Appendix

A. Downloading cast information

As briefly described in the main paper Section 4.3, We download cast information from IMDb[5]. Specifically, we first query the movie based on its IMDb ID, *e.g.* tt0120780, which is provided by the datasets like AudioVault-AD [18] or MAD [50]. Next, we download the cast list under the HTML element 'Top Cast', where each item in the list contains the actor name, the character name and a portrait picture of the actor. For each movie, we download such information for up to 10 characters.

Special Cases. Some characters in the cast list do not have corresponding portrait pictures. Among 488 movies from MAD-train, we find 293 movies have missing portrait pictures in their top-10 cast list. By manual verification, we find it is typically because the actors are less known and therefore do not have an IMDb profile page – since most of the IMDb data source is contributed by volunteers, there exists an inevitable bias towards celebrities or well-known movies. In such cases, we remove the characters in our data collection pipeline. Overall, among 488 movies from MAD-train, there are 17 movies with less than 5 characters downloaded, and one movie has an empty character list, which is *Human Flow (2017)*[6], a documentary.

B. Statistics of movie AD and subtitles

Frequency of names and pronouns. Table A.1 and A.2 show the frequency of names and pronouns on AD and subtitles respectively. The frequency is calculated on a per-sentence basis, that is, if any name (from Named-Entity Recognition (NER) outputs) or pronoun exists in the AD/subtitle sentence, the count is accumulated by one. The tables show that a substantial 39.1% of AD sentences contain character names, compared to only 13.3% for subtitles. Generating sentences with correct names is an important aspect of AD quality. Note that in this analysis, we discard the intro and outro of the movie for more reliable frequencies. The AD during those periods mainly performs an OCR task – introducing the producers, the name of the studio or reading movie credits at the end, which includes a large number of '[PER]' tags from the NER outputs.

Unique names within each movie. From the NER output of AD sentences, we aggregate the unique words with '[PER]' tags for each movie. For 488 movies in MAD-train, we found on average there are 69 unique names for each movie, with a maximum of 176 unique names and a minimum of 3 unique names. The number is much higher than the length of a typical cast list because (i) characters could be mentioned in different ways, *e.g.* by their first-name, last-name or titles, (ii) the names mentioned in AD do not correspond to characters, *e.g.* Gryffindor for the college name, (iii) errors or noises of the NER pipeline that the words are partitioned incorrectly.

from 488 **MAD-train** movies	quantity	ratio
all AD sentences	310,494	100%
AD with [PER] tag	121,557	39.1% (40.7%†)
AD with pronouns*	111,974	36.1%
AD with ([PER] tag *or* pronouns)	202,256	65.1%

Table A.1. Frequency of names or pronouns in the **AD sentences**. The numbers are based on MAD-train movies *after removing the intro and outro* of the movies. The '[PER]' is the entity category for 'person' from NER outputs. '†': If including AD from intro and outro, the percentage of AD with [PER] tag is 40.7%, which is reported in the main paper page-4 and 8. '*': We count the occurrence of any one of six pronouns {she, her, he, him, they, them}.

from 488 **MAD-train** movies	quantity	ratio
all subtitle sentences	628,613	100%
subtitles with [PER] tag	83,904	13.3%
subtitles with pronouns*	150,564	24.0%
subtitles with ([PER] tag *or* pronouns)	216,410	34.4%

Table A.2. Frequency of names or pronouns in the **subtitles**. The numbers are based on MAD-train movies. The '[PER]' is the entity category for 'person' from NER outputs. '*': We count the occurrence of any one of eight pronouns {she, her, he, him, they, them, i, me}.

Visualization of AD and subtitles on the time axis. Following The Web Content Accessibility Guidelines 2.0 [9] (also introduced in the main paper Section 3.3), successful AD should be added during existing pauses in movie dialogues. In Figure A.1, we visualize both ground-truth AD and movie subtitles on the timeline for 15-second and 10-minute movie clips to illustrate this interleaved property of ground-truth AD and subtitles.

Stats of inter-annotator agreement. As briefly described in Section 3.3, the timestamps of human-generated AD vary for the same movie, especially during long pauses in dialogue. On the AudioVault website, a small portion (less than 20%) of movies have more than one AD versions or multi-lingual AD versions. Figure A.2 shows an example movie clip with its two AD versions on AudioVault-AD. Those two versions describe the same movie but are provided by annotators from the US and UK respectively. Comparing the middle blocks with the lower blocks in Fig. A.2, it can be seen that AD sentences from the two versions have different start/end timestamps (both shown in orange blocks). We also notice that character names are referred to differently in both AD versions, *e.g.* the AD at

[5] https://www.imdb.com/
[6] https://www.imdb.com/title/tt6573444/

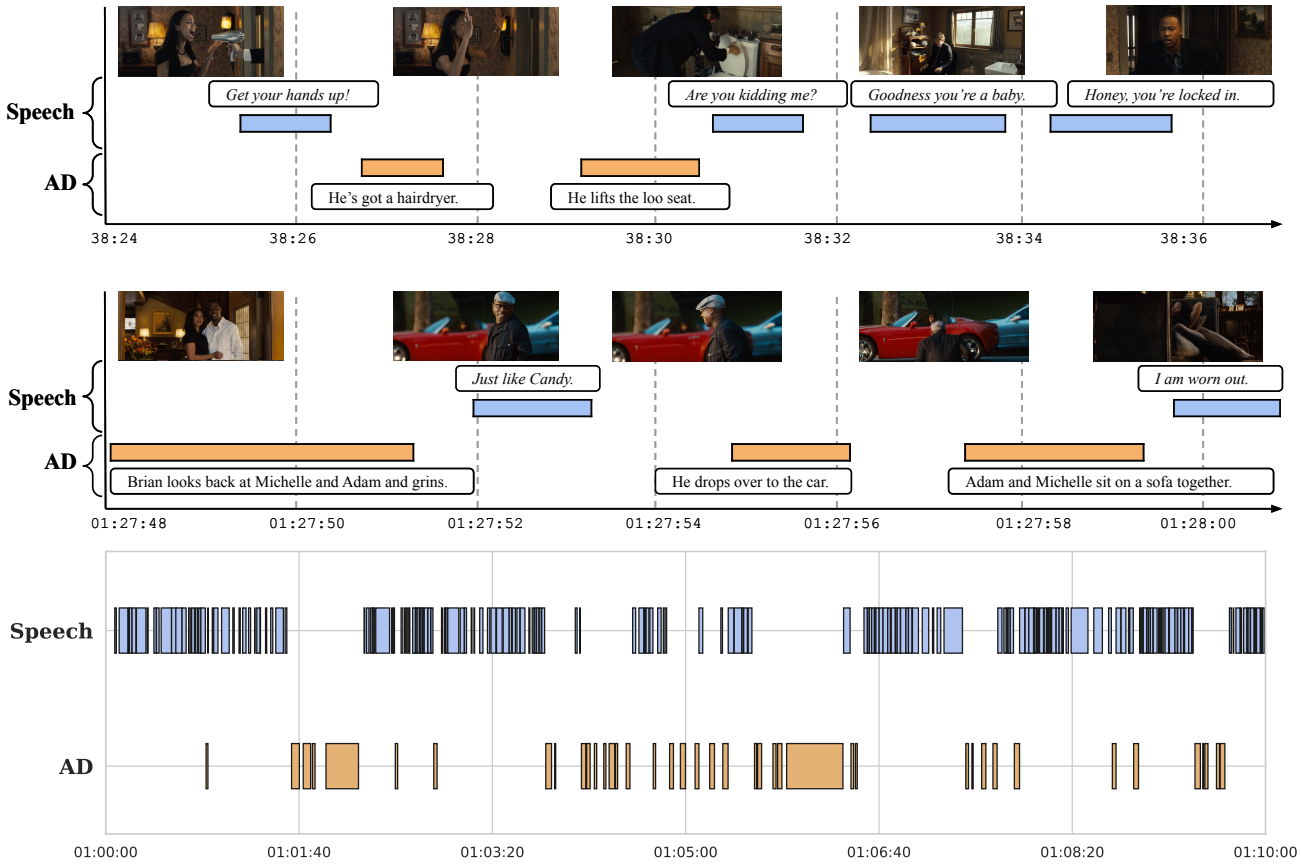

Figure A.1. Timeline visualization of a movie with its original dialogue (speech) and human-generated Audio Description (AD). AD is inserted at appropriate times between speech, describing relevant visual elements in the frames. The top and mid figures show movie clips spanning 15 seconds with corresponding frames and texts, the bottom figure shows a movie clip spanning 10 minutes with only timestamps. The movie shown here is *Death at a Funeral (2010)* with IMDb ID tt1321509. The corresponding AD is sourced from AudioVault-AD (ID 17295).

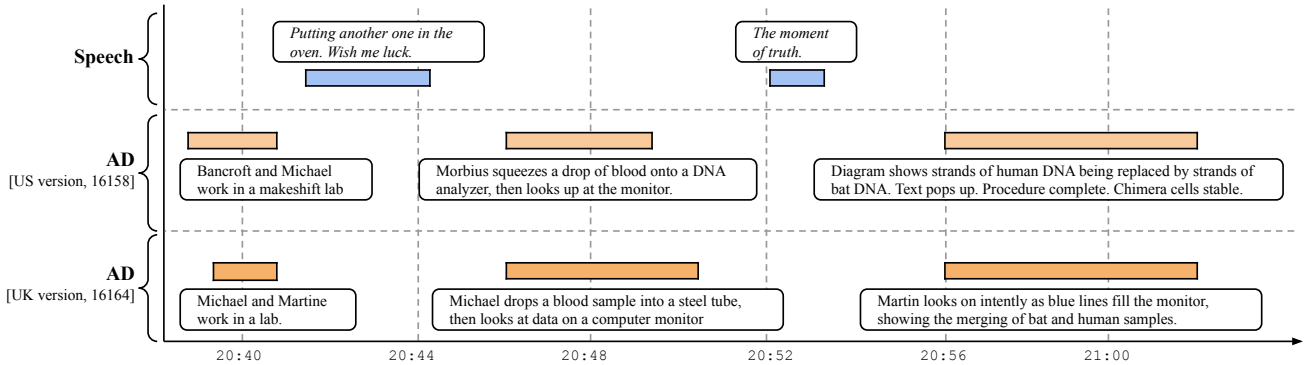

Figure A.2. Timeline visualization of the **same movie clip** with its original dialogue (speech) and **two versions** of human-generated Audio Description (AD). Note that disagreements of timestamps exist between different versions of AD for the same movie clip. The movie clip is from *Morbius (2022)* with IMDb ID tt5108870. The two versions of AD are from AudioVault-AD with ID 16158 (US annotator) and 16164 (UK annotator). The characters who appeared in the scene are *Dr. Michael Morbius* and *Martine Bancroft*.

20:40. Incorporating multiple versions of AD of the same movie would be an interesting research direction. In this paper, we only consider one AD version for each movie by choosing the version with a lower AudioVault ID.

C. Training details

C.1. Character recognition module

Architecture details. See Table A.3 for the details of character recognition module.

linear projection layer	$512 \rightarrow 512$
num blocks	2
channel	512
num head	8
ff dimension	2048

Table A.3. The architecture details of the character recognition module, which consists of a 2-layer transformer decoder.

Training recipe. The character recognition module is trained with binary labels derived from MovieNet face annotations, as described in the main paper Sections 3.2 and 4.1. The model is trained with AdamW optimizer with a learning rate of 10^{-4} for 10 epochs with a batch size of 512 movie clips. The loss is binary cross-entropy with label balancing.

C.2. Other pretraining with partial data.

We follow [18] for the pretraining with partial data. Specifically, we use the text-only AudioVault-AD dataset to finetune the last 6 blocks of a Web-Text pretrained GPT2 for 5 epochs. We also use the video-text data from WebVid to pretrain the perceiver resampler and X-Attn blocks for 5 epochs, but with GPT2 weights frozen. Both pretraining procedures can be achieved in parallel, and the trained weights from both settings can be combined as an initialization for the AD generation finetuning.

C.3. The final finetuning.

Architecture details. See Table A.4 for the details of the perceiver resampler and X-Attn blocks.

Perceiver Resampler	projection layer†	$512 \rightarrow 768$
	num latent	10
	num blocks	2
	channel	768
	num head	12
	ff dimension	3072
X-Attn	num blocks	12*
	channel	768
	num head	12
	ff dimension	3072

Table A.4. The architecture details of perceiver resampler and X-Attn blocks. †: The perceiver resampler takes 512-d CLIP visual features as input. Those features are first projected to 768-d for further computation. *: We insert 12 X-Attn blocks into 12-block GPT2-small model, that is one X-Attn block for each GPT2 block.

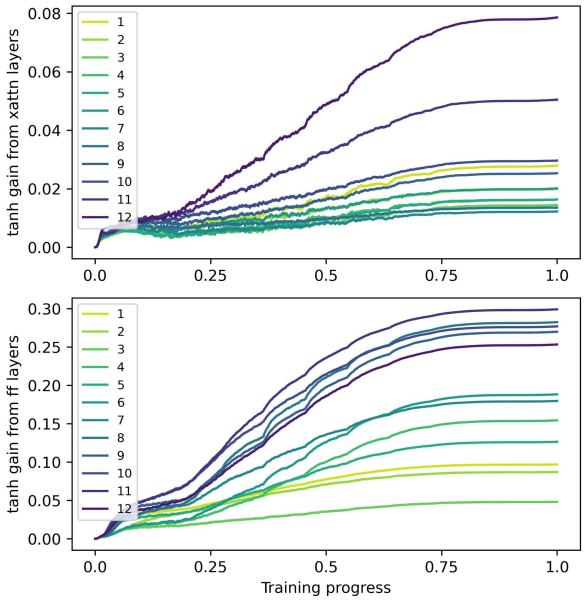

Figure A.3. Monitoring Tanh gating during the training process. There are two Tanh gates for each X-Attn block: one for X-Attn operation and the other for feed-forward operation. Please refer to [1] for details. In this figure, the X-Attn blocks are trained from randomly initialized weights, thus the gating value starts from zero.

Training recipe. The AD generation pipeline is trained (or finetuned) on MAD-train data with a batch size of 64 movie clips for 10 epochs. We use the AdamW optimizer with a cosine-decayed learning rate schedule with a linear warm-up. The default learning rate is 10^{-4}. The GPT2 weights are frozen when training for AD generation. The trainable parameters are the perceiver resampler and the X-Attn blocks. For the textual character information (*e.g.* Jack played by Leonardo DiCaprio ...), we right-pad the sequences of text tokens for up to 64 tokens. For the contextual AD information, we right-pad the sequences for up to 32 tokens. For the character's exemplar features, we pad with zero values for up to 10 characters.

C.4. Temporal Proposal Classification

Architecture Details. A pretrained BERT *base-uncased* model is used, with special tokens added to the vocabulary for the timestamps tokens, `|<|t01|>,...,<|t59|>` to indicate each 0.5-second bin in the 30-second context window. The visual CLIP features are first projected through a linear layer (512→768), whereas the audio features are simply zero-padded from 128→768. BERT positional embeddings are added to both features.

Training recipe. The model is trained with a batch size of 64 context windows for 3 epochs on MAD-train movies. We use AdamW optimizer with a learning rate of 10^{-4}.

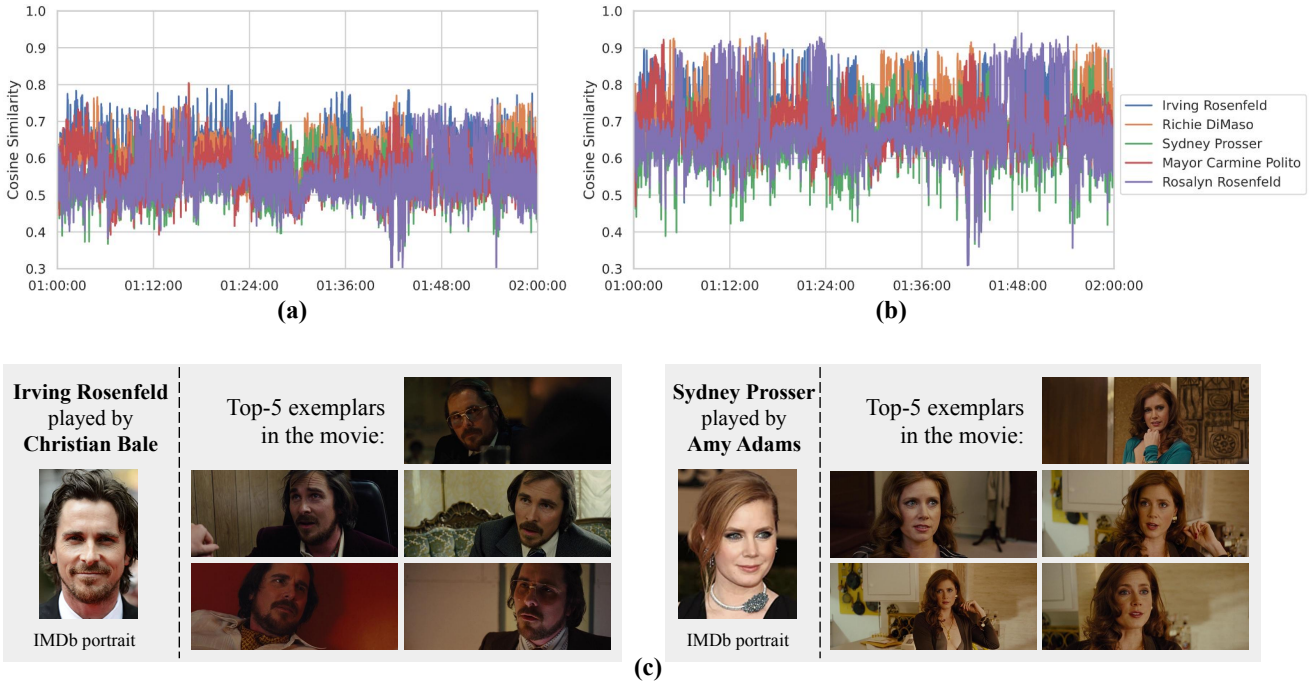

Figure A.4. Details of calibrating cosine distance and leveraging IMDb portrait images. **(a)** Cosine similarity between actors' IMDb portrait images and the movie features *before* calibration (only a one-hour clip is shown for clarity). **(b)** Cosine similarity between characters' in-movie exemplar features with the movie features, *i.e. after* calibration. The same one-hour clip is shown. **(c)** Visualization of top-5 exemplars for two characters, which are simply obtained by taking the top-5 peaks from Fig.(a) for each actor. The movie samples are from *American Hustle (2013)* with IMDb ID tt1800241.

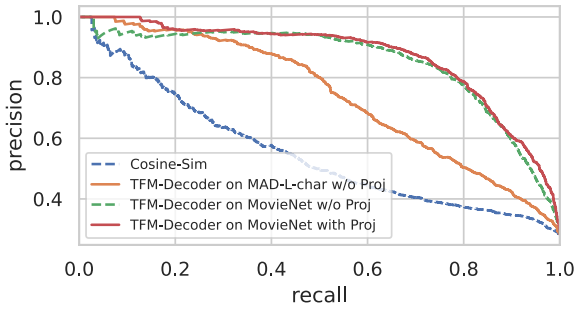

Figure A.5. More Precision-Recall curves for the character recognition methods. We show three methods: thresholding actor-movie cosine similarity, transformer decoder on MAD-L-char w/o linear projection layer, transformer decoder on MovieNet w/o linear projection layer, and transformer decoder on MovieNet with linear projection layer. The precision/recall is calculated on a per-character basis, *i.e.* the precision/recall of the cosine thresholds to correctly find a character name mentioned in the AD.

Baseline. For the binary temporal proposal classification task described in Section 3.3, we propose a simple decision-based baseline whereby any speech gap with a duration greater than a fixed threshold is classified to have AD inserted, and not AD inserted otherwise. In Table 3, the Average Precision and ROC AUC is calculated by varying the fixed threshold at 100 values equally spaced between 2 and 6 seconds.

D. Analysis

D.1. Tanh gating during training

Following Flamingo [1], we visualize the absolute value of tanh gating for each X-Attn block during training, which could be a rough indicator showing how much visual information is conditioned by the GPT-2 model. In contrast to Flamingo Fig. 6 that their tanh gating values are much closer to 1, our Figure A.3 shows the tanh values have a similar increasing trend during training but the final value is much lower. It indicates a longer training schedule with a larger dataset would further benefit our model.

D.2. Character recognition module

Cosine distance and calibration. As shown in Figure A.4-(a), the cosine similarity between actors' IMDb portrait images and the movie features is not a good indicator of in-screen or off-screen actors. For example, the peaks of the blue curve (Irving Rosenfeld) are always higher than that of the purple curve (Rosalyn Rosenfeld). As introduced in the main paper page 4, in order to compensate for the variance of appearance from IMDb portrait images, we find exemplars of the actors in the same movie as a calibra-

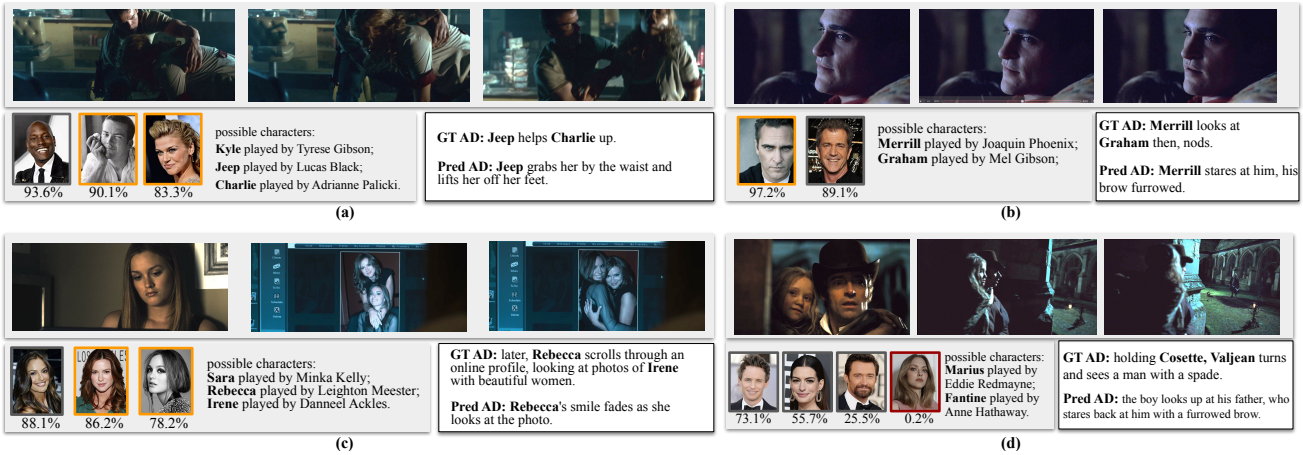

Figure A.6. Following the same style as the main paper Figure 6, we show qualitative results with the character bank. The probability shown below the characters' portraits is the output of our character recognition module, with correctly recognized characters marked using an orange border. We use 50% as the decision boundary for active characters. The movies are from **(a)**: Legion (2010), **(b)**: Signs (2002), **(c)**: The Roommate (2011), **(d)**: Les Misérables (2012).

tion process. Figure A.4-(c) shows two examples of exemplar searching, which is achieved by simply taking the top-5 peaks for each actor in Fig. A.4-(a). Next, we use the averaged exemplar features to replace the original IMDb portrait features and re-compute the cosine similarity. As shown in Figure A.4-(b), the calibration process normalizes the cosine similarity and makes the comparison between actors more meaningful.

Other character annotation dataset. In the main paper, we use the manually annotated character annotation from the MovieNet dataset. But the character labels can also be obtained with weakly annotated data, such as the AD annotation.

We propose a dataset named **MAD-L-char** for movie character recognition, which is sourced from MAD-train and LSMDC-train. The character names in **MAD-L-char** are automatically mined in two steps: (1) running named entity recognition (NER) [38] on the AD annotation, and (2) computing the intersection with the movie's cast list. Specifically, the NER on MAD-train is sourced by running an open-sourced model[7], and the NER from 139 LSMDC-train movies can be obtained from the LSMDC annotations.

P-R curve for character recognition. In addition to the main paper Table 1 and Fig. 5, here in Fig. A.5, we compare four PR curves as detailed in the figure caption. The PR curves show that the model trained on the manually annotated MovieNet dataset clearly outperforms the same model trained on the automatically mined MAD-L-char dataset. Additionally, the extra linear project layer brings a clear performance gain. Note that it is difficult to achieve perfect PR curves, partially because for some movies, even the top

[7]https://huggingface.co/Jean-Baptiste/camembert-ner

Methods	Training Data	Linear Proj	ROC AUC	Average Precision
Cosine-Sim	-	-	0.72	0.55
TFM Decoder	MAD-L-char	✗	0.84	0.74
TFM Decoder	MovieNet	✗	0.92	0.85
TFM Decoder	MovieNet	✓	**0.93**	**0.87**

Table A.5. Quantitative comparison of various character recognition modules.

10 characters downloaded from IMDb may not cover the main characters, such as the Harry Potter series which has a very large cast list. The corresponding quantitative metrics of these methods are shown in Table A.5.

Statistics of recognized active characters. After the character recognition module is trained, we simply choose the standard probability of 0.5 as the threshold for the decision boundary. With a threshold of 0.5, the character recognition module achieves 0.83 recall and 0.75 precision on MAD-eval movies (read from Figure A.5). Next, this module can be used to recognize active characters in any public movie, either offline or on-the-fly. Among more than 300k AD sentences in MAD-train, the character recognition module predicts 1.3 active characters on average per AD sentence, with 94.8% AD sentences having no more than 5 predicted active characters and 14.6% AD sentences having zero active characters.

D.3. Learning with subtitles

In addition to the character bank, we find feeding in subtitles as model inputs does not further improve performance. There are two possible reasons: (i) usually the subtitles do not describe the scene or characters, and (ii) the character names are already supplied by the character bank. Leveraging movie subtitles effectively is a promising future direction.

E. More qualitative results

More qualitative results are shown in Figure A.6. Note that in **(d)**, the girl in the scene (*young* Cosette played by Isabelle Allen) is not in the top cast whereas our top cast contains the *adult* Cosette played by Amanda Seyfried, shown in red border. Recognizing characters in such cases is challenging but it indicates the character recognition module has a large space for improvement.

F. Video captioning results on TVC

TVC [30] is a video captioning dataset consisting of TV series, which contains character names in captions. There are some domain gaps between TV series and movies: *e.g*. the frequent character bank is smaller for TV series, and scene locations may be less varied. In Figure A.7, we provide qualitative results of adapting our AutoAD-II model on TVC *without* any further training on TV series. Different from TVC which provides captions for a relatively long video clip spanning a few minutes, we feed in short clips spanning just a few seconds to match our training distribution.

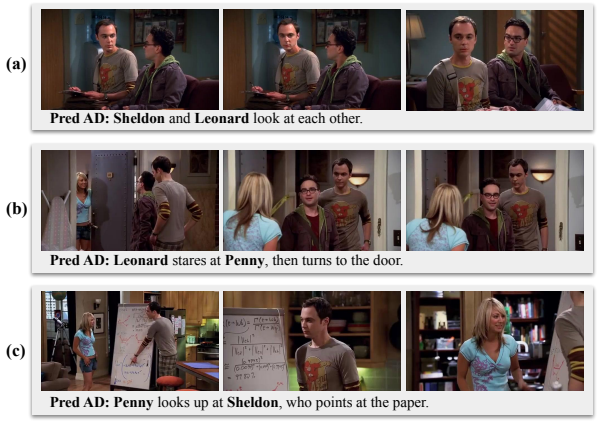

Figure A.7. Qualitative results on TVC samples without any specific training. The characters' portraits are downloaded from IMDb page of *The Big Bang Theory* https://www.imdb.com/title/tt0898266/.