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

Junyu Xie [1] Tengda Han[1] Max Bain[1] Arsha Nagrani[1] Eshika Khandelwal [2,3]
Gül Varol[1,3] Weidi Xie[1,4] Andrew Zisserman[1]

[1]Visual Geometry Group, University of Oxford [2] CVIT, IIIT Hyderabad
[3]LIGM, École des Ponts ParisTech [4]SAI, Shanghai Jiao Tong University

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

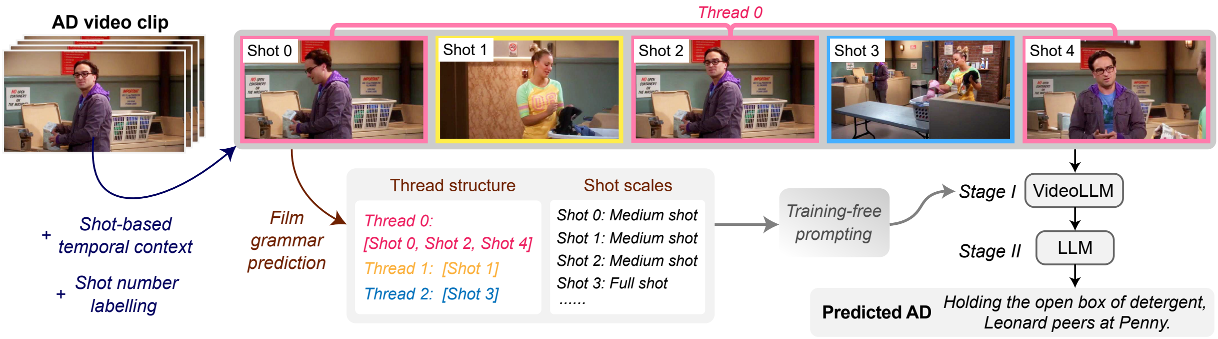


Figure 1. **Overview of our training-free Audio Description (AD) generation framework.** The input video clip (corresponding to the AD interval) is extended to include adjacent shots, providing richer temporal visual context. Each video frame is labeled with its corresponding shot number. The extended 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 neighboring 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 [16, 23–25, 45, 72, 86, 89] and TV material [19, 82]. 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 [37, 50, 52]. 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 [68]. 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 [82], 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 [47], 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 [8, 44, 55, 71], several new AD metrics [23, 25, 89] have been proposed, including CRITIC [25] 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 [25] and previous user studies [78]. 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: **(i)** We incorporate shot-based temporal context into AD generation via training-free prompting techniques including shot number referral and dynamic frame sampling; **(ii)** 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; **(iii)** We improve the current AD evaluation by introducing the character-free action score, and a new assistance-oriented evaluation protocol; **(iv)** 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 [25, 67] and TV series [82], with human annotations sourced from platforms such as AudioVault [2].

For automatic AD generation, prior works [19, 23–25, 45, 72] 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 [16, 82, 86, 89] 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 [72] and DistinctAD [19] 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 [5, 31, 36, 62, 65] categorise their characteristics based on camera setups, including shot scales (examples shown in Fig. 4) and camera movements. Correspondingly, various models [15, 42, 46, 62, 70] have been developed for shot type classification.

Regarding shot-wise relationships, a few datasets [5, 10, 56] have been proposed to explore transitions (i.e. cuts) between shots, which have also been leveraged in video content generation [20, 57, 63, 92]. Beyond pairwise shot transitions, research has also investigated longer temporal contexts with thread-based editing structures. Notably, Hoai et al. introduced the Thread-Safe [27] dataset, demonstrating that thread information can enhance action recognition. These structures have also been utilised in video-based face and people clustering [9, 69].

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 [32, 33, 39, 74, 76] in video captioning typically treat event localisation and captioning as independent stages, whereas more recent approaches [12, 18, 34, 38, 51, 58, 61, 77, 80, 85, 96, 97] integrate these tasks in an end-to-end manner. Video captioning benchmarks cover a range of domains, including cooking [95], actions [39], movies [64], TV series [41], and open-domain settings [11, 14, 26, 49, 66, 83, 84, 88, 93].

Temporal grounding in VLM. To equip conventional VLMs with temporal grounding capability, some studies [29, 43, 79] 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 [13, 17, 21, 22, 28, 30, 59, 73] or embedding time information within visual tokens [29, 43, 79].

Training-free approaches [60, 81, 94] have also been explored for temporally grounded understanding. Notably, a recent work [81] 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.

3. Training-free AD generation framework

Given a video clip $\mathcal{V}=\{\mathcal{I}_0,...,\mathcal{I}_T\}$, 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 $\mathcal{P}_{\text{LLM}}$) 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 formuation.

3.1. Leveraging shot-based temporal context

Regarding the visual inputs to VideoLLM, prior works [25, 45] often sample frames $\{\mathcal{I}_{t_A},...,\mathcal{I}_{t_B}\}$ 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.

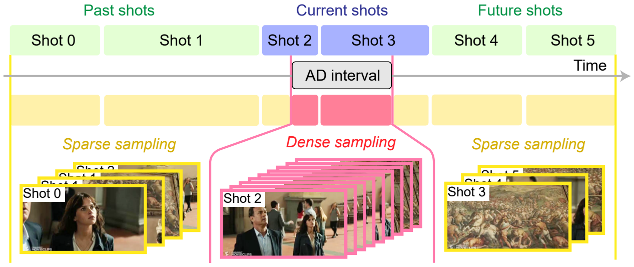


Figure 2. **Shot-based temporal context,** where *current shots* are defined as those temporally overlapping with the AD interval. *Past shots* and *future shots* provide extended contextual information. Shot numbers are visually overlaid on the top-left of each frame, and frames within the AD interval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

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 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"*.

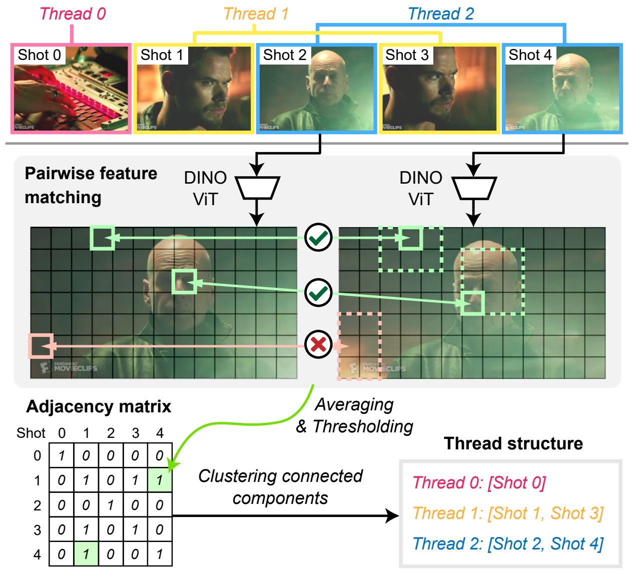
Emphasising the targeted AD interval. To ensure that the VideoLLM focuses on describing visual content within the AD interval, we propose two strategies: dynamic frame sampling and shot number referral.

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 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.



Shot	0	1	2	3	4
0	1	0	0	0	0
1	0	1	0	1	1
2	0	0	1	0	0
3	0	1	0	1	0
4	0	1	0	0	1

Thread 0: [Shot 0]
Thread 1: [Shot 1, Shot 3]
Thread 2: [Shot 2, Shot 4]

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.

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 ($\mathcal{I}^i_{T_i}$) with the first frame of Shot j ($\mathcal{I}^j_0$), extracting their DINOv2 [54] features as $f^i_{T_i}$ and $f^j_0 \in \mathbb{R}^{h\times w\times c}$.

Inspired by [35], we assess frame-wise dense correlations by computing a cost volume between their feature maps:

$$\mathcal{C}^{i,j}_{p,q} = m(p,q) \circ (\hat{f}^i_{T_i;p} \cdot \hat{f}^j_{0;q}) \tag{3}$$

where $\hat{f}^i_{T_i;p}$ and $\hat{f}^j_{0;q} \in \mathbb{R}^c$ 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:

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

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 (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.

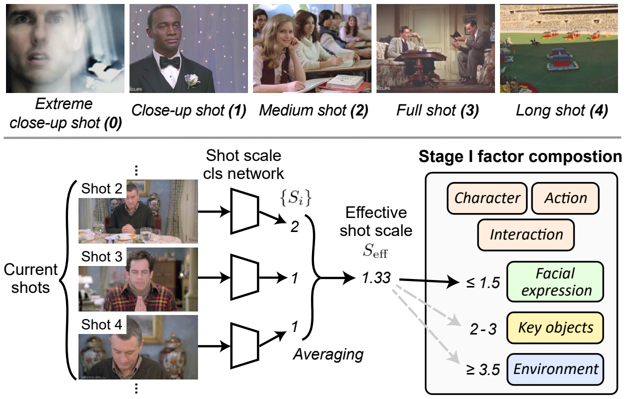


Figure 4. **Shot scales. Top:** Examples of five different shot scales. **Bottom:** Stage I factor composition based on shot scale classification. The scales of the current shots in the clip are predicted and averaged. The resulting effective shot scale then guides the formulation of the Stage I prompt,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

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,...,Shot j], we formulate the prompt as: *"[Shot i,...,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 [54] 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

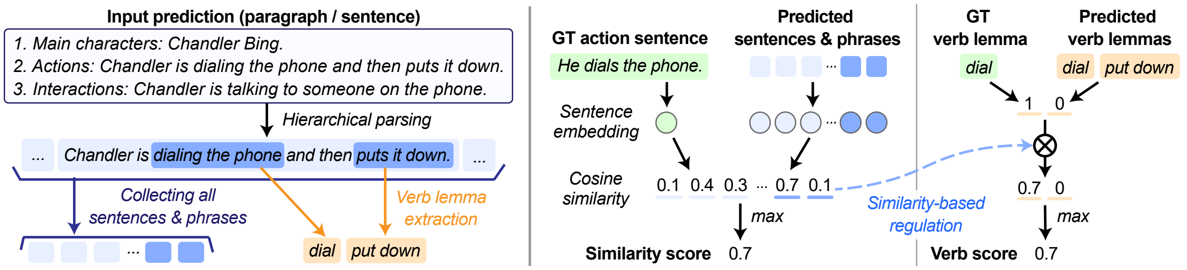

Input prediction (paragraph / sentence)
1. Main characters: Chandler Bing.
2. Actions: Chandler is dialing the phone and then puts it down.
3. Interactions: Chandler is talking to someone on the phone.
Hierarchical parsing
...
Chandler is dialing the phone and then puts it down.
...
Collecting all sentences & phrases
Verb lemma extraction
...
dial
put down
GT action sentence
He dials the phone.
Predicted sentences & phrases
...
Sentence embedding
...
Cosine similarity
0.1 0.4 0.3 ... 0.7 0.1
max
Similarity score 0.7
Similarity-based regulation
GT verb lemma
dial
Predicted verb lemmas
dial
put down
1 0
0.7 0
max
Verb score 0.7

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.

($S_{\text{eff}} \geq 3.5$), 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 [25], MAD-Eval [23]) and TV series (TV-AD [82]). In more detail, CMD-AD is constructed by aligning ground truth ADs with the Condensed Movie Dataset (CMD) [7], 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 [64]. On the other hand, TV-AD features 34k AD annotations from 13 TV series, with its test set sourced from TVQA [40], consisting of 3k ADs.

For AD evaluation, we follow the prior works [19, 25] to assess general prediction quality using CIDEr [71], Recall@k/N [24], and LLM-AD-Eval [25]. We also consider character recognition accuracy using CRITIC [25] 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 the Supp. Mat.

Shot detection. To segment the video clip into shots, we use PySceneDetect [3] with the "Adaptive Detection" method,

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.

Metric	Input	Accuracy	Macro-F1
ViViT [6]	RGB	0.747	0.751
SGNet [62]	RGB + Flow	0.875	–
Lu et al. [46]	RGB + mask	0.892	–
Li et al. [42]	RGB	0.895	0.897
Ours	RGB	**0.897**	**0.899**

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

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 [82], which applies coloured circles around faces for visual character indication. For simplicity, the visualisations in this paper do not display these circle labels.

Regarding the base models, we use Qwen2-VL-7B [75] as the VideoLLM in Stage I and LLaMA3-8B [48] 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 [53] 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 [27], 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) [69].

Metric	Paragraph		Sentence	
	Pearson	Spearman	Pearson	Spearman
CIDEr [71]	0.205	0.264	0.412	0.528
ROUGE-L [44]	0.305	0.280	0.526	0.512
SPICE [4]	0.022	0.048	0.031	0.012
BERTScore [91]	0.377	0.393	0.508	0.507
LLM-based (GPT-4o [53])	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.

Tab. 1 verifies the choice of DINOv2-g14 [54] 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 [42, 46], we use the MovieShots [62] dataset, which consists of 46k shots (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 [42] 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 [54], 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

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** ↑0.8	**48.9** ↑1.4	**27.7** ↑0.3	**30.7** ↑1.9	**42.7** ↑0.7	**22.9** ↑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).

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 human agreement study, please refer to the Supp. Mat.

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 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.

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.

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.

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. Additionally, we also report our performance on the MAD-Eval benchmark in Supp. Mat.

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

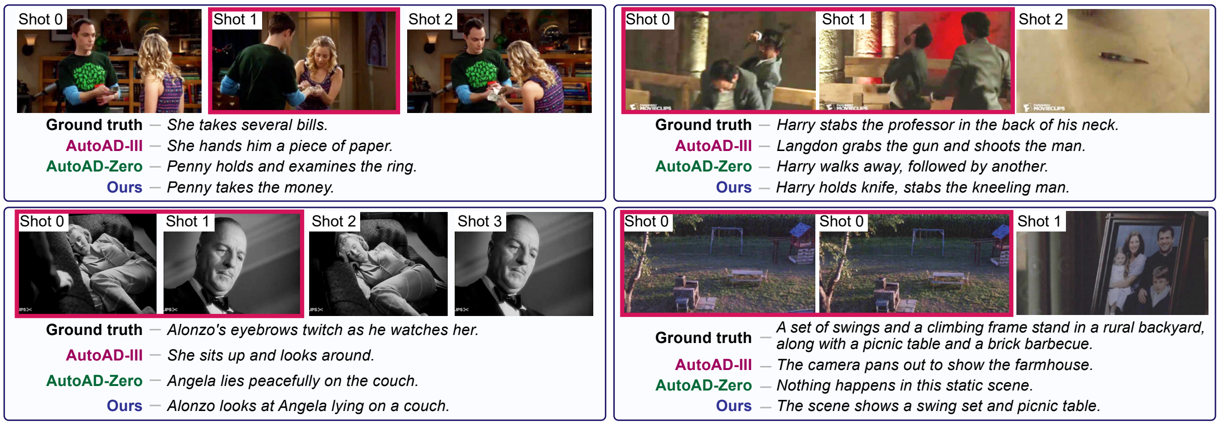
5.8. Assisted AD generation

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.

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 [24]	CLIP-B32	GPT-2	✗	✗	13.5	8.2	–	26.1	2.08 | –	–	–	–	–	–
AutoAD-III [25]	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 [19]	CLIP$_{\text{AD}}$-B16	LLaMA3-8B	✗	✗	22.7	–	–	33.0	2.88 | 2.03	27.4	–	–	32.1	2.89 | 2.00
Video-LLaMA [90]	Video-LLaMA-7B	–	✓	✗	4.8	0.0	–	22.0	1.89 | –	–	–	–	–	–
VideoBLIP [87]	VideoBLIP	–	✓	✗	5.2	0.0	–	23.6	1.91 | –	–	–	–	–	–
AutoAD-Zero [82]	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 [82]	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 [82]	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.



Ground truth — *She takes several bills.*
AutoAD-III — *She hands him a piece of paper.*
AutoAD-Zero — *Penny holds and examines the ring.*
Ours — *Penny takes the money.*

Ground truth — *Harry stabs the professor in the back of his neck.*
AutoAD-III — *Langdon grabs the gun and shoots the man.*
AutoAD-Zero — *Harry walks away, followed by another.*
Ours — *Harry holds knife, stabs the kneeling man.*

Ground truth — *Alonzo's eyebrows twitch as he watches her.*
AutoAD-III — *She sits up and looks around.*
AutoAD-Zero — *Angela lies peacefully on the couch.*
Ours — *Alonzo looks at Angela lying on a couch.*

Ground truth — *A set of swings and a climbing frame stand in a rural backyard, along with a picnic table and a brick barbecue.*
AutoAD-III — *The camera pans out to show the farmhouse.*
AutoAD-Zero — *Nothing happens in this static scene.*
Ours — *The scene shows a swing set and picnic table.*

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 [82]	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 8. **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 hyper-parameter for top-p (nucleus) sampling and τ_p as the sampling temperature. "Joint output" generates five ADs simultaneously in a single run.

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. 8, 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 the Supp. Mat.

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/`. 5

[4] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016. 6

[5] 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

[6] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021. 6

[7] Max Bain, Arsha Nagrani, Andrew Brown, and Andrew Zisserman. Condensed movies: Story based retrieval with contextual embeddings. In *ACCV*, 2020. 5

[8] Satanjeev 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

[9] 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. 2

[10] Boris Chen, Amir Ziai, Rebecca S. Tucker, and Yuchen Xie. Match cutting: Finding cuts with smooth visual transitions. In *WACV*, 2023. 2

[11] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, pages 190–200, 2011. 3

[12] Shaoxiang Chen and Yu-Gang Jiang. Towards bridging event captioner and sentence localizer for weakly supervised dense event captioning. In *CVPR*, 2021. 3

[13] 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

[14] 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

[15] 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

[16] Peng Chu, Jiang Wang, and Andre Abrantes. LLM-AD: Large language model based audio description system. *arXiv preprint arXiv:2405.00983*, 2024. 1, 2

[17] Andong Deng, Zhongpai Gao, Anwesa Choudhuri, Benjamin Planche, Meng Zheng, Bin Wang, Terrence Chen, Chen Chen, and Ziyan Wu. Seq2time: Sequential knowledge transfer for video llm temporal grounding. *arXiv preprint arXiv:2411.16932*, 2024. 3

[18] Chaorui Deng, Shizhe Chen, Da Chen, Yuan He, and Qi Wu. Sketch, ground, and refine: Top-down dense video captioning. In *CVPR*, 2021. 3

[19] 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, 5, 8

[20] 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

[21] 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

[22] 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

[23] Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. Autoad: Movie description in context. In *CVPR*, 2023. 1, 2, 5

[24] 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. 5, 8

[25] 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, 5, 8

[26] 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

[27] Minh Hoai and Andrew Zisserman. Thread-safe: Towards recognizing human actions across shot boundaries. In *ACCV*, 2014. 2, 6

[28] Wenyi Hong, Weihan 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

[29] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments. In *CVPR*, 2024. 3

[30] 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

[31] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *ECCV*, 2020. 2

[32] Vladimir Iashin and Esa Rahtu. A better use of audio-visual cues: Dense video captioning with bi-modal transformer. In *BMVC*, 2020. 2

[33] Vladimir Iashin and Esa Rahtu. Multi-modal dense video captioning. In *CVPR Workshops on Multimodal Learning*, 2020. 2

[34] 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

[35] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. In *NeurIPS*, 2020. 4

[36] 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

[37] Steve D. Katz. *Film Directing Shot by Shot: Visualizing from Concept to Screen*. Michael Wiese Productions, ISBN: 1615932976, 2019. 1

[38] 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

[39] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017. 2, 3

[40] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. TVQA: Localized, compositional video question answering. In *EMNLP*, 2018. 5

[41] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *ECCV*, 2020. 3

[42] Yuzhi Li, Feng Tian, Haojun Xu, and Tianfeng Lu. Toward unified and quantitative cinematic shot attribute analysis. *Electronics*, 2023. 2, 6

[43] 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

[44] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, 2004. 2, 6

[45] 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

[46] 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

[47] 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

[48] Meta. The llama 3 herd of models. *arXiv preprint arXiv: 2407.21783*, 2024. 6

[49] 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

[50] James Monaco. *How to Read a Film: Movies, Media, and Beyond*. OUP, ISBN: 0571168973, 2009. 1

[51] Jonghwan Mun, Linjie Yang, Zhou Ren, Ning Xu, and Bohyung Han. Streamlined dense video captioning. In *CVPR*, 2019. 3

[52] Walter Murch. *In the Blink of an Eye: A Perspective on Film Editing*. Silman-James Press, ISBN: 1879505622, 2021. 1

[53] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv: 2303.08774*, 2024. 6

[54] 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, 6

[55] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 2

[56] 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

[57] 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

[58] Toby Perrett, Tengda Han, Dima Damen, and Andrew Zisserman. It's just another day: Unique video captioning by discriminative prompting. In *ACCV*, 2024. 3

[59] 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

[60] 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

[61] Tanzila Rahman, Bicheng Xu, and Leonid Sigal. Watch, listen and tell: Multi-modal weakly supervised dense event captioning. In *ICCV*, 2019. 3

[62] Anyi Rao, Jiaze 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

[63] 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

[64] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. In *CVPR*, 2015. 3, 5

[65] 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

[66] 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

[67] 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

[68] 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. 1

[69] 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. 2, 6

[70] Bartolomeo Vacchetti and Tania Cerquitelli. Cinematographic shot classification with deep ensemble learning. *Electronics*, 2022. 2

[71] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. In *CVPR*, 2015. 2, 5, 6

[72] Hanlin Wang, Zhan Tong, Kecheng Zheng, Yujun Shen, and Limin Wang. Contextual ad narration with interleaved multimodal sequence. *arXiv preprint arXiv:2403.12922*, 2024. 1, 2

[73] 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

[74] Jingwen Wang, Wenhao Jiang, Lin Ma, Wei Liu, and Yong Xu. Bidirectional attentive fusion with context gating for dense video captioning. In *CVPR*, 2018. 2

[75] 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

[76] 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. 2

[77] 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

[78] 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

[79] 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

[80] 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

[81] 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

[82] 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, 5, 6, 8

[83] 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

[84] 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

[85] 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

[86] 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

[87] Keunwoo Peter Yu. Videoblip, 2023. 8

[88] 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

[89] 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

[90] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *ENNLP*, 2023. 8

[91] Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *ICLR*, 2020. 6

[92] 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

[93] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *CVPR*, 2023. 3

[94] Minghang Zheng, Xinhao Cai, Qingchao Chen, Yuxin Peng, and Yang Liu. Training-free video temporal grounding using large-scale pre-trained models. *arXiv preprint arXiv:2408.16219*, 2024. 3

[95] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 3

[96] 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

[97] 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