

Shot-by-Shot: Film-Grammar-Aware Training-Free Audio Description Generation

Junyu Xie¹ Tengda Han¹ Max Bain¹ Arsha Nagrani¹ Eshika Khandelwal^{2,3}
Gül Varol^{1,3} Weidi Xie^{1,4} Andrew Zisserman¹

¹Visual Geometry Group, University of Oxford ²CVIT, IIT Hyderabad

³LIGM, École des Ponts ParisTech ⁴SAI, Shanghai Jiao Tong University

<https://www.robots.ox.ac.uk/vgg/research/shot-by-shot/>

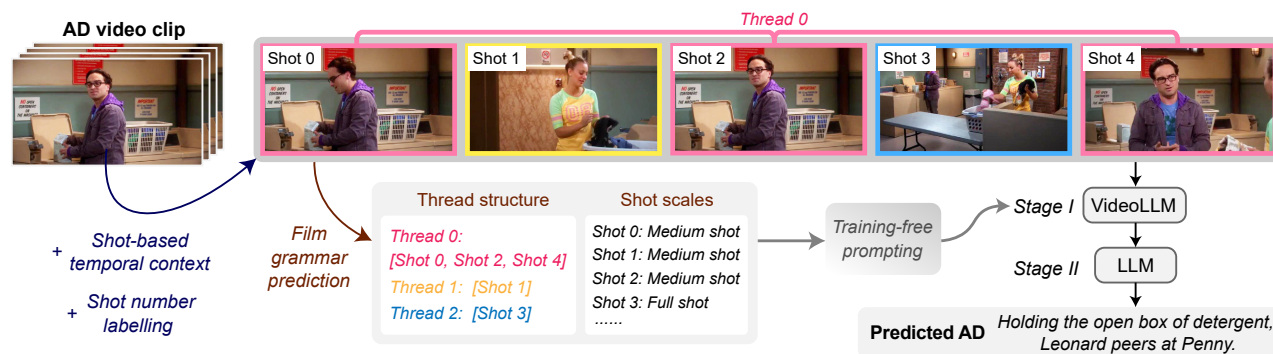


Figure 1. **Overview of our training-free Audio Description (AD) generation framework.** The input video clip (corresponding to the AD interval) is extended to include adjacent shots, providing richer temporal visual context. Each video frame is labelled with its corresponding shot number. The extended clip then undergoes a film grammar prediction process, where the thread structure and shot scales are estimated. The AD is generated in two stages: Stage I utilises the predicted cinematic information as training-free prompt guidance to produce dense video descriptions. Stage II then employs an LLM to generate a summarised AD output.

Abstract

Our objective is the automatic generation of Audio Descriptions (ADs) for edited video material, such as movies and TV series. To achieve this, we propose a two-stage framework that leverages “shots” as the fundamental units of video understanding. This includes extending temporal context to neighbouring shots and incorporating film grammar devices, such as shot scales and thread structures, to guide AD generation. Our method is compatible with both open-source and proprietary Visual-Language Models (VLMs), integrating expert knowledge from add-on modules without requiring additional training of the VLMs. We achieve state-of-the-art performance among all prior training-free approaches and even surpass fine-tuned methods on several benchmarks. To evaluate the quality of predicted ADs, we introduce a new evaluation measure – an action score – specifically targeted to assessing this important aspect of AD. Additionally, we propose a novel evaluation protocol that treats automatic frameworks as AD generation assistants and asks them to generate multiple candidate ADs for selection.

1. Introduction

In movies and TV series, Audio Descriptions (ADs) are narrations provided for the visually impaired, conveying visual information to complement the original soundtrack. Their purpose is to ensure a continuous and coherent narrative flow, enabling audiences to follow the plot effectively. Unlike video captions, ADs are constrained by length, prioritising the most visually salient and story-centric information, such as character dynamics and significant objects, whilst omitting redundant details like background figures or unchanging locations. Additionally, ADs are typically produced by professional narrators in a specific style and format, ensuring coherence while not interfering with the original audio.

With the advent of Visual-Language Models (VLMs), there has been a growth of interest in automatically generating ADs for both movies [19, 26–28, 50, 82, 97, 101] and TV material [22, 92]. However, as anyone who has ever watched a movie or read a book about film editing knows – the fundamental unit of edited video material is the *shot*, not the frame [40, 56, 58]. Shots are used to *structure* the video material, defining the granularity and temporal context through choices in its scale (close-up, long shot, etc.) and its movement (panning, tracking,

etc.). The “film grammar” is then used to convey meaning through specific editing choices on shot transitions (cuts, fades, etc.), durations, and composition (threads, montage, etc.).

Current approaches to automating AD generation, and video large language models (VideoLLMs) in general, are not shot-aware, hindering interpretation of edited video material that has frequent shot transitions [76]. In this paper, our principal objective is to incorporate shot information and editing structure into the AD generation process. To this end, we consider temporal context in terms of shots, and take account of the two key properties of *thread structure* and *shot scale*.

Thread structure identifies the sequence of shots captured with the same camera. An example of a scene with multiple simultaneous threads in space-time is shown in Fig. 3. We develop a robust thread clustering method and use its predictions to guide VLMs in understanding shot-wise relationships.

Shot scale typically implies the type of content of the frame. For instance, close-up shots often highlight characters’ facial expressions or gaze interactions, while long shots tend to depict the overall environment and ambience of the scene, as shown in Fig. 4 (top). We leverage this property by developing an off-the-shelf shot scale classifier and employing a scale-dependent prompting strategy to improve the contextual relevance of VLM-generated descriptions.

In this paper, we develop a *training-free* AD generation approach. Inspired by AutoAD-Zero [92], as illustrated in Fig. 1, we adopt a two-stage pipeline in which a VideoLLM generates dense text descriptions in the first stage, followed by an LLM that produces the final AD outputs from this text. We improve the effectiveness of training-free approaches by incorporating two key improvements based on: (i) shot-based temporal context; and (ii) the shot scale and thread structure, that are both crucial to cinematic composition and understanding. Previous methods could describe human-object interactions or human-human interactions, such as looks [52], within a frame or shot. With the improvements we introduce, the generated AD also includes such interactions when they are implied by the shot structure.

As well as the challenge of generating ADs, another challenge is *how to evaluate* the predicted ADs. Apart from conventional metrics [10, 48, 62, 81], several new AD metrics [26, 28, 101] have been proposed, including CRITIC [28] measuring *who* is mentioned in the AD. However, these metrics fail to emphasise character *actions* – one of the most critical aspects of ADs. To address this, we introduce an *action score*, assessing whether the predictions accurately capture the actions described in GT AD, independent of character information.

Moreover, as ADs are time-limited, there is always a choice on which visual aspect (characters, actions, objects, etc.) should be included. Consequently, a single video clip can have multiple equally valid ADs, each highlighting a different aspect. This observation is supported by the inter-rater agreement experiments in AutoAD-III [28] and previous user studies [88]. Therefore, beyond evaluating single AD performance, we assess our framework’s capability as an *assistant* to generate multiple AD

candidates, and report the performance of the selected *one* AD.

In summary, we propose an enhanced training-free AD generation framework, with the following contributions:

- We incorporate shot-based temporal context into AD generation via training-free prompting techniques including shot number referral and dynamic frame sampling.
- We develop state-of-the-art methods for thread structure and shot scale predictions, and demonstrate that incorporating predicted film grammar knowledge enhances AD generation.
- We improve the current AD evaluation by introducing the character-free action score, and a new assistance-oriented evaluation protocol.
- Our approach achieves state-of-the-art performance in training-free AD generation, and furthermore, surpasses fine-tuned models on multiple benchmarks. This is the first time a training-free approach has achieved superior performance to fine-tuned methods.

2. Related work

Audio Description generation. Efforts have been made to curate Audio Description (AD) datasets for both movies [28, 75] and TV series [92], with human annotations sourced from platforms such as AudioVault [2].

For automatic AD generation, prior works [22, 26–28, 50, 82] fine-tune pre-trained VLMs on AD annotations to produce descriptions in an end-to-end manner. However, these methods face challenges due to limited high-quality AD annotations and the high computational cost of fine-tuning each new backbone. Alternatively, training-free approaches [19, 92, 97, 101] have gained traction for their scalability and flexibility, allowing customised AD output based on official guidelines [1] or specific needs. Yet, these methods still lag in performance, while our approach is the first to achieve results on par with fine-tuned methods.

Instead of limiting the video input to each AD interval, UniAD [82] and DistinctAD [22] fine-tune VLMs with multiple AD clip inputs to incorporate broader temporal context. In contrast, our method systematically extends AD clip to adjacent shots and introduces a training-free approach that enables pre-trained VideoLLMs to better capture localised temporal context.

Film grammar analysis. Prior research has sought to understand film grammar from two major perspectives: (i) intra-shot properties, (ii) shot-wise relationships.

For individual shots, several datasets [6, 34, 39, 69, 73] categorise their characteristics based on camera setups, including shot scales (examples shown in Fig. 4) and camera movements. Correspondingly, various models [17, 45, 51, 69, 80] have been developed for shot type classification.

Regarding shot-wise relationships, a few datasets [6, 12, 63] have been proposed to explore transitions (i.e. cuts) between shots, which have also been leveraged in video content generation [23, 64, 70, 104]. Beyond pairwise shot transitions,

research has also investigated longer temporal contexts with thread-based editing structures. Notably, Hoai et al. introduced the Thread-Safe [30] dataset, demonstrating that thread information can enhance action recognition. These structures have also been utilised in video-based face and people clustering [11, 77].

While prior work on film grammar has mainly focused on classification and generation tasks, we specifically utilise shot scale and thread structure information to enhance AD generation in movies and TV series.

Dense video captioning. Dense video captioning is closely related to AD generation. Early works [35, 36, 42, 84, 86] in video captioning typically treat event localisation and captioning as independent stages, whereas more recent approaches [14, 21, 37, 41, 57, 65, 68, 87, 90, 95, 108, 109] integrate these tasks in an end-to-end manner. Video captioning benchmarks cover a range of domains, including cooking [107], actions [42], movies [72], TV series [44], and open-domain settings [13, 16, 29, 54, 74, 93, 94, 99, 105].

Temporal grounding in VLM. To equip conventional VLMs with temporal grounding capability, some studies [32, 47, 89] generate additional data with enhanced temporal information for fine-tuning. A more common approach explicitly incorporates temporal information into inputs, either by inserting temporal tokens [15, 20, 24, 25, 31, 33, 66, 83] or embedding time information within visual tokens [32, 47, 89].

Training-free approaches [67, 91, 106] have also been explored for temporally grounded understanding. Notably, a recent work [91] achieves temporal grounding by overlaying frame numbers as visual prompts. Instead of using uniformly sampled timestamps, we leverage the natural shot structures in movies and TV series, adopting them as fundamental units for training-free temporal referral.

Evaluation for Audio Description. Early AD evaluation often adopts captioning metrics, including CIDEr [81], ROUGE [48], BLEU [62], and METEOR [10], which measure word/n-gram overlap with different weighting strategies. Other metrics focus on semantics, leveraging rule-based scene graph matching (SPICE [5]), word embedding similarity (BERTScore [103], Recall@k/N [26]), or LLM-based evaluation [28, 101].

However, the fixed and distinct set of character names in each video can bias conventional captioning metrics. For instance, TF-IDF [71] weighting in CIDEr assigns high importance to character names. To provide a more independent measure of character names and other AD content, CRITIC [28] assesses character recognition in predicted ADs, while we develop an action-centric metric with minimised dependence on character names.

Most prior works evaluate single AD outputs; however, training-free methods are compatible with generating multiple candidates for selection. While early studies [88] explored this through user studies, we propose a quantitative evaluation protocol to further assess AD generation methods as assistive tools.

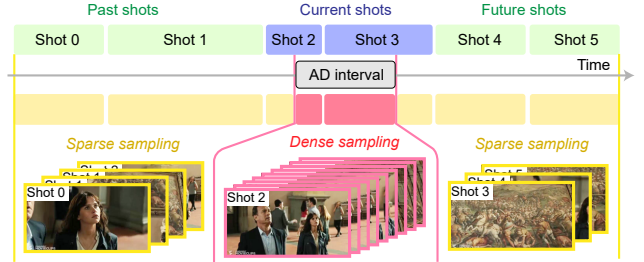


Figure 2. **Shot-based temporal context**, where *current shots* are defined as those temporally overlapping with the AD interval. *Past shots* and *future shots* provide extended contextual information. Shot numbers are visually overlaid on the top-left of each frame, and frames within the AD interval are sampled more densely than context frames.

3. Training-free AD generation framework

Given a video clip $\mathcal{V} = \{\mathcal{I}_0, \dots, \mathcal{I}_T\}$, the task of audio description is to generate a concise narration \mathcal{N} describing what happens around a given AD interval $[t_A, t_B]$. In this work, we propose a two-stage framework that leverages VideoLLMs and LLMs to predict ADs in a training-free manner.

In Stage I, we employ a VideoLLM that takes a sequence of frames (from multiple shots) as input and generates a dense description \mathcal{D} , guided by instructions $\mathcal{P}_{\text{VideoLLM}}$:

$$\mathcal{D} = \text{VideoLLM}(\mathcal{V}, [t_A, t_B], \mathcal{P}_{\text{VideoLLM}}) \quad (1)$$

In Stage II, we then prompt an LLM (with instructions \mathcal{P}_{LLM}) to extract key information from the dense Stage I description and format it into an AD-style narration \mathcal{N} :

$$\mathcal{N} = \text{LLM}(\mathcal{D}, [t_A, t_B], \mathcal{P}_{\text{LLM}}) \quad (2)$$

In this section, we focus on enhancing the visual understanding of the Stage I VideoLLM for edited video material, and make three innovations: In Sec. 3.1, we incorporate shot-based temporal context into Stage I visual inputs. In Sec. 3.2, we leverage the thread structure to enrich cross-shot understanding. Finally, in Sec. 3.3, we incorporate shot-scale awareness into Stage I prompt formulation.

3.1. Leveraging shot-based temporal context

Regarding the visual inputs to VideoLLM, prior works [28, 50] often sample frames $\{\mathcal{I}_{t_A}, \dots, \mathcal{I}_{t_B}\}$ that directly correspond to the AD interval $[t_A, t_B]$. However, this approach can be problematic due to (i) misalignment between the AD interval and the actual timestamps when the action occurs, and (ii) the lack of contextual information from adjacent shots. Therefore, we investigate how incorporating temporal context can enhance the understanding of the video clip.

Structuring temporal context with shots. To obtain more visual context information, instead of simply extending the AD interval by fixed timestamps, we explore a more structured approach that treats shots as the fundamental units.

Specifically, we apply an off-the-shelf shot segmentation

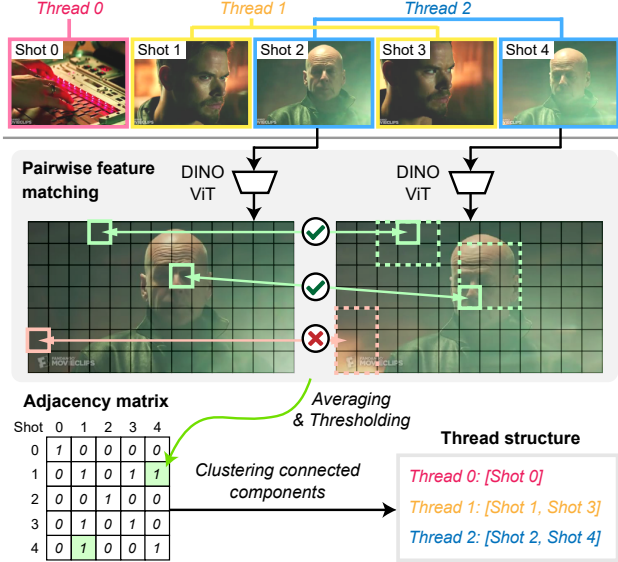


Figure 3. **Thread structure.** **Top:** Example of a thread structure with interleaving shots. **Bottom:** A training-free approach for thread clustering, where shots are pairwise compared using dense feature matching to construct an adjacency matrix, which is then used to predict the thread structure.

model to partition the entire video clip into individual shots. For each AD interval, we first identify the shots that (partially) overlap with it, referred to as “current shots”, as illustrated in Fig. 2. We then consider *at most* two “past shots” and two “future shots” adjacent to current shots as temporal context. For all shots included, we label them sequentially from past to future with a number starting from “Shot 0”.

Emphasising the targeted AD interval. To ensure that the VideoLLM focuses on describing visual content within the AD interval, we propose two strategies: dynamic frame sampling and shot number referral.

Dynamic frame sampling. As shown in Fig. 2, to emphasise the frames of interest, we adopt denser sampling within the AD interval (red region) and sparser sampling for the surrounding context frames (yellow region). In practice, we specify fixed numbers of frames to be sampled within and outside the AD interval and apply uniform sampling according to these constraints.

Shot number referral. To further enhance the attention towards the current shot content, we label each sampled frame with its shot number (e.g. “Shot 0”) at the top-left. During the formulation of the text prompt, instead of prompting the VideoLLM to “describe what happened in the video clip”, we ask it to “describe what happened in [Shot 2, Shot 3]” (i.e. current shots). Through this visual-textual prompting strategy, we found that the VideoLLM could successfully interpret the meaning of shot numbers and refer to the correct shots.

3.2. Leveraging thread structure

Movies are generally edited such that viewpoints from two or more cameras are intertwined in shot *threads*, as illustrated in

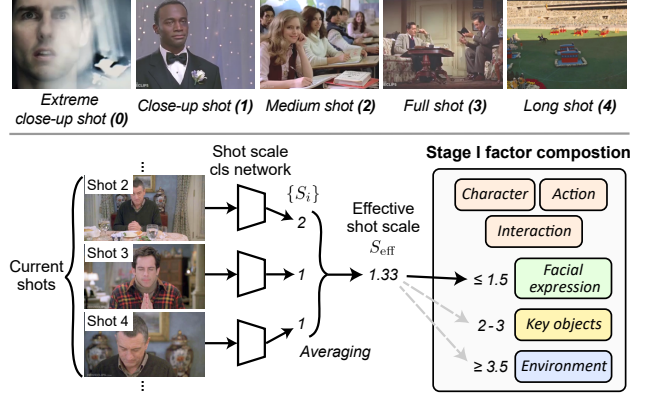


Figure 4. **Shot scales.** **Top:** Examples of five different shot scales. **Bottom:** Stage I factor composition based on shot scale classification. The scales of the current shots in the clip are predicted and averaged. The resulting effective shot scale then guides the formulation of the Stage I prompt, incorporating additional factors such as facial expressions, etc.

Fig. 3 (top). These interleaved arrangements of shot threads often imply relationships between objects and characters (e.g. gaze interactions) and their 3D arrangement. To leverage this information for Stage I description, we first determine the thread structure using a separate module, then incorporate it into the VideoLLM through prompt guidance.

Thread structure prediction. We develop a training-free method to predict thread clustering. The problem here is to determine if two shots correspond to the same viewpoint or not. Specifically, given two shots (Shot i and a later Shot j), we compare the last frame of Shot i ($\mathcal{I}_{T_i}^i$) with the first frame of Shot j ($\mathcal{I}_{T_j}^j$), extracting their DINOv2 [61] features as $f_{T_i}^i$ and $f_{T_j}^j \in \mathbb{R}^{h \times w \times c}$.

Inspired by [38], we assess frame-wise dense feature correlations by computing a cost volume between their feature maps:

$$C_{p,q}^{i,j} = m(p,q) \circ (\hat{f}_{T_i;p}^i \cdot \hat{f}_{T_j;q}^j) \quad (3)$$

where $\hat{f}_{T_i;p}^i$ and $\hat{f}_{T_j;q}^j \in \mathbb{R}^c$ are the normalised p -th and q -th spatial elements in the respective feature maps. The binary attention mask $m(p,q)$ is set to 1 only if the spatial position of the q -th element is within an $n \times n$ neighbourhood of the p -th element.

We then apply a softmax operation along the last dimension (q) and find the maximum similarity for each p , followed by averaging over all p -th elements to obtain a matching score between Shot i and Shot j :

$$s^{i,j} = \frac{1}{N} \sum_p \max_q \left(\frac{\exp(C_{p,q}^{i,j}/\tau)}{\sum_l \exp(C_{p,l}^{i,j}/\tau)} \right) \quad (4)$$

where N denotes the number of feature patches, and τ is the softmax temperature.

Intuitively, as shown in Fig. 3, this process effectively checks whether each patch (p) in one shot frame matches with a patch (q) in the other shot frame at a roughly similar spatial position

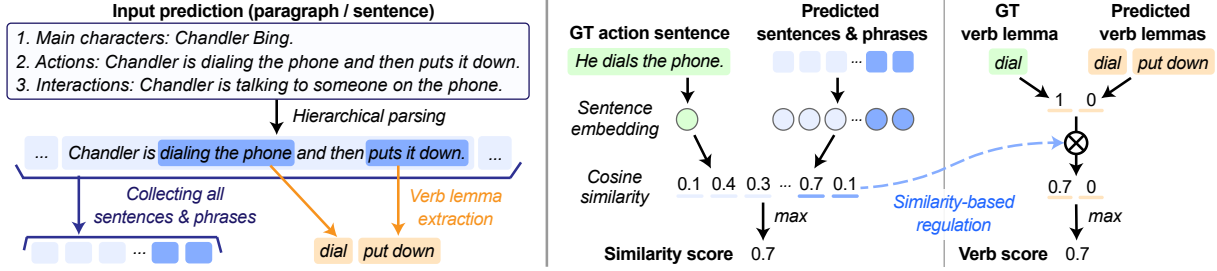


Figure 5. **Action score.** **Left:** Hierarchical prediction parsing. Given a predicted paragraph or sentence, it is first divided into a set of sentences (light blue) and action phrases (dark blue). Each action phrase is further processed to obtain its verb lemma (light orange). **Middle:** Scoring based on semantic similarity. Sentence embeddings for the GT action sentence and the set of predicted sentences and phrases are extracted, with the maximum cosine similarity defined as the similarity score. **Right:** Scoring based on verb matching. The predicted verb lemma is compared with the GT lemma and multiplied by the corresponding semantic-based similarity. The highest resultant score is defined as the verb score.

(i.e., within an $n \times n$ neighbourhood).

Finally, we construct an adjacency matrix based on the predicted scores $s^{i,j}$ for all possible pairs of Shot i and Shot j . By thresholding the adjacency matrix (threshold ϵ) and identifying the largest connected components, we cluster the set of shots into multiple threads.

Thread structure injection. Once the thread information is obtained, we inject it into the text prompt $\mathcal{P}_{\text{VideoLLM}}$ for the Stage I VideoLLM. In practice, we conduct this information injection only to clips that exhibit thread structures (i.e. $N_{\text{thread}} < N_{\text{shot}}$). For each given thread [Shot $i, \dots, \text{Shot } j$], we formulate the prompt as: “[Shot $i, \dots, \text{Shot } j$] share the same camera setup”. This statement implies that the cameras in these shots maintain consistent angles and scales.

Rather than simply providing this information, we further engage the VideoLLM by asking it to explain why the given thread structure is correct. This effectively corresponds to a Chain-of-Thought (CoT) process, enhancing its understanding of these repetitive thread structures.

3.3. Leveraging shot scale information

In movies and TV series, shot scales are often carefully designed during filming or post-editing to implicitly convey information to the audience. Our objective is to use the shot scale to choose what should be included in the Stage I prompts.

Shot scale classification. We first build a shot scale classification network by fine-tuning a pre-trained DINOv2 [61] model. For all *current* shots, we classify their shot scales $\{S_i\}$ into one of *five* classes, represented by values 0–4, as shown in Fig. 4 (top). We then compute their average to obtain the effective shot scale S_{eff} .

Stage I factor composition. We then leverage the predicted shot scale to determine the factors to include in Stage I instructions. We first consider three fixed factors that form the basis of ADs, namely characters, actions, and interactions. The additional factor can be determined through applying a set of thresholds to S_{eff} , as detailed in Fig. 4 (bottom). For example, for close-up shots ($S_{\text{eff}} \leq 1.5$), we would ask the VideoLLM to additionally describe the “facial expression”, whereas for long shots

($S_{\text{eff}} \geq 3.5$), a description on the “environment” will be included.

4. Action score

In this section, we introduce a new metric, termed “action score”, which focuses on whether a specific ground truth (GT) action is captured within a predicted Stage I description paragraph or a Stage II AD output. For instance, for the GT action “*He dials the phone*”, we want the metric to measure the action of “*dial the phone*”, but not be sensitive to character names and other predicted content. Therefore, this metric is designed to possess two key properties: (i) it is character-free, meaning that the presence of character names has minimal impact on the evaluation, and (ii) it is recall-oriented, without penalising additional action information in the prediction.

Preprocessing of GT actions. For each GT AD, we extract the character-free GT actions by (i) replacing character names with pronouns, and (ii) splitting the AD into subsentences, each containing one action (verb). For example, given a GT AD “*Chandler dials the phone, then hurriedly hangs up.*”, the extracted GT actions are “*He dials the phone.*” and “*He hurriedly hangs up.*”

Hierarchical parsing of predictions. To process the predicted paragraph during evaluation, we first decompose it into individual sentences (light blue), as shown in the left column of Fig. 5. For each predicted sentence, we then perform rule-based dependency parsing to obtain action phrases (deep blue). Then, all extracted sentences and action phrases are collected to form a prediction set. Additionally, for each action phrase, we extract the corresponding verb lemma (i.e. the root form of verbs).

Action score computation. The action score features a combination of semantic-based and verb-matching-based components, as illustrated in the middle and right columns of Fig. 5, respectively.

Semantic-based evaluation. For each GT action sentence, we assess its semantic similarity with the set of predicted sentences and phrases. Specifically, we employ a general text embedding (GTE) model to compute *sentence-level* embeddings for the GT action sentence e_{GT} and each element in the predicted set $\{e_{\text{pred};i}\}$. By computing cosine similarities and taking the maximum value, the *similarity score* is defined

as $s_{\text{sim}} = \max_i (s_{\text{sim};i}) = \max_i [(e_{\text{GT}} \cdot e_{\text{pred};i}) / (\|e_{\text{GT}}\| \|e_{\text{pred};i}\|)]$.

Verb-matching-based evaluation. In addition to semantic matching, we credit predictions that contain the same verbs as those in GT actions. Practically, we compare the predicted verb lemmas with GT verb lemma and compute binary matching scores $\{m_j\}$. These matching scores are further weighted by the corresponding similarity scores $\{s_{\text{sim};j}\}$ through element-wise multiplication. Finally, we take the maximum of the regulated scores to compute the *verb score*, i.e. $s_{\text{verb}} = \max_j (s_{\text{sim};j} \circ m_j)$.

To obtain the final action score, we combine the scores from both sources using a weighted average $s_{\text{action}} = \alpha_{\text{sim}} \circ s_{\text{sim}} + \alpha_{\text{verb}} \circ s_{\text{verb}}$, with α_{sim} and α_{verb} denoting the weighted factors.

5. Experiments

This section begins with datasets and metrics for AD generation in Sec. 5.1, followed by implementation details in Sec. 5.2. Results on film grammar predictions are presented in Secs. 5.3 and 5.4, while Sec. 5.5 analyses human alignment with action scores. AD generation results are detailed in Secs. 5.6 to 5.8.

5.1. AD generation datasets and metrics

We evaluate our framework on AD generation datasets for both movies (CMD-AD [28], MAD-Eval [26]) and TV series (TV-AD [92]). In more detail, CMD-AD is constructed by aligning ground truth ADs with the Condensed Movie Dataset (CMD) [9], comprising 101k ADs (94k for training and 7k for testing) from 1.4k movies. MAD-Eval consists of 6.5k ADs sampled from 10 movies within LSMDC [72]. On the other hand, TV-AD features 34k AD annotations from 13 TV series, with its test set sourced from TVQA [43], consisting of 3k ADs.

For AD evaluation, we follow the prior works [22, 28] to assess general prediction quality using CIDEr [81], Recall@k/N [27], and LLM-AD-Eval [28]. We also consider character recognition accuracy using CRITIC [28] and character-free action evaluation using action scores (described in Sec. 4).

5.2. Implementation details

In this section, we provide key implementation details for AD generation and action score computation. For additional information on film grammar predictions and other details, please refer to Appendix A.

Shot detection. To segment the video clip into shots, we use PySceneDetect [3] with the ‘‘Adaptive Detection’’ method, which compares the ratio of pixel changes with the neighbouring frames. On average, each shot spans 3.5s, 3.3s, 3.8s in CMD-AD, TV-AD, and MAD-Eval, respectively.

AD generation setting. During dynamic frame sampling, we select a total of 32 frames, with 16 frames uniformly sampled within and outside the AD interval, respectively. For character recognition, we adopt the same visual-textual prompting method proposed by [92], which applies coloured circles around faces for visual character indication. For simplicity, the visualisations in this paper do not display these circle labels.

Feature	Frame setup	Precision	Recall	AP	WCP
CLIP-L14 CLS	Side	0.691	0.635	0.705	0.922
DINOv2-L14 CLS	Side	0.759	0.683	0.788	0.933
DINOv2-g14 CLS	Side	0.761	0.675	0.786	0.936
DINOv2-g14 spatial	Mid	0.808	0.717	0.822	0.953
DINOv2-g14 spatial	All	0.870	0.795	0.896	0.964
DINOv2-g14 spatial	Side	0.878	0.799	0.902	0.965

Table 1. **Thread structure prediction on Thread-Safe.** ‘‘Side’’ refers to comparisons between the temporally nearest frames in two shots; ‘‘Mid’’ refers to comparisons between the middle frames of each shot; ‘‘All’’ refers to comparisons between all frame pairs from the two shots.

Regarding the base models, we use Qwen2-VL-7B [85] as the VideoLLM in Stage I and LLaMA3-8B [53] as the LLM in Stage II. This setup is used as the *default* unless stated otherwise. Additionally, we explore the framework with the proprietary GPT-4o [60] model for both stages.

Action score evaluation setting. For action score computation, we set the weight factors as $\alpha_{\text{sim}} = 0.8$ and $\alpha_{\text{verb}} = 0.2$. When aggregating the action score results, we first average over multiple GT actions within each GT AD, and then perform global averaging across all AD samples. Moreover, in practice, we find that most action scores are clustered within the range of 0.25 – 0.75. To improve clarity, we apply further rescaling $f(x) = (x - 0.25) \times 2$ as post-processing. For evaluations in this paper, unless otherwise specified, we use the action score to assess the Stage II AD outputs.

5.3. Thread structure prediction

We evaluate thread structure prediction on Thread-Safe [30], which consists of approximately 4.7k video clips collected from 15 TV series. Each video clip contains a multi-shot scene with corresponding thread clusters manually annotated.

For evaluation, we first construct an adjacency matrix from the GT clusters and extract binary GT labels $\hat{s}_{i,j}$ from the off-diagonal entries, where each label indicates whether a given pair of shots belong to the same cluster. We then compute the Average Precision (AP) between the predicted relationships $\{s_{i,j}\}$ and the ground truth $\{\hat{s}_{i,j}\}$, as well as report the precision and recall values. Additionally, we directly compare the predicted clusters with ground truth clusters by reporting the weighted clustering purity (WCP) [77].

Tab. 1 verifies the choice of DINOv2-g14 [61] as the feature extractor for frame pair comparison. Compared to abstract CLS tokens, dense spatial matching achieves higher AP in frame pair relationship prediction and higher WCP in thread clustering. Additionally, we observe that using the temporally closest frames from two shots (‘‘Side’’) leads to a noticeable performance improvement. This can be attributed to the continuous story flow across shots within the same thread.

5.4. Shot scale classification

Following prior work on shot scale classification [45, 51], we use the MovieShots [69] dataset, which consists of 46k shots

Metric	Input	Accuracy	Macro-F1
ViViT [7]	RGB	0.747	0.751
SGNet [69]	RGB + Flow	0.875	–
Lu et al. [51]	RGB + mask	0.892	–
Li et al. [45]	RGB	0.895	0.897
Ours	RGB	0.897	0.899

Table 2. **Shot scale classification on MovieShots.**

Metric	Paragraph		Sentence	
	Pearson	Spearman	Pearson	Spearman
CIDEr [81]	0.205	0.264	0.412	0.528
ROUGE-L [48]	0.305	0.280	0.526	0.512
METEOR [10]	0.462	0.406	0.602	0.641
BLEU-1 [62]	0.265	0.264	0.477	0.481
SPICE [5]	0.022	0.048	0.031	0.012
BERTScore [103]	0.377	0.393	0.508	0.507
LLM-based (LLaMA3.1-70B [53])	0.569	0.491	0.779	0.798
LLM-based (GPT-4o [60])	0.742	0.678	0.797	0.807
Action Score (w/o verb matching)	0.735	0.728	0.765	0.790
Action Score (w verb matching)	0.749	0.729	0.806	0.820

Table 3. **Comparison of action score with other metrics.** The listed metrics measure the similarity between predicted paragraphs/sentences and ground truth actions. The reported values indicate the correlation (i.e. alignment) between these metrics and human-annotated scores.

(train:val:test = 7:1:2) collected from over 7k movie trailers. We follow its definition of shot scales, categorising shots into five classes ranging from extreme close-up to long shots, as illustrated in Fig. 4 (top). To evaluate the model performance, we report classification accuracy and Macro-F1 [45] scores on the MovieShots test set.

Since previous state-of-the-art methods on shot scale classification are not open-sourced, we develop a new network by fine-tuning DINOv2 [61], achieving superior performance over prior approaches that rely on additional optical flow or SAM-based mask inputs, as demonstrated in Tab. 2.

5.5. Human alignment with action scores

Action scores aim to evaluate whether a GT action is captured within a predicted description, making them recall-oriented. Such descriptions can be in the form of paragraphs (Stage I descriptions) or single sentences (Stage II ADs). To assess whether action scores align with human judgments, we create a dataset containing pairs of predicted descriptions and GT actions. For each GT action, human annotators *manually annotate* the quality of predictions into $\{0,1,2,3\}$ based on the relevance towards GT action ranging from “unrelated” (0) to “exact matching” (3).

Comparison with other metrics. Next, we use the human-annotated scores as a reference to compare different metrics in terms of human agreement (measured by correlations), as reported in Tab. 3. Additionally, we consider an LLM-based metric to predict scores, following the same scoring criteria as human annotations. In general, the action score achieves the best correlations with human annotations. Note, it is also more efficient than LLM-based metrics, with 0.15s compared to 6s per prediction evaluation. For more details regarding this

Exp.	Temporal context	Frame sampling	Shot label	CMD-AD			TV-AD		
				CIDEr	CRITIC	Action	CIDEr	CRITIC	Action
A	–	–	–	22.4	45.7	27.0	26.0	42.2	22.3
B	1 shot	Dyn.	Shot num.	24.7	46.8	27.8	28.9	41.6	23.1
C	2 shots	Dyn.	–	24.5	46.6	27.4	27.8	42.3	22.3
D	2 shots	Dyn.	Frame box	24.8	46.4	27.4	25.0	41.8	22.8
E	2 shots	Uni.	Shot num.	24.1	47.5	26.2	26.5	42.2	22.0
F	2 shots	Dyn.	Shot num.	25.1	47.5	27.8	28.9	42.1	23.0

Table 4. **Leveraging shot-based temporal context.** Key changes relative to the default setting (Exp. F) are highlighted in light blue. “Temporal context” indicates the number of context (past & future) shots. “Shot num.” refers to overlaying the shot number at the top-left of each current shot frame, while “frame box” represents highlighting the boundary of each current shot frame with a red box.

Thread structure	CMD-AD subset			TV-AD subset		
	CIDEr	CRITIC	Action	CIDEr	CRITIC	Action
✗	29.9	47.5	27.4	28.8	42.0	22.6
✓	30.7 $\uparrow 0.8$	48.9 $\uparrow 1.4$	27.7 $\uparrow 0.3$	30.7 $\uparrow 1.9$	42.7 $\uparrow 0.7$	22.9 $\uparrow 0.3$

Table 5. **Thread structure injection.** Thread structure information is injected only into subsets predicted to exhibit thread structures ($\sim 30\%$ in CMD-AD and $\sim 60\%$ in TV-AD).

Stage I factors	CMD-AD			TV-AD		
	CIDEr	CRITIC	Action	CIDEr	CRITIC	Action
Base	25.4	47.4	27.7	29.2	42.0	23.0
Base + Face (AutoAD-Zero)	25.2	46.8	27.8	30.0	42.3	22.5
Base + Obj.	26.1	45.8	27.9	30.4	42.9	22.8
Base + Env.	25.2	46.7	27.4	29.8	40.7	22.2
Base + Face + Obj. + Env.	26.0	47.2	27.4	30.1	42.7	22.9
Scale-dependent (Ours)	26.3	47.8	28.4	31.1	42.2	23.9

Table 6. **Factors included in Stage I description.** Base: character + action + interaction; Face: facial expression; Env.: environment; Obj.: object. “Scale-dependent” refers to our approach, which leverages shot scale predictions to determine the relevant factors for each clip.

human agreement study, please refer to Appendix D.

5.6. AD generation – evaluation of components

Shot-based temporal context. We investigate different setups for leveraging temporal context in AD generation, as shown in Tab. 4, leading to the following observations: (i) Expanding the temporal context range noticeably boosts the performance, with gains saturating around “2 shots” (Exp. A, B, and F); (ii) “Shot number referral” is the most effective strategy for outlining the current shot. (Exp. C, D, and F); (iii) Dynamically sampling the current shots at a higher frame rate boosts AD generation (Exp. E and F). The latter two improvements can be attributed to a more efficient focus on the visual content around the targeted AD interval.

Thread structure injection. After extending the context information with neighbouring shots, we further enhance the VideoLLM’s understanding by incorporating thread structures. Note that this guidance is applied only to video sequences exhibiting thread structures. As shown in Tab. 5, this additional information improves AD generation performance across both datasets.

Method	VLM	LLM	Training -free	Propriet. model	CMD-AD					TV-AD						
					CIDEr	CRITIC	Action	R@1/5	LLM-AD-Eval	CIDEr	CRITIC	Action	R@1/5	LLM-AD-Eval		
AutoAD-II [27]	CLIP-B32	GPT-2	✗	✗	13.5	8.2	—	26.1	2.08	—	—	—	—	—		
AutoAD-III [28]	EVA-CLIP	LLaMA2-7B	✗	✗	25.0	32.7	31.5	31.2	2.89	2.01	26.1	28.8	26.4	30.1	2.78	1.99
DistinctAD [22]	CLIP _{AD} -B16	LLaMA3-8B	✗	✗	22.7	—	—	33.0	2.88	2.03	27.4	—	—	32.1	2.89	2.00
Video-LLaMA [102]	Video-LLaMA-7B	—	✓	✗	4.8	0.0	—	22.0	1.89	—	—	—	—	—	—	—
VideoBLIP [98]	VideoBLIP	—	✓	✗	5.2	0.0	—	23.6	1.91	—	—	—	—	—	—	—
AutoAD-Zero [92]	VideoLLaMA2-7B	LLaMA3-8B	✓	✗	17.7	43.7	25.5	26.9	2.83	1.96	22.6	39.4	21.7	27.4	2.94	2.00
AutoAD-Zero [92]	Qwen2-VL-7B	LLaMA3-8B	✓	✗	21.9	44.3	26.9	30.8	3.00	2.20	26.4	41.6	22.1	30.4	3.05	2.27
Ours	Qwen2-VL-7B	LLaMA3-8B	✓	✗	26.3	47.8	28.4	33.0	3.15	2.42	31.1	42.2	23.9	33.1	3.09	2.35
AutoAD-Zero [92]	GPT-4o	GPT-4o	✓	✓	22.4	45.1	30.7	32.9	3.08	2.49	30.9	44.4	26.8	34.7	3.12	2.57
Ours	GPT-4o	GPT-4o	✓	✓	26.1	49.1	32.5	36.5	3.17	2.66	34.2	46.5	27.4	36.6	3.12	2.59

Table 7. **Quantitative comparison on CMD-AD and TV-AD.** For training-free methods, “VLM” and “LLM” refer to the models used in separate stages, while for fine-tuned models, they denote the pre-trained components within an end-to-end model.

Method	VLM	LLM	Training -free	Propriet. model	MAD-Eval					
					CIDEr	R@5/16	Rouge-L	SPICE	METEOR	BLEU-1
ClipCap [55]	CLIP-B32	GPT-2	✗	✗	4.4	36.5	8.5	1.1	—	—
CapDec [59]	—	—	✗	✗	6.7	—	8.2	1.4	—	—
AutoAD-I [26]	CLIP-B32	GPT-2	✗	✗	13.4	42.1	11.9	4.4	—	—
AutoAD-II [27]	CLIP-B32	GPT-2	✗	✗	19.5	51.3	13.4	—	—	—
AutoAD-III [28]	EVA-CLIP	LLaMA2-7B	✗	✗	24.0	52.8	13.9	6.1	5.5	13.1
MovieSeq [50]	CLIP-B16	LLaMA2-7B	✗	✗	24.4	51.6	15.5	7.0	—	—
DistinctAD [22]	CLIP _{AD} -B16	LLaMA3-8B	✗	✗	27.3	56.0	17.6	8.3	—	—
UniAD [82]	CLIP-L14	LLaMA-8B	✗	✗	28.2	54.9	17.2	—	—	—
Video-LLaMA [102]	Video-LLaMA-7B	—	✓	✗	4.8	33.8	—	—	—	—
Video-BLIP [98]	Video-BLIP	—	✓	✗	5.0	35.2	—	—	—	—
AutoAD-Zero [92]	VideoLLaMA2-7B	LLaMA3-8B	✓	✗	22.4	47.0	14.4	7.3	6.6	15.1
AutoAD-Zero [92]	Qwen2-VL-7B	LLaMA3-8B	✓	✗	23.6	51.3	14.6	7.8	6.6	13.6
Ours	Qwen2-VL-7B	LLaMA3-8B	✓	✗	25.0	50.6	14.7	7.8	7.2	16.2
VLog [4]	BLIP-2 + GRIT	GPT-4	✓	✓	1.3	42.3	7.5	2.1	—	—
MM-Vid [49]	GPT-4V	—	✓	✓	6.1	46.1	9.8	3.8	—	—
MM-Narrator [101]	Azure API + CLIP-L14	GPT-4V	✓	✓	9.8	—	12.8	—	7.1	10.9
MM-Narrator [101]	Azure API + CLIP-L14	GPT-4	✓	✓	13.9	49.0	13.4	5.2	6.7	12.8
LLM-AD [19]	GPT-4V	—	✓	✓	20.5	—	13.5	—	—	—
AutoAD-Zero [92]	GPT-4o	GPT-4o	✓	✓	25.4	54.3	14.3	8.1	6.7	13.7
Ours	GPT-4o	GPT-4o	✓	✓	26.9	56.4	15.0	8.5	7.4	15.9

Table 8. **Quantitative comparison on MAD-Eval.** For training-free methods, “VLM” and “LLM” refer to the models used in separate stages, while for fine-tuned models, they denote the pre-trained components within an end-to-end model.

Scale-dependent Stage I factors. Tab. 6 explores the impact of different Stage I factors on final AD performance. Using shot scales as guidance for Stage I factor formulation (i.e. scale-dependent) not only outperforms configurations with single fixed factors but also surpasses the case where all factors are included in Stage I. This could be attributed to that the scale-dependent description contains more relevant and less redundant information, enabling more efficient AD extraction in Stage II.

5.7. AD generation – comparison with SotA

Tab. 7 provides a comprehensive summary of AD generation performance on CMD-AD and TV-AD, comparing across training-free methods with and without proprietary models, as well as models fine-tuned on human-annotated ADs. Notably, with the same base model setup (Qwen2-VL-7B + LLaMA3-8B), our training-free framework significantly outperforms AutoAD-Zero, primarily due to the usage of temporal context and film grammar information. By incorporating the more powerful GPT-4o models, our performance scales up further,

surpassing even existing fine-tuned models.

Regarding the MAD-Eval results, as shown in Tab. 8, our framework achieves state-of-the-art performance among all training-free approaches, also demonstrating competitiveness with fine-tuned models.

Qualitative visualisations. Fig. 6 presents several qualitative examples, where the top two cases illustrate how temporal context information aids in identifying key objects.

In the bottom-left example, prior methods fail to associate characters, leading to the omission of the man (*Alonzo*). In contrast, our method recognises the thread structure (i.e. *[Shot 0, Shot 2], [Shot 1, Shot 3]*), which guides the correct prediction of the man’s gaze direction towards the lying woman.

The bottom-right example highlights the effectiveness of scale-dependent Stage I factor formulation. AutoAD-Zero, designed to query characters, actions, interactions, and facial expressions, sometimes overlooks environmental details. Our method, in contrast, correctly identifies the shot as a long shot



Figure 6. **Qualitative visualisations.** Current shots (corresponding to AD intervals) are outlined by red boxes for illustration purposes only. For simplicity, not all context shots are shown. Training-free methods adopt Qwen2-VL + LLaMA3. Examples are taken from *The Big Bang Theory* (S2E14) (top left), *Inferno* (2016) (top right), *The Asphalt Jungle* (1950) (bottom left), and *Signs* (2002) (bottom right). The top-left example demonstrates the benefits of shot-based temporal context, where the objects (i.e. bills) in Penny’s hands are not clearly visible within the AD interval (Shot 1), leading to ambiguous or incorrect predictions by AutoAD-Zero. In contrast, our method successfully identifies the objects from the context shot (Shot 2). The top-right example describing the action of *Harry Sims* similarly verifies the effectiveness of incorporating context shots.

Method	Candidate sampling	CMD-AD			TV-AD		
		CIDEr	CRITIC	Action	CIDEr	CRITIC	Action
Ours	Single AD (Ref.)	26.3	47.8	28.4	31.1	42.2	23.9
Ours	Indep. output ($p=0.90; \tau_p=0.6$)	33.3	49.6	32.5	41.2	44.3	28.6
Ours	Indep. output ($p=0.95; \tau_p=1.5$)	37.0	50.3	35.1	45.5	46.4	31.5
AutoAD-Zero [92]	Joint output	31.6	46.4	33.8	43.2	46.8	30.1
Ours	Joint output	38.4	49.2	35.7	51.3	47.3	31.8

Table 9. **Assisted AD generation results.** All methods adopt Qwen2-VL + LLaMA3-8B as base models. The first row provides single AD generation results as references (labelled in gray), the rest rows report the performance of one selected AD out of five candidates. “Indep. output” denotes five random independent Stage II runs, with p as the hyperparameter for top-p (nucleus) sampling and τ_p as the sampling temperature. “Joint output” generates five ADs simultaneously in a single run.

and instructs the VideoLLM to incorporate environmental context, resulting in more accurate scene descriptions. For more visualisations, please refer to Appendix E.

5.8. Assisted AD generation

The subjective nature of AD sets a practical limit on metric scores, lower than the theoretical maximum, because human annotators often provide different but valid descriptions. Therefore, beyond enforcing generating a *single* AD sentence, we also consider employing our framework as an *assistant* to produce *multiple* candidate AD sentences.

To standardise such a protocol, we consider five candidate ADs generated by an *assistant* and employ an *expert* to select the best one. To effectively benchmark performance against existing GT ADs, we define the “*expert*” as an automatic selection mechanism that chooses the candidate with the highest average CIDEr and action score.

To develop an AD generation assistant, we fix the Stage I dense descriptions and explore generating multiple candidates in Stage II. This can be achieved by either running Stage II independently five times (termed the “independent output” setup) or generating five AD outputs simultaneously within a single run (termed the “joint output” setup). As observed in Tab. 9, the assistant-based setup significantly improves upon the single AD performance, highlighting the potential of training-free methods in effectively capturing the desired content for AD generation. Within the “independent output” setup, increasing the randomness of sampling (i.e. higher p and τ_p) enhances the quality of the selected AD, owing to greater candidate diversity. Meanwhile, the “joint output” setup achieves superior performance, which could be attributed to reduced information redundancy across the simultaneously generated ADs. For additional visualisations, discussions, and detailed text prompts, please refer to Appendix E.

6. Discussion – summary and limitations

We have demonstrated the benefit of shot-based context and film grammar awareness in AD generation – our training-free two-stage framework achieves state-of-the-art performance among all training-free counterparts, even surpassing fine-tuned models on multiple benchmarks.

The current framework has two main limitations: (i) the performance depends on the base VideoLLM, which may occasionally hallucinate details inconsistent with the visual content; and (ii) story-level context is not incorporated into the AD generation process. These limitations could potentially be addressed in future work by improving visual grounding and extending the visual and textual context to include the plot.

Acknowledgements. This research is supported by the UK EPSRC Programme Grant Visual AI (EP/T028572/1), a Royal Society Research Professorship RSRP\R\241003, a Clarendon Scholarship, and ANR-21-CE23-0003-01 CorVis.

References

- [1] Guidelines for audio describers. <https://adp.acb.org/guidelines.html>. 2
- [2] Audiovault. <https://audiovault.net/>. 2
- [3] Pyscenedetect. <https://www.scenedetect.com/>. 6
- [4] Vlog: Video as a long document. <https://github.com/showlab/VLog>, 2023. 8
- [5] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016. 3, 7, 15
- [6] Dawit Mureja Argaw, Fabian Caba Heilbron, Joon-Young Lee, Markus Woodson, and In So Kweon. The anatomy of video editing: A dataset and benchmark suite for ai-assisted video editing. In *ECCV*, 2022. 2
- [7] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021. 7
- [8] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 15
- [9] Max Bain, Arsha Nagrani, Andrew Brown, and Andrew Zisserman. Condensed movies: Story based retrieval with contextual embeddings. In *ACCV*, 2020. 6
- [10] Satyanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 2005. 2, 3, 7, 15
- [11] Andrew Brown, Vicky Kalogeiton, and Andrew Zisserman. Face, body, voice: Video person-clustering with multiple modalities. In *ICCV 2021 Workshop on AI for Creative Video Editing and Understanding*, 2021. 3
- [12] Boris Chen, Amir Ziai, Rebecca S. Tucker, and Yuchen Xie. Match cutting: Finding cuts with smooth visual transitions. In *WACV*, 2023. 2
- [13] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, pages 190–200, 2011. 3
- [14] Shaoxiang Chen and Yu-Gang Jiang. Towards bridging event captioner and sentence localizer for weakly supervised dense event captioning. In *CVPR*, 2021. 3
- [15] Shimin Chen, Xiaohan Lan, Yitian Yuan, Zequn Jie, and Lin Ma. Timemarker: A versatile video-llm for long and short video understanding with superior temporal localization ability. *arXiv preprint arXiv:2411.18211*, 2024. 3
- [16] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, and Sergey Tulyakov. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *CVPR*, 2024. 3
- [17] Zeyu Chen, Yana Zhang, Suya Zhang, and Cheng Yang. Study on location bias of cnn for shot scale classification. *Multimedia Tools and Applications*, 2022. 2
- [18] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024. 15
- [19] Peng Chu, Jiang Wang, and Andre Abrantes. LLM-AD: Large language model based audio description system. *arXiv preprint arXiv:2405.00983*, 2024. 1, 2, 8
- [20] Andong Deng, Zhongpai Gao, Anwesa Choudhuri, Benjamin Planche, Meng Zheng, Bin Wang, Terrence Chen, Chen Chen, and Ziyang Wu. Seq2time: Sequential knowledge transfer for video llm temporal grounding. *arXiv preprint arXiv:2411.16932*, 2024. 3
- [21] Chaorui Deng, Shizhe Chen, Da Chen, Yuan He, and Qi Wu. Sketch, ground, and refine: Top-down dense video captioning. In *CVPR*, 2021. 3
- [22] Bo Fang, Wenhao Wu, Qiangqiang Wu, Yuxin Song, and Antoni B. Chan. DistinctAD: Distinctive audio description generation in contexts. *arXiv preprint arXiv:2411.18180*, 2024. 1, 2, 6, 8, 14
- [23] Dennis Fedorishin, Lie Lu, Srirangaraj Setlur, and Venu Govindaraju. Audio match cutting: Finding and creating matching audio transitions in movies and videos. In *ICASSP*, 2024. 2
- [24] Yongxin Guo, Jingyu Liu, Mingda Li, Xiaoying Tang, Xi Chen, and Bo Zhao. Vtg-llm: Integrating timestamp knowledge into video llms for enhanced video temporal grounding. *arXiv preprint arXiv:2405.13382*, 2024. 3
- [25] Yongxin Guo, Jingyu Liu, Mingda Li, Xiaoying Tang, Qingbin Liu, and Xi Chen. Trace: Temporal grounding video llm via causal event modeling. *arXiv preprint arXiv:2410.05643*, 2024. 3
- [26] Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. Autoad: Movie description in context. In *CVPR*, 2023. 1, 2, 3, 6, 8, 14
- [27] Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. AutoAD II: The sequel – who, when, and what in movie audio description. In *ICCV*, 2023. 6, 8
- [28] Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. AutoAD III: The prequel – back to the pixels. In *CVPR*, 2024. 1, 2, 3, 6, 8, 14, 16
- [29] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with temporal language. In *EMNLP*, 2018. 3
- [30] Minh Hoai and Andrew Zisserman. Thread-safe: Towards recognizing human actions across shot boundaries. In *ACCV*, 2014. 3, 6
- [31] Wenyi Hong, Weihang Wang, Ming Ding, Wenmeng Yu, Qingsong Lv, Yan Wang, Yean Cheng, Shiyu Huang, Junhui Ji, Zhao Xue, et al. Cogvlm2: Visual language models for image and video understanding. *arXiv preprint arXiv:2408.16500*, 2024. 3
- [32] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments. In *CVPR*, 2024. 3

- [33] De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, and Jan Kautz. Lita: Language instructed temporal-localization assistant. In *ECCV*, 2024. 3
- [34] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiase Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *ECCV*, 2020. 2
- [35] Vladimir Iashin and Esa Rahtu. A better use of audio-visual cues: Dense video captioning with bi-modal transformer. In *BMVC*, 2020. 3
- [36] Vladimir Iashin and Esa Rahtu. Multi-modal dense video captioning. In *CVPR Workshops on Multimodal Learning*, 2020. 3
- [37] Md Mohaiminul Islam, Ngan Ho, Xitong Yang, Tushar Nagarajan, Lorenzo Torresani, and Gedas Bertasius. Video recap: Recursive captioning of hour-long videos. In *CVPR*, 2024. 3
- [38] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. In *NeurIPS*, 2020. 4
- [39] Xuekun Jiang, Libiao Jin, Anyi Rao, Linning Xu, and Dahua Lin. Jointly learning the attributes and composition of shots for boundary detection in videos. *Trans. Multi.*, 2022. 2
- [40] Steve D. Katz. *Film Directing Shot by Shot: Visualizing from Concept to Screen*. Michael Wiese Productions, ISBN: 1615932976, 2019. 1
- [41] Minkuk Kim, Hyeon Bae Kim, Jinyoung Moon, Jinwoo Choi, and Seong Tae Kim. Do you remember? dense video captioning with cross-modal memory retrieval. In *CVPR*, 2024. 3
- [42] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017. 3
- [43] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. TVQA: Localized, compositional video question answering. In *EMNLP*, 2018. 6
- [44] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *ECCV*, 2020. 3
- [45] Yuzhi Li, Feng Tian, Haojun Xu, and Tianfeng Lu. Toward unified and quantitative cinematic shot attribute analysis. *Electronics*, 2023. 2, 6, 7
- [46] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023. 14
- [47] Zhaowei Li, Qi Xu, Dong Zhang, Hang Song, Yiqing Cai, Qi Qi, Ran Zhou, Junting Pan, Zefeng Li, Vu Tu, et al. Groundinggpt: Language enhanced multi-modal grounding model. In *ACL*, 2024. 3
- [48] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, 2004. 2, 3, 7, 15
- [49] Kevin Lin, Faisal Ahmed, Linjie Li, Chung-Ching Lin, Ehsan Azarnasab, Zhengyuan Yang, Jianfeng Wang, Lin Liang, Zicheng Liu, Yumao Lu, Ce Liu, and Lijuan Wang. MM-Vid: Advancing video understanding with GPT-4V(ision). *arXiv preprint arXiv:2310.19773*, 2023. 8
- [50] Kevin Qinghong Lin, Pengchuan Zhang, Difei Gao, Xide Xia, Joya Chen, Ziteng Gao, Jinheng Xie, Xuhong Xiao, and Mike Zheng Shou. Learning video context as interleaved multimodal sequences. In *ECCV*, 2024. 1, 2, 3, 8
- [51] Fengtian Lu, Yuzhi Li, and Feng Tian. Exploring challenge and explainable shot type classification using sam-guided approaches. *Signal, Image and Video Processing*, 2024. 2, 6, 7
- [52] Manuel Marin-Jimenez, Vicky Kalogeiton, Pablo Medina-Suarez, and Andrew Zisserman. Laeo-net++: revisiting people looking at each other in videos. *IEEE TPAMI*, pages 1–1, 2020. 2
- [53] Meta. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 6, 7, 14, 15
- [54] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019. 3
- [55] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. 8
- [56] James Monaco. *How to Read a Film: Movies, Media, and Beyond*. OUP, ISBN: 0571168973, 2009. 1
- [57] Jonghwan Mun, Linjie Yang, Zhou Ren, Ning Xu, and Bohyung Han. Streamlined dense video captioning. In *CVPR*, 2019. 3
- [58] Walter Murch. *In the Blink of an Eye: A Perspective on Film Editing*. Silman-James Press, ISBN: 1879505622, 2021. 1
- [59] David Nukrai, Ron Mokady, and Amir Globerson. Text-only training for image captioning using noise-injected clip. In *ENNLP*, 2022. 8
- [60] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2024. 6, 7, 14, 15
- [61] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 4, 5, 6, 7, 14
- [62] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 2, 3, 7, 15
- [63] Alejandro Pardo, Fabian Caba Heilbron, Juan León Alcázar, Ali Thabet, and Bernard Ghanem. Moviecuts: A new dataset and benchmark for cut type recognition. In *ECCV*, 2022. 2
- [64] Alejandro Pardo, Fabio Pizzati, Tong Zhang, Alexander Pondaven, Philip Torr, Juan Camilo Perez, and Bernard Ghanem. Matchdiffusion: Training-free generation of match-cuts. *arXiv preprint arXiv:2411.18677*, 2024. 2
- [65] Toby Perrett, Tengda Han, Dima Damen, and Andrew Zisserman. It's just another day: Unique video captioning by discriminative prompting. In *ACCV*, 2024. 3
- [66] Long Qian, Juncheng Li, Yu Wu, Yaobo Ye, Hao Fei, Tat-Seng Chua, Yueting Zhuang, and Siliang Tang. Momentor: Advancing video large language model with fine-grained temporal reasoning. In *ICML*, 2024. 3
- [67] Mengxue Qu, Xiaodong Chen, Wu Liu, Alicia Li, and Yao Zhao. Chatvtg: Video temporal grounding via chat with video dialogue large language models. In *CVPRW*, 2024. 3

- [68] Tanzila Rahman, Bicheng Xu, and Leonid Sigal. Watch, listen and tell: Multi-modal weakly supervised dense event captioning. In *ICCV*, 2019. 3
- [69] Anyi Rao, Jiase Wang, Linning Xu, Xuekun Jiang, Qingqiu Huang, Bolei Zhou, and Dahua Lin. A unified framework for shot type classification based on subject centric lens. In *ECCV*, 2020. 2, 6, 7
- [70] Anyi Rao, Xuekun Jiang, Sichen Wang, Yuwei Guo, Zihao Liu, Bo Dai, Long Pang, Xiaoyu Wu, Dahua Lin, and Libiao Jin. Temporal and contextual transformer for multi-camera editing of tv shows. In *Proceedings of the CVEU Workshop at ICCV*, 2022. 2
- [71] Stephen Robertson. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of Documentation*, 2004. 3, 14
- [72] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. In *CVPR*, 2015. 3, 6
- [73] Mattia Savardi, András Bálint Kovács, Alberto Signoroni, and Sergio Benini. Recognition of camera angle and camera level in movies from single frames. In *Proceedings of the ACM International Conference on Interactive Media Experiences Workshops*, 2023. 2
- [74] Nina Shvetsova, Anna Kukleva, Xudong Hong, Christian Rupprecht, Bernt Schiele, and Hilde Kuehne. HowToCaption: Prompting LLMs to transform video annotations at scale. *arXiv:2310.04900*, 2023. 3
- [75] Mattia Soldan, Alejandro Pardo, Juan León Alcázar, Fabian Caba, Chen Zhao, Silvio Giancola, and Bernard Ghanem. MAD: A scalable dataset for language grounding in videos from movie audio descriptions. In *CVPR*, 2022. 2
- [76] Yunlong Tang, Junjia Guo, Hang Hua, Susan Liang, Mingqian Feng, Xinyang Li, Rui Mao, Chao Huang, Jing Bi, Zeliang Zhang, Pooyan Fazli, and Chenliang Xu. Vidcomposition: Can mllms analyze compositions in compiled videos? *arXiv preprint arXiv:2411.10979*, 2024. 2
- [77] Makarand Tapaswi, Omkar M. Parkhi, Esa Rahtu, Eric Sommerlade, Rainer Stiefelhagen, and Andrew Zisserman. Total cluster: A person agnostic clustering method for broadcast videos. In *Proceedings of the Indian Conference on Computer Vision Graphics and Image Processing*, 2014. 3, 6
- [78] Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. 15
- [79] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruiti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 14
- [80] Bartolomeo Vacchetti and Tania Cerquitelli. Cinematographic shot classification with deep ensemble learning. *Electronics*, 2022. 2
- [81] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. In *CVPR*, 2015. 2, 3, 6, 7, 14
- [82] Hanlin Wang, Zhan Tong, Kecheng Zheng, Yujun Shen, and Limin Wang. Contextual ad narration with interleaved multi-modal sequence. *arXiv preprint arXiv:2403.12922*, 2024. 1, 2, 8
- [83] Haibo Wang, Zhiyang Xu, Yu Cheng, Shizhe Diao, Yufan Zhou, Yixin Cao, Qifan Wang, Weifeng Ge, and Lifu Huang. Grounded-videollm: Sharpening fine-grained temporal grounding in video large language models. *arXiv preprint arXiv:2410.03290*, 2024. 3
- [84] Jingwen Wang, Wenhao Jiang, Lin Ma, Wei Liu, and Yong Xu. Bidirectional attentive fusion with context gating for dense video captioning. In *CVPR*, 2018. 3
- [85] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 6, 15
- [86] Teng Wang, Huicheng Zheng, Mingjing Yu, Qian Tian, and Haifeng Hu. Event-centric hierarchical representation for dense video captioning. *IEEE Transactions on Circuits and Systems for Video Technology*, 2020. 3
- [87] Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. End-to-end dense video captioning with parallel decoding. In *ICCV*, 2021. 3
- [88] Yujia Wang, Wei Liang, Haikun Huang, Yongqi Zhang, Dingzeyu Li, and Lap-Fai Yu. Toward automatic audio description generation for accessible videos. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021. 2, 3
- [89] Yueqian Wang, Xiaojun Meng, Jianxin Liang, Yuxuan Wang, Qun Liu, and Dongyan Zhao. Hawkeye: Training video-text llms for grounding text in videos. *arXiv preprint arXiv:2403.10228*, 2024. 3
- [90] Hao Wu, Huabin Liu, Yu Qiao, and Xiao Sun. Dibs: Enhancing dense video captioning with unlabeled videos via pseudo boundary enrichment and online refinement. In *CVPR*, 2024. 3
- [91] Yongliang Wu, Xinting Hu, Yuyang Sun, Yizhou Zhou, Wenbo Zhu, Fengyun Rao, Bernt Schiele, and Xu Yang. Number it: Temporal grounding videos like flipping manga. In *CVPR*, 2025. 3
- [92] Junyu Xie, Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. Autoad-zero: A training-free framework for zero-shot audio description. In *ACCV*, 2024. 1, 2, 6, 8, 9
- [93] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*, 2016. 3

- [94] Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu, and Baining Guo. Advancing high-resolution video-language representation with large-scale video transcriptions. In *CVPR*, 2022. 3
- [95] Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. *arXiv preprint arXiv:2302.14115*, 2023. 3
- [96] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 15
- [97] Xiaojun Ye, Junhao Chen, Xiang Li, Haidong Xin, Chao Li, Sheng Zhou, and Jiajun Bu. MMAD: Multi-modal movie audio description. In *LREC-COLING*, 2024. 1, 2
- [98] Keunwoo Peter Yu. Videoblip, 2023. 8
- [99] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. MERLOT reserve: Multimodal neural script knowledge through vision and language and sound. In *CVPR*, 2022. 3
- [100] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, Peng Jin, Wenqi Zhang, Fan Wang, Lidong Bing, and Deli Zhao. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025. 15
- [101] Chaoyi Zhang, Kevin Lin, Zhengyuan Yang, Jianfeng Wang, Linjie Li, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. MM-Narrator: Narrating long-form videos with multimodal in-context learning. In *CVPR*, 2024. 1, 2, 3, 8
- [102] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *ENNLP*, 2023. 8
- [103] Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *ICLR*, 2020. 3, 7
- [104] Yuqi Zhang, Bin Guo, Nuo Li, Ying Zhang, Qianru Wang, and Zhiwen Yu. Representation learning of next shot selection for vlog editing. In *Proceedings of the CVEU Workshop at ICCV*, 2023. 2
- [105] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *CVPR*, 2023. 3
- [106] Minghang Zheng, Xinhao Cai, Qingchao Chen, Yuxin Peng, and Yang Liu. Training-free video temporal ground-ing using large-scale pre-trained models. *arXiv preprint arXiv:2408.16219*, 2024. 3
- [107] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 3
- [108] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *CVPR*, 2018. 3
- [109] Xingyi Zhou, Anurag Arnab, Shyamal Buch, Shen Yan, Austin Myers, Xuehan Xiong, Arsha Nagrani, and Cordelia Schmid. Streaming dense video captioning. In *CVPR*, 2024. 3

Shot-by-Shot: Film-Grammar-Aware Training-Free Audio Description Generation

Appendix

This appendix is organised as follows:

- **Implementation details:** In Appendix A, we provide additional details on film grammar prediction, action score evaluation settings, and the exact text instructions.
- **Evaluation metrics for AD generation:** In Appendix B, we elaborate on the key metrics used to measure AD quality.
- **Additional experimental results:** In Appendix C, we provide additional results for AD generation.
- **Human alignment with action scores:** In Appendix D, we provide a thorough description of the human agreement study for action scores, including curated test sets, a correlation analysis, and an inter-rater agreement study.
- **Qualitative visualisations:** Appendix E includes more detailed visualisations, along with an in-depth analysis of failure cases and assisted AD generation.

A. Implementation details

Thread structure prediction setting. To predict thread structure in a zero-shot manner, we resize video frames to 224p and employ DINOv2 [61] ViT-g/14 w. reg to extract spatial features. During matching score prediction, we consider a 5×5 mask neighbourhood for cost volume computation and set the softmax temperature to $\tau = 0.1$. To determine the relationship between each pair of shots, we apply a threshold of $\epsilon = 0.3$ to the matching score $s^{i,j}$.

Shot scale classification setting. For shot scale classification, we fine-tune the last 6 layers of the DINOv2 [61] ViT-B/14 on the MovieShots training set. During evaluation, we use the middle frame of each shot as input.

The averaged shot scale of the current shots (i.e. the effective shot scale S_{eff}) is used to guide the incorporation of additional factors in Stage I prompts. Specifically,

$$\text{Stage I factor} += \begin{cases} \text{Facial expression,} & \text{if } S_{\text{eff}} \leq 1.5 \\ \text{Key object,} & \text{if } 2 \leq S_{\text{eff}} \leq 3 \\ \text{Environment,} & \text{if } S_{\text{eff}} \geq 3.5 \\ \text{None,} & \text{otherwise} \end{cases}$$

Action score evaluation setting. To obtain character-free GT action sentences, we employ LLaMA3.1-70B [53] for pre-processing in two steps: (i) *Character information removal:* Character names are replaced with appropriate pronouns using the LLM with the prompt provided in Algorithm 3; (ii) *Action sentence extraction:* Each AD sentence is split into subsentences, each containing a single action. To achieve this, the LLM is prompted with instructions in Algorithm 4.

During hierarchical prediction parsing, we use spaCy¹ to

¹<https://spacy.io/models/en>

extract action phrases and corresponding verb lemmas from predicted sentences.

During similarity score computation, to extract sentence embeddings, we apply gte-Qwen2-7B-instruct [46], which supports optional text prompt input as guidance. We set the prompt to: “Retrieve relevant passages that involve similar actions, with particular focus on the verbs.”, further emphasising actions and verbs during similarity matching.

Additionally, to establish the LLM-based baseline, we define evaluation criteria as outlined in Algorithm 5 and use them to prompt LLaMA-3.1-70B and GPT-4o.

GPT-4o setup for AD generation. For both stages, we use gpt-4o-2024-08-06 [60] as the base model. For visual token extraction, the “detail” parameter is set to “low”.

Text instructions for AD generation. The prompts for AD generation are provided in Algorithms 1 and 2 for Stage I and Stage II, respectively. The Stage I prompt is designed for both Qwen2-VL and GPT-4o, while the Stage II prompt is tailored for LLaMA3 and GPT-4o.

B. Evaluation metrics for AD generation

CIDeR [81] measures text similarity by computing a weighted word-matching score, emphasising n-gram overlap while accounting for term frequency and importance through TF-IDF [71] weighting.

Recall@k/N [26] is a retrieval-based metric that evaluates whether predicted texts can be distinguished from their temporal neighbours. Specifically, for each predicted AD, it checks whether the AD can be retrieved at a top-k position within a neighbourhood of N ADs. Following prior work [22, 28], we report Recall@1/5 on CMD-AD and TV-AD, and Recall@5/16 on MAD-Eval.

LLM-AD-Eval [28] employs LLM agents (LLaMA2-7B [79] | LLaMA3-8B [53]) as evaluators to compare ground truth ADs with predictions, generating a matching score ranging from 1 (lowest) to 5 (highest).

CRITIC [28] measures the accuracy of character names in predicted ADs. It first resolves character ambiguity in GT ADs by applying a coreference model to replace pronouns with corresponding character names. During evaluation, the intersection-over-union (IoU) of predicted and ground truth character names is computed.

Action Score, as detailed in Sec. 4, evaluates the quality of predicted actions (i.e. verbs, object nouns, etc.) while minimising the influence of character name variations. Throughout this paper, unless otherwise specified, we use the action score to assess Stage II AD outputs, i.e. “Action” refers to “Action (Stage II)”.

Stage I VideoLLM	Stage II LLM	CMD-AD				TV-AD			
		CIDEr	CRITIC	Action (Stage I)	Action (Stage II)	CIDEr	CRITIC	Action (Stage I)	Action (Stage II)
Qwen2.5-VL-7B [8]	LLaMA3-8B [53]	24.1	49.7	35.5	27.2	26.5	43.6	36.4	23.8
VideoLLaMA3-7B [100]	LLaMA3-8B [53]	22.1	45.1	34.2	24.8	26.8	41.3	39.9	23.8
InternVL2.5-8B [18]	LLaMA3-8B [53]	24.0	46.0	35.0	28.1	28.3	41.2	36.0	24.2
Qwen2-VL-7B [85]	LLaMA3-8B [53]	26.3	47.8	38.2	28.4	31.1	42.2	38.2	23.9
Qwen2-VL-7B [85]	Gemma3-12B [78]	26.4	45.9	–	30.6	30.0	43.2	–	24.7
Qwen2-VL-7B [85]	Qwen3-8B [96]	26.3	45.2	–	28.4	30.7	43.4	–	24.7
Qwen2-VL-7B [85]	LLaMA3-8B [53]	26.3	47.8	–	28.4	31.1	42.2	–	23.9
GPT-4o [60]	GPT-4o [60]	26.1	49.1	40.2	32.5	34.2	46.5	41.0	27.4

Table A1. **Different open-source VideoLLMs (for Stage I) and LLMs (for Stage II).** As a reference, the last row reports results using the proprietary GPT-4o model. Note, we additionally report action scores for predicted Stage I description, whereas other metrics, including CIDEr, CRITIC, and Action (Stage II), measure the Stage II AD quality.

Thread structure	Stage I VideoLLM	TV-AD subset		
		CIDEr	CRITIC	Action
✗	Qwen2.5-VL-7B [8]	26.7	42.8	23.7
✓	Qwen2.5-VL-7B [8]	27.9 \uparrow 1.2	43.4 \uparrow 0.6	23.7 0.0
✗	VideoLLaMA3-7B [100]	24.9	42.1	21.9
✓	VideoLLaMA3-7B [100]	25.5 \uparrow 0.6	41.1 \downarrow 1.0	22.7 \uparrow 0.8
✗	InternVL2.5-8B [18]	25.9	40.3	21.8
✓	InternVL2.5-8B [18]	27.1 \uparrow 1.2	41.9 \uparrow 1.6	23.1 \uparrow 1.3
✗	Qwen2-VL-7B [85]	28.8	42.0	22.6
✓	Qwen2-VL-7B [85]	30.7 \uparrow 1.9	42.7 \uparrow 0.7	22.9 \uparrow 0.3

Table A2. **Thread structure injection for different open-source VideoLLMs.** The base model for Stage II is LLaMA3-8B. Thread structure information is injected only into subsets predicted to exhibit thread structures (\sim 60% in TV-AD).

For MAD-Eval, we additionally report the performance on conventional metrics including ROUGE-L [48], SPICE [5], METEOR [10], and BLEU-1 [62].

C. Additional experimental results

Different VideoLLMs for Stage I. Tab. A1 compares our AD generation performance using different open-source VideoLLMs in Stage I, validating our choice of Qwen2-VL-7B as the default model.

Beyond the Stage II action scores presented in the main text, we also report Stage I action scores as a direct indicator of dense description performance. In general, Stage I action scores are noticeably higher than their Stage II counterparts, suggesting that some ground truth actions are captured in dense descriptions but are not selected for the final AD outputs. This further supports the validity of our assisted AD generation protocol, where multiple candidate ADs with different actions are extracted from dense descriptions and await further selection.

Different LLMs for Stage II. Tab. A1 also compares different options for the Stage II LLM. Overall, the choice of LLM has a relatively minor impact compared to the Stage I VideoLLM. The default LLaMA3-8B achieves overall strong performance.

Exp.	CMD-AD			TV-AD		
	CIDEr	CRITIC	Action	CIDEr	CRITIC	Action
1	26.3	47.8	28.4	31.1	42.2	23.9
2	26.9	47.7	28.4	30.7	41.9	23.8
3	26.4	48.3	28.3	30.5	43.0	23.7
4	26.3	47.3	28.5	30.8	42.0	23.5
5	26.5	48.3	28.4	31.5	41.9	23.7
Mean	26.5	47.9	28.4	30.9	42.2	23.7
STD	0.2	0.4	0.1	0.4	0.5	0.1

Table A3. **Repeated (multi-run) experiments.** Results shown are from five independent runs (Stage I + Stage II) using different random seeds.

Method	Shot Partition	Film Grammar	Stage I VideoLLM	Stage II LLM	Total
AutoAD-Zero	–	–	2.18s	0.64s	2.82s
Ours	0.09s	0.12s	2.82s	0.72s	3.75s

Table A4. **Inference time analysis.**

Thread structure injection for different VideoLLMs.

We investigate how injecting thread structure into different open-source VideoLLMs affects AD generation. Specifically, we evaluate performance on TV-AD, which contains a large proportion of thread-structured video clips. As shown in Tab. A2, incorporating thread information leads to general performance boosts across various VideoLLMs.

Repeated AD generation. Tab. A3 reports results from five independent runs of our two-stage AD generation pipeline, each adopting a different random seed. The results are largely consistent across runs, indicating the stability of the AD generation process. In particular, the CRITIC results exhibit the highest variance, followed by CIDEr, while the action score remains relatively stable across repeated experiments.

Inference time analysis. We report the inference time *per AD* in Tab. A4. Our added components (e.g. shot partitioning, film grammar prediction) incur minimal overhead. The main cost increase arises from sampling more contextual frames for Stage I inputs. Overall, our method maintains reasonable ef-

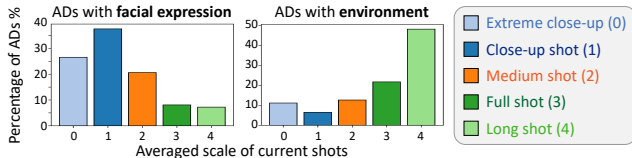


Figure A1. **Correlations between descriptive factors in ADs and averaged shot scales.**

iciency—given averaged input clip duration (including contextual shots) is 19.82s—while delivering clear performance gains.

Investigation on factor–scale correlations. To verify our assumptions about the strong correlations between descriptive factors in ADs and shot scales (as mentioned in Sec. 3.3), we used GPT-4o to extract key elements (e.g. environments) mentioned in GT ADs from 100 movies, and analysed their correlation with shot scales. As shown in Fig. A1, facial expressions predominantly occur in (extreme) close-up shots, while environmental cues align with full/long shots. These observations align with the assumption and generalise across real-world movies.

D. Human alignment with action scores

To assess how the proposed action score aligns with human knowledge, we curate a test set that measures the correlation between action scores and human scoring.

Scoring criteria. The human-annotated scores measure whether the ground truth (GT) action is described in the descriptions, ranging from 0 to 3 based on the following scoring criteria:

- Score 0 - *GT action is unrelated to any action in PD*
- Score 1 - *GT action is loosely related to an action in PD*
- Score 2 - *GT action is similar in meaning to an action in PD*
- Score 3 - *GT action exactly matches with an action in PD, using the same verb*

where PD stands for Predicted Description. Additional guidelines and scoring examples are provided in Tab. A5.

Test Set Formulation. We construct two test sets, namely the “paragraph set” and the “sentence set,” corresponding to the scoring of (Stage I) dense descriptions and (Stage II) AD sentences, respectively.

The paragraph set consists of 300 ground truth (GT) ADs, each paired with a predicted paragraph. In total, around 460 character-free GT actions are extracted, with each action-paragraph pair manually annotated by five workers using the 0–3 scoring scale, as described in the previous section. For ADs containing multiple GT actions, the final human score is obtained by averaging the manually assigned scores across different actions, meaning the resultant score may not always be an integer.

The sentence set contains 500 GT ADs with approximately 890 actions. Given the video clip described by the GT AD, instead of generating AD predictions from a VLM, we use a human-narrated AD from alternative sources for the same clip as the prediction. Similarly, the final human score is computed by averaging the scores across different actions within each GT AD.

Correlation between human scoring and metrics. We plot the human-annotated scores against the scores reported by each metric, as shown in Fig. A2. Most conventional metrics (blue and pink) fail to align with human evaluations of action predictions. In contrast, both LLM-based metrics and our action scores effectively assess the quality of action predictions in AD sentences (Fig. A2, bottom).

However, when evaluating longer paragraphs against the GT action, LLM-based metrics struggle, whereas action scores maintain a high correlation with human judgments (Fig. A2, top). Quantitative results on human-metric correlations are provided in Tab. 3.

Inter-rater analysis for action score. To obtain an upper bound on action scores for predicted AD sentences, we measure the agreement between two versions of human-annotated ADs for the same movie clip at different temporal IoUs (i.e. inter-rater agreement [28]). As shown in Tab. A6, the action score increases monotonically as the temporal IoU increases.

E. Qualitative visualisations

Fig. A3 presents additional visualisations for CMD-AD and TV-AD, comparing our method against other approaches.

Example failure cases are illustrated in Fig. A4. The top example highlights a hallucination issue in the prediction by Qwen2-VL + LLaMA3 (Ours), where the VideoLLM model misidentifies a “gift” as “candy” and infers an incorrect action of “eating,” which does not occur. When a stronger base model is used (Ours* with GPT-4o), this issue is mitigated.

Additionally, the current method struggles to incorporate broader (story-level) contextual understanding, as demonstrated in the bottom example of Fig. A4. Specifically, the model fails to describe the return of an insect and instead focuses on the sleeping woman. This limitation could potentially be addressed by incorporating more abstract information, which could be extracted from existing movie plots or summarised from a longer temporal context.

Visualisation of intermediate outputs. Fig. A5 provides more detailed visualisations, including intermediate results on thread structure and shot scale predictions, as well as Stage I descriptions. It also illustrates how the predicted shot scales influence the formulation of Stage I factors.

Visualisation for assisted AD generation. Fig. A6 presents additional examples with Stage I descriptions, from which multiple AD candidates are extracted. Among the five AD predictions, the one that best aligns with the ground truth (based on the averaged CIDEr and action score) is highlighted.

In the top example, multiple actions are present in the Stage I dense description (e.g. “kiss”, “hand on neck”, “eyes closed”, etc.), resulting in AD candidates that differ in both subjects and actions. In contrast, the middle example contains fewer actions (e.g. “shoot blue energy” and “look”/“observe”). In this case, the AD candidates primarily vary in style, such as changes in

Score	Note	GT action	Prediction
0	Unrelated: Completely irrelevant in actions, objects, environments, etc.	<i>He indicates that he was just joking.</i>	<i>The video begins with a man sitting under an umbrella in the rain, wearing a dark jacket with red accents and a blue shirt underneath. The scene transitions to show him lying on the ground, wet from the rain, with his arms outstretched and his eyes closed. Another person, dressed in a white shirt and black vest, reaches towards him from above.</i>
		<i>His new companion cuddles back up to him.</i>	<i>The flames start to die down.</i>
1	Loosely related: Semantically or causally relevant actions; Similar objects, environments, etc.	<i>He points the remote at the word's main menu.</i>	<i>1. Main characters: Morty and Michael Newman. 2. Actions: Both characters are standing in a room with futuristic digital screens and symbols around them. They appear to be engaged in an activity involving the screens, possibly interacting with them or using devices. 3. Character-character interactions: The two characters seem to be focused on the screens, possibly discussing or collaborating on something related to the content displayed.</i>
		<i>They step away from the window.</i>	<i>The four of them leave the cabin and go to their horses tethered to a rail at the side.</i>
2	Similar meaning: Similar verb, objects, environments, etc.	<i>He dives into an Olympic-sized swimming pool.</i>	<i>The movie clip shows a young boy running out of a house wearing only a diaper. He then jumps over a fence and runs across a lawn. The boy continues running and jumps into a swimming pool. Throughout the video, the boy's movements are energetic and playful, and he appears to be having fun.</i>
		<i>She carries a laundry basket outside the house.</i>	<i>Caroline brings some washing into the garden.</i>
3	Exactly same match: Exactly the same verb; Same/similar objects, environments, etc.	<i>He lights a cigar.</i>	<i>In the movie clip, a man is seen sitting at a table, while another man enters the room and approaches him. The second man takes out a cigarette and lights it, while the first man watches him. The scene is set in a dimly lit room with a table and chairs in the foreground, and a window in the background.</i>
		<i>She stares glumly at the night sky.</i>	<i>At the palace, Jasmine wanders out into her balcony and stares up at the stars.</i>

Table A5. **Example of human-annotated scores** assessing whether the ground truth (GT) action is accurately captured in the predictions. For each score, examples of a paragraph prediction and a sentence prediction are provided.

tIoU	#movies	#AD pairs	CIDEr	R@1/5	Action	LLM-AD-Eval
0.8	315	4447	61.5	71.2	45.6	3.04 3.24
0.9	267	999	69.8	80.4	47.6	3.53 3.34
0.95	148	229	73.9	—	47.8	3.57 3.45

Table A6. **Inter-rater analysis on CMD-AD**, where two versions of human-annotated ADs for the same movie clip are compared under different temporal intersection-over-union (tIoU) thresholds.

subjects or sentence structures, providing varied options for selection.

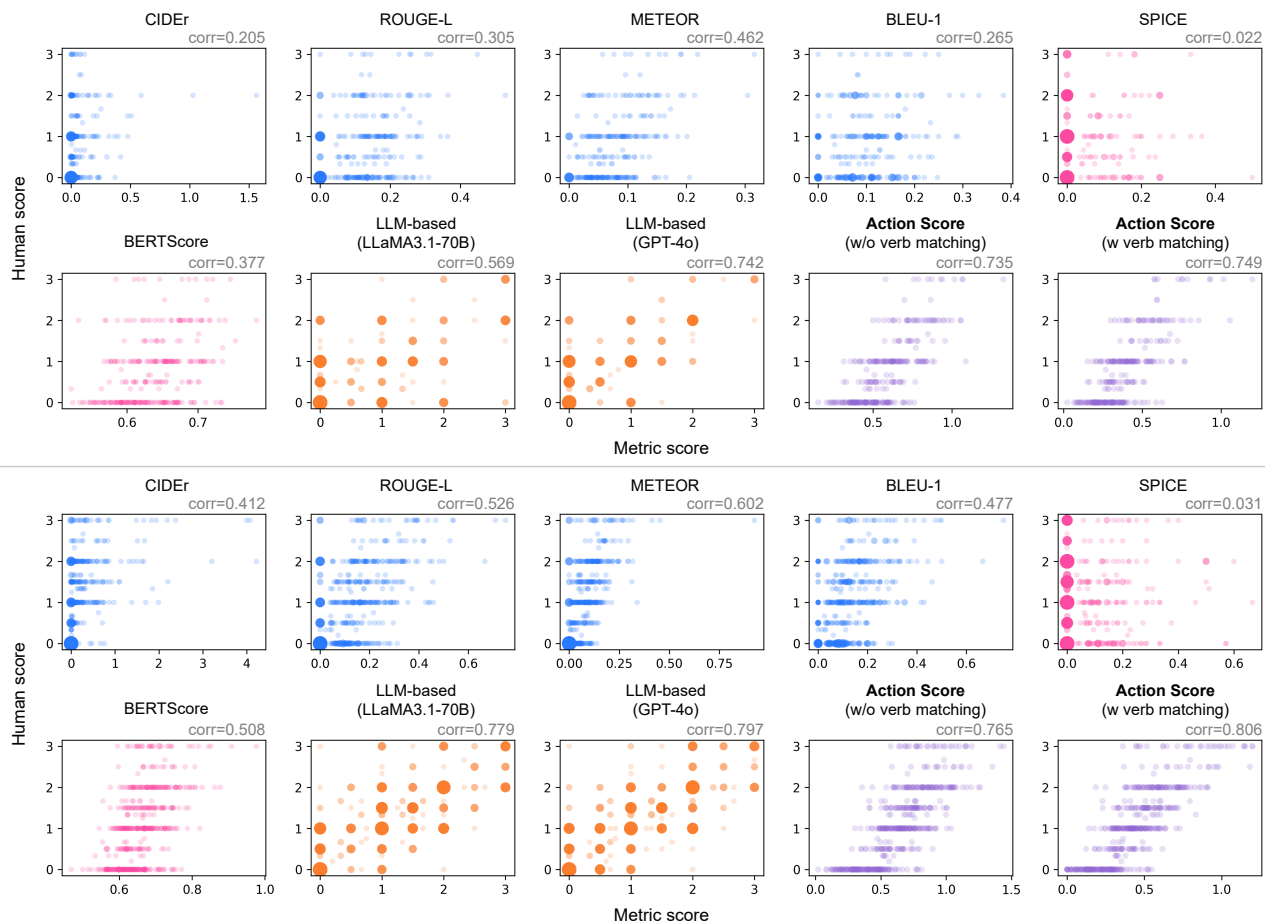


Figure A2. **Human agreement of different metrics for action evaluation**, where human-annotated scores are compared with metric scores. These scores measure the quality of the Stage I description paragraph (top) or the Stage II output AD (bottom). We consider various metrics, including word-matching-based metrics (blue), semantic-based metrics (pink), LLM-based metrics (orange), and our proposed action scores (purple). The Pearson correlation between human and metric scoring is reported. Within the scatter plots, we use colour depth and marker size to indicate density – larger and darker markers represent more data points at a single position. Zoom in for a clearer view.

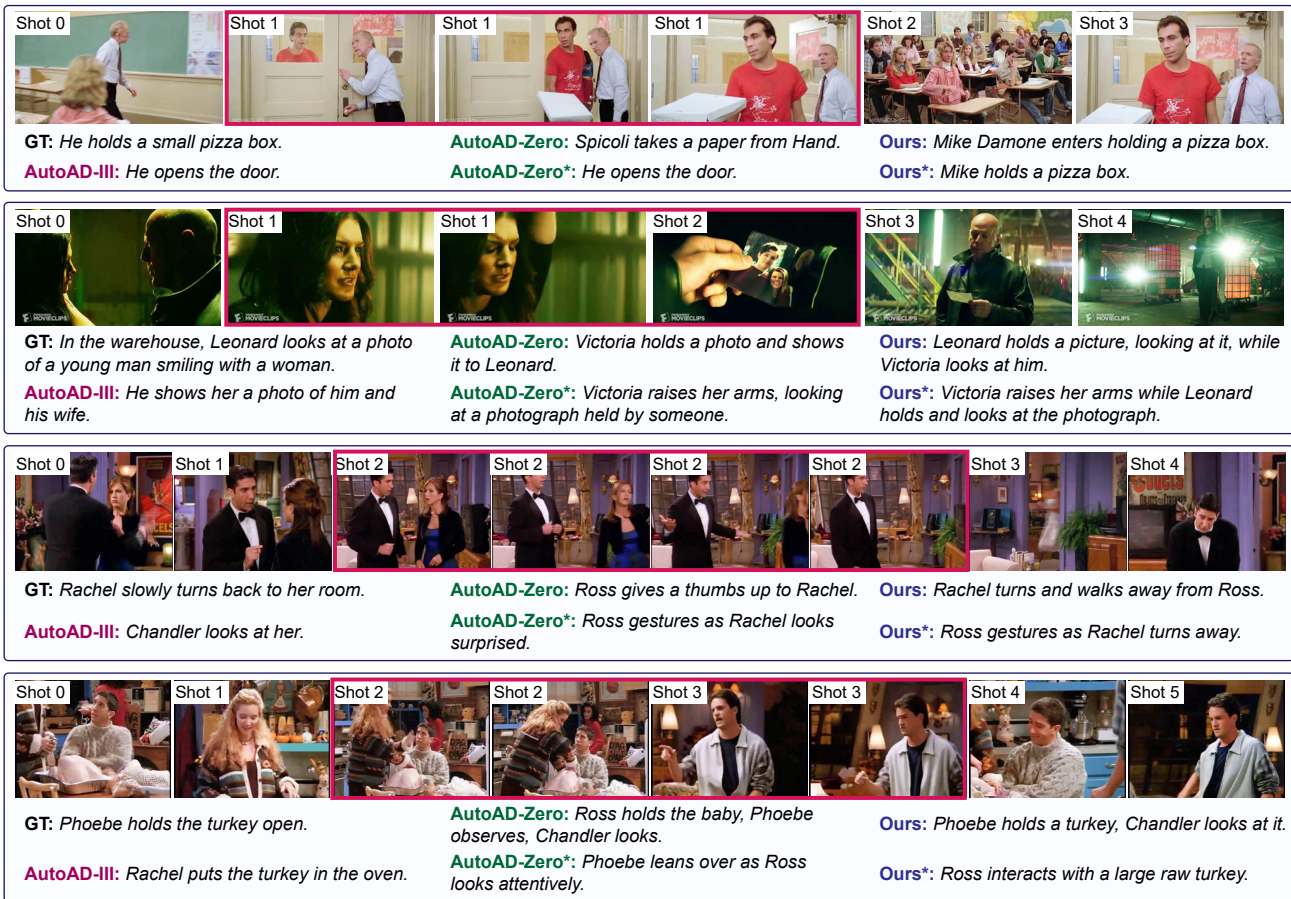


Figure A3. **More qualitative visualisations.** Current shots (corresponding to AD intervals) are outlined by red boxes for illustration purposes only. Training-free methods with “*” adopt GPT-4o (otherwise Qwen2-VL + LLaMA3). Examples from top to bottom are taken from *Fast Times at Ridgemont High* (1982), *Extraction* (2015), *Friends* (S3E2), and *Friends* (S1E9), respectively.

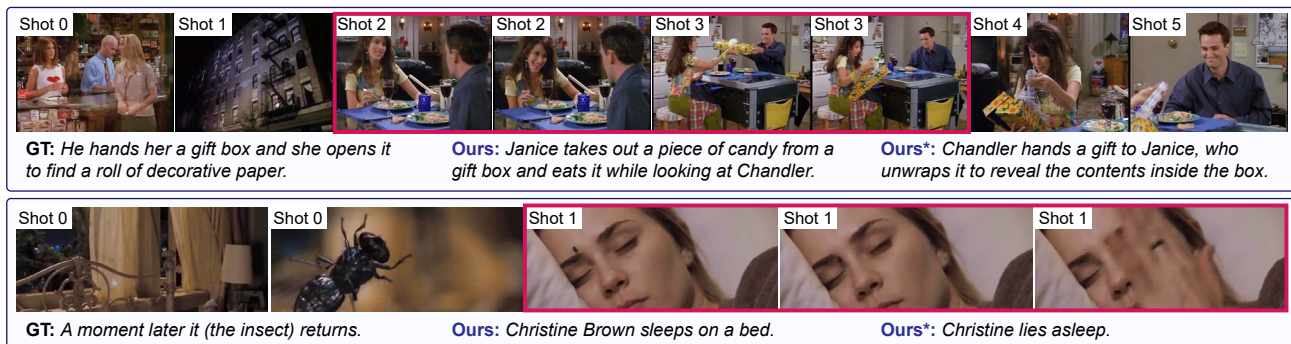


Figure A4. **Failure case visualisations.** Current shots (corresponding to AD intervals) are outlined by red boxes for illustration purposes only. Training-free methods with “*” adopt GPT-4o (otherwise Qwen2-VL + LLaMA3). Examples from top to bottom are taken from *Friends* (S3E4), and *Drag Me to Hell* (2009), respectively.

Predicted shot scale: Long Shot (Shot 2) **Predicted thread structure:** [Shot 0], [Shot 1], [Shot 2], [Shot 3], [Shot 4]

Ours (Stage I dense prediction):

1. Main characters: There are no circles available to indicate characters in Shot 2.
2. Actions: The camera pans over a valley surrounded by snow-covered mountains. There is a helicopter flying in the sky, and a person is seen sitting in the helicopter.
3. Character-character interactions: There are no interactions between characters in this shot.
4. Environment: The environment consists of a valley with snow-covered mountains in the background.

Ours (Stage II AD): A person sits in a helicopter flying over snow-covered mountains in a valley. **AutoAD-Zero:** The camera pans across a vast, snow-covered mountain range.

GT: Now the chopper flies over a barren valley with almost no ice or snow. **AutoAD-III:** The sun shines brightly on the snow-capped peaks of the Himalayas.

Predicted shot scale: Medium Shot (Shot 2, Shot 3) **Predicted thread structure:** [Shot 0, Shot 3], [Shot 1, Shot 5], [Shot 2, Shot 4]

Ours (Stage I dense prediction):

1. Main characters: Leslie Winkle, Leonard Hofstadter;
2. Actions: Leslie Winkle is playing the violin, while Leonard Hofstadter is playing the cello. Both characters are looking at each other.
3. Character-character interactions: Leslie Winkle and Leonard Hofstadter are looking at each other.
4. Key objects: violin, cello.

Ours (Stage II AD): Leslie plays the violin, gazing at Leonard on cello. **AutoAD-Zero:** Leslie looks at Leonard as they play violin and cello.

GT: Leslie and Leonard gaze at each other as they play faster and faster. **AutoAD-III:** Sheldon looks at Penny.

Predicted shot scale: Medium Shot (Shot 2) **Predicted thread structure:** [Shot 0, Shot 2, Shot 4], [Shot 1, Shot 3]

Ours (Stage I dense prediction):

1. Main characters: Phoebe Buffay, Rachel Green;
2. Actions: Phoebe Buffay sits down at the table, Rachel Green talks to Phoebe Buffay, Phoebe Buffay talks to Rachel Green;
3. Character-character interactions: Phoebe Buffay and Rachel Green are talking to each other;
4. Key objects: table, chairs, lamp.

Ours (Stage II AD): Phoebe sits with Rachel at a table. **AutoAD-Zero:** Phoebe turns her head to the right.

GT: Phoebe sits beside her. **AutoAD-III:** Rachel looks at Phoebe.

Figure A5. **Detailed visualisations including intermediate results**, such as predicted thread structures and shot scales, as well as Stage I dense descriptions. Current shots (corresponding to AD intervals) are outlined by red boxes for illustration purposes only. Training-free methods adopt Qwen2-VL + LLaMA3 as base models. Examples from top to bottom are taken from *Everest* (2015), *The Big Bang Theory* (S1E5), and *Friends* (S1E12), respectively.



Ours (Stage I dense prediction):

1. Main characters: Nathan and Karen.
2. Actions: In Shot 2, Nathan and Karen are kissing passionately, with Karen's hand on Nathan's neck. In Shot 3, they continue kissing, with Karen's hand still on Nathan's neck.
3. Character-character interactions: Karen's hand on Nathan's neck indicates a sense of intimacy and affection between the two characters.
4. Facial expressions: Both characters appear to be deeply engaged in the kiss, with their eyes closed and mouths open.

Ours (Stage II AD candidates):

- Nathan and Karen kiss passionately, hands entwined.
- Karen's hand on Nathan's neck indicates intimacy.
- Nathan and Karen kiss, eyes closed, mouths open.
- **Karen's hand on Nathan's neck during a romantic kiss. (Selected)**
- Nathan and Karen share a passionate, neck-touching kiss.

GT: She puts her hands on the back of his neck.



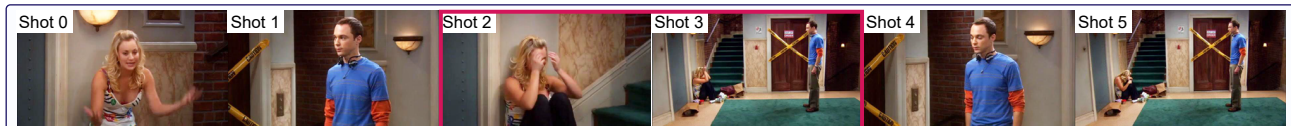
Ours (Stage I dense prediction):

1. Main characters: A large, white, furry creature is seen on a bridge, while a person is standing on a tower.
2. Actions: The white creature is emitting blue energy from its mouth, and the person on the tower is observing the creature.
3. Character-character interactions: The person on the tower is looking at the white creature.
4. Key objects: The white creature, the tower, and the bridge are the main objects in the scene.

Ours (Stage II AD candidates):

- The creature emits blue energy from its green mouth, shooting up into the sky.
- **The creature's blue energy shoots up into the sky, captivating the person's attention. (Selected)**
- The creature's green mouth glows as it shoots blue energy into the air, drawing the person's gaze.
- The person on the ground watches as the creature's blue energy blasts into the air.
- The person observes the creature's blue energy blasts, looking up at the sky.

GT: The Yeti's magical glow spirals into the sky above him.



Ours (Stage I dense prediction):

1. Main characters: Penny, Sheldon Cooper;
2. Actions: Penny is sitting on the floor, crying, and covering her face with her hands. Sheldon is standing in front of her, looking at her.
3. Character-character interactions: Penny is looking at Sheldon, and Sheldon is looking at Penny.
4. Key objects: caution tape, stairs, door, and bags on the floor.

Ours (Stage II AD candidates):

- **Penny sits crying on the floor. (Selected)**
- Sheldon stares at Penny crying.
- Penny and Sheldon gaze at each other.
- Sheldon stands looking at Penny.
- Penny looks at Sheldon on the floor.

GT: Penny sinks to the floor and cries.

Figure A6. **Visualisations for assisted AD generation**, where multiple AD candidates are extracted from the dense description, and the best among the five candidates is highlighted in **bold**. Current shots (corresponding to AD intervals) are outlined with red boxes for illustration purposes only. Our method uses Qwen2-VL and LLaMA3 as base models. The examples, from top to bottom, are taken from *Abduction* (2011), *Abominable* (2019), and *The Big Bang Theory* (S2E3), respectively.

Algorithm 1 Stage I text prompt

```
# Thread information injection
if exist(thread_structure):
    # {thread_structure}: e.g., "[Shot 1, Shot 3] share the same camera setup."
    thread_structure_text = "Finally, in one sentence, briefly explain why {thread_structure}\n"
else:
    thread_structure_text = ""

# Additional factors suggested by shot scales
factor_numbers = "four"
if effect_shot_scale <= 1.5:
    additional_factor_text = "4. Describe the facial expressions of characters.\n"
elif effect_shot_scale >= 2 and effect_shot_scale <= 3:
    additional_factor_text = "4. Describe the key objects that characters interact with.\n"
elif effect_shot_scale >= 3.5:
    additional_factor_text
    = "4. Describe the environment, focusing on the location, furniture, entrances and exits, etc.\n"
else:
    additional_factor_text = ""
    factor_numbers = "three"

# Stage I prompt
# {video_type}: "movie" or "TV series"
# {key_shots}: the middle shots (e.g., "[Shot 2, Shot 3]")
# {label_type}: "circles"
# {char_text}: character information (e.g., "Possible characters: Sheldon Cooper (red), ...")
prompt = (
    "Please watch the following
    {video_type} clip, where different shot numbers are labelled on the top-left of each frame.\n"
    f"Please briefly describe what happened in {key_shots} in the {factor_numbers} steps below:\n"
    f"1. Identify main characters (if {label_type} are available){char_text};\n"
    "2. Describe the actions of characters, i.e., who is doing what, focusing on the movements;\n"
    "3. Describe the interactions between characters, such as looking;\n"
    f"{additional_factor_text}"
    f>Note, colored {label_type
    } are provided for character indications only, DO NOT mention them in the description. "
    f"{thread_structure_text}"
    "Make sure you do not hallucinate information.\n"
    "### Answer Template ###\n" # Base format
    , need to be adjusted based on additionally factors, and whether the thread structure is injected
    "Description:\n"
    "1. Main characters: '';\n"
    "2. Actions: '';\n"
    "3. Character-character interactions: ''."
)
```

Algorithm 2 Stage II text prompt

```
# Stage II system prompt
# {video_type}: "movie" or "TV series"
sys_prompt = (
    f"[INST] <<
        SYS>>\nYou are an intelligent chatbot designed for summarizing {video_type} audio descriptions. "
        "Here's how you can accomplish
        the task:-----##INSTRUCTIONS: you should convert the predicted descriptions into one sentence. "
        "You should directly start the answer with the converted results
        WITHOUT providing ANY more sentences at the beginning or at the end. \n<</SYS>>\n\n{} [/INST] "
    )

# Dataset dependent information
if dataset == "CMD-AD":
    verb_list = ['look', 'turn', 'take', 'hold', 'pull', 'walk', 'run', 'watch', 'stare', 'grab',
                ', 'fall', 'get', 'go', 'open', 'smile'] # top-15 lemma verb in the corresponding training set
    speed_factor = 0.275 # averaged (duration / number of words in AD) in the training set
elif dataset == "TV-AD":
    verb_list = ['look', 'walk', 'turn',
                ', 'stare', 'take', 'hold', 'smile', 'leave', 'pull', 'watch', 'open', 'go', 'step', 'get', 'enter']
    speed_factor = 0.2695
elif dataset == "MAD-Eval":
    verb_list = ['look', 'turn', 'sit',
                ', 'walk', 'take', 'stand', 'watch', 'hold', 'pull', 'see', 'go', 'open', 'smile', 'run', 'get']
    speed_factor = 0.5102

# Single AD generation / multiple AD candidate outputs (as an assistant)
if not assistant_mode: # Single AD
    pred_text = "Provide the AD from a narrator perspective.\n"
else: # Multiple ADs
    pred_text = "Provide 5 possible ADs from a narrator perspective, each offering a valid and distinct
                summary by emphasizing different key characters, actions, and movements present in the scene.\n"

# Stage II user prompt
# {text_pred}: Stage I dense description outputs
# {duration}: duration of the AD interval
# {example_sentence}: 10 randomly sampled AD sentences from training sets
user_prompt = (
    "Please summarize the
    following description for one movie clip into ONE succinct audio description (AD) sentence.\n"
    f"Description: {text_pred}\n\n"

    "Focus on the most attractive characters, their actions, and related key objects.\n"
    "For characters, use their first names, remove titles such as 'Mr.' and 'Dr.'. If names
    are not available, use pronouns such as 'He' and 'her', do not use expression such as 'a man'.\n"
    "For actions, avoid mentioning the camera, and do not focus on 'talking'.\n"
    "For objects,
    especially when no characters are involved, prioritize describing concrete and specific ones.\n"
    "Do not mention characters' mood.\n"
    "Do not hallucinate information that is not mentioned in the input.\n"
    f"Try to identify the
    following motions (with decreasing priorities): {verb_list}, and use them in the description.\n"
    f"{pred_text}"
    f"Limit the length of the output within {int(duration / speed_factor + 1)} words.\n\n"

    "Output template (in JSON
    format): \"summarized_AD\": \"\".\n" # Adjust the template for single / multiple AD generation.
    "Here are some example outputs:\n"
    f"{example_sentence}"
    )
)
```

Algorithm 3 LLM-based character information removal text prompt

```
# System prompt for LLM-based character information removal in GT ADs
sys_prompt = (
    "You are an intelligent chatbot designed for removing character information of a sentence. "
    "Here's how you can accomplish the task: "
    "You should replace all character information
        including names, roles, and jobs into pronouns (e.g., he, she, they, her, him, them). "
    "Note, objects
        , locations, and animals are not counted as character information and should be kept as-is. "
    "You should output
        the result in JSON format WITHOUT providing ANY more sentences at the beginning or at the end."
)

# User prompt for LLM-based character information removal in GT ADs
# {text_gt}: GT AD
user_prompt = (
    "Please read the sentence below that describes a video clip:\n\n"
    f"Input sentence: \"{text_gt}\"\n\n"

    "Replace all character information
        including names, roles, and jobs into pronouns (e.g., he, she, they, her, him, them).\n"
    "Note, objects
        , locations, and animals are not counted as character information and should be kept as-is.\n"

    "**Examples:**\n"
    " - Example 1:\n"
    "   - Input sentence: \"Spicoli watches Mr. Hand pass out the schedule.\"\n"
    "   - Output: \"He watches him pass out the schedule.\"\n"
    " - Example 2:\n"
    "   - Input sentence: \"Waiting
        for a reply, the inspector has a look of smug satisfaction as he combs his neat moustache.\"\n"
    "   - Output
        : \"Waiting for a reply, he has a look of smug satisfaction as he combs his neat moustache.\"\n"
    " - Example 3:\n"
    "   - Input sentence: \"Emmerich's eyebrows twitch as he watches her.\"\n"
    "   - Output: \"His eyebrows twitch as he watches her.\"\n"
    " - Example 4:\n"
    "   - Input sentence: \"Inside is a second pair of doors.\"\n"
    "   - Output: \"Inside is a second pair of doors.\"\n"
    " - Example 5:\n"
    "   - Input
        sentence: \"The blonde saunters over to him in her striped pantsuit and leans in for a kiss.\"\n"
    "   - Output: \"She saunters over to him in her striped pantsuit and leans in for a kiss.\"\n"
    "... # More examples, omitted here for simplicity

    "**Output Format:**\n"
    "{\n"
    "  \"Output\": <output>\n"
    "}\n\n"
)
```

Algorithm 4 LLM-based action sentence extraction text prompt

```
# System prompt for LLM-based action sentence extraction from GT ADs
sys_prompt = (
    "You are an intelligent chatbot designed for decompose the sentence into subsentences. "
    "Here's how you can accomplish the task: "
    "You should split (rewrite if needed
      ) the sentence into subsentences, each containing only one action phrase (i.e., verb phrase). "
    "You should output
      your answer in JSON format WITHOUT providing ANY more sentences at the beginning or at the end."
)

# User prompt for LLM-based action sentence extraction from GT ADs
# {text_gt}: GT AD after character information removal
user_prompt = (
    "Please read the sentence below that describes a video clip:\n\n"
    f"Input sentence: \"{text_gt}\"\n\n"
    "Split and rewrite the sentence into subsentences, each containing only one action (i.e.,
      verb phrase) and preserving all other information (e.g., locations, time, affections, etc.).\n"
    "Do not output repeating actions.\n"
    "**Examples:**\n"
    "  - Example 1:\n"
    "    - Input sentence: \"He watches him pass out the schedule.\"\n"
    "    - Subsentences: [\"He watches him.\", \"He passes out the schedule.\"]\n"
    "  - Example 2:\n"
    "    - Input sentence
      : \"Waiting for a reply, he has a look of smug satisfaction as he combs his neat moustache.\"\n"
    "    - Subsentences: [\"He waits
      for a reply.\", \"He has a look of smug satisfaction.\", \"He combs his neat moustache.\"]\n"
    "  - Example 3:\n"
    "    - Input sentence: \"He
      swings in front of Kingpin, then bounces off a building and kicks the criminal into the air.\"\n"
    "    - Subsentences: [\"
      He swings in front of him.\", \"He bounces off a building.\", \"He kicks him into the air.\"]\n"
    "  - Example 4:\n"
    "    - Input sentence: \"His eyebrows twitch as he watches her.\"\n"
    "    - Subsentences: [\"His eyebrows twitch.\", \"He watches her.\"]\n"
    "  - Example 5:\n"
    "    - Input sentence: \"Inside is a second pair of doors.\"\n"
    "    - Subsentences: [\"Inside is a second pair of doors.\"]\n"
    "... # More examples, omitted here for simplicity

    "**Output Format:**\n"
    "{\n"
    "  \"Subsentences\": \n"
    "    [\n"
    "      <subsentence1>,\n"
    "      <subsentence2>,\n"
    "      <subsentence3>,\n"
    "      ... \n"
    "    ]\n"
    "}\n\n"
)
```

Algorithm 5 LLM-based action metric text prompt

```
# System prompt for LLM-based action evaluation
sys_prompt = (
    "You are an evaluation assistant designed to assess the accuracy of a description
    (Des) in capturing the action specified in a reference sentence (Ref) for a movie clip. "
    "Focus only on the presence
    of the referenced action and ignore any additional, unrelated actions in the description. "
    "Ignore any character information in the description. "
    "Avoid assumptions
    about action details beyond what is explicitly provided in either the reference or description. "
    "Output the
    result exclusively in JSON format, with a score (0 to 3) and a brief explanation describing the
    relationship between the actions in Ref and Des, without any introductory or concluding text."
)

# User prompt for LLM-based action evaluation
# {text_gt}: character-free action sentence extracted from GT AD
# {text_pred}: predicted dense description (paragraph) or AD sentence
user_prompt = (
    "You will be provided with a reference action sentence (Ref) and a description (Des) for a clip. "
    "Your task is to evaluate if the action described in Ref is explicitly or clearly implied in Des. "
    "Focus only on the presence
    of the referenced action, without considering any additional actions and character information
    that may appear in Des, and do not assume any actions beyond those stated in Ref or Des. "
    "The output should be a score (0 to 3) with a brief
    one-sentence explanation describing the relationship between the actions in Ref and Des.\n\n"

    "# Scoring Criteria:\n"
    "- **Score 0:** The action in Ref is completely unrelated to actions in Des.\n"
    "- **Score 1:** The action in Ref is loosely related to an action in Des.\n"
    "- **Score 2:** The action in Ref is similar in meaning to an action in Des.\n"
    "- **Score 3:** The action in Ref exactly matches an action in Des, using the same verb.\n\n"
    "# Examples:\n"
    "- Example 1:\n"
    "  - Ref: 'He runs across the street.'\n"
    "  - Des: 'Tom is jogging down the street.'\n"
    "  - Output: {\n"
    "    'score': 2,\n"
    "    'explanation': 'The
    action \"runs across the street\" in Ref is similar to \"jogging down the street\" in Des.'\n"
    "  }\n\n"
    "- Example 2:\n"
    "  - Ref: 'He pours wine into a glass.'\n"
    "  - Des: 'The woman drinks.'\n"
    "  - Output: {\n"
    "    'score': 1,\n"
    "    'explanation
    ': 'The action \"pours wine into a glass\" in Ref is loosely related to \"drinks\" in Des.'\n"
    "  }\n\n"
    "... # More examples, omitted here for simplicity
    "# Output Format:\n"
    "{\n"
    "  'score': <score>,\n"
    "  'explanation': '<explanation>'\n"
    "}\n\n"
    "# Now, apply these instructions to the following texts:\n\n"
    f"  - # Reference (Ref): '{text_gt}'\n"
    f"  - # Description (Des): '{text_pred}'"
)

```
