Now You See Me: Context-Aware Automatic Audio Description

Seon-Ho Lee* Korea University
Jue Wang Amazon AGI
David Fan† Meta FAIR
Zhikang Zhang Amazon AGI
Linda Liu Amazon Prime Video
Xiang Hao Amazon Prime Video
Vimal Bhat Amazon Prime Video
Xinyu Li Amazon AGI

Abstract

Audio Description (AD) plays a pivotal role as an application system aimed at guaranteeing accessibility in multimedia content, which provides additional narrations at suitable intervals to describe visual elements, catering specifically to the needs of visually impaired audiences. In this paper, we introduce $\mathrm{CA}^3\mathrm{D}$*, the pioneering unified Context-Aware Automatic Audio Description system that provides AD event scripts with precise locations in the long cinematic content. Specifically,* $\mathrm{CA}^3\mathrm{D}$ *system consists of: 1) a Temporal Feature Enhancement Module to efficiently capture longer term dependencies, 2) an anchor-based AD event detector with feature suppression module that localizes the AD events and extracts discriminative feature for AD generation, and 3) a self-refinement module that leverages the generated output to tweak AD event boundaries from coarse to fine. Unlike conventional methods which rely on metadata and ground truth AD timestamp for AD detection and generation tasks, the proposed* $\mathrm{CA}^3\mathrm{D}$ *is the first end-to-end trainable system that only uses visual cue. Extensive experiments demonstrate that the proposed* $\mathrm{CA}^3\mathrm{D}$ *improves existing architectures for both AD event detection and script generation metrics, establishing the new state-of-the-art performances in the AD automation.*

1. Introduction

Movie audio description (AD) is the verbal narration which describes the visual elements in the movie. Movie AD aims to enable people to understand the story of the movie only with the sounds. Hence, it is especially important for that visually impaired people can have the equal opportunity to enjoy movies. However, the manual generation of AD is time-consuming and expensive; up to $75 per minute of content [1]. To reduce the enormous cost for AD generation, various techniques [13, 14] have been proposed. However, the automated AD generation is still challenging and requires more improvements for practical usage.

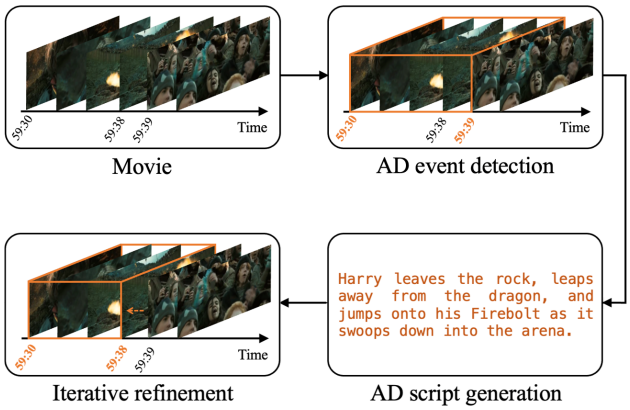


Figure 1. **CA³D**: We propose a one-stage AD Automation system that detects AD events, generates AD scripts and refines AD detection on a whole movie.

To build an automated AD system for movies, two subproblems should be addressed: AD event detection and AD script generation. Given a movie, AD event detection aims to find the temporal locations where AD should be provided while the AD script generation aims to create textual description for the detected AD event window. Compared to the standard (dense) video captioning [15, 18, 45, 48], the AD automation is more challenging because the boundary between AD and non-AD events are ambiguous. Hence, it requires long-term context from movies to decide when and where to generate AD. Despite of its importance, there is no previous work addressing the AD automation properly.

Recently, AutoAD [14] and its improved version, Au-

*Work done during an internship at Amazon Prime Video.
†Work done while at Amazon Prime Video.

[1] https://www.3playmedia.com/blog/how-much-does-audio-description-cost/

toAD II [13] have been proposed to tackle the AD automation. However, they decouple the AD detection and generation as two separate tasks and simplify each of them. Specifically, AutoAD II leverages subtitle information as the context to exclude non-AD events and predicts if at least one AD event exists in the speech gap, which cannot produce precise AD timestamp and may be biased on the data statistic. For the AD generation, conventional methods [13,14] assume the access to the ground truth AD events locations. To follow the story more accurately, they also leverage the additional information such as the previous AD scripts, subtitles, and character data to generate AD scripts. However, such metadata is not always available for all movies. Moreover, the AD detection and generation should be coupled as a whole system in the real industry, where has no access to any prior.

In this paper, we propose an end-to-end trainable system to achieve the context-aware automatic audio description ($\mathrm{CA}^3\mathrm{D}$), which is the first unified algorithm to both AD event detection and AD script generation. Specifically, we first propose a temporal feature enhancement module to capture longer term dependencies to expand the temporal horizon of input by employing the structured state space sequence (S4) model, and then we introduce an anchor-based AD event detector associated with a feature suppression module which inhibits the following AD generator from using the information irrelevant to the AD event. At last, we also propose a self-refinement module as an option for fine-tuning the AD event location and scripts. Through extensive experiments, we demonstrate the superior performance of our proposed system on academia public benchmark (MADv2 [13, 39] dataset). We summarize the contribution of this paper as following:

- We propose the first unified AD automation system, $\mathrm{CA}^3\mathrm{D}$, which generates AD scripts with precise timestamp on the entire movie.
- We first employ the S4 model as a temporal feature enhancement module and sub-sequentially propose anchor-based detector, differentiable feature suppression module and an optional self-refinement module to work seamlessly in the AD automation system.
- We achieve the promising performances on both AD detection and generation tasks. Notably, even without leveraging external data and ground truth locations, the proposed algorithm shows competitive or better results to the previous methods which exploit those additional information.

2. Related Work

2.1. Dense Video Captioning

The dense video captioning (DVC) is one of the closest applications to the AD automation, both necessitating the detection of events within an untrimmed video and the subsequent generation of descriptive content for each identified event. Previous research efforts can be broadly categorized into two groups: 1) two-stage models such as those presented in works like [15,18,45,48], which bifurcate the task into event detection and trimmed video captioning; 2) joint models exemplified by [5,6,21,28], which concurrently optimize detection and generation tasks by exploiting cross-modal alignment and events connection. While the workflow of DVC and AD automation shares similarities, the latter is more challenging due to the fact that AD automation necessitates capturing significantly longer contextual information from movies, in contrast to the average video length of 150 seconds in the Activity Net Captions [18], which serves as a de facto benchmark for DVC. Unlike video captions, AD automation engages in auditory story understanding, imposing supplementary requirements for both location and content considerations.

2.2. Long-form Video Understanding with S4

As mentioned above, the AD automation needs to capture the long-term dependencies from movies. In the long-form video understanding, there are two major challenges concluded from the previous researches [4,17,41,46,50,51]: efficiency and effectiveness. Efficiency issue comes from the large memory and computational cost of long input while the effectiveness challenge represents how to learn discriminative feature from redundant video sequences. To tackle these challenges, Gu et al. [10] proposed a structured state-space sequence model, a novel alternative to CNNs and transformers, which models the long-range dependencies by simulating a linear time invariant (LTI) system. Subsequently, S4ND [29] and ViS4mer [17] extend S4 model to the video classification task. Finally, Wang et al. [47] further improve the efficiency of S4 model with additional selective module formulating the S5 model. In contrast to previous applications of the S4 model that leverage it to model long sequential inputs, we uniquely employ the S4 module as a visual enhancement encoder. Our approach aims to distill longer-term memory into visual features from shorter clips. Notably, our design differs from LSTCL [44], LSMCL [46], and BraVe [34] by abandoning the dual-encoder with symmetric contrastive learning. This departure significantly improves efficiency and practicality.

2.3. Audio Description Automation

There are initial explorations [13, 14, 37, 39] that try to generate AD and predict AD locations, but the current solutions are still far from being practically useful to scale up the AD automation. Starting from the data curation, LSMDC [36], M-VAD [42], QuerYD [30], and MPII-MD [35] gather linguistic information from movies to curate clip-level video captioning task. The size of these datasets are either small-scaled [35, 36, 42] or not from the

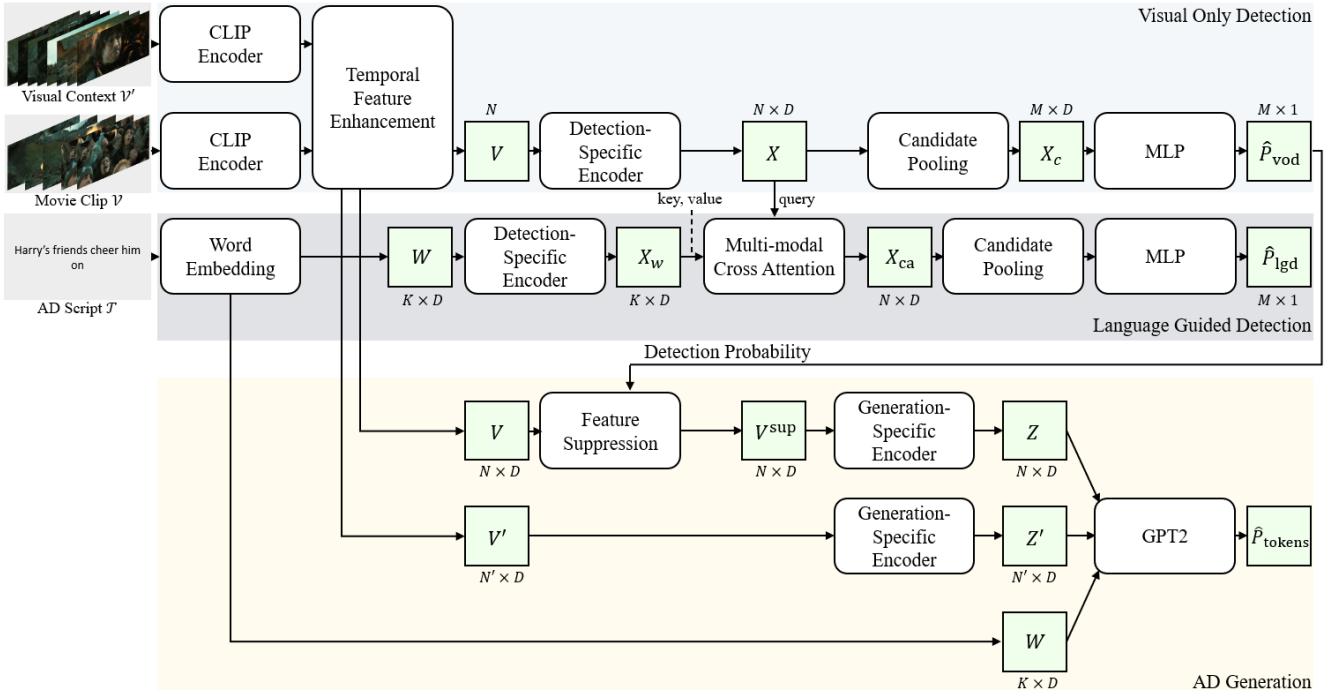


Figure 2. An overview of CA^{3}D network. The upper part shows the detector architecture and the bottom part depicts the generator architecture.

cinematic data [30]. To improve these, Soldan et al. [39] propose the MAD dataset which is a large-scaled benchmark with cinematic content on visual grounding task. Based on which, Han et al. [14] propose a cleaner version, named MADv2 dataset, and introduce AutoAD for AD generation on trimmed AD events. Followed by AutoAD [14], AutoAD II [13] includes a new temporal segment proposing module indicating whether or not a AD should be generated within a speech gap. In addition, this work also improves the generation part with a Flamingo-style [1] architecture to generate better AD scripts. However, prior works decouple the AD detection and generation as two separate tasks and each task is simplified. For example, AutoAD II [13] only provides a binary decision within a speech gap and the duration of the gap is prefixed based on the statistic of MADv2 dataset [39]. Moreover, both AutoAD [14] and AutoAD II [13] generate AD scripts on the ground truth AD locations, which is impractical in the real world scenarios. Thus, we propose an unified AD automation system in this paper, that can automatically detect and generate AD on cinematic data at scale.

3. Method

3.1. Problem Definition

Given a movie clip $\mathcal{V} = \{I_1, I_2, \ldots, I_N\}$ with N consecutive frames and its visual context $\mathcal{V}'$, the proposed CA^{3}D first enriches $\mathcal{V}$ with $\mathcal{V}'$. From $\mathcal{V}$ and $\mathcal{V}'$, we obtain the enhanced visual feature V by using the image encoder and the feature enhancement module. Then, V is sent to the AD detector[2] to predict $\{y_1, y_2, \ldots, y_N\}$, where $y_i \in \{0, 1\}$ indicates whether the i-th frame I_i belongs to AD events. Followed by the detection results, a feature suppression module is applied to extract AD related representation: V_{sup} from V. Lastly, the AD script $\mathcal{T} = \{t_1, t_2, \ldots t_K\}$ of K words is generated, which describes the content in the way that people can enjoy the movie by hearing the Text To Speech (TTS)[3] of it.

3.2. Temporal Feature Enhancement Module

As observed in previous researches [13, 14], both the location and content of AD events adhere to the story line, necessitating the model to accurately grasp long-term reasoning. Therefore, conventional algorithms [13, 14] leverage meta data, such as subtitle or character bank, to exploit context information in the AD automation. However, such information may not be available for all movies which makes it difficult to scale up. On the other hand, recent study on the S4 model [11, 17] has shown its superior performance in modeling long-form video, with the linear complexity to the input length. Compared to the transformers and RNNs, S4 model can capture longer history with cheaper cost.

[2]We assume only one AD event within $\mathcal{V}$ during the training.

[3]Please note the TTS generation is out of scope of this work. This work only focuses on AD detection and generation.

Preliminaries – S4 Model: We start from the state-space model, *i.e.*, a linear time invariant system, which can be written as:

$$\begin{aligned} x'(t) &= Ax(t) + Bu(t) \\ y(t) &= Dx(t) + Eu(t). \end{aligned} \tag{1}$$

This formulation simply projects an input signal $u(t)$ from one-dimensional space to an N-dimensional latent space $x(t)$, which is then mapped back to a one-dimensional output signal $y(t)$. To implement (1) with discrete inputs like image/word tokens, it can be discredited by using a learnable step size Δ [12], which can be rewritten as:

$$\begin{aligned} x_k &= \bar{A}x_{k-1} + \bar{B}u_k \\ y_k &= \bar{D}x_k, \end{aligned} \tag{2}$$

where $\bar{A} = (I + \frac{\Delta \cdot A}{2})/(I - \frac{\Delta \cdot A}{2})$, $\bar{B} = \Delta \cdot B/(I - \frac{\Delta \cdot A}{2})$, $\bar{D} = D$ and E can be replaced by residual connection. Furthermore, (2) can be solved using a discrete convolution [10]:

$$y = \bar{\mathcal{K}} \circledast \mathcal{U}, \tag{3}$$

where $\mathcal{U} = \{u_0, u_1, \ldots, u_{k-1}, u_k\}$ and $\bar{\mathcal{K}} = \{\bar{D}\bar{B}, \bar{D}\bar{A}\bar{B}, \ldots, \bar{D}\bar{A}^{\mathrm{L}-1}\bar{B}\}$ is a discredited convolutional kernel and L is the sequence length.

It is found in [10] that $\bar{\mathcal{K}}$ can become a closed-form expression if the matrix A becomes diagonal and low-rank (structured by the HiPPO theory [9]). As a result, (3) is linear to the input length and can be efficiently computed using fast Fourier transform (FFT) and inverse FFT, without multiplying the matrix A by $L-1$ times[4]. This advantage shapes the S4 model as an efficient architecture capturing long temporal dependencies.

Context-aware Temporal Feature Enhancement: Since AD involves storytelling, the context of the movie is crucial for both AD detection and generation; For instance, to generate the AD script for 'Dooku fires *again*' in Star Wars, it is essential to understand Dooku's actions in the preceding scenes. Therefore, we incorporate visual context, represented by the movie clip $\mathcal{V}' = \{I'_1, \ldots, I'_{N'}\}$ with N' frames, immediately preceding the movie clip $\mathcal{V}$. Subsequently, we enhance the features from both $\mathcal{V}'$ and $\mathcal{V}$ by conveying context information through the feature enhancement module h_{S4}, which comprises S4 layers:

$$[V', V] = h_{\mathrm{S4}}(h_{\mathrm{CLIP}}(\mathcal{V}'), h_{\mathrm{CLIP}}(\mathcal{V})), \tag{4}$$

Here, h_{CLIP} represents the CLIP image encoder [32][5]. $V' = [v'_1, v'_2, \ldots, v'_{N'}] \in \mathbb{R}^{N' \times D}$ and $V = [v_1, v_2, \ldots, v_N] \in \mathbb{R}^{N \times D}$ are the enhanced feature maps of the visual context $\mathcal{V}'$ and movie clip $\mathcal{V}$. Leveraging the robust temporal capacity of the S4 model [10, 11, 17], the proposed feature enhancement module h_{S4} distills long-term dependencies into the subsequent visual features. Consequently, the enhanced movie clip feature V (length of N) encapsulates visual context information from $N + N'$ frames. In Section 4, we demonstrate the superior performance of the temporally enhanced feature map.

[4]Please refer to [9] for more details and relevant proofs.

[5]We opt for the CLIP image encoder because the MADv2 [39] dataset provides only frame-level CLIP-encoded features.

3.3. AD Detection Module

We first propose the visual-only detection (VOD) scheme in the upper part of Figure 2, which solely utilizes the enhanced visual features V for AD event detection. It's worth noting that V is also employed for AD script generation. We use a detection-specific encoder h_{det}, comprising S4 layers [11], to efficiently derive feature representations for AD event detection. This can be expressed as:

$$X = [x_1, x_2, \ldots, x_N] = h_{\mathrm{det}}(v_1, v_2, \ldots, v_N), \tag{5}$$

where $X \in \mathbb{R}^{N \times D}$ is the sequence of the detection-specific features $x_i \in \mathbb{R}^D$ for $i \in \{1, 2, \ldots, N\}$.

Anchor-based detection: To facilitate the continuous prediction of AD frames with soft probability assignment, we employ the anchor-based detection framework. This framework casts the problem as a classification task over all potential AD event locations in $\mathcal{V}$. Specifically, during training, we assume there is only one AD event in the clip $\mathcal{V}$ with N frames. Then, the number of possible locations of an AD event with length l is $N - l + 1$, and the total number of all possible AD events is $M = \sum_{l=1,\ldots,N}(N - l + 1) = \frac{N(N+1)}{2}$. We define the AD event candidate c_m for $m \in \{1, 2, \ldots, M\}$ as:

$$C = [c_1, c_2, \ldots, c_M]^\top = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix} \in \mathbb{Z}^{M \times N}, \tag{6}$$

where 1 indicates that the corresponding frame belongs to an AD event and 0 represents the opposite.

Then, we obtain the candidate-wise features by aggregating the features at AD event locations of each candidate via candidate pooling, which is defined as:

$$X_c = [x^c_1, x^c_2, \ldots, x^c_M] = \bar{C}X \in \mathbb{R}^{M \times D}, \tag{7}$$

Here, $\bar{C} = [\bar{c}_1, \bar{c}_2, \ldots, \bar{c}_M]^\top$ is the normalized candidate from C where $\bar{c}_m = \frac{c_m}{c_m^\top c_M}$. Note that c_M is the vector of 1's. The probability that each candidate matches the ground-truth (GT) AD event $[y_1, y_2, \ldots, y_N]$ is obtained by

$$\hat{P}_{\mathrm{vod}} = [\hat{p}^{\mathrm{vod}}_1, \hat{p}^{\mathrm{vod}}_2, \ldots, \hat{p}^{\mathrm{vod}}_M] = h_{\mathrm{MLP}}(X_c), \tag{8}$$

where h_{MLP} is a multi-layer perceptron (MLP). For event candidate c_m, $\hat{p}^{\mathrm{vod}}_m$ informs the probability that it is the AD event.

3.4. Feature Suppression and Generation Module

In most cases, only a portion of frames in a movie clip $\mathcal{V}$ belong to an AD event, while others do not. For reliable AD script generation, we suppress the information of frames that are not part of the AD event, as they may provide unnecessary information to the script generator.

Feature Suppression: To achieve this, we employ the feature suppression module to obtain discriminative visual features based on the AD detection results. Let $\theta \subset \{1, 2, \ldots, M\}$ be the set of candidate indices i, where $\hat{p}_i^{\text{vod}}$ represents the top k probabilities in $\hat{P}_{\text{vod}}$. Note that $|\theta| = k$. Then, from V, we obtain the suppressed features V_{sup} by:

$$V_{\text{sup}} = \frac{1}{|\theta|} \sum_{i \in \theta} \hat{p}_i^{\text{vod}} c_i \mathbf{1}^\top \odot V, \tag{9}$$

where $\mathbf{1}$ denotes a D-dimensional vector of ones and $\odot$ is the Hadamard product operator. Hence, it filters out the features corresponding to the frames which are estimated as the non AD event.

Script Generation: For script generation, we follow the basic design of AutoAD [14]. However, we employ visual context information for more reliable AD script generation. Similar to (5), we first obtain the generation-specific context feature $Z' = f_{V'}(V') \in \mathbb{R}^{N' \times D}$ from the enhanced visual context V'. Also, we map the suppressed visual features V_{sup} to generation-specific features $Z = f_V(V_{\text{sup}}) \in \mathbb{R}^{N \times D}$. We note that $f_{V'}$ and f_V are generation-specific encoders for visual context and visual features, respectively. Both encoders consist of S4 layers. Then, from Z' and Z, the frozen GPT2 f_{GPT2} generates the probability for each of K tokens in $\mathcal{T}$ as

$$\hat{P}_{\text{tokens}} = [\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_K] = f_{\text{GPT2}}(Z', Z). \tag{10}$$

Note that we use the word embeddings W as additional input to f_{GPT2} during training.

3.5. Self-Refinement Module

Unlike other video events that have distinctive content to detect, such as action and anomalies [20, 26], the boundaries of AD events are naturally ambiguous. Meanwhile, we believe the detection and generation tasks are complementary to each other in the AD automation system. Thus, in addition to the temporally enhanced visual input, we also propose an optional self-refinement module to leverage the generated AD scripts to further improve the reliability of both detection and generation.

To this end, we introduce the language-guided detection (LGD) scheme that leverages the AD script $\mathcal{T}$ along with the visual information in V, providing a complementary cue for AD event detection. In alignment with our AD generation module, we convert the AD script $\mathcal{T}$ into a sequence of GPT-2 [33] word embeddings, $W = [w_1, w_2, \ldots, w_K]$. Similar to (5), we obtain the language features for detection $X_w = [x_1^w, \ldots, x_K^w] = h_{\text{det}}^w(w_1, \ldots, w_K)$ by using another detection-specific encoder h_{det}^w, which has the same architecture as h_{det}. Then, we use the cross-attention module h_{ca} to capture the correlation between language features (serving as keys and values) and visual features (serving as queries) by:

$$X_{\text{ca}} = h_{\text{ca}}(X, X_w). \tag{11}$$

Here, X_{ca} denotes the cross-attended features. Similarly in visual only detection scheme, we predict the detection probability by using X_{ca} as input to (7) and (8). Note that, in language guided scheme, we use the GT AD script $\mathcal{T}$ and the generated one $\hat{\mathcal{T}}$ for training and inference, respectively.

Iterative Refinement: Given an entire movie $\mathcal{M}$, the movie clips $\{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_T\}$ and the corresponding visual contexts $\{\mathcal{V}'_1, \mathcal{V}'_2, \ldots, \mathcal{V}'_T\}$ are sampled using a sliding window approach, progressing from the beginning to the end of the movie with a specified step size of S. For each $\mathcal{V}_i$ and $\mathcal{V}'_i$, CA^{3}D iteratively refines the detection and generation results. In the initial iteration, CA^{3}D utilizes the visual detection scheme to predict the location of AD events. Subsequently, it generates the AD script for $\mathcal{V}_i$ based on the obtained detection results. Starting from the second iteration, we employ the language guided detection scheme using the AD script generated in the previous iteration to achieve more accurate AD event localization. Additionally, we generate the AD script based on the refined AD event location. This refinement process is iteratively repeated for a predefined number of iterations. Algorithm 1 provides a detailed description of the evaluation process.

3.6. Objective Function

We define the training loss on the detector outputs as

$$\ell_{\text{det}} = \ell_{\text{focal}}(\hat{P}_{\text{vod}}, P_{\text{vod}}) + \ell_{\text{focal}}(\hat{P}_{\text{lgd}}, P_{\text{lgd}}) \tag{12}$$

where $\hat{P}_{\text{vod}}$ denotes the detection probabilities of visual only detection and $\hat{P}_{\text{lgd}}$ denotes those of language guided detection. Also, P_{vod} and P_{lgd} are their GT probabilities and ℓ_{focal} is the focal loss [24] over binary classes.

The training loss on the generator outputs is defined as

$$\ell_{\text{gen}} = \ell_{\text{ce}}(\hat{P}_{\text{tokens}}, P_{\text{tokens}}). \tag{13}$$

where ℓ_{ce} is the cross-entropy loss. Therefore, the total training loss is defined as $\ell_{\text{total}} = \ell_{\text{det}} + \ell_{\text{gen}}$. It's noteworthy that the parameters of the detector are optimized by ℓ_{gen} as well, owing to the differentiable feature suppression in (9). In other words, the detector is incentivized to identify the precise AD event location for improved script generation.

Algorithm 1 CA^{3}D

Input: Whole movie $\mathcal{M}$

Sample the movie clips $\{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_T\}$ and the visual contexts $\{\mathcal{V}'_1, \mathcal{V}'_2, \ldots, \mathcal{V}'_T\}$ from $\mathcal{M}$;
for all $i \in \{1, 2, \ldots, T\}$ **do**
  Obtain V and V' from $\mathcal{V}_i$ and $\mathcal{V}'_i$ via (4);
  for all $j \in \{1, \ldots, J\}$ **do**
    if $j = 1$ **then**
      Obtain X_c from V via (5) and (7);
      Obtain $\hat{P}^{ij}_{\text{vod}}$ from X_c via (8); ▷ *VOD*
      Obtain V_{sup} based on $\hat{P}^{ij}_{\text{vod}}$ via (9);
      Generate $\hat{\mathcal{T}}^j_i$ from Z and Z'; ▷ *Generation*
      $\mathcal{D} \leftarrow \hat{P}^{ij}_{\text{vod}}$; $\mathcal{G} \leftarrow \hat{\mathcal{T}}^j_i$;
    else if $j > 1$ **then** ▷ *Iterative refinement*
      Obtain X_w from $\hat{\mathcal{T}}^{j-1}_i$;
      Obtain X_c from X and X_w via (11) and (7);
      Obtain $\hat{P}^{ij}_{\text{lgd}}$ from X_c via (8); ▷ *LGD*
      Obtain V_{sup} based on $\hat{P}^{ij}_{\text{lgd}}$ via (9);
      Generate $\hat{\mathcal{T}}^j_i$ from Z and Z'; ▷ *Generation*
      $\mathcal{D} \leftarrow \hat{P}^{ij}_{\text{lgd}}$; $\mathcal{G} \leftarrow \mathcal{T}^j_i$;
    end if
  end for
end for

Output: Detection results $\mathcal{D}$, Generation results $\mathcal{G}$

4. Experiments

4.1. Experimental Setup

Dataset: We assess the performance of our model on widely used AD benchmarks. Specifically, **MADv2 [14]** consists of 498 movies, with 488 designated for training and 10 for evaluation. We note that the evaluation split of the MADv2 dataset is identical to the MAD-eval dataset [14]. The dataset comprises pre-extracted frame-wise CLIP features [31] and AD scripts with corresponding timestamps. Additionally, an anonymized version is available, where character names are substituted with the placeholder 'someone'. By default, we employ the named version of the MADv2 dataset for both training and evaluation purposes. Moreover, **AudioVault [14]** encompasses 3.3 million AD events derived from scripts and timestamps across 7,000 movies. Consistent with the methodology outlined in [14], we exclusively utilize this dataset for pretraining GPT2.

Implementation Details: During training, we employ the AdamW optimizer [25] with a batch size of 128 and a weight decay of 0. The cosine learning rate scheduler [8] is initialized at 0.0001, and the networks are trained for 10 epochs with a linear warm-up period of 2,000 steps (equivalent to 0.75 epoch). The experiments are conducted using PyTorch [16] on four Tesla V100 GPUs. For text generation, we utilize beam search [19] with a beam size of 5, reporting results based on the top-1 outputs from the beam search by default. Text generation stops upon predicting a full stop mark; otherwise, we limit the sequence length to 67 tokens, as introduced in [14]. Additionally, we set $N = 32$, $N' = 64$, and $K = 36$. Further details can be found in the supplementary materials.

Evaluation Metrics: As the proposed CA^{3}D is the first unified system which tackles both AD detection and generation tasks, we evaluate two tasks separately to demonstrate the effectiveness. To assess the performance of AD event detection, we compute *Precision*, *Recall*, and *F1* scores. An identified AD event throughout the entire movie is deemed correct if its intersection-over-union (IoU) ratio with the ground truth surpasses 0.1. For the evaluation of AD script generation, we employ Rouge-L [22] and CIDEr scores [43] metrics, consistent with previous works [13, 14]. For each GT AD script, we compute both metrics with the generated AD script whose detected temporal location is closest to it.

Table 1. Comparison of AD detection results on MADv2 dataset. Here, 'V' and 'AD' denote the visual inputs and previous AD context inputs, respectively.

Algorithm	Inputs	Precision	Recall	F1
Random	V	29.3	43.4	35.1
TriDet [38]	V	52.1	76.5	61.9
MIGCN [52]	V + AD	69.8	69.8	69.8
LGI [27]	V + AD	**71.4**	71.4	**71.4**
CA^{3}D	V	55.2	**80.1**	65.3

4.2. Main Results

AD Event Detection: Table 1 presents the performance of AD event detection on the MADv2 dataset. For comparison, we include scores of random estimation as the lower bounds while the performance of LGI [27] and MIGCN [52] as the upper bounds performance. Specifically, LGI [27] and MIGCN are the state-of-the-art methods for the Natural Language Video Grounding (NLVG) task [2, 7, 40, 49], which aims to accurately locate the video moment semantically corresponding to a specific linguistic query. In Table 1, they are evaluated using ground-truth AD locations and scripts. As LGI and MIGCN localizes one AD event per linguistic query, the values for *False Positive* and *False Negative* are identical. So the value of *Precision*, *Recall* and *F1* for LGI [27] are the same. Besides, we also include TriDet [38] as one baseline which is the state-of-the-art method for action detection.

In Table 1, the proposed CA^{3}D demonstrates superior performances compared to TriDet [38] across all metrics. Furthermore, it achieves competitive results with LGI [27] and MIGCN [52], even though LGI [27] benefits from access to subsequent and preceding AD ground-truth scripts and visual information. It's noteworthy that LGI [27]

Table 2. Comparison of AD generation results on the named version of MADv2 dataset.

Algorithm	Pretrain	Inputs	AD location	Rouge-L	CIDEr
SwinBERT [23]	AudioVault	V	✓	8.5	6.7
AutoAD [14]	WebVid + AudioVault	V	✓	9.9	10.0
AutoAD II [13]	WebVid + AudioVault	V	✓	9.7	10.0
CA^3D	AudioVault	V	✓	**11.3**	**10.8**
CA^3D	AudioVault	V		**11.3**	9.4
AutoAD [14]	WebVid + AudioVault	V + AD	✓	13.9	19.0
AutoAD II [13]	WebVid + AudioVault	V + Char.	✓	13.1	19.2
CA^3D	AudioVault	V + AD	✓	**14.0**	**20.4**
CA^3D	AudioVault	V + AD		13.5	17.7

Table 3. Comparison of AD generation results on the unnamed version of MADv2 dataset.

Algorithm	Inputs	AD location	Rouge-L	CIDEr
AutoAD [14]	V + AD	✓	15.9	14.5
CA^3D	V + AD	✓	**16.2**	**14.8**
CA^3D	V + AD		15.8	13.9

achieves an F1 score of 71.4, underscoring the challenging nature of the AD detection task even with access to ground truth AD scripts. Importantly, CA^3D outperforms LGI [27] in *Recall* by 8.7%. While the *Precision* may be lower than the upper bound, a high *Recall* is advantageous in the context of AD automation. This is because *False Positive* detection can be addressed during post-processing, while *False Negatives* are challenging to rediscover in long movies.

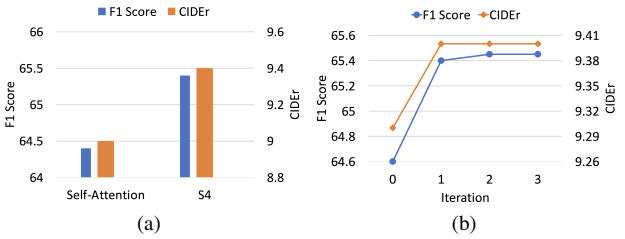
AD Script Generation: Table 2 presents a comparison of the performance of AD script generation on the named version of the MADv2 dataset [13]. We include AutoAD [14], AutoAD II [13] and SwinBERT [23] as baselines which are the state-of-the-art methods in AD generation and video captioning. Please note they generate AD scripts by using the ground truth AD locations. When all methods are provided with the oracle AD location, the proposed algorithm exhibits consistent better performance for both visual-only input and visual plus linguistic context (either AD context or character name). Remarkably, CA^3D achieves a CIDEr score 1.2 higher than AutoAD II, which leverages additional metadata and employs an additional network for character name recognition. Moreover, CA^3D still maintains competitive performances with AutoAD and AutoAD II when generating AD scripts within the detected AD event windows (without GT AD location). Additionally, Table 3 provides results on the unnamed version of the MADv2 dataset, where CA^3D achieves the best scores across all metrics. The lower performances of TriDet and SwinBERT suggest that the simple adoption of SOTA methods in other applications is not practically useful in solving AD automation. Notably, unlike AutoAD and AutoAD II, CA^3D is **not** pretrained on the **WebVid** dataset [3], which contains **2.5M** short video-text pairs. The proposed algorithm not only establishes new state-of-the-art performance but also demonstrates significant data efficiency.

Table 4. Ablation studies for modules in the detector on the MADv2 dataset.

Method	Enhancement	Anchor	Precision	Recall	F1
I	✓		51.9	77.2	62.1
II		✓	53.3	79.4	64.5
III	✓	✓	54.4	82.2	65.4

Table 5. Ablation studies for modules in the generator on the MADv2 dataset.

Method	Enhancement	Suppression	Rouge-L	CIDEr
I	✓		10.8	8.9
II		✓	11.0	9.2
III	✓	✓	11.3	9.4



(a) (b)

Figure 3. (a) Comparison of AD generation and detection performances at different iterations, (b) Comparison of AD generation and detection performances at different iterations.

4.3. Ablation Study

Context-aware Temporal Feature Enhancement: In Tables 4 and 5, we assess the effectiveness of the proposed feature enhancement module. In this work, we advocate the use of a simple S4 module to integrate longer temporal cues into visual features with concise content, which is more efficient than previous architecture. From Tables 4 and 5, it is evident that the feature enhancement module enhances the performance of CA^3D in both detection and generation tasks. Figure 3a shows that the proposed feature enhancement module achieves better performances with S4 layers than standard self-attention layers. We note that the computational complexity of S4 is $\mathcal{O}(n)$ while the one of transformer layer is $\mathcal{O}(n^2)$, where n is the sequence length. The linear complexity of S4 enables the success of our proposed temporal feature enhancement module in the long-form video domain.

Anchor-based Prediction Head: In Table 4, we conduct an ablation study on the proposed anchor-based prediction head in the AD event detector. Method I does not utilize AD event candidates and directly predicts whether each frame belongs to an AD event from frame-wise features. This can be considered a segmentation-based AD detection baseline.

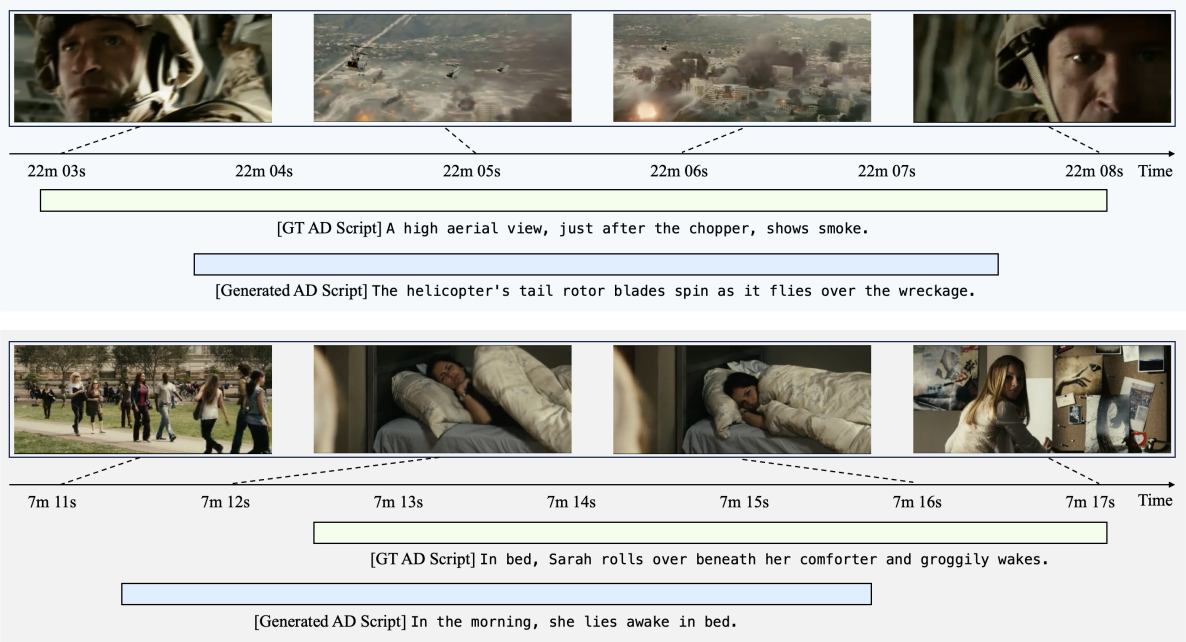


Figure 4. Examples of AD detection and generation results on the MADv2 test set.

By comparing Method I and Method III in Table 4, it is evident that the proposed anchor-based prediction head significantly improves the detection results, with improvements of $+2.5$, $+5.0$, and $+3.3$ in *Precision*, *Recall*, and *F1*.

Feature Suppression: In Table 5, we conduct an ablation study on the proposed feature suppression module in the AD script generator. Method I does not utilize the feature suppression module, using V instead of V_{sup} in Algorithm 1 as the baseline model. By comparing Method I and Method III in Table 5, it is evident that the feature suppression module brings clear benefits, improving both Rouge-L and CIDEr by 0.5. This result shows that it is crucial to eliminate visual information not corresponding to the AD events.

Self-refinement: Figure 3b outlines the detection and generation scores on the MADv2 dataset at each iteration. We employ the visual detection scheme for the initial prediction (iteration 0) and then use the language-guided detection in later iterations. The detection performance improves as the iteration goes on, indicating that the generated scripts contribute to enhanced AD event detection. Moreover, the proposed algorithm achieves higher generation scores when opt-in the self-refinement module which enables more accurate detection results. The performance saturates after one iteration, which suggests the effectiveness of the refinement module. These results underscore the complementary nature of AD event detection and AD script generation within the proposed automation system.

4.4. Visualizations

Figure 4 shows the examples of AD detection and generation results on the MADv2 test set. CA^{3}D yields satisfactory outcomes for both detection and generation tasks. We note that the determination of AD event locations is subjective and somewhat ambiguous, as different individuals may prefer different locations (e.g., commencing from the beginning or the middle of a scene). Moreover, within the MADv2 dataset, some AD event timestamps are not accurate, as illustrated in the upper part of Figure 4. Even though the detected AD event locations may not align perfectly with the ground truth, CA^{3}D provides useful results which are precise enough to facilitate the AD process. Similarly, while the generated AD scripts may not exactly match the GT AD scripts, they describe the scenes properly.

5. Conclusion

In this paper, we tackle the AD automation through an unified system, CA^{3}D, which identifies the AD events and simultaneously generates the corresponding AD scripts on cinematic data. CA^{3}D involves a novel temporal enhancement module to first expand the temporal horizon of the input, an anchor-based AD detector working seamlessly with the feature suppression module to extract discriminative representation and a self-refinement module to further boost the performance. CA^{3}D operates directly on long-form videos, establishing the new state-of-the-art performance in the AD automation.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 3

[2] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pages 5803–5812, 2017. 6

[3] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738, 2021. 7

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 813–824. PMLR, 2021. 2

[5] Shaoxiang Chen and Yu-Gang Jiang. Towards bridging event captioner and sentence localizer for weakly supervised dense event captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8425–8435, 2021. 2

[6] Chaorui Deng, Shizhe Chen, Da Chen, Yuan He, and Qi Wu. Sketch, ground, and refine: Top-down dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 234–243, 2021. 2

[7] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, pages 5267–5275, 2017. 6

[8] Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. *arXiv preprint arXiv:1810.13243*, 2018. 6

[9] Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. *Advances in Neural Information Processing Systems*, 33:1474–1487, 2020. 4

[10] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021. 2, 4

[11] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *ICLR*, 2022. 3, 4

[12] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021. 4

[13] Tengda Han, Max Bain, Arsha Nagrani, Gul Varol, Weidi Xie, and Andrew Zisserman. AutoAD II: The sequel-who, when, and what in movie audio description. In *ICCV*, 2023. 1, 2, 3, 6, 7

[14] Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. AutoAD: Movie description in context. In *CVPR*, 2023. 1, 2, 3, 5, 6, 7

[15] Vladimir Iashin and Esa Rahtu. Multi-modal dense video captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 958–959, 2020. 1, 2

[16] Sagar Imambi, Kolla Bhanu Prakash, and GR Kanagachidambaresan. Pytorch. *Programming with TensorFlow: Solution for Edge Computing Applications*, pages 87–104, 2021. 6

[17] Md Mohaiminul Islam and Gedas Bertasius. Long movie clip classification with state-space video models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2, 3, 4

[18] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715, 2017. 1, 2

[19] Abhishek Kumar, Shankar Vembu, Aditya Krishna Menon, and Charles Elkan. Beam search algorithms for multilabel learning. *Machine learning*, 92:65–89, 2013. 6

[20] Kuan-Ting Lai, Felix X Yu, Ming-Syan Chen, and Shih-Fu Chang. Video event detection by inferring temporal instance labels. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 2243–2250, 2014. 5

[21] Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei. Jointly localizing and describing events for dense video captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7492–7500, 2018. 2

[22] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 6

[23] Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang. SwinBERT: End-to-end transformers with sparse attention for video captioning. In *CVPR*, 2022. 7

[24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 5

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[26] Gérard Medioni, Isaac Cohen, François Brémond, Somboon Hongeng, and Ramakant Nevatia. Event detection and analysis from video streams. *IEEE Transactions on pattern analysis and machine intelligence*, 23(8):873–889, 2001. 5

[27] Jonghwan Mun, Minsu Cho, and Bohyung Han. Local-global video-text interactions for temporal grounding. In *CVPR*, 2020. 6, 7

[28] Jonghwan Mun, Linjie Yang, Zhou Ren, Ning Xu, and Bohyung Han. Streamlined dense video captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6588–6597, 2019. 2

[29] Eric Nguyen, Karan Goel, Albert Gu, Gordon W Downs, Preey Shah, Tri Dao, Stephen A Baccus, and Christopher Ré. S4nd: Modeling images and videos as multidimensional signals using state spaces. *Advances in neural information processing systems*, 2022. 2

[30] Andreea-Maria Oncescu, Joao F Henriques, Yang Liu, Andrew Zisserman, and Samuel Albanie. Queryd: A video dataset with high-quality text and audio narrations. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2265–2269. IEEE, 2021. 2, 3

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 6

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. 2021. 4

[33] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 5

[34] Adrià Recasens, Pauline Luc, Jean-Baptiste Alayrac, Luyu Wang, Florian Strub, Corentin Tallec, Mateusz Malinowski, Viorica Patraucean, Florent Altché, Michal Valko, et al. Broaden your views for self-supervised video learning. *arXiv preprint arXiv:2103.16559*, 2021. 2

[35] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3202–3212, 2015. 2

[36] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *International Journal of Computer Vision*, 123:94–120, 2017. 2

[37] Xuyang Shen, Dong Li, Jinxing Zhou, Zhen Qin, Bowen He, Xiaodong Han, Aixuan Li, Yuchao Dai, Lingpeng Kong, Meng Wang, et al. Fine-grained audible video description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10585–10596, 2023. 2

[38] Dingfeng Shi, Yujie Zhong, Qiong Cao, Lin Ma, Jia Li, and Dacheng Tao. TriDet: Temporal action detection with relative boundary modeling. In *CVPR*, 2023. 6

[39] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5026–5035, 2022. 2, 3, 4

[40] Xin Sun, Xuan Wang, Jialin Gao, Qiong Liu, and Xi Zhou. You need to read again: Multi-granularity perception network for moment retrieval in videos. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1022–1032, 2022. 6

[41] Yuchong Sun, Bei Liu, Hongwei Xue, Ruihua Sone, Huan Yang, and Jianlong Fu. Long-form video-language pre-training with multimodal temporal contrastive learning. *Advances in neural information processing systems*, 2022. 2

[42] Atousa Torabi, Christopher Pal, Hugo Larochelle, and Aaron Courville. Using descriptive video services to create a large data source for video annotation research. *arXiv preprint arXiv:1503.01070*, 2015. 2

[43] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015. 6

[44] Jue Wang, Gedas Bertasius, Du Tran, and Lorenzo Torresani. Long-short temporal contrastive learning of video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14010–14020, 2022. 2

[45] Jingwen Wang, Wenhao Jiang, Lin Ma, Wei Liu, and Yong Xu. Bidirectional attentive fusion with context gating for dense video captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7190–7198, 2018. 1, 2

[46] Jue Wang and Lorenzo Torresani. Deformable video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14053–14062, 2022. 2

[47] Jue Wang, Wentao Zhu, Pichao Wang, Xiang Yu, Linda Liu, Mohamed Omar, and Raffay Hamid. Selective structured state-spaces for long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6387–6397, 2023. 2

[48] 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*, 31(5):1890–1900, 2020. 1, 2

[49] Zhenzhi Wang, Limin Wang, Tao Wu, Tianhao Li, and Gangshan Wu. Negative sample matters: A renaissance of metric learning for temporal grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2613–2623, 2022. 6

[50] Chao-Yuan Wu and Philipp Krähenbühl. Towards Long-Form Video Understanding. In *CVPR*, 2021. 2

[51] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13587–13597, 2022. 2

[52] Zongmeng Zhang, Xianjing Han, Xuemeng Song, Yan Yan, and Liqiang Nie. Multi-modal interaction graph convolutional network for temporal language localization in videos. *IEEE TIP*, 30:8265–8277, 2021. 6