AutoAD: Movie Description in Context

Tengda Han[1*] Max Bain[1*] Arsha Nagrani[1†] Gül Varol[1,2] Weidi Xie[1,3] Andrew Zisserman[1]
[1]Visual Geometry Group, University of Oxford
[2]LIGM, École des Ponts, Univ Gustave Eiffel, CNRS [3]CMIC, Shanghai Jiao Tong University
https://www.robots.ox.ac.uk/vgg/research/autoad/

Abstract

The objective of this paper is an automatic Audio Description (AD) model that ingests movies and outputs AD in text form. Generating high-quality movie AD is challenging due to the dependency of the descriptions on context, and the limited amount of training data available. In this work, we leverage the power of pretrained foundation models, such as GPT and CLIP, and only train a mapping network that bridges the two models for visually-conditioned text generation. In order to obtain high-quality AD, we make the following four contributions: (i) we incorporate context from the movie clip, AD from previous clips, as well as the subtitles; (ii) we address the lack of training data by pretraining on large-scale datasets, where visual or contextual information is unavailable, e.g. text-only AD without movies or visual captioning datasets without context; (iii) we improve on the currently available AD datasets, by removing label noise in the MAD dataset, and adding character naming information; and (iv) we obtain strong results on the movie AD task compared with previous methods.

1. Introduction

That of all the arts, the most important for us is the cinema.
Vladimir Lenin

One of the long-term aims of computer vision is to understand long-form feature films. There has been steady progress towards this aim with the identification of characters by their face and voice [12, 15, 25, 29, 79], the recognition of their actions and inter-actions [38, 50, 60, 85], of their relationships [37], and 3D pose [61]. However, this is still a long way away from story understanding. Movie *Audio Description (AD)*, the narration describing visual elements in movies, provides a means to evaluate current movie understanding capabilities. AD was developed to aid visually impaired audiences, and is typically generated by experienced annotators. The amount of AD on the internet is growing due to more societal support for visually impaired communities and its inclusion is becoming an emerging legal requirement.

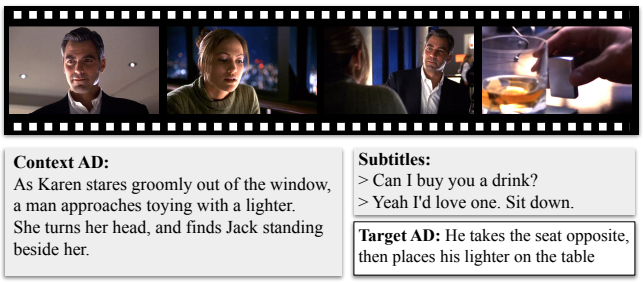


Figure 1. **Movie audio description (AD)** consists of sentences describing movies for the visually impaired. Note how it is heavily influenced by various types of context – the visual frames, the previous AD, and the subtitles of the movie.

AD differs from image or video captioning in several significant respects [67], bringing its own challenges. First, AD provides dense descriptions of important visual elements *over time*. Second, AD is always provided on a separate soundtrack to the original audio track and is highly *complementary* to it. It is complementary in two ways: it does not need to provide descriptions of events that can be understood from the soundtrack alone (such as dialogue and ambient sounds), and it is constrained in time to intervals that do not overlap with the dialogue. Third, unlike dense video captioning, AD aims at *storytelling*; therefore, it typically includes factors like a character's name, emotion, and action descriptions.

In this work, our objective is automatic AD generation – a model that takes continuous movie frames as input and outputs AD in text form. Specifically, we generate text given a temporal interval of an AD, and evaluate its quality by comparing with the ground-truth AD. This is a relatively unexplored task in the vision community with previous work targeting ActivityNet videos [88], a very different domain to long-term feature films with storylines, and the LSMDC challenge [68], where the descriptions and character names are treated separately.

As usual, one of the challenges holding back progress

∗: equal contribution. †: also at Google Research

is the lack of suitable training data. Paired image-text or video-text data that is available at scale, such as alt-text [63,72] or stock footage with captions [7], does not generalize well to the movie domain [8]. However, collecting high-quality data for movie understanding is also difficult. Researchers have tried to hire human annotators to describe video clips [21, 36, 90] but this does not scale well. Movie scripts, books and plots have also been used as learning signals [12, 75, 97] but they do not ground on vision closely and are limited in number.

In this paper we address the AD and training data challenges by – Spoiler Alert – developing a model that uses temporal context together with a visually conditioned generative language model, while providing new and cleaner sources of training data. To achieve this, we leverage the strength of large-scale language models (LLMs), like GPT [64], and vision-language models, like CLIP [63], and integrate them into a video captioning pipeline that can be effectively trained with AD data.

Our contributions are the following: (i) inspired by ClipCap [52] we propose a model that is effectively able to leverage both temporal context (from previously generated AD) and dialogue context (in particular the names of characters) to improve AD generation. This is done by bridging foundation models with lightweight adapters to integrate both types of context; (ii) we address the lack of large-scale training data for AD by pretraining components of our model on partially missing data which are typically available in large quantities e.g. text-only AD without movie frames, or visual captioning datasets without multiple sentences as context; (iii) we propose an automatic pipeline for collecting AD narrations at scale using speaker-based separation; and finally (iv) we show promising results on automatic AD, as seen from both qualitative and quantitative evaluations, and also achieve impressive *zero-shot* results on the LSMDC multi-description benchmark comparable to the finetuned state-of-the-art.

2. Related Works

Image Captioning. Image captioning is a long-standing problem in computer vision [3, 21, 22, 24, 33, 34, 47]. Early pioneering works learn to associate images and words within a limited vocabulary and a set of images [9, 10, 39]. Large-scale image captioning datasets have been collected by scraping images from the internet and their corresponding alt-texts with quality filters as a post-processing [72]. In doing so, strong joint image-text representations can be learned [63], and image captioning from raw pixels, with impressive results [41, 92]. Recent work [52, 56] learns a bridge between strong joint image-text representations (CLIP) and the natural language representation (GPT-2) for image captioning, obtaining promising results that generalise well across domains. In this work, we extend this approach to perform automatic AD from videos.

Video Captioning. Video captioning presents additional challenges due to the lack of quality large-scale video-text data and increased complexity from the temporal axis. Early video caption datasets [19, 90] adopt manual annotations, a far from scalable collection method. ASR (automated speech recognition) from YouTube instructional videos is collected at scale for video-language datasets [51], but contains high levels of noise due to the weak correspondence between the narration and visual content. VideoCC [55] transfers captions from images to videos, but this method is still limited by the existing seed image captioning dataset used. Earlier video captioning models lack generalisation capabilities due to limited training data [59, 84]. Some recent methods [28, 48, 71] train on ASR from the HowTo100M dataset, while others expand image-text representations [78] to multiple frames.

A task more related to AD is that of dense video captioning [35], which involves producing a number of captions and their corresponding grounded timestamps in the video. To enrich inter-task interactions, recent works for this task [18, 20, 23, 44, 54, 65, 73, 74, 86, 87, 96] jointly train both a captioning and localization module. Our task differs in that the captions are: made with the intent to aid storytelling; specific to the movie domain; and complementary to the audio track.

Visual Storytelling. Most similar in vein to the AD task is visual storytelling [30, 42, 66], in which the goal is to generate coherent sentences for a sequence of video clips or images. LSMDC [70] proposes the multi-description task of generating captions for a set of clips from a movie, with character names anonymized. In contrast, movie AD takes as input a continuous long video and describes the visual happenings complementary to the story, characters, dialogue and audio. Most similar to our model is TPAM [93] which prompts a frozen GPT-2 with local visual features. Ours differs in that: (i) it is not restricted to local visual context but rather global by recurrently conditioning on previous outputs; and (ii) we additionally pretrain GPT on in-domain text-only AD data.

Movie Understanding. Previous works investigate storyline understanding by aligning movies to additional data sources such as plots [77, 89], books [80, 97], scripts [58], and YouTube summaries [6]. However, these sources are limited in number and often do not closely relate to the visual elements in the frame. Using existing movie AD as the data source for videos is an emergent direction for movie understanding. LSMDC [68], M-VAD dataset [81] and MPII-MD [69], gather AD and scripts from movies to provide captions for short video clips, several seconds in duration. QuerYD [57] provides high-quality textual descriptions for longer videos by scraping AD from YouDescribe [67], an online community of AD contributors. Re-

cently, the MAD dataset [76] collects movie AD at scale to provide dense textual annotations for movies with a focus on visual grounding task.

Prompt Tuning and Adapters. Originally for language modelling, prompt tuning is a lightweight approach to adapt a pretrained model to perform a downstream task. Early works [16, 32, 40, 43] learn prompt vectors that are shared within the targeted dataset and task. A similar line of works to ours is *visual-conditioned* prompt tuning, in which the prompt vectors are conditioned on the visual inputs. Visual-conditioned prompts are used for adapting pretrained image-language models [4, 31], and for few-shot learning [1, 82]. Training lightweight feature adapters between pretrained vision and text encoders is another approach to adapt pretrained models [26,94]. The adapter layers can also be inserted into the pretrained language model in an interleaved way [91]. Our work adopts prompt tuning in order to condition a language generation model on visual information (frames), and textual context (subtitles and previous AD).

3. Method

Given a long-form movie $\mathcal{V}$ segmented into multiple short clips $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$, our goal is to generate the audio description (AD) in text form for every movie clip. Note that each movie clip is cut from the raw movie based on the timestamp $[t_{\text{start}}, t_{\text{end}}]$ given by the AD annotation. Specifically, for the i-th movie clip consisting of multiple frames $\mathbf{x}_i = \{\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_N\}$, we aim to produce text $\mathcal{T}_i$ that describes the visual elements in such a way that helps the visually impaired follow the storyline. To this purpose, an ideal AD generation system must be able to exploit the full contextual information leading up to the i-th movie clip. One method for this, which we adopt, is to use previous AD $\mathcal{T}_{t<i}$ and subtitles $\mathcal{S}_{t\leq i}$ to generate the text $\mathcal{T}_i$. In the following sections, we first give an overview of our visual captioning pipeline with prompt tuning (Sec. 3.1), followed by our contextual components (Sec. 3.2), and finally the pretraining methods with partial data (Sec. 3.3).

3.1. Visual Captioning with Prompt Tuning

In order to describe our method, we first present the typical pipeline for an image captioning model, and then detail how we extend this to ingest multiple frames and additional text context. Given an image-caption pair $\{\mathcal{I}_i, \mathcal{C}_i\}$, where the caption consists of a sequence of language tokens $\mathcal{C}_i = \{c_1, c_2, ..., c_k\}$, the standard objective of an image captioning model is to generate text tokens $\hat{\mathcal{C}}_i$ that are close to the target $\mathcal{C}_i$. Technically, the captioning models are trained to maximize the joint probability of predicting the ground-truth language tokens, or equivalently minimize the following negative log-likelihood (NLL) loss,

$$\mathcal{L}_{\text{NLL}} = -\log p_\theta(\mathcal{C}_i|\mathbf{h}_{\mathcal{I}_i}) = -\log p_\theta(c_1, c_2, ..., c_k|\mathbf{h}_{\mathcal{I}_i})$$

where θ denotes the parameters of the model, and $\mathbf{h}_{\mathcal{I}_i}$ denotes the extracted image features of $\mathcal{I}_i$. Previous works like ClipCap [52] fit a powerful text generation model and visual encoding model into this image captioning pipeline. Specifically, strong visual encoding models, such as CLIP [63], are used to extract the visual features from the input image $\mathbf{z}_i = f_{\text{CLIP}}(\mathcal{I}_i)$, then a visual mapping network $\mathcal{M}_{\text{V}}$ is trained to map the visual features to 'prompt vectors' that adapt to the text generation model, $\mathbf{h}_{\mathcal{I}_i} = \mathcal{M}_{\text{V}}(\mathbf{z}_i)$. Finally these prompt vectors $\mathbf{h}_{\mathcal{I}_i}$ are fed to a pretrained text generation model, such as GPT [64], for the captioning task. We adapt this visual captioning pipeline, which uses pretrained feature extractor CLIP and langauge model GPT, for movie AD generation and propose key components that support contextual understanding.

3.2. Benefiting from Temporal Context

Here, we describe how we extend this single-frame captioning model to include different forms of context, including multiple frames, previous AD text, and subtitles. Compared to image captioning where the annotation describes 'what is in the image', movie AD describes the visual happenings in the scene that are relevant to the broader story – often centered around events, characters and the interactions between them. Factors like these cannot be accurately described from a static image alone and therefore a successful automatic AD system must utilize the context of prior events and character interactions.

To tackle these temporal dependencies, we propose to include three components to incorporate the essential contextual information from movies: (i) immediate visual context in the current movie clip (multiple frames), (ii) the previous movie AD, and (iii) the movie subtitles. The architecture of our model is shown in Fig. 2.

Multiple frames (immediate visual context). In contrast to the image captioning method, the visual mapping network $\mathcal{M}_{\text{V}}$ takes as input multiple frame features from the current movie clip $\mathbf{x}_i$ rather than a single image feature, and outputs prompt vectors for the movie clip,

$$\mathbf{h}_{\mathbf{x}_i} = \mathcal{M}_{\text{V}}(\{\mathbf{z}_1, ..., \mathbf{z}_N\}); \quad \mathbf{z}_i = f_{\text{CLIP}}(\mathcal{I}_i).$$

In detail, the mapping network consists of a multi-layer transformer encoder that enables modelling temporal relations among multiple frame features, as shown in Fig 2.

Previous AD text. The sequence of events leading up to the present contain contextual information which are crucial for generating AD of current scene that helps the viewer follow the story. We input this contextual knowledge to our model in the form of the past ADs. Specifically, our model takes the past K movie ADs $\{\mathcal{T}_{i-K}, ..., \mathcal{T}_{i-1}\}$ to generate the AD for the current clip. The past movie ADs are a few sentences, which are first concatenated into a single paragraph, then tokenized and converted to a sequence of word embeddings. Inspired by the design of special tokens in language

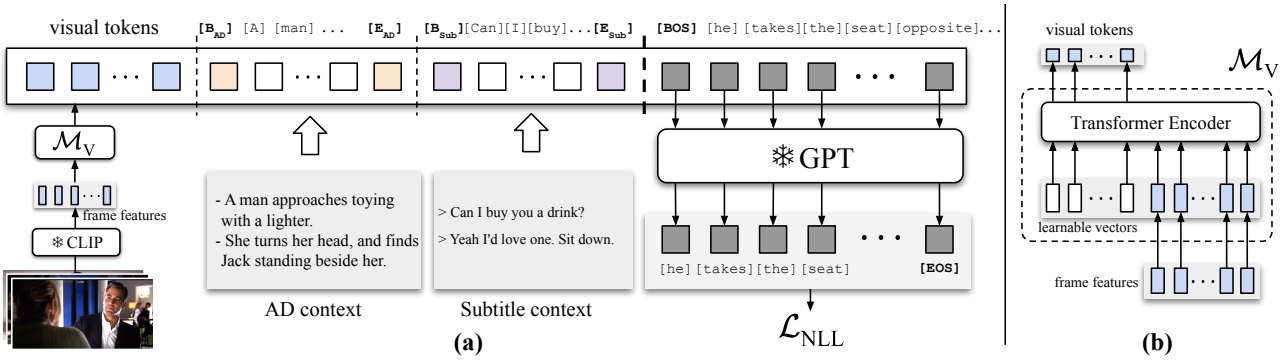


Figure 2. **(a) Overview of AutoAD:** AutoAD consists of a *frozen* visual encoder (CLIP) and a *frozen* LLM (GPT) for generating captions. We introduce a lightweight mapping network to map CLIP features into visual tokens, which are then combined with previous AD context and subtitle context, before being fed into the GPT model. $\mathcal{M}_V$ refers to the visual mapping network, $[B_*]$ and $[E_*]$ denote the learnable special tokens for contextual AD and subtitle sequences. **(b) Detail of the visual mapping network:** A transformer encoder takes as input multiple frame features and outputs a few visual tokens which are further fed to a text generation model.

models, we wrap the context AD embeddings with *learnable* special tokens to indicate the beginning and end of the AD sequence. Formally, the contextual AD embedding is a sequence,

$$\mathbf{h}_{\text{AD}} = [\text{B}_{\text{AD}}; \mathbf{h}_{\mathcal{T}_{i-K}}; ...; \mathbf{h}_{\mathcal{T}_{i-1}}; \text{E}_{\text{AD}}] \quad (1)$$

where B_{AD} and E_{AD} are the learnable special tokens indicating the beginning and end, the symbol ';' denotes concatenation, and $\mathbf{h}_{\mathcal{T}_j} \in \mathbb{R}^{n \times C}$ denotes the word embedding of the j-th movie ADs.

Previous subtitles. Our model also takes the movie subtitles as additional contextual information, which can be sourced either from the official movie metadata or automatically transcribed with an ASR model. The character dialogues, contained with the subtitles, provide complementary information to movie description, including the character names, relationships and emotions. Similar to the context ADs, we concatenate multiple subtitle sentences into a single paragraph and wrap them with learnable special tokens. Practically, since the timing of movie AD does not overlap with the subtitles, we take the most recent L subtitles within a certain time range as the context,

$$\mathbf{h}_{\text{Sub}} = [\text{B}_{\text{Sub}}; \mathbf{h}_{\mathcal{S}_{i-L}}; ...; \mathbf{h}_{\mathcal{S}_{i-1}}; \text{E}_{\text{Sub}}]$$

Due to the weak correlation between the subtitles and the visual elements in the scene, we also experiment with a variant that only encodes the character names occurring in the recent subtitles.

Summary. Overall, the movie AD for the current movie clip $\mathcal{T}_{\mathbf{x}_i}$ is generated by conditioning on all the previously described visual and contextual information using a pretrained GPT. The conditional information is fed to GPT as prompt vectors as shown in Fig. 2. The model is trained with NLL loss,

$$\mathcal{L}_{\text{NLL}} = -\log p_\Theta(\mathcal{T}_{\mathbf{x}_i} | \mathbf{h}_{\mathbf{x}_i}, \mathbf{h}_{\text{AD}}, \mathbf{h}_{\text{Sub}}). \quad (2)$$

During training, we input the ground-truth past AD. During inference, we experiment with two methods to incorporate the past AD: an **oracle** setting where the *ground-truth* past ADs are used in Eq. 2 to generate the current AD, and a **recurrent** setting where the *predicted* past ADs are used instead.

3.3. Pretraining with Partial Data

A major challenge for generating AD is the lack of training data, since the model requires the corresponding visual, textual and contextual data to all be jointly trained. However since our model is modular, components of it can be pretrained with *partial data* – when a certain type of data is missing, the remaining modules can still be trained. We experiment with partial-data pretraining under two settings: visual-only pretraining and AD-only pretraining.

Visual-only Pretraining. In the absence of contextual data, the visual mapping network can be pretrained with abundant image captioning or (short) video captioning datasets. In this case, the context modules (both contextual AD and subtitles) are deactivated. The training objective of Eq. 2 is turned into $\mathcal{L} = -\log p_\Theta(\mathcal{T}_{\mathbf{x}_i} | \mathbf{h}_{\mathbf{x}_i})$ for visual-only pretraining. Note that the language model is kept frozen here since we find image/video captioning datasets have a clear domain gap with movie AD in both the vision and text modalities.

AD-only Pretraining. Movie AD datasets with corresponding visual information (*e.g.* frames or frame features) are limited at scale due to potential copyright issues. However, abundant *text-only* movie ADs are available online as described in Sec. 5. In the absence of visual data, the contextual AD module and the language model can still be pretrained. The training objective in this case becomes $\mathcal{L} = -\log p_\Theta(\mathcal{T}_{\mathbf{x}_i} | \mathbf{h}_{\text{AD}})$, which is similar to training a story completion objective [53] by finetuning GPT on *text-only* movie AD data but with a few additional special tokens. This text-only movie AD pretraining is also related to [27], which shows a second stage of language model pretraining on in-domain data improves downstream performance.

Manual Verification*	She stands and the little warrior takes in her size, about twice his own.	As she steps past him, he defensively grips his spear	Leia sits on a moss covered log.
MAD-v1 [76]	Angola, she stands in the Little Warrior, takes in her size about twice his own.	As she steps past me. Defensively grips his spear.	I'm not gon na. Leah sits on a Moss covered log.
MAD-v2 (ours)	She stands and the little warrior takes in her size about twice his own.	As she steps past him, he defensively grips his spear.	Leia sits on a moss-covered log.

Figure 3. **Qualitative comparison of MAD annotations.** We compare the original MAD-v1 [76] and our proposed MAD-v2. Note MAD-v1's erroneous transcriptions of AD and dialogue leakage (highlighted in red text). The samples are taken from Star Wars VI: Return of the Jedi (1983) [49]. *We verify this example by manually transcribing the AD narration from the audio track.

4. Denoising MAD Dataset

Our main objective is to generate movie audio descriptions. For this goal, the model is trained on the MAD training set [76], a dataset of AD caption-video clip pairs from 488 movies. MAD provides the video data in the form of CLIP visual features in order to avoid copyright restrictions. The AD annotations for each movie are automatically collected from AudioVault[1], a large open-source database of audio files containing the full-length original movie track mixed with the AD narrator's voice. The MAD authors transcribe a subset of this data using ASR, and also have access to the official DVD subtitles. Their automated method then uses *text-based* speaker separation of the transcribed audio by using subtitles to know when dialogue is present, and assuming all other speech is AD.

This however introduces *significant noise* because (i) the outdated ASR model results in erroneous transcriptions; and (ii) official DVD subtitles are not exhaustive of all speech in the movie and thus such a method frequently misidentifies character dialogue as AD narration (an example is provided in Fig. 3). Further, obtaining official subtitles from DVDs presents additional challenges when collecting this data at scale.

We propose an improved automated data collection method for AD, requiring only the audio track as input (no DVD subtitles), that tackles both issues by using *audio-based* speaker separation and an improved ASR model. We then use this method to collect improved annotations for the MAD dataset. Briefly, taking the mixed audio containing both AD narrations and original movie sound track as input, our automated AD collection pipeline contains five stages: (1) speech recognition using WhisperX [5] resulting in punctuated transcriptions with word-level timestamps; (2) sentence tokenization using nltk [11] to provide sentence-level segmentation; (3) speaker diarization [13,14] to assign speaker labels to each sentence, where the sentence timestamps are used as oracle voice-activity-detection (VAD); (4) labelling the speaker ID of the AD narrator by selecting the cluster with the lowest proportion of first-person pronouns (*e.g.* 'I' and 'we'); and finally (5) synchronization of the segment timestamps with the visual features by comparing audio. Further details are in the Appendix.

[1] https://audiovault.net

Dataset	Total movies	Total duration (hrs)	Total AD captions	Subtitles	Visual Features
QueryD [57]	-	207	31K	✗	✓
LSMDC [70]	200	147	128K	✗	✓
MAD-v1 [76]	488	892	280K	✓	✓
MAD-v2 (ours)	488	892	264K	✓	✓
AudioVault (ours)	7,057	12,510	3.3M	✓	✗

Table 1. **Statistics of Audio Description datasets.** We report relevant statistics to compare our MAD-v2 and Audiovault datasets.

Henceforth we refer to the original MAD annotation [76] as **MAD-v1** and our new denoised annotations as **MAD-v2**. A qualitative comparison is shown in Fig. 5, we find that our MAD-v2 is much more robust and contains less errors and less character dialogue leakage. Both LSMDC and MAD-v1 post-process their annotations by replacing character names in the annotations with 'someone' via entity recognition, and release both variants of annotations which we refer to as **`Named`** and **`Unnamed`**. Similarly, we propose two variants of our denoised annotations:

MAD-v2-`Named`: It contains the raw collected AD narrations *without* any post-processing on the character names.

MAD-v2-`Unnamed`: Following the character name anonymisation performed in earlier works, we identify character names using a Named Entity Recognition (NER) model [62] and replace them with 'someone'.

5. Partial Pretraining with AudioVault Dataset

Paired AD and corresponding visual data are difficult to obtain especially due to movie copyrights, whereas a large number of movie ADs audio tracks are available online for free (*e.g.* AudioVault). To demonstrate the effect of partial pretraining in Sec. 3.3, we collect a large-scale *text-only* movie AD dataset from AudioVault. In detail, we source mixed audio files from over 7,000 movies from AudioVault that are not included in MAD-v1, and use a denoising pipeline similar to that described in Sec. 4 to obtain the movie ADs (detailed in Appendix). Additionally we obtain a proxy for the movie subtitles by assuming the ASR from all the non-AD speakers are the characters' dialogues. To ensure no test-time leakage, we remove all movies present in either LSMDC or MAD from the dataset.

Overall, our AudioVault dataset is an order of magnitude larger than prior AD datasets (see Table 1), from which we provide two sets of data:

AudioVault-AD. The AD narrations from AudioVault and their corresponding timestamps within each movie, totalling 3.3 million AD utterances.

AudioVault-Sub. The subtitles data from AudioVault and their corresponding timestamps within each movie, totalling 8.7 million subtitle utterances.

6. Experiments

In this section we first outline the experimental details for the AD task, the datasets used for training & testing, the architectural details, and the evaluation metrics (Sec. 6.1). We then report results and discuss the findings, perform ablations on our model, and compare to prior works (Sec. 6.2).

6.1. Implementation Details

6.1.1 Datasets

Training Datasets. **CC3M** (Conceptual Caption) [72] is a large image alt-text dataset that contains 3.3M web images. **WebVid** [7] is a large video-caption dataset that contains 2.5M short stock footage videos. We use them for the partial-data pretraining for visual modules. Additionally, we use our **AudioVault-AD** to pretrain the textual modules, as described in Sec. 3.3. For the main Movie AD task, we train with original **MAD-v1** and our cleaned version **MAD-v2**, detailed in Sec. 4.

Test Datasets. **LSMDC** [68] contains 118K short video clips with descriptions from 202 movies, of which 182 of them are public. The original MAD-val&test split inherits its LSMDC annotations after filtering out 20 lower-quality movies, resulting in 162 movies from all the LSMDC-train/val/test splits. We propose an evaluation split named **MAD-eval** by further excluding LSMDC train&test movies from these 162 movies, which gives a subset consisting of 10 movies. The reason is twofold: (i) LSMDC-train is commonly used by other works as training data, and (ii) the character names of LSMDC-test are not public. Similarly, we use both **MAD-eval-`Named`** and **MAD-eval-`Unnamed`** versions. The 'Unnamed' version corresponds to the standard LSMDC annotation style – where the characters' titles and names in the descriptions are replaced by the word 'someone'; the 'Named' version is constructed from the original character names provided by LSMDC. Additionally, subtitles are not provided with MAD-val/test or LSMDC, so we transcribe them from the full-length audio tracks using WhisperX [5].

6.1.2 Architecture

For **visual features**, we use the CLIP ViT-B-32 model [63], which is a 12-layer transformer encoder that outputs 1×512 feature vectors for each input frame. These features are provided by the MAD dataset. For the **visual mapping network**, we use a 2-layer transformer encoder with 8 attention heads and 512 hidden dimensions, followed by a linear projection layer that projects 512-d features into 768-d. We use ten prompt vectors. For the **language model**, we use GPT-2 [64], specifically the version from HuggingFace. The GPT-2 model takes as input 768-d token embeddings, passes through a 12-layer transformer with a causal attention map, and outputs the next token embedding for every input token. We limit the generated number of tokens to 36, since most movie ADs are less than 36 tokens. The GPT-2 is frozen in most of our experiments unless otherwise stated. Each special token (*e.g.* $\texttt{B}_{\texttt{AD}}$) is a learnable 768-d vector. We take at most 64 past AD tokens and 32 subtitle tokens, and short text samples are padded. Specifically for subtitles, we take the most recent four dialogues within a one-minute time window.

6.1.3 Training and Inference Details

On the MAD-v1 and MAD-v2 datasets, we use a batch size of 8 sequences, each of which contains 16 consecutive video-AD pairs from a movie. Overall that gives 8×16 video-AD pairs for every batch. From each video clip, 8 frame features are uniformly sampled. By default, the model is trained for 10 epochs. One epoch means the model has seen *all* the audio descriptions once. Additional implementation details are in the Appendix.

We use the AdamW optimizer [46] and a cosine-decay learning rate schedule with a linear warm-up. The starting learning rate is 10^{-4} and is decayed to 0. For each experiment, we use a single Nvidia A-40 for training. For text generation, greedy search and beam search are commonly used sampling methods. We stop the text generation when a full stop mark is predicted, otherwise we limit the sequence length to 67 tokens. We use beam search with a beam size of 5 and mainly report results by the top-1 beam-searched outputs, since beam search performs slightly better than greedy search on multiple scenarios. Note that under the 'recurrent' setting, we feed the past greedy-searched text outputs to the model to generate the current AD, which we find gives more stable results.

6.1.4 Evaluation Metrics

To evaluate the quality of text compared with the ground-truth, we use classic metrics including ROUGE-L [45] (**R-L**), CIDEr [83] (**C**) and SPICE [2] (**S**). We also report BertScore [95] (**BertS**), which evaluates word matching between a candidate sentence and reference sentence with pre-trained BERT embeddings. A higher value indicates better text generation compared with the ground-truth.

6.2. Experiments on Movie Audio Descriptions

Effect of Temporal Context. In Table 2 we show that visual context from multiple frames brings a clear gain for

Temporal Context	Partial Data Pretrain	R-L	C	S	BertS
None (1 frame)	None	7.1	4.0	1.0	13.2
V (8 frames)	None	9.3	6.7	2.4	15.6
	CC3M [72]	9.9	8.4	2.4	16.8
	WV [7]	9.9	10.0	2.0	17.3
V+AD	None	11.1 (13.3)	12.6 (17.8)	5.1 (5.8)	18.6 (22.1)
	AV-AD	12.1 (13.9)	14.1 (19.0)	4.2 (4.8)	23.0 (23.7)
	AV-AD, WV	11.9 (13.9)	14.3 (21.9)	4.4 (4.8)	24.2 (23.8)
V+AD+Sub	AV-AD, WV	11.3	13.3	4.7	22.2
V+AD+SubN*	AV-AD, WV	11.9	14.2	5.1	23.6

Table 2. **Ablative experiments of our AD captioning method.** We ablate our model with different types of temporal context and partial pretraining. All models are trained on MAD-v2-Named and evaluated on MAD-eval-Named. For models with AD context we report recurrent results with oracle in parentheses. 'V' refers to visual context by taking multi-frame inputs, 'WV' refers to WebVid2M dataset, 'AV-AD' here refers to our partial-data pretraining with text-only AudioVault-AD dataset. *'SubN' denotes the variant of subtitle module that only takes names as input.

MAD Train Set		MAD-eval-Unnamed				MAD-eval-Named			
		R-L	C	S	BertS	R-L	C	S	BertS
v1	Unnamed	15.1	12.7	9.5	22.4	12.7	15.9	4.7	22.0
	Named	11.3	10.9	3.0	24.0	12.8	17.0	5.2	21.8
v2	Unnamed	**15.9**	**14.5**	**10.5**	**26.7**	12.9	18.0	4.7	22.0
	Named	11.4	10.0	3.1	22.5	**13.3**	**17.8**	**5.8**	**22.1**

Table 3. **Effect of denoising MAD training data annotation.** We train a model with 6 contextual ADs on MAD-v1 [76] or MAD-v2 sources without any pretraining. The model is evaluated on both the **Named** and **Unnamed** versions of MAD-eval under the **oracle** setting. Cross-domain testing results (when the model is trained and tested on different types of annotations) are provided for reference and marked in gray.

Methods	Pretraining Data	R-L	C	S	BertS
ClipCap [52]	CC3M	8.5	4.4	1.1	11.8
CapDec* [56]	AV-AD	8.2	6.7	1.4	14.3
AutoAD (ours)	AV-AD	**12.1**	14.1	4.2	23.0
AutoAD (ours)	AV-AD & WebVid	11.9	**14.3**	**4.4**	**24.2**

Table 4. Compared with other works on movie AD generation task on MAD-v2. We obtain results from other methods by fine-tuning their models on MAD-v2-Named dataset, and evaluated on MAD-eval-Named. *CapDec [56] proposes text-only pretraining to adapt the style for text generation, we pretrained their model on the text-only AudioVault-AD dataset then applied it to MAD-v2.

the AD task (C 6.7 vs 4.0). AD context provides a consistent performance improvement under both oracle (C 17.8 vs 6.7) and recurrent settings (C 12.6 vs 6.7). Note that we find feeding AD context as text tokens works better than training a textual feature mapping network, we conjecture the ADs in their original text form carry the most key information like the names and places. However, subtitle context provides no gain for our model (C 13.3 vs 14.3) under the recurrent setting, which we attribute to the very weak correspondence between the visual elements in the scene and the character dialogue. When the subtitles are filtered and contain only character names (denoted as 'SubN'), they provide a slight performance gain (C 14.2 vs 13.3). Since the subtitles used are without speaker identities, the model may struggle to know which character in the frame spoke each subtitle. Overcoming these challenges will be considered in future work.

Methods	Paired Training Data	C	M
Baseline [59]	LSMDC	11.9	8.3
TAPM [93]	LSMDC	15.4	**8.4**
AutoAD (ours)	MAD-v2-Unnamed	16.7	7.4
AutoAD (ours)	MAD-v2-Unnamed & LSMDC	**17.5**	7.5

Table 5. **Results on the LSMDC 2019 Multi-Sentence Description public test set.** We report our method with different amounts of training data and without subtitles for comparison under similar settings. Official challenge metrics (CIDEr and METEOR) are reported with the 'sentence' setting as described in [68, 93].

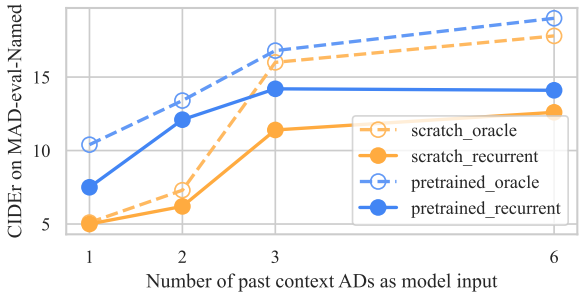
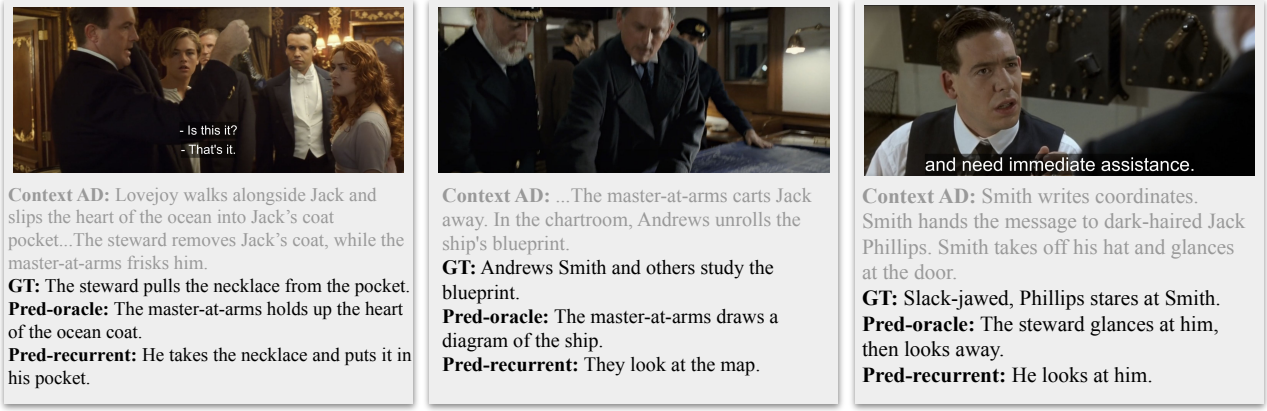


Figure 4. **Effect of the length of context AD.** We use the model 'V+AD' in Table 2, and train with different number of past AD sentences. 'scratch' indicates no partial-data pretraining; 'pretrained' refers to pretraining with text-only AudioVault-AD.

Effect of MAD data cleaning. Table 3 demonstrates the benefit of our MAD v2 annotations over v1, confirming the qualitative findings. Training the AD model with context on v2 outperforms training on v1 under all settings (both named and unnamed) by a significant margin. Since the v2 annotations are fewer in number than MAD-v1, this suggests they are indeed less noisy and result in AD captioning models with improved performance.

Effect of Pretraining with Partial Data. In Table 2, we find that **visual-only pretraining** on open-domain vision-text data provides clear gains (CIDEr 8.4 vs 6.7 for CC3M, and 10.0 vs 6.7 for WebVid). But considering the size of visual samples, the improvement is not data-efficient. We attribute this to the large domain gap between movie AD and classical visual caption annotations like CC3M or WebVid2M. The **text-only pretraining** of our model also improves performance. For the recurrent AD context model, AudioVault-AD pretraining increases CIDEr from 12.6 to 14.1, which indicates the great importance of adapting to the text style and context. The combination of the visual



Context AD: Lovejoy walks alongside Jack and slips the heart of the ocean into Jack's coat pocket...The steward removes Jack's coat, while the master-at-arms frisks him.
GT: The steward pulls the necklace from the pocket.
Pred-oracle: The master-at-arms holds up the heart of the ocean coat.
Pred-recurrent: He takes the necklace and puts it in his pocket.

Context AD: ...The master-at-arms carts Jack away. In the chartroom, Andrews unrolls the ship's blueprint.
GT: Andrews Smith and others study the blueprint.
Pred-oracle: The master-at-arms draws a diagram of the ship.
Pred-recurrent: They look at the map.

Context AD: Smith writes coordinates. Smith hands the message to dark-haired Jack Phillips. Smith takes off his hat and glances at the door.
GT: Slack-jawed, Phillips stares at Smith.
Pred-oracle: The steward glances at him, then looks away.
Pred-recurrent: He looks at him.

Figure 5. **Qualitative examples of automatically generated AD by AutoAD.** We highlight AD predictions under both the oracle and recurrent settings. Previous AD context is shown in gray. For ease of visualisation, a single frame from each movie clip is shown with subtitles overlaid. Samples are taken from Titanic (1997) [17].

module after visual-only pretraining (WebVid) and the textual modules after text-only pretraining (AV-AD) gives a further performance gain (C 21.9 vs 19.0 for the oracle setting, and 14.3 vs 14.1 for recurrent).

Length of Context. In Figure 4 we show the effect of varying the number of context ADs given to the model. Longer AD context improves performance almost consistently across all settings, but it brings extra computational cost due to the quadratic complexity of the attention operation in GPT-2. Note that we experiment with at most 6 contextual AD sentences, which is equivalent to about 70-word embeddings in Eq. 1. The trend for the recurrent setting flattens when the context ADs are longer than 3 sentences, which is probably due to the limited power of processing long context for the GPT2 model.

6.2.1 Qualitative Results

Fig. 5 shows qualitative examples of our model. Under the oracle setting, the model can use the character identities easily from the past ground-truth AD (*e.g.* "master-at-arms"). Whereas under the recurrent setting, the model can only learn names from the subtitles but names appear very sparsely in subtitles, therefore the model mostly predicts pronouns (*e.g.* "he", "they") but still gets the actions ("looks") or objects ("necklace") correct.

6.3. Comparison with Other Works

In Table 4, we compare our method with previous visual captioning methods. Note that since the MAD dataset only releases the CLIP visual features, rather than the movie frames, our comparison is limited to methods that build on frozen CLIP features. We show a clear performance improvement compared to ClipCap [52] and CapDec [56], for the latter the language model is also adapted to the movie AD domain by text-only pretraining. The results highlight the importance of context for movie AD.

In Table 5, we adapt our method to the Multi-Sentence Description task on LSMDC, in which the model takes five consecutive clips and generates five corresponding descriptions. Since the task is performed on the *unnamed* annotations, we finetune our best model in Table 4 with varying 0-4 context ADs as input on MAD-v2-`Unnamed` dataset and test with the *recurrent* setting. To make minimal changes, our model still takes a single clip feature at each step, whereas previous methods take all five clips together for movie description. Despite this disadvantage, we obtain competitive results on this task even without using the *manually-cleaned* LSMDC training set (C 16.7 vs 15.4), effectively *zero-shot*. The performance of the model can be further improved by additionally training on LSMDC data.

7. Conclusion and Future Work

This paper focuses on the automatic generation of movie AD for a given time interval, and has made significant progress. We propose an AutoAD pipeline that incorporates contextual information. Additionally, we demonstrate the effectiveness of partial-data pretraining, a technique that could be widely applicable when full data is difficult to obtain. Further, we clean up the previous MAD dataset and collect a new text-only movie AD dataset as a pretraining resource. However, a clear limitation of this AutoAD pipeline is character naming – referencing *who* is doing *what*, a necessary ingredient for story-coherent movie AD. Additionally, future work could tackle the problem of *when* to generate AD, instead of relying on the annotated AD timestamps.

Acknowledgements. We thank Mattia Soldan for helping with the MAD dataset, Anna Rohrbach for the LSMDC dataset, and the AudioVault team for their priceless contribution to the visually impaired. This research is funded by EPSRC PG VisualAI EP/T028572/1, a Google-Deepmind Scholarship, and ANR-21-CE23-0003-01 CorVis.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022. 3

[2] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *Proc. ECCV*, pages 382–398. Springer, 2016. 6

[3] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018. 2

[4] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv:2203.17274*, 2022. 3

[5] Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. WhisperX: Time-accurate speech transcription of long-form audio. *arXiv preprint, arXiv:2303.00747*, 2023. 5, 6

[6] Max Bain, Arsha Nagrani, Andrew Brown, and Andrew Zisserman. Condensed movies: Story based retrieval with contextual embeddings. In *Proc. ACCV*, 2020. 2

[7] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proc. ICCV*, 2021. 2, 6, 7

[8] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. A clip-hitchhiker's guide to long video retrieval. *arXiv preprint arXiv:2205.08508*, 2022. 2

[9] Kobus Barnard, Pinar Duygulu, David Forsyth, Nando De Freitas, David M Blei, and Michael I Jordan. Matching words and pictures. *The Journal of Machine Learning Research*, 3:1107–1135, 2003. 2

[10] Kobus Barnard and David Forsyth. Learning the semantics of words and pictures. In *Proc. ICCV*, volume 2, pages 408–415. IEEE, 2001. 2

[11] Steven Bird. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72, 2006. 5

[12] Piotr Bojanowski, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Finding actors and actions in movies. In *Proc. ICCV*, pages 2280–2287, 2013. 1, 2

[13] Hervé Bredin and Antoine Laurent. End-to-end speaker segmentation for overlap-aware resegmentation. In *Proc. Interspeech 2021*, 2021. 5

[14] Hervé Bredin, Ruiqing Yin, Juan Manuel Coria, Gregory Gelly, Pavel Korshunov, Marvin Lavechin, Diego Fustes, Hadrien Titeux, Wassim Bouaziz, and Marie-Philippe Gill. pyannote.audio: neural building blocks for speaker diarization. In *Proc. ICASSP*, 2020. 5

[15] Andrew Brown, Ernesto Coto, and Andrew Zisserman. Automated video labelling: Identifying faces by corroborative evidence. In *International Conference on Multimedia Information Processing and Retrieval*, 2021. 1

[16] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020. 3

[17] James Cameron. Titanic. Paramount Pictures, 1997. 8

[18] Aman Chadha, Gurneet Arora, and Navpreet Kaloty. iPerceive: Applying common-sense reasoning to multi-modal dense video captioning and video question answering. In *Proc. WACV*, 2021. 2

[19] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Association for Computational Linguistics*, pages 190–200, 2011. 2

[20] Shaoxiang Chen and Yu-Gang Jiang. Towards bridging event captioner and sentence localizer for weakly supervised dense event captioning. In *Proc. CVPR*, 2021. 2

[21] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 2

[22] Xinlei Chen and C. Lawrence Zitnick. Learning a recurrent visual representation for image caption generation. *arXiv*, abs/1411.5654, 2014. 2

[23] Chaorui Deng, Shizhe Chen, Da Chen, Yuan He, and Qi Wu. Sketch, ground, and refine: Top-down dense video captioning. In *CVPR*, 2021. 2

[24] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proc. CVPR*, pages 2625–2634, 2015. 2

[25] Mark Everingham, Josef Sivic, and Andrew Zisserman. "Hello! My name is... Buffy" – automatic naming of characters in TV video. In *Proc. BMVC*, 2006. 1

[26] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv:2110.04544*, 2021. 3

[27] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020. 4

[28] Gabriel Huang, Bo Pang, Zhenhai Zhu, Clara Rivera, and Radu Soricut. Multimodal pretraining for dense video captioning. *arXiv preprint arXiv:2011.11760*, 2020. 2

[29] Qingqiu Huang, Wentao Liu, and Dahua Lin. Person search in videos with one portrait through visual and temporal links. In *Proc. ECCV*. Springer-Verlag, 2018. 1

[30] Ting-Hao Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. Visual storytelling. In *Association for Computational Linguistics*, pages 1233–1239, 2016. 2

[31] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. *Proc. ECCV*, 2022. 3

[32] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *Proc. ECCV*, 2022. 3

[33] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proc. CVPR*, pages 3128–3137, 2015. 2

[34] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014. 2

[35] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proc. ICCV*, pages 706–715, 2017. 2

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

[37] Anna Kukleva, Makarand Tapaswi, and Ivan Laptev. Learning interactions and relationships between movie characters. In *Proc. CVPR*, June 2020. 1

[38] Ivan Laptev, Marcin Marszałek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *Proc. CVPR*, 2008. 1

[39] Victor Lavrenko, Raghavan Manmatha, and Jiwoon Jeon. A model for learning the semantics of pictures. In *NeurIPS*, volume 16, 2003. 2

[40] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *EMNLP*, 2021. 3

[41] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 2

[42] Junnan Li, Yongkang Wong, Qi Zhao, and M. Kankanhalli. Video storytelling: Textual summaries for events. *IEEE Transactions on Multimedia*, 22:554–565, 2020. 2

[43] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL*, 2021. 3

[44] Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei. Jointly localizing and describing events for dense video captioning. In *CVPR*, 2018. 2

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

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

[47] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Neural baby talk. In *CVPR*, 2018. 2

[48] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020. 2

[49] Richard Marquand. Star wars: Episode vi – return of the jedi. 20th Century Fox, 1983. 5

[50] Marcin Marszałek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *Proc. CVPR*, 2009. 1

[51] 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 *Proc. ICCV*, pages 2630–2640, 2019. 2

[52] Ron Mokady, Amir Hertz, and Amit H Bermano. ClipCap: CLIP prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. 2, 3, 7, 8

[53] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Association for Computational Linguistics*, pages 839–849, 2016. 4

[54] Jonghwan Mun, Linjie Yang, Zhou Ren, Ning Xu, and Bohyung Han. Streamlined dense video captioning. In *Proc. CVPR*, 2019. 2

[55] Arsha Nagrani, Paul Hongsuck Seo, Bryan Seybold, Anja Hauth, Santiago Manen, Chen Sun, and Cordelia Schmid. Learning audio-video modalities from image captions. *Proc. ECCV*, 2022. 2

[56] David Nukrai, Ron Mokady, and Amir Globerson. Text-only training for image captioning using noise-injected CLIP. *arXiv preprint arXiv:2211.00575*, 2022. 2, 7, 8

[57] 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 *Proc. ICASSP*, pages 2265–2269. IEEE, 2021. 2, 5

[58] Pinelopi Papalampidi, Frank Keller, and Mirella Lapata. Movie plot analysis via turning point identification. *arXiv preprint arXiv:1908.10328*, 2019. 2

[59] Jae Sung Park, Marcus Rohrbach, Trevor Darrell, and Anna Rohrbach. Adversarial inference for multi-sentence video description. In *Proc. CVPR*, 2019. 2, 7

[60] Alonso Patron-Perez, M. Marszałek, Andrew Zisserman, and Ian D. Reid. High five: Recognising human interactions in TV shows. In *Proc. BMVC*, 2010. 1

[61] Georgios Pavlakos, Ethan Weber, Matthew Tancik, and Angjoo Kanazawa. The one where they reconstructed 3d humans and environments in TV shows. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Proc. ECCV*, 2022. 1

[62] Jean Baptiste Polle. Camembert-ner: model fine-tuned from camembert for ner task. https://huggingface.co/Jean-Baptiste/camembert-ner. Accessed: 2022-11-01. 5

[63] 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. In *Proc. ICML*, 2021. 2, 3, 6

[64] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019. 2, 3, 6

[65] Tanzila Rahman, Bicheng Xu, and Leonid Sigal. Watch, listen and tell: Multi-modal weakly supervised dense event captioning. In *Proc. ICCV*, 2019. 2

[66] Hareesh Ravi, Kushal Kafle, Scott Cohen, Jonathan Brandt, and Mubbasir Kapadia. Aesop: Abstract encoding of stories, objects, and pictures. In *Proc. ICCV*, pages 2052–2063, 2021. 2

[67] Video Description Research and Development Center. YouDescribe, 2013. 1, 2

[68] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. In *Proc. CVPR*, 2015. 1, 2, 6, 7

[69] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. In *Proc. CVPR*, pages 3202–3212, 2015. 2

[70] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *IJCV*, 123(1):94–120, 2017. 2, 5

[71] Paul Hongsuck Seo, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. End-to-end generative pretraining for multimodal video captioning. In *Proc. CVPR*, pages 17959–17968, 2022. 2

[72] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Association for Computational Linguistics*, 2018. 2, 6, 7

[73] Zhiqiang Shen, Jianguo Li, Zhou Su, Minjun Li, Yurong Chen, Yu-Gang Jiang, and Xiangyang Xue. Weakly supervised dense video captioning. In *Proc. CVPR*, 2017. 2

[74] Botian Shi, Lei Ji, Yaobo Liang, Nan Duan, Peng Chen, Zhendong Niu, and Ming Zhou. Dense procedure captioning in narrated instructional videos. In *Association for Computational Linguistics*, 2019. 2

[75] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Proc. ECCV*, pages 510–526. Springer, 2016. 2

[76] 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 *Proc. CVPR*, 2022. 3, 5, 7

[77] Yidan Sun, Qin Chao, and Boyang Li. Synopses of movie narratives: a video-language dataset for story understanding. *arXiv preprint arXiv:2203.05711*, 2022. 2

[78] Mingkang Tang, Zhanyu Wang, Zhenhua Liu, Fengyun Rao, Dian Li, and Xiu Li. Clip4caption: Clip for video caption. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4858–4862, 2021. 2

[79] Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. "knock! knock! who is it?" probabilistic person identification in TV series. In *Proc. CVPR*, 2012. 1

[80] Makarand Tapaswi, Martin Bauml, and Rainer Stiefelhagen. Book2movie: Aligning video scenes with book chapters. In *Proc. CVPR*, pages 1827–1835, 2015. 2

[81] 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

[82] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. In *NeurIPS*, 2021. 3

[83] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proc. CVPR*, pages 4566–4575, 2015. 6

[84] Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond J. Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence – video to text. In *ICCV*, 2015. 2

[85] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *Proc. CVPR*, pages 98–106, 2016. 1

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

[87] Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. End-to-end dense video captioning with parallel decoding. In *Proc. ICCV*, 2021. 2

[88] Yujia Wang, Wei Liang, Haikun Huang, Yongqi Zhang, Dingzeyu Li, and Lap-Fai Yu. Toward automatic audio description generation for accessible videos. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2021. 1

[89] Yu Xiong, Qingqiu Huang, Lingfeng Guo, Hang Zhou, Bolei Zhou, and Dahua Lin. A graph-based framework to bridge movies and synopses. In *Proc. ICCV*, pages 4592–4601, 2019. 2

[90] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *Proc. CVPR*, pages 5288–5296, 2016. 2

[91] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. *arXiv preprint arXiv:2206.08155*, 2022. 3

[92] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. CoCa: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*, 2022. 2

[93] Youngjae Yu, Jiwan Chung, Heeseung Yun, Jongseok Kim, and Gunhee Kim. Transitional adaptation of pretrained models for visual storytelling. In *Proc. CVPR*, pages 12658–12668, 2021. 2, 7

[94] Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv:2111.03930*, 2021. 3

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

[96] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proc. CVPR*, 2018. 2

[97] Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proc. ICCV*, 2015. 2