, J., Zisserman, A.: Look, listen and recognise: Character-aware audio-visual subtitling. arXiv preprint arXiv:2401.12039 (2024)
21. Lei, J., Yu, L., Bansal, M., Berg, T.L.: TVQA: Localized, compositional video question answering. In: EMNLP (2018)
22. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691 (2021)

23. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190 (2021)
24. Lin, K., Ahmed, F., Li, L., Lin, C.C., Azarnasab, E., Yang, Z., Wang, J., Liang, L., Liu, Z., Lu, Y., Liu, C., Wang, L.: MM-Vid: Advancing video understanding with GPT-4V(ision). arXiv preprint arXiv:2310.19773 (2023)
25. Lin, K., Li, L., Lin, C.C., Ahmed, F., Gan, Z., Liu, Z., Lu, Y., Wang, L.: SwinBERT: End-to-end transformers with sparse attention for video captioning. In: CVPR (2022)
26. Lu, Z., Grauman, K.: Story-driven summarization for egocentric video. In: CVPR (2013)
27. Luo, H., Ji, L., Shi, B., Huang, H., Duan, N., Li, T., Chen, X., Zhou, M.: UniViLM: A unified video and language pre-training model for multimodal understanding and generation. arXiv preprint arXiv:2002.06353 (2020)
28. Raajesh, H., Desanur, N.R., Khan, Z., Tapaswi, M.: Micap: A unified model for identity-aware movie descriptions. In: CVPR (2024)
29. Rohrbach, A., Rohrbach, M., Tandon, N., Schiele, B.: A dataset for movie description. In: CVPR (2015)
30. Rohrbach, A., Torabi, A., Rohrbach, M., Tandon, N., Pal, C., Larochelle, H., Courville, A., Schiele, B.: Movie description. IJCV (2017)
31. Salewski, L., Alaniz, S., Rio-Torto, I., Schulz, E., Akata, Z.: In-context impersonation reveals large language models' strengths and biases. NeurIPS (2024)
32. Seo, P.H., Nagrani, A., Arnab, A., Schmid, C.: End-to-end generative pretraining for multimodal video captioning. In: CVPR (2022)
33. Shtedritski, A., Rupprecht, C., Vedaldi, A.: What does CLIP know about a red circle? visual prompt engineering for VLMs. In: ICCV (2023)
34. Soldan, M., Pardo, A., Alcázar, J.L., Caba, F., Zhao, C., Giancola, S., Ghanem, B.: MAD: A scalable dataset for language grounding in videos from movie audio descriptions. In: CVPR (2022)
35. Srivastava, D., Singh, A.K., Tapaswi, M.: How you feelin'? Learning emotions and mental states in movie scenes. In: CVPR (2023)
36. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C.C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P.S., Lachaux, M.A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E.M., Subramanian, R., Tan, X.E., Tang, B., Taylor, R., Williams, A., Kuan, J.X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., Scialom, T.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv: 2307.09288 (2023)
37. Vedantam, R., Lawrence Zitnick, C., Parikh, D.: CIDEr: Consensus-based image description evaluation. In: CVPR (2015)
38. Wang, H., Tong, Z., Zheng, K., Shen, Y., Wang, L.: Contextual AD narration with interleaved multimodal sequence. arXiv preprint arXiv: 2403.12922 (2024)
39. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. NeurIPS (2022)

40. Wu, C., Lei, J., Zheng, Q., Zhao, W., Lin, W., Zhang, X., Zhou, X., Zhao, Z., Zhang, Y., Wang, Y., et al.: Can GPT-4V(ision) serve medical applications? Case studies on GPT-4V for multimodal medical diagnosis. arXiv preprint arXiv:2310.09909 (2023)
41. Xie, J., Han, T., Bain, M., Nagrani, A., Varol, G., Xie, W., Zisserman, A.: AutoAD-Zero: A training-free framework for zero-shot audio description. arXiv preprint arXiv:2407.15850 (2024), https://arxiv.org/abs/2407.15850
42. Yan, Z., Zhang, K., Zhou, R., He, L., Li, X., Sun, L.: Multimodal ChatGPT for medical applications: an experimental study of GPT-4V. arXiv preprint arXiv:2310.19061 (2023)
43. Yang, J., Zhang, H., Li, F., Zou, X., Li, C., Gao, J.: Set-of-mark prompting unleashes extraordinary visual grounding in GPT-4V. arXiv preprint arXiv:2310.11441 (2023)
44. Yang, Z., Li, L., Lin, K., Wang, J., Lin, C.C., Liu, Z., Wang, L.: The dawn of LMMs: Preliminary explorations with gpt-4v (ision). arXiv preprint arXiv:2309.17421 (2023)
45. Ye, X., Chen, J., Li, X., Xin, H., Li, C., Zhou, S., Bu, J.: MMAD: Multi-modal movie audio description. In: LREC-COLING (2024)
46. Zhang, C., Lin, K., Yang, Z., Wang, J., Li, L., Lin, C.C., Liu, Z., Wang, L.: MM-Narrator: Narrating long-form videos with multimodal in-context learning. In: CVPR (2024)
47. Zhang, K., Chao, W.L., Sha, F., Grauman, K.: Video summarization with long short-term memory. In: ECCV (2016)
48. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: CVPR (2022)
49. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. IJCV **130**(9) (2022)