As the Story Unfolds: Watching a Film and Identifying Characters as a Human Does

Zhongrui Gui [1] **Junyu Xie**[1] **Tengda Han**[1] **Weidi Xie**[1,2] **Andrew Zisserman**[1]
[1]Visual Geometry Group, Department of Engineering Science, University of Oxford, UK
[2]School of Artificial Intelligence (SAI), Shanghai Jiao Tong University, China
{zhongrui,jyx,htd,weidi,az}@robots.ox.ac.uk

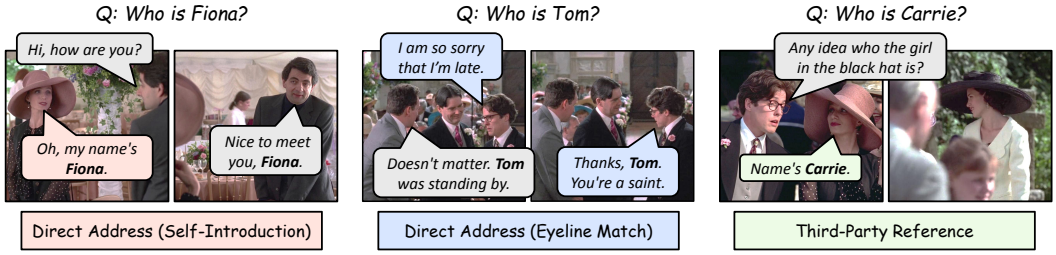


Figure 1: **Character naming task.** We associate character names to their facial appearances in movies, mirroring how viewers infer who is who while watching. Characters may be introduced through diverse cues, including self-introduction, direct address, and third-party reference. We show three representative examples from *Four Weddings and a Funeral* (1994). Correctly linking a character to their name is central to movie understanding, enabling coherent character tracking across scenes as the story unfolds.

Abstract

Films introduce characters incrementally as they progress. We study the problem of identifying principal characters and assigning their names directly from the movie itself, without using external cast lists, actor photographs, or pre-built character banks. We introduce the character naming task: given a name mentioned in dialogue and the associated video context, the model must determine which visible character bears that name. This task is challenging because name resolution often depends on cinematic cues such as shot–reverse-shot structure, self-introduction, direct address, third-party reference, gaze, and temporal continuity. We provide a dataset and annotations for evaluating character naming and first character appearances, develop a video-and-dialogue model for the task, and compare it with proprietary multimodal large language models. We further introduce a progressive framework for building a character bank as the film unfolds. Together, these contributions support online, human-like movie understanding in which character identities are inferred from the film itself. Website: https://www.robots.ox.ac.uk/~vgg/research/StreamCast/.

1 Introduction

The objective of this paper is to watch a film as a human would – ingesting the film frame-by-frame in an online mode, and identifying the principal characters and their names *as the story unfolds*. Traditional films, such as a romantic comedy, consist of characters and their story arcs. In order for the viewer to quickly understand who these principal characters are, the film must introduce and name them towards the start of the film. This is achieved by a combination of cinematic devices (such as the camera following the person, and close-ups) and using the dialogue or other information to name them, as illustrated in Fig. 1.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Obtaining character names explicitly only from the film is in stark contrast to the methods employed recently in the literature for tasks like generating audio descriptions and transcripts [21, 24–26, 43, 67, 68]. In these works, a *character bank* is first constructed using a cast list and actor photos from IMDB, Fandom and other online sources. The character bank is then used to label the principal characters throughout the film using face recognition (or voice recognition for animated films [21]).

To operationalize our goal, we first investigate how the viewer is presented with the name of a character as they watch the film – almost always it is from the dialogue, and very occasionally from other sources, such as name badges. This enables the *character naming* task to be automated and evaluated – given a name in the dialogue, the associated shots can be used to ask the question: which of the people appearing in this clip has this name? To this end, the first contribution of this paper is to build a dataset from MovieNet [28] and Movie101 [73] films that can be used to train and evaluate this task. We develop a method to solve this task, using the video and dialogue, and compare its performance to proprietary Multimodal LLMs (MLLMs), such as Gemini [19]. This is a challenging task, as correct inference often requires an understanding of film editing style (as the person addressed by name may be shown in a different shot) and also character gaze and eyeline (to know who they are looking at when saying a name).

This establishes that characters can be named in films, using the information presented by the screenwriter and director. Often, the name of a principal character is presented multiple times across different dialogues throughout the film, providing opportunities for error correction (by both human and machine) if the first presentation had been missed or incorrectly assigned. Once the principal characters are named (possibly multiple times), they can be recognized throughout the rest of the movie, and retrospectively named from their first appearance (using human or machine memory). To this end, we introduce two further tasks: (i) annotating the movies with the first appearance of the principal characters (which may well be before they are named in the dialogue); and (ii) progressively building up a *character bank* as the film progresses. The character bank records the appearance, voice and names of the principal characters (as in previous work). We provide annotations to evaluate the character first appearance over a number of movies. For example, in the movie *Juno*, the protagonist *Juno* is introduced in the first shot of the film, and appears in almost every scene, but is not named until 7.2 minutes into the film.

In summary, we make the following four contributions: (i) we introduce the new task of *character naming* from dialogue alone, a key enabling step for watching and understanding a film as a human does; (ii) we provide the Movie Character Naming (MCN) dataset with $23{,}792$ instances of multiple-choice character naming over 407 movies, and also 20 movies with the first appearance of the principal characters annotated; (iii) we develop a character naming model using open-source MLLMs, and compare its performance to proprietary MLLMs, analysing what is difficult to solve; and (iv) we introduce the first automatic method to progressively build up a character bank for a feature-length movie. These capabilities put in place the foundation for understanding character arcs and, ultimately, the story of a film. The dataset and models are available at: `https://www.robots.ox.ac.uk/~vgg/research/StreamCast/`.

2 Movie Character Naming (MCN) Dataset

In this section, we introduce the Movie Character Naming (MCN) Dataset, which provides annotations for two complementary tasks: *Character Naming* and *First Occurrence*. The *Character Naming* annotations consist of video segments in which character names are uttered in the dialogue, each paired with ground-truth name-to-track mappings. The *First Occurrence* annotations, by contrast, are designed to evaluate the streaming pipeline for character bank construction, and provide two (possibly coinciding) timestamps per principal character: the moment they first appear on screen, and the moment their name is first revealed.

Fig. 2 shows a representative MCN sample for the character *Carrie* from *Four Weddings and a Funeral*. The movie timeline is annotated with *Character Naming* and *First Occurrence* labels. Additional examples are provided in Appendix H. In the following, we first describe how the dataset is annotated given character tracks and dialogue. The implementation details for obtaining the character face-tracks and dialogue are given in Sec. 2.3.

2.1 Data Collection and Curation

We build MCN on top of MovieNet [28] and Movie101 [73], which already provide ground-truth mappings between character names and body bounding-box coordinates.

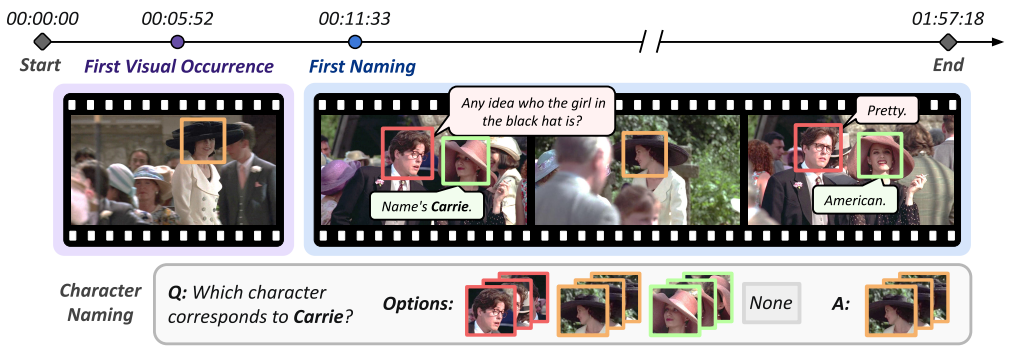


Figure 2: **Example annotations from the MCN dataset.** The MCN dataset provides manually annotated timestamps for each character's first visual occurrence and first naming, along with ground-truth name-to-face correspondences for the *Character Naming* task.

Table 1: **Statistics of the Character Naming (CN) and First Occurrence (FO) annotations.** Each question corresponds to a name-to-track mapping query or a first-occurrence timestamp annotation.

Task	*Split*	*# Movies*	*Duration (h)*	*# Characters*	*# Questions*
CN	Train	377	108.13	1853	22210/21150
CN	Test	30	6.51	226	1582/1399
FO	Test	20	39.98	287	574

Annotations for Character Naming (CN). We construct the associations between character names and face tracks in three steps. *First,* since MovieNet and Movie101 already provide ground-truth name–body associations, we align character body boxes with their face tracks in space and time, yielding name–face associations for *all* characters appearing on screen. *Second,* we keep only those associations whose corresponding names are also mentioned in the dialogue, since these are the only ones inferable from the audio–visual signal we use. *Third,* we handle the case where a name is mentioned in the dialogue but the corresponding face track is unavailable (e.g., when the character is referred to in absentia). In this case, we label the association as `none`, since no in-clip cues support the assignment. As a result, we provide the final name–face associations as ground-truth labels, along with all character-level face tracks in the video clip as candidate options, to construct a multiple-choice instance.

Annotations for First Occurrence (FO). For every character in the movie, we manually annotate two timestamps: the *first visual occurrence*, i.e. the moment the character first appears on screen, and the *first naming*, i.e. the moment at which a human viewer can first associate the character's visual appearance with their name based solely on the preceding content.

Quality Control. To ensure a high-quality test set, we manually inspect every test sample and correct errors in face-track clustering and name labelling. We then run state-of-the-art proprietary models on the curated set and examine cases where their outputs disagree with the annotation, fixing any remaining errors (or confirming the annotation is correct but the model fails).

2.2 Dataset Statistics

The statistics of both annotation types in the MCN dataset are summarised in Tab. 1. For the CN task, the training splits comprise clips from 377 training movies (108.13 hours), covering 1,853 characters and 22,210 name-to-track questions, of which 21,150 have non-`none` answers. The training movies are disjoint from those in both the CN and FO test sets, ensuring no movie-level leakage across splits. The CN test split contains clips from 30 movies (6.51 hours), covering 226 characters and 1,582 manually verified questions, of which 1,399 have non-`none` answers. In particular, the CN questions in the test set can be broken down by evidence type: approximately 12% involve self-introduction, 70% involve direct address by other characters, 16% involve third-party reference, and 2% require reasoning from textual cues in the movie, such as name badges.

The FO annotations form a test set of 20 movies (39.98 hours) with 287 annotated characters, yielding 574 timestamp annotations in total. Each character has two annotations, corresponding to the first visual occurrence and the first naming. Together, the MCN dataset spans a diverse range of genres, casts, and narrative styles, ensuring that our evaluation captures performance across varied cinematic settings.

2.3 Preprocessing – Character-Level Face-Tracks and Associated Speech

To obtain multiple-choice instances for *Character Naming*, we need to extract character-level tracks and speech from video clips around name occurrences in the dialogue, therefore enabling us to ask models to select one of them for each query name. This involves extracting a face-track cluster f_i for each character and linking it with the corresponding voice-track cluster v_j. Note, this pre-processing is done at the clip level when annotating the dataset, but results are also made available over the entire movie for downstream tasks (e.g., character bank construction).

Face Track Extraction. For each recurring character i, we detect and track faces across shots and group the tracks into a character-level face-track cluster f_i. The detailed procedure is given in Appendix A.1.

Linking Voices to Face Tracks. For the clean voice segments in the video, we first apply speaker diarization to group them into voice clusters, with each cluster denoted v_j. To match each voice cluster to an existing face track (i.e. to identify who is speaking on screen), we apply TalkNet [57] to obtain synchronization scores $s_{i,j}$ between faces f_i and voices v_j. Each voice cluster is then assigned to the face-track cluster with the highest synchronization score, yielding complete character identities with paired faces and voices.

As a further preprocessing step, we transcribe the voice tracks into textual dialogue with Parakeet [47]. Since each voice track is linked to a face cluster, this naturally yields visually grounded subtitles. Finally, the character names n emerge as a natural by-product of subtitle extraction. We retain clips whose subtitles contain at least one character name, passing them to the subsequent character naming step.

3 Assigning Names to Characters

The task of assigning names to characters, i.e., Character Naming (CN), is performed naturally by humans while watching a film, but is not trivial for deep learning systems owing to multiple challenges: first, a character may be addressed directly by name, or referred to in the third person while off-screen, or revealed only through what others say about them; second, who is being addressed may only be indicated by gaze or by the editing structure, for example, as one character says a name, and the named character then reacts in the subsequent shot. We therefore treat CN as a new multimodal task that requires joint reasoning over textual cues (e.g., conversational logic, vocative address) and visual cues (e.g., eye contact, on-screen presence).

Formally, given a video clip V containing a set of face-track clusters $\{f_i\}$ (each optionally linked to a voice cluster v_j, as described in Sec. 2.3) and a candidate name n extracted from the dialogue, the Character Naming (CN) task is to predict an assignment $\psi(n) \in \{f_i\} \cup \{\varnothing\}$. Here, $\psi(n) = f_i$ indicates that the name n refers to the character represented by face-track cluster f_i, while $\psi(n) = \varnothing$ indicates that none of the visible characters corresponds to n (e.g., the named character is off-screen).

In this section, we first describe how we operationalize Character Naming for MLLMs through visual prompting alone (Sec. 3.1). We then introduce a fine-tuning recipe that improves the MLLMs' reasoning on this task, enabling more reliable assignments of character names (Sec. 3.2).

3.1 MLLM-Based Character Naming

To adapt MLLMs for character naming in a zero-shot manner, we use the following prompting strategies.

Prompting the MLLM with Visual Identities. Rather than providing the MLLM with bounding-box coordinates of the face tracks, we adopt the color-based visual prompting scheme of Xie et al. [67]. Each face track is assigned a unique color, and its bounding boxes are drawn in that color on the raw video frames. The boxes are then referred to by color in the text prompt, allowing the MLLM to associate identities with their spatial locations.

Prompting the MLLM with Visually Grounded Subtitles. Subtitles are embedded directly into the text prompt, with each speaker referred to by the color tag of their associated face track. We additionally include timestamps to enable more temporally accurate inference. For example, a line spoken by the character with blue face bounding boxes would be formatted as `<693.2s> [BLUE] *"Any idea who the girl in the black hat is?"* <695.0s>`.

Prompting the MLLM for Character Naming. Given the visual identities and visually grounded subtitles, we directly ask the MLLM to match each character name from the subtitles with a color,

e.g., "*Which color corresponds to John Constantine?*" The color code thus bridges each visual identity's spatial position to its corresponding name.

3.2 Fine-tuning the MLLM for Character Naming

Following the prompting strategies described above, we adopt a two-stage R1-style fine-tuning recipe [22] to further improve the naming accuracy.

Stage I: Supervised Fine-tuning (SFT). The first stage is a cold start that supervises the MLLM with synthetic reasoning traces. These traces are generated by a teacher MLLM, which is given the video frames, visually grounded subtitles, and ground-truth name-to-color mapping, and asked to produce a reasoning trace that leads to the correct answer. This teaches the student MLLM to follow the desired output format (i.e. a reasoning trace followed by the answer) before the reinforcement learning stage.

Stage II: Reinforcement Learning. The second stage applies GRPO [22] with a composite reward $R = r_f + r_a$, where r_f rewards adherence to the expected response structure and r_a measures the correctness of the predicted name–face pairs. Letting $\hat{i}(n)$ and $i^*(n)$ denote the face track assigned to name n under the prediction $\hat{\psi}$ and the ground truth ψ^*, we set $r_a = \sum_{n \in N} \mathbf{1}[\hat{i}(n) = i^*(n)]$. The GRPO advantage and policy update follow the standard formulation [22], summarized in Appendix B.

A naive application of GRPO to this task suffers from a subtle shortcut: the model can latch onto color-specific cues. For instance, it may always associate the protagonist with "red" simply because "red" is the most frequently used color in the training distribution. We address this with a simple but effective *color-permutation augmentation*: during RL training, we randomly permute the color assignment for each sample. The ground-truth name–face mapping is invariant under such permutations, so consistent rewards across permuted variants are achievable only by reasoning about identity itself. This regularizes the policy toward color-agnostic reasoning and, empirically, accounts for a significant share of our gains over the cold-start baseline.

4 Streaming Pipeline for Character Bank Construction

Given a full movie, our goal is to construct a character bank in an online manner, taking the shot as the elementary unit. The *character bank* $\mathcal{B}_t$ serves as a memory of all characters observed so far. At each shot t, it aggregates the corresponding character information as triplets (f,n,v), where f is a face-track cluster, n a character name, and v a voice cluster.

Formally, the bank is updated auto-regressively (as illustrated in Fig. 3):

$$\mathcal{B}_t = \phi_{\mathcal{B}}(\mathcal{B}_{t-1}, \psi(V_t)), \tag{1}$$

where V_t denotes the video clip at shot t, and ψ is the MLLM fine-tuned for CN, which outputs character names and corresponding face tracks and voices $\{(n,f,v)\}$. We also retain identities with no assigned name, i.e. $\{(\varnothing,f,v)\}$, mirroring how a human viewer can visually remember a character even before they are introduced by name. These triplets are then merged into the previous bank by the update operator $\phi_{\mathcal{B}}$.

Overall, this streaming pipeline maintains and auto-regressively updates the character bank $\mathcal{B}$. We detail the bank update process in Sec. 4.1 and introduce an error-correction mechanism in Sec. 4.2 that keeps the update robust to error accumulation over multiple steps.

4.1 Character Bank Update

Given the names and corresponding face and voice tracks $\{(n,f,v)\}$ extracted from a new shot (with n possibly empty), we merge them into the existing entries of the previous bank in two steps.

Track Assignment. We treat face tracks as the primary signal and assign each f_i to an existing face cluster in the bank. We extract the corresponding face embedding e_i and find the closest cluster by distance to its embedding centroid $\bar{e}_b$ as in Eq. 2. A smaller distance indicates greater similarity between the face embedding e_i and the cluster centroid $\bar{e}_b$.

$$b^* = \arg\min_{b \in \mathcal{B}_{t-1}} d_{\cos}(e_i, \bar{e}_b). \tag{2}$$

If $d_{\cos}(e_i, \bar{e}_{b^*}) < \tau_f$, the track is assigned to entry b^*. Otherwise, we fall back to voice as the secondary signal: if the shot's speaker embedding u_i matches an entry's voice centroid $\bar{u}_b$ under a tighter distance

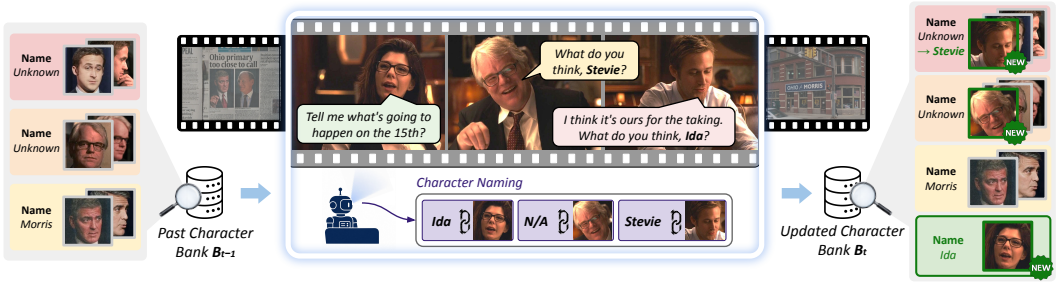


Figure 3: **Streaming pipeline for character bank construction.** As new shots arrive, we extract observations (character tracks, subtitles, and mentioned names) and reason jointly over them to link face tracks to names. Each step incrementally updates the bank through three operations: *naming* an existing cluster (*Stevie*, first cluster), *enriching* a cluster with additional face exemplars (second cluster), and *instantiating* a new entry for a previously unseen character (fourth cluster). The sample is drawn from *The Ides of March* (2011).

threshold $\tau_v < \tau_f$, the track is assigned there. If neither threshold is satisfied, a new entry is instantiated in the bank. Upon assignment, the matched centroids are updated.

Name Binding. We then bind a name to each entry in the character bank. For each entry, we maintain a histogram over the names associated with its face tracks across all observed shots, and use the majority-voted name as its canonical name. This handles aliases and nicknames; for example, in *Juno*, the character *Juno* may be addressed as *Mama Bear*, *Home Skillet*, or *General MacGuff*. Majority voting helps recover the most representative name for each character. It also improves robustness to noisy predictions from the character naming module, since occasional erroneous name assignments are unlikely to dominate the histogram.

Retrospective Assignment is a special case of name binding: when the CN module assigns a name to a previously unnamed face-track cluster, the new name is added to that cluster's histogram with a count of one, and the canonical name is updated accordingly. Thus, clusters that are initially unnamed can still acquire names once sufficient naming evidence becomes available in later shots.

4.2 Error Correction during Character Bank Construction

The Markov structure of the update process $\phi_{\mathcal{B}}$ makes it susceptible to error accumulation from upstream components (e.g. transcription, active-speaker detection, and identity grounding). To mitigate this, we introduce three error correction mechanisms deployed along the streaming chain. First, we use an MLLM to filter out noisy or ambiguous face tracks before clustering, ensuring high-quality embeddings. Second, we use an LLM to filter out query names from the subtitles that are unlikely to refer to on-screen characters. Third, we resolve cross-entry name conflicts after each shot using a greedy voting scheme that enforces one-to-one name assignment. Together, these mechanisms address distinct error modes and stabilize the evolution of $\mathcal{B}_t$ without revisiting past observations. We refer the reader to Appendix C for further details.

5 Experiments

This section presents our evaluation. We first introduce the metrics (Sec. 5.1), then evaluate each task. Sec. 5.2 evaluates our character naming (CN) module on the MCN test set against state-of-the-art open- and closed-source models. Sec. 5.3 evaluates the full streaming character bank pipeline, measuring how well our method tracks each character's first visual appearance and the first moment it can be linked to a name, and compares the final character bank against the IMDB ground truth. We ablate key design choices, including the two-stage fine-tuning strategy and data augmentation for CN, and the error correction mechanisms for the character bank. Implementation details and ablations are provided in Appendix D and Appendix E.

5.1 Evaluation Metrics

Character Naming Accuracy. We evaluate for two settings. *Overall accuracy* is computed over all name queries, including those labelled `none` when the mentioned character cannot be linked to

Table 2: **Character naming accuracy on the MCN test set.** We report overall accuracy (including `none` labels) and known accuracy (restricted to queries with a valid face track). Our fine-tuned Qwen3VL-4B-Thinking closes most of the gap to proprietary frontier models despite being substantially smaller in terms of model size.

Model	*Propriet.*	*Fine-Tuned*	*Overall Acc.*	*Known Acc.*
Random Baseline	×	×	29.90	33.74
Gemini-3-Flash-Preview	✓	×	79.57	86.62
Gemini-3.1-Pro-Preview	✓	×	**82.47**	**89.69**
Qwen3VL-4B-Thinking	×	×	52.15	56.68
Qwen3VL-8B-Instruct	×	×	53.60	58.83
Qwen3.5-9B	×	×	61.57	67.69
Qwen3VL-8B-Instruct (FT)	×	✓	64.92	71.77
Qwen3VL-4B-Thinking (SFT + RL)	×	✓	**72.76**	**81.06**

any on-screen face track. *Known accuracy* is computed only over queries whose ground-truth label is a valid face track, thereby examining the model's ability to correctly ground on-screen characters.

Character Bank Mean Temporal IoU. To evaluate the character bank streaming pipeline, we propose a *Mean Temporal IoU* (mTIoU) metric that measures how closely the bank tracks ground-truth human knowledge of each character over time.

For a movie of duration T and character c, let $H_c \subseteq [0,T]$ denote the interval during which a human viewer is aware of c, and $P_c \subseteq [0,T]$ the interval during which the model correctly identifies c in the bank. We define $\mathrm{TIoU}(c) = |H_c \cap P_c| / |H_c \cup P_c|$, which penalizes both *late* identification and *incorrect* identification (e.g. an entry later overwritten with a wrong name). To reflect narrative importance, we weight each character by their subtitle mention frequency w_c and report $\mathrm{mTIoU} = \sum_c w_c \mathrm{TIoU}(c) / \sum_c w_c$.

We instantiate two variants from the MCN timestamps: the *visual* variant mTIoU_v uses $H_c = [t_c^v, T]$ with t_c^v the first visual occurrence of c, and the *name* variant mTIoU_n uses $H_c = [t_c^n, T]$ with t_c^n the first naming timestamp (when a human can first link a face to a name).

Principal Character Recall (PCR@K). To evaluate the quality of the final character bank, we propose *Principal Character Recall* at K (PCR@K), measuring how reliably the pipeline recovers a movie's most important characters with both correct names and correct visual exemplars.

For each movie, we take the top-K principals from IMDB, ranked by their narrative importance. A ground-truth principal c is *recovered* iff some single entry $b \in \mathcal{B}$ satisfies both of the following conditions: (i) the canonical name of b phonetically matches the IMDB name of c, and (ii) the distance between its face centroid $\bar{e}_b$ and the IMDB reference embedding $\bar{e}_c^*$ is below the threshold τ_{eval}. We report per-movie recall as the fraction of recovered principals and average across movies.

5.2 Results on Character Naming

We evaluate Character Naming (CN) with three sets of methods. First, we compare against several baselines, including a random baseline that uniformly selects from the candidate colors, non-proprietary open-source MLLMs (Qwen3VL-4B-Thinking, Qwen3VL-8B-Instruct, and Qwen3.5-9B [2]), and a naive fine-tuning baseline obtained by directly fine-tuning Qwen3VL-8B-Instruct on the MCN training set. Second, we report the result of our CN model, which is obtained by two-stage R1-style fine-tuning of Qwen3VL-4B-Thinking on the MCN training set. Third, we compare against state-of-the-art proprietary MLLMs: Gemini-3-Flash-Preview and Gemini-3.1-Pro-Preview [19]. All models receive the same inputs with a visually prompted video clip, visually grounded subtitles, and queried character names, and are asked to output the corresponding box colors.

Results. We report Character Naming results in Tab. 2. The pretrained Qwen3VL-8B-Instruct achieves 53.60% overall accuracy, improving to 64.92% via naive fine-tuning. In contrast, our two-stage SFT+GRPO approach with *color-permutation augmentation*, applied to a smaller 4B-Thinking backbone, yields a $+7.84\%$ gain over the naive 8B baseline. Despite the smaller model size, our approach closes most of the gap to Gemini-3.1-Pro-Preview and clearly outperforms all open-source alternatives. Qualitative examples are shown in Fig. 4.

Discussion. Compared with open-source models such as Qwen3.5-9B, our two-stage fine-tuned model more accurately names characters in direct address scenarios (e.g. shot–reverse-shot, telephone

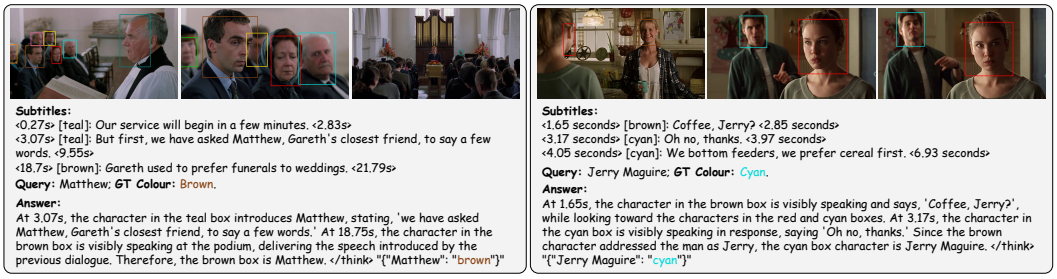


Figure 4: **Qualitative results for Character Naming.** Examples are from *Four Weddings and a Funeral* and *Jerry Maguire*. In each case, our default "Qwen3VL-4B-Thinking (SFT + RL)" model reasons jointly over subtitles and video frames to link a queried name to the correct face — in the first, by following a verbal introduction to a later podium shot; in the second, by resolving direct address across a shot–reverse-shot exchange.

Table 3: **Character naming accuracy on the MCN test subset from unseen movies.** We report the accuracy results on a test subset derived from unseen movies (released after the models were trained) by all evaluated models to isolate the effects of knowledge leakage. Overall accuracy includes `none` labels, and known accuracy is restricted to queries with a valid face track.

Model	*Propriet.*	*Fine-Tuned*	*Overall Acc.*	*Known Acc.*
Random Baseline	×	×	18.42	20.93
Gemini-3-Flash-Preview	✓	×	78.62	**81.34**
Gemini-3.1-Pro-Preview	✓	×	**80.50**	78.36
Qwen3VL-8B-Instruct	×	×	45.28	45.52
Qwen3.5-9B	×	×	50.94	54.48
Qwen3VL-4B-Thinking (SFT + RL)	×	✓	**64.78**	**71.64**

conversations), better exploits visual cues (e.g. pointing, handshakes), and improves temporal grounding between subtitles and frames. A qualitative comparison is provided in Appendix I.

Nevertheless, a gap remains to the strongest proprietary model, likely driven by errors in transcription and active-speaker detection. We hypothesise that models such as Gemini [19] can partially recover from these errors by leveraging lip motion, whereas our model is more sensitive to such upstream noise.

There are also cases where both Gemini [19] and our model fail. In particular, both struggle with (i) subtle cues (e.g. brief pointing among multiple characters), (ii) complex scenes with motion, and (iii) over-predicting on-screen characters when the ground-truth character is absent. We refer readers to Appendix I for further analysis of these failure modes.

Evaluation on Unseen Movies. We also observe that proprietary models and pre-trained open-source models might have seen the movies that we test on. Therefore, they could infer the character identities without using cues from the clips but from their previous knowledge. This introduces a risk of data contamination, potentially leading to an overestimation of the models' actual capabilities on our benchmark. To address this concern, we collect 102 manually annotated clips from four films released from March 2026 onward (*Voicemails for Isabelle*, 19 June 2026; *In the Grey*, 15 May 2026; *The Death of Robin Hood*, 19 June 2026; *Heartstopper Forever*, 17 July 2026), after the reported training cut-offs of both Gemini-3.1-Pro (February 2026) and Qwen3-VL (September 2025), yielding 159 character-naming questions, 25 of which have none as the answer. Neither model can therefore have encountered these films during pre-training.

The results are given in Tab. 3, and it can be seen that the relative ordering of methods given in Tab. 2 is unchanged on both metrics. Our fine-tuned 4B model again outperforms all open-source baselines by a wide margin and trails only the proprietary Gemini models. Accuracy drops for all methods by a margin comparable to that of the random baseline, reflecting the greater difficulty of these clips, which frequently feature fast-moving action and dense background crowds, compared to the MCN test set used in Tab. 2. Our model retains most of its original accuracy, confirming that it relies on in-clip reasoning rather than prior familiarity with the films.

Table 4: **Results on the MCN First Occurrence dataset.** We report mean Temporal IoU (mTIoU) against two reference intervals, first visual occurrence and first naming, and Principal Character Recall (PCR@K) on the final character bank for $K \in \{5,10,15\}$. Our method operates in a streaming setting, whereas proprietary baselines are given full-movie access.

Method	*Streaming*	*mTIoU$_v$*	*mTIoU$_n$*	*PCR@5*	*PCR@10*	*PCR@15*
Gemini-3-Flash-Preview	×	51.81	30.92	35.00	32.04	29.84
Gemini-3.1-Pro-Preview	×	49.05	30.10	44.00	37.15	34.93
Ours	✓	**84.90**	**54.40**	**63.00**	**54.54**	**48.19**

5.3 Results on Streaming Character Bank Construction

Baseline. For streaming character bank construction, we use Gemini-3-Flash-Preview and Gemini-3.1-Pro-Preview [19] as baselines, prompting them to output each character's name, first visual occurrence timestamp, and first naming timestamp. To obtain face clusters, we ask the models to identify timestamps with close-up or clear solo shots, from which we perform face detection, extract features, and cluster faces, selecting the largest connected component as the predicted cluster. As these models are not designed to directly output face crops or clusters, we consider a prediction correct if any face in the cluster matches the ground-truth identity.

Results. As shown in Tab. 4, our pipeline substantially outperforms both proprietary baselines across all metrics, despite operating in the more challenging streaming setting. These improvements reflect more accurate identification of when characters first appear and are named (mTIoU), as well as stronger alignment of the final character bank with canonical identities from external sources (PCR). These gains are especially notable given that the baselines have full-movie access while ours processes the video in a single streaming pass. Appendix J shows two examples of the final constructed character bank.

Discussion. Our pipeline performs well on movies such as *Juno* and *Meet the Parents*, which feature relatively simple scenes with few participants and a limited set of recurring characters. In these cases, identity clustering is generally reliable, aided by our face-track filtering. Even when Character Naming errors occur, our method can often recover as additional evidence emerges over time.

Performance degrades, however, in movies with many scenes containing numerous minor or background characters. In *Harry Potter and the Goblet of Fire*, for example, background characters introduce noise into clustering, which distracts the Character Naming process. Movies with very large named casts, such as *Four Weddings and a Funeral*, with over 50 named characters among hundreds of recurring faces, are also particularly challenging, as the model must maintain and disambiguate many identities over long temporal contexts.

6 Related Work

Character-Centric Movie Understanding. Movie understanding has emerged as a milestone task for video understanding, owing to its long temporal span and rich multimodal narrative structure, motivating a variety of datasets [3, 28, 55, 59, 61]. A representative line of research is *Audio Description (AD) generation* [21, 24–26, 31, 48, 67, 68], where Han et al. [25] identifies character recognition as central to the task. Existing work typically takes one of two shortcuts: relying on curated sources like IMDB cast lists [24–26, 43, 67, 68] or fan wikis [21], which are often incomplete or unavailable for new films, or sidestepping naming entirely with placeholders like "Person A" [18, 56, 75]. A separate line of work links names to faces directly from video. Early approaches align shooting scripts with subtitles for weak labels and use active-speaker detection to associate them with face tracks [15, 54]. Extensions add probabilistic models over face, speech, and clothing cues [58], coreference resolution [45], and script-free dialogue-based naming for recurring TV casts [12, 27, 40]. A complementary line assumes identities are known and analyses interactions via gaze and eyelines [20, 36, 37, 46]. We revisit this problem in the era of MLLMs. Prior dialogue-based approaches, including Haurilet et al. [27], cast naming as an optimisation over many mentions across clips or episodes of a recurring TV cast; we instead resolve identity from the single clip surrounding the name as a human viewer does, generalising to feature-length movies in the wild where names are typically mined from one-off dialogue. In concurrent work, Pu et al. [44] track identities via textual appearance descriptions, a lossy representation that fails to distinguish visually similar characters and degrades under costume or lighting changes. Our character bank instead grounds each identity in visual exemplars, retaining the fine-grained cues text abstracts away.

Visual Reasoning in Vision-Language Models (MLLMs). Recent work extends reasoning from Large Language Models (LLMs) [32, 65] to the multimodal setting. Early approaches prompt frozen MLLMs to produce intermediate reasoning steps [39, 62], while more recent methods train explicit reasoning via reinforcement learning. Inspired by [22], Group Relative Policy Optimization (GRPO) has been adapted to MLLMs, showing that RL with verifiable rewards improves reasoning ability [29, 38, 51, 74], including across perception tasks such as detection [35], segmentation [34], and fine-grained perception [72]. For video understanding specifically, prior work incorporates temporal reasoning into this paradigm, e.g. via temporal-aware GRPO [16], joint spatio-temporal reasoning [33], and long-form temporal grounding [63]. In parallel, visual prompting introduces structured cues, such as colored masks [69], boxes [6, 53], region tokens [10, 71], or identity markers [67], to link visual entities with language. Our approach builds on both lines of work, combining color-based visual prompting with a two-stage SFT + GRPO fine-tuning strategy to enable an MLLM to reason about character identity from visual and dialogue cues jointly.

Person Clustering in Long Videos. Grouping person appearances by identity is a long-standing problem in video understanding, with applications to cast listing, character-based browsing, and story understanding [11, 15, 17, 60]. Early work focuses on *face clustering*, using metric or representation learning to group face tracks under variations in pose, lighting, and expression [50, 52, 64, 70]. These methods exploit spatio-temporal structure in edited videos via co-occurrence constraints [4, 30, 66], or apply nearest-neighbor clustering over pretrained embeddings [42, 49]. However, they rely on visible faces, limiting performance when characters are occluded, off-screen, or viewed from behind. To address this, Brown et al. [5] introduce Video Person-Clustering, grouping face, body, and voice cues via multimodal hierarchical clustering, with later work extending this direction using graph-based formulations [41]. Our pipeline follows a similar multimodal approach, using face embeddings as the primary signal with voice as fallback. Unlike prior work, we operate in a *streaming* setting, where clusters are instantiated, updated, and named as the film unfolds, extending online clustering [1, 7] to feature-length videos.

7 Discussion

In this paper, we introduce the task of *Character Naming* from dialogue and close the gap between open-source models and the best proprietary models. Building on this task, we further introduce a streaming pipeline to construct a *character bank*, enabling online, human-like movie understanding. For both tasks, we compare our method with Gemini [19], though neither task is fully solved, even by proprietary models. A natural next step is to develop models or an agentic framework to overcome the current failure modes, and thereby improve performance.

Acknowledgements

This research is supported by UK EPSRC Programme Grant VisualAI (EP/T028572/1), a Royal Society Research Professorship RSRP\R\241003, and Clarendon Scholarships.

References

[1] Charu C Aggarwal, S Yu Philip, Jiawei Han, and Jianyong Wang. A framework for clustering evolving data streams. In *Proceedings 2003 VLDB conference*, pages 81–92. Elsevier, 2003.

[2] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[3] Max Bain, Arsha Nagrani, Andrew Brown, and Andrew Zisserman. Condensed movies: Story based retrieval with contextual embeddings. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[4] Martin Bauml, Makarand Tapaswi, and Rainer Stiefelhagen. Semi-supervised learning with constraints for person identification in multimedia data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3602–3609, 2013.

[5] Andrew Brown, Vicky Kalogeiton, and Andrew Zisserman. Face, body, voice: Video person-clustering with multiple modalities. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3184–3194, 2021.

[6] Mu Cai, Haotian Liu, Siva Karthik Mustikovela, Gregory P Meyer, Yuning Chai, Dennis Park, and Yong Jae Lee. Vip-llava: Making large multimodal models understand arbitrary visual prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12914–12923, 2024.

[7] Feng Cao, Martin Estert, Weining Qian, and Aoying Zhou. Density-based clustering over an evolving data stream with noise. In *Proceedings of the 2006 SIAM international conference on data mining*, pages 328–339. SIAM, 2006.

[8] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

[9] Brandon Castellano. PySceneDetect: Video scene cut detection and analysis tool. `https://github.com/Breakthrough/PySceneDetect`, 2024. Accessed: 2026-04-19.

[10] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023.

[11] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Unsupervised metric learning for face identification in tv video. In *2011 International Conference on Computer Vision*, pages 1559–1566. IEEE, 2011.

[12] Timothee Cour, Benjamin Sapp, Akash Nagle, and Ben Taskar. Talking pictures: Temporal grouping and dialog-supervised person recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1014–1021. IEEE, 2010.

[13] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *CVPR*, 2020.

[14] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv preprint arXiv:2005.07143*, 2020.

[15] Mark Everingham, Josef Sivic, and Andrew Zisserman. Hello! my name is... buffy"–automatic naming of characters in tv video. In *BMVC*, volume 2, page 6, 2006.

[16] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025.

[17] Andrew Fitzgibbon and Andrew Zisserman. On affine invariant clustering and automatic cast listing in movies. In *European Conference on Computer Vision*, pages 304–320. Springer, 2002.

[18] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24108–24118, 2025.

[19] Google DeepMind. Gemini 3 Pro model card. Technical report, Google, December 2025. URL `https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Model-Card.pdf`.

[20] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6047–6056, 2018.

[21] Zhongrui Gui, Junyu Xie, Tengda Han, Weidi Xie, and Andrew Zisserman. Character-centric understanding of animated movies. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 3300–3309, 2025.

[22] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[23] Jiangyu Han, Petr Pálka, Marc Delcroix, Federico Landini, Johan Rohdin, Jan Cernockỳ, and Lukáš Burget. Efficient and generalizable speaker diarization via structured pruning of self-supervised models. *arXiv preprint arXiv:2506.18623*, 2025.

[24] Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. Autoad: Movie description in context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18930–18940, 2023.

[25] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13645–13655, 2023.

[26] Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. Autoad iii: The prequel-back to the pixels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18164–18174, 2024.

[27] Monica-Laura Haurilet, Makarand Tapaswi, Ziad Al-Halah, and Rainer Stiefelhagen. Naming tv characters by watching and analyzing dialogs. In *2016 IEEE winter conference on applications of computer vision (WACV)*, pages 1–9. IEEE, 2016.

[28] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *European conference on computer vision*, pages 709–727. Springer, 2020.

[29] Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Xu Tang, Yao Hu, and Shaohui Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*, 2025.

[30] Vicky Kalogeiton and Andrew Zisserman. Constrained video face clustering using 1nn relations. *https://ora.ox.ac.uk/objects/uuid:23f369b3-794b-42a6-a6af-ab4b0d726ba4/files/spn89d688d*, 2020.

[31] Eshika Khandelwal, Junyu Xie, Tengda Han, Max Bain, Arsha Nagrani, Andrew Zisserman, Gül Varol, and Makarand Tapaswi. More than a moment: Towards coherent sequences of audio descriptions. *arXiv preprint arXiv:2510.25440*, 2025.

[32] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

[33] Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. Videochat-r1: Enhancing spatio-temporal perception via reinforcement fine-tuning. *arXiv preprint arXiv:2504.06958*, 2025.

[34] Yuqi Liu, Bohao Peng, Zhisheng Zhong, Zihao Yue, Fanbin Lu, Bei Yu, and Jiaya Jia. Seg-zero: Reasoning-chain guided segmentation via cognitive reinforcement. *arXiv preprint arXiv:2503.06520*, 2025.

[35] Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-rft: Visual reinforcement fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2034–2044, 2025.

[36] Manuel J Marin-Jimenez, Vicky Kalogeiton, Pablo Medina-Suarez, and Andrew Zisserman. Laeo-net: revisiting people looking at each other in videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3477–3485, 2019.

[37] Manuel Jesús Marin-Jimenez, Andrew Zisserman, Marcin Eichner, and Vittorio Ferrari. Detecting people looking at each other in videos. *International Journal of Computer Vision*, 106(3):282–296, 2014.

[38] Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, et al. Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025.

[39] Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. Compositional chain-of-thought prompting for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14420–14431, 2024.

[40] Arsha Nagrani and Andrew Zisserman. From benedict cumberbatch to sherlock holmes: Character identification in tv series without a script. *arXiv preprint arXiv:1801.10442*, 2018.

[41] Arsha Nagrani, Samuel Albanie, and Andrew Zisserman. Learnable pins: Cross-modal embeddings for person identity. In *Proceedings of the European conference on computer vision (ECCV)*, pages 71–88, 2018.

[42] Charles Otto, Dayong Wang, and Anil K Jain. Clustering millions of faces by identity. *IEEE transactions on pattern analysis and machine intelligence*, 40(2):289–303, 2017.

[43] Stefano Pini, Marcella Cornia, Federico Bolelli, Lorenzo Baraldi, and Rita Cucchiara. M-vad names: a dataset for video captioning with naming. *Multimedia Tools and Applications*, 78(10):14007–14027, 2019.

[44] Junfu Pu, Yuxin Chen, Teng Wang, and Ying Shan. Omniscript: Towards audio-visual script generation for long-form cinematic video. *arXiv preprint arXiv:2604.11102*, 2026.

[45] Vignesh Ramanathan, Armand Joulin, Percy Liang, and Li Fei-Fei. Linking people in videos with "their" names using coreference resolution. In *European conference on computer vision*, pages 95–110. Springer, 2014.

[46] Adria Recasens, Carl Vondrick, Aditya Khosla, and Antonio Torralba. Following gaze in video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1435–1443, 2017.

[47] Dima Rekesh, Nithin Rao Koluguri, Samuel Kriman, Somshubra Majumdar, Vahid Noroozi, He Huang, Oleksii Hrinchuk, Krishna Puvvada, Ankur Kumar, Jagadeesh Balam, et al. Fast conformer with linearly scalable attention for efficient speech recognition. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8. IEEE, 2023.

[48] 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.

[49] Saquib Sarfraz, Vivek Sharma, and Rainer Stiefelhagen. Efficient parameter-free clustering using first neighbor relations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8934–8943, 2019.

[50] Vivek Sharma, Makarand Tapaswi, M Saquib Sarfraz, and Rainer Stiefelhagen. Self-supervised learning of face representations for video face clustering. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pages 1–8. IEEE, 2019.

[51] Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.

[52] Shuai Shen, Wanhua Li, Zheng Zhu, Guan Huang, Dalong Du, Jiwen Lu, and Jie Zhou. Structure-aware face clustering on a large-scale graph with 107 nodes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9085–9094, 2021.

[53] Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. What does clip know about a red circle? visual prompt engineering for vlms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11987–11997, 2023.

[54] Josef Sivic, Mark Everingham, and Andrew Zisserman. "who are you?"-learning person specific classifiers from video. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1145–1152. IEEE, 2009.

[55] 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.

[56] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232, 2024.

[57] Ruijie Tao, Zexu Pan, Rohan Kumar Das, Xinyuan Qian, Mike Zheng Shou, and Haizhou Li. Is someone speaking? exploring long-term temporal features for audio-visual active speaker detection. In *Proceedings of the 29th ACM international conference on multimedia*, pages 3927–3935, 2021.

[58] Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. "knock! knock! who is it?" probabilistic person identification in tv-series. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2658–2665. IEEE, 2012.

[59] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4631–4640, 2016.

[60] Makarand Tapaswi, Marc T Law, and Sanja Fidler. Video face clustering with unknown number of clusters. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5027–5036, 2019.

[61] Paul Vicol, Makarand Tapaswi, Lluis Castrejon, and Sanja Fidler. Moviegraphs: Towards understanding human-centric situations from videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8581–8590, 2018.

[62] Yaoting Wang, Shengqiong Wu, Yuecheng Zhang, Shuicheng Yan, Ziwei Liu, Jiebo Luo, and Hao Fei. Multimodal chain-of-thought reasoning: A comprehensive survey. *arXiv preprint arXiv:2503.12605*, 2025.

[63] Ye Wang, Ziheng Wang, Boshen Xu, Yang Du, Kejun Lin, Zihan Xiao, Zihao Yue, Jianzhong Ju, Liang Zhang, Dingyi Yang, et al. Time-r1: Post-training large vision language model for temporal video grounding. *arXiv preprint arXiv:2503.13377*, 2025.

[64] Zhongdao Wang, Liang Zheng, Yali Li, and Shengjin Wang. Linkage based face clustering via graph convolution network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1117–1125, 2019.

[65] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[66] Baoyuan Wu, Siwei Lyu, Bao-Gang Hu, and Qiang Ji. Simultaneous clustering and tracklet linking for multi-face tracking in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 2856–2863, 2013.

[67] Junyu Xie, Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. Autoad-zero: A training-free framework for zero-shot audio description. In *Proceedings of the Asian Conference on Computer Vision*, pages 2265–2281, 2024.

[68] Junyu Xie, Tengda Han, Max Bain, Arsha Nagrani, Eshika Khandelwal, Gül Varol, Weidi Xie, and Andrew Zisserman. Shot-by-shot: Film-grammar-aware training-free audio description generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16503–16513, 2025.

[69] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023.

[70] Lei Yang, Xiaohang Zhan, Dapeng Chen, Junjie Yan, Chen Change Loy, and Dahua Lin. Learning to cluster faces on an affinity graph. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2298–2306, 2019.

[71] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. *arXiv preprint arXiv:2310.07704*, 2023.

[72] En Yu, Kangheng Lin, Liang Zhao, Jisheng Yin, Yana Wei, Yuang Peng, Haoran Wei, Jianjian Sun, Chunrui Han, Zheng Ge, et al. Perception-r1: Pioneering perception policy with reinforcement learning. *arXiv preprint arXiv:2504.07954*, 2025.

[73] Zihao Yue, Qi Zhang, Anwen Hu, Liang Zhang, Ziheng Wang, and Qin Jin. Movie101: A new movie understanding benchmark. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4669–4684, 2023.

[74] Hengguang Zhou, Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh. R1-zero's" aha moment" in visual reasoning on a 2b non-sft model. *arXiv preprint arXiv:2503.05132*, 2025.

[75] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, et al. Mlvu: Benchmarking multi-task long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13691–13701, 2025.

Appendix

This Appendix contains the following sections:

- In Appendix A, we elaborate on the preprocessing stage, covering face track extraction and clustering as well as the linking of voice clusters to face tracks via active-speaker synchronization.
- In Appendix B, we present the full GRPO formulation used to fine-tune our character-naming MLLM, including the composite reward, advantage normalization, and policy update.
- In Appendix C, we provide detailed descriptions of the error correction mechanism during streaming character bank construction.
- In Appendix D, we specify implementation details and hyperparameter settings for clustering, fine-tuning, and streaming character bank construction.
- In Appendix E, we conduct ablation studies on the fine-tuning strategy, color-permutation augmentation, and error-correction mechanisms to verify key design choices.
- In Appendix F, we introduce an additional baseline for streaming character bank construction that natively produces face exemplars, and is therefore better matched to our output format than the MLLM baselines.
- In Appendix G, we provide additional details on the MCN dataset collection, including the coarse-to-fine alignment pipeline used to recover MovieNet annotations at native resolution.
- In Appendix H, we present representative dataset examples illustrating the diversity of films and naming cues covered by the MCN benchmark.
- In Appendices I and J, we provide additional qualitative results for character naming and streaming character bank construction, including comparisons with baselines and failure cases.
- In Appendix K, we describe the compute resources used for the experiments, including MLLM fine-tuning and inference.
- In Appendix L, we discuss the broader impacts of our work, including potential risks and the scope of the released artefacts.
- We also include a *supplementary video* that demonstrates our character naming results and visualizes the streaming character bank construction process on the movie *Juno*.

A Details on Preprocessing

This section provides additional details on the preprocessing stage of Sec. 2.3. We first describe how face tracks are detected, extracted, and merged into face clusters (Sec. A.1), and then explain how voice clusters are linked to these face clusters via active-speaker synchronization (Sec. A.2).

A.1 Face Track Extraction and Clustering

To obtain complete and accurate face tracks, we detect shot boundaries using SceneDetect [9] and query SAM 3 [8] with the text prompt "face" at the beginning, middle, and end of each shot to handle occlusions. We then extract face features with InsightFace [13] by sampling 15 frames per track. Following Brown et al. [5], we cluster the tracks into face clusters $\{f_i\}$ under two constraints: (1) a *spatio-temporal cannot-link constraint*, preventing tracks with (partial) temporal overlap from being grouped, since co-occurring tracks in the same frame must correspond to distinct characters; and (2) a *nearest-neighbor distance constraint*, which requires the distance between the features of merged tracks to fall below a threshold τ_{cluster}.

A.2 Linking Voices to Face Tracks

We transcribe movie dialogue using Parakeet [47] and group voice segments into voice clusters $\{v_j\}$ via speaker diarization [23]. To match each voice cluster to a face cluster, we apply TalkNet [57] to compute a per-frame active-speaker synchronization score $s_{k,t}$, indicating how likely face track k is speaking at timestamp t. We rely on diarization because TalkNet, which is predominantly trained on frontal faces, fails when speakers are off-screen or not facing the camera.

For each voice cluster v_j with subtitle lines $\{e\}$ (each e has start and end times t_e^{s} and t_e^{e}), we aggregate the per-frame scores at the cluster level to obtain an association score $S_{i,j}$ between face cluster f_i and voice cluster v_j:

$$s_{e,i} = \max_{\text{track } k \in f_i} \frac{1}{|\mathcal{T}_{k,e}|} \sum_{t \in \mathcal{T}_{k,e}} s_{k,t}, \quad S_{i,j} = \max_{e \in v_j} s_{e,i}, \quad \mathcal{T}_{k,e} = \{t : t_e^{\mathrm{s}} \le t \le t_e^{\mathrm{e}}\}. \tag{3}$$

We assign each v_j to the face cluster $f_{i^*(j)}$ with the highest score, $i^*(j) = \mathrm{argmax}_i S_{i,j}$, provided that $S_{i^*(j),j}$ exceeds a threshold τ_{ASD}. As an additional override, whenever $s_{k,t}$ exceeds τ_{override} for some face track k within the time span of subtitle line e, we directly assign e to the face cluster containing k, bypassing the diarization-based grouping. This handles cases where TalkNet provides high-confidence per-frame evidence that contradicts the diarization output.

B GRPO Reinforcement Learning Fine-Tuning

This section provides the full GRPO formulation summarised in Sec. 3.2.

Reward. We apply GRPO [22] with a composite reward that combines a format reward r_f and an accuracy reward r_a. The format reward r_f is granted whenever the output o_i adheres to the expected response structure. The accuracy reward r_a measures how well the predicted set of name–identity pairs $\hat{\psi}(V)$ agrees with the ground-truth pairs $\psi^*(V)$. Letting $\hat{i}(n)$ and $i^*(n)$ denote the identities paired with name n under $\hat{\psi}$ and ψ^*, respectively, we set

$$r_a = \sum_{n \in N} \mathbf{1}\left[\hat{i}(n) = i^*(n)\right], \tag{4}$$

and define the full reward as $R_i = r_f + r_a$.

Advantage and Policy Update. The advantage A_i is computed by normalizing rewards within each group of G sampled responses:

$$A_i = \frac{R_i - \mathrm{mean}(\{R_j\})}{\mathrm{std}(\{R_j\})}. \tag{5}$$

Following Guo et al. [22], the policy is updated by maximising

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^{G} \min(\rho_i(\theta)A_i, \mathrm{clip}(\rho_i(\theta), 1-\epsilon, 1+\epsilon)A_i) - \beta D_{\mathrm{KL}}(\pi_\theta \| \pi_{\text{ref}})\right], \tag{6}$$

where $\rho_i(\theta) = \pi_\theta(o_i \mid q)/\pi_{\theta_{\text{old}}}(o_i \mid q)$ is the importance ratio, ϵ is the clipping threshold, and β controls the strength of the KL regularisation against the reference policy π_{ref}.

C Error Correction for Character Bank Construction

Additional Details on Error Mitigation. As outlined in Sec. 4.2, the Markovian update process $\phi_{\mathcal{B}}$ is inherently sensitive to upstream noise, where errors from components such as transcription, active-speaker detection, and identity grounding can propagate over time. To address this, we elaborate on the three interventions applied along the streaming pipeline: face-track filtering, intra-entry name merging, and inter-entry name disambiguation.

Face-Track Filtering. Prior to clustering, we introduce an MLLM-based filtering stage to discard unreliable or ambiguous face tracks, thereby maintaining clean and identifiable clusters in the bank. This step removes low-quality observations, including blurred faces and extreme profile views. Empirically, embeddings derived from the retained high-quality tracks are sufficiently stable, rendering clustering errors negligible under our operating thresholds.

Query Name Filtering. As previously discussed, when no on-screen character actually corresponds to the candidate name, both our model and Gemini tend to over-assign one of the visible faces to an off-screen name rather than abstaining. To circumvent this, we filter the candidate name pool upstream so that off-screen names rarely reach the naming step in the first place. Concretely, we use an LLM (Qwen3.5-9B) to inspect each subtitle window and keep only those candidate names likely to refer to a character currently on screen. This requires reasoning over the flow of the surrounding dialogue rather

than classifying each line in isolation, since whether a named person is actually present or merely being discussed is rarely decidable from a single utterance.

Inter-Entry Name Disambiguation. Ideally, each name should be uniquely associated with a single entry; however, MLLM-induced errors can lead to conflicts across entries. We resolve these after each shot using a greedy voting procedure: for names assigned to multiple entries, the entry with the highest vote retains the name, while others revert to their next-best candidates. This process is repeated until convergence.

Overall, these mechanisms address complementary failure modes in the streaming process and prevent error accumulation as $\mathcal{B}_t$ evolves, without requiring access to past observations.

D Implementation Details

We set the face-clustering distance threshold to $\tau_{\text{cluster}} = 0.4$, the active-speaker association threshold to $\tau_{\text{ASD}} = -0.075$, and the per-frame override threshold to $\tau_{\text{override}} = 0.1$. We use DiariZen-Large-s80-v2 [23] as our diarization model.

For training the MLLM for speaker identity recognition, we adopt Qwen3-VL-4B-Thinking [2] as our base model. To construct the first-stage supervised fine-tuning (SFT) dataset, we sample approximately 10% of the training set and prompt Gemini-3-Flash-Preview [19] to generate reasoning traces that lead to the given ground-truth name-to-track mappings. For SFT, we fine-tune only the language model component of Qwen3-VL-4B-Thinking while keeping both the vision encoder and the multimodal projector frozen, for one epoch with a batch size of 4, a learning rate of 1×10^{-5}, and a cosine schedule with warmup ratio 0.03. Then, we perform GRPO fine-tuning on the remaining training set for one epoch, using a learning rate of 1×10^{-6}, a batch size of 4, $G = 4$ sampled responses per prompt, and a KL coefficient of $\beta = 0.004$.

For streaming character bank construction, we set the face matching threshold to $\tau_f = 0.55$ and the voice fallback threshold to $\tau_v = 0.25$. Voice embeddings are extracted with ECAPA-TDNN [14] over speaker-aligned subtitle segments; both face and voice centroids are maintained as L2-normalized running means over all cluster members. Candidate names are extracted from subtitles using spaCy transformer NER pipelines (`en_core_web_trf`), and off-screen mentions are filtered out by Qwen3.5-9B [2]. The remaining names are linked to face clusters by our fine-tuned MLLM, which receives frames sampled at 2 FPS. Low-quality faces are filtered by Qwen3-VL-8B-Instruct [2] with stricter prompting before they are committed to the bank.

For PCR@K evaluation, we set the face-matching threshold $\tau_{\text{eval}} = 0.55$ and use the Double Metaphone algorithm for phonetic matching between predicted canonical names and IMDB ground-truth names. The IMDB reference embedding $\bar{e}_c^*$ is the L2-normalized InsightFace embedding extracted from the actor's IMDB headshot.

No hyperparameter mentioned above was tuned on the test set. In the absence of a validation split, all thresholds were set following prior work: τ_{cluster}, τ_f and τ_v follow the multimodal clustering steup of Brown et al. [5], and τ_{ASD} and τ_{override} follow TalkNet [57]. τ_{eval} is set equal to τ_f (i.e. to 0.55), the value conventionally used to decide whether two InsightFace embeddings correspond to the same identity.

E Ablation Study

We ablate the key design choices of our method on the MCN dataset to isolate their individual contributions. Using Qwen3-VL-4B-Thinking as the base model, we evaluate the fine-tuning strategy and data augmentation for character naming. We also ablate on error correction mechanisms to measure their effects on the full streaming pipeline.

Comparing with Direct SFT on an Instruct Model. To assess whether our two-stage recipe can be replaced by directly fine-tuning a larger Instruct model, we compare against Qwen3VL-8B-Instruct trained with SFT on the answer labels. While SFT alone on the 4B Thinking model performs slightly worse than the 8B baseline (64.16% vs. 64.92%), adding the RL stage pushes the smaller model to 72.76% overall accuracy, clearly surpassing the 8B Instruct model. Reasoning-based training therefore provides gains that cannot be recovered by model scale alone.

Table 5: **Ablation study on Character Naming.** We compare our two-stage (SFT + RL) fine-tuning of Qwen3VL-4B-Thinking against direct SFT on a larger Qwen3VL-8B-Instruct model, and further ablate the contributions of the RL stage and color-permutation augmentation.

Model	*SFT*	*RL*	*Augment.*	*Overall Acc.*	*Known Acc.*
Qwen3VL-8B-Instruct	✓	×	×	64.92	71.77
Qwen3VL-4B-Thinking	✓	×	×	62.39	67.98
Qwen3VL-4B-Thinking	✓	×	✓	64.16	70.41
Qwen3VL-4B-Thinking	✓	✓	×	71.37	79.56
Qwen3VL-4B-Thinking	✓	✓	✓	**72.76**	**81.06**

Effect of Color-Permutation Augmentation. As shown in Tab. 5, perturbing the cluster-to-color mapping during training consistently improves performance across both training stages. At the SFT stage, it lifts overall accuracy from 62.39% to 64.16% and known accuracy from 67.98% to 70.41%. The gains persist after RL, with overall accuracy improving from 71.37% to 72.76% and known accuracy from 79.56% to 81.06%. These consistent improvements indicate that the augmentation prevents the model from exploiting spurious shortcuts tied to specific color assignments, encouraging it to genuinely reason about character identity from visual and dialogue cues.

Table 6: **Ablation of error-correction mechanisms.** We start from our full model and disable one mechanism at a time, measuring the impact on mTIoUs and PCR@Ks.

Variant	*mTIoU$_v$*	*mTIoU$_n$*	*PCR@5*	*PCR@10*	*PCR@15*
Ours (Full)	84.90	**54.40**	**63.00**	**54.54**	**48.19**
w/o Face-Track Filtering	83.30	52.27	60.00	53.01	47.47
w/o Query Name Filtering	**85.13**	52.16	57.00	49.46	44.99
w/o Inter-Entry Name Disambiguation	84.90	52.40	62.00	53.00	47.48

Error-Correction Mechanisms. Tab. 6 summarises the contribution of each error-correction stage. Removing face-track filtering degrades both mTIoU_v (from 84.90 to 83.30) and mTIoU_n (from 54.40 to 52.27), suggesting that noisy or unreliable tracks introduce low-quality face features that weaken face clustering and subsequently affect name assignment. Removing query name filtering does not reduce mTIoU_v (from 84.90 to 85.13), but substantially reduces mTIoU_n (from 54.40 to 52.16) and $\mathrm{PCR@10}$ (from 54.54 to 49.46). This indicates that off-screen mentions mainly contribute spurious candidate names, which interfere with the MLLM's name-binding step rather than character localisation. Finally, removing inter-entry name disambiguation primarily affects name assignment: mTIoU_n decreases from 54.40 to 52.40, while mTIoU_v remains unchanged and $\mathrm{PCR@}K$ decreases slightly (e.g., $\mathrm{PCR@5}$: $63.00 \rightarrow 62.00$). This shows that inter-entry disambiguation improves the consistency of name assignments across entries without changing localising characters' first occurrences. Overall, the ablation distinguishes two input-stage filtering mechanisms that suppress noisy evidence before name binding from an inter-entry resolution mechanism that refines the final name assignments.

F Additional Baseline for Streaming Character Bank Construction

In Sec. 5.3, we use Gemini-3-Flash-Preview and Gemini-3.1-Pro-Preview [19] as baselines for streaming character bank construction. We note that these baselines are disadvantaged by our requirement that they emit face exemplars, a form of output they are not designed to produce. No existing method targets this setting directly: prior work either presupposes a pre-built character bank or assumes a recurring television cast. Proprietary MLLMs are therefore the closest available point of comparison, notwithstanding the mismatch in output format. The lenient scoring criterion was adopted to compensate that.

To provide a comparison that does not rely on lenient scoring to compensate for the output-format mismatch, we have additionally implemented a classical weakly-supervised baseline [27]. It uses exactly the same face tracking, clustering, and majority-vote name binding as our pipeline, replacing only the mechanism by which names are linked to faces, and is given access to the full film rather than operating in a streaming pass. The weakly-supervised baseline uses subtitle text alone, with no audio, no pixels, and no vision-language model. A character vocabulary is extracted from the transcript by NER, and each mention is typed as first-, second-, or third-person using the five grammar rules

Table 7: **Results of the MIL baseline on the MCN First Occurrence dataset.** We report an additional baseline for streaming character bank construction.

Method	*Streaming*	*mTIoU$_v$*	*mTIoU$_n$*	*PCR@5*	*PCR@10*	*PCR@15*
Gemini-3-Flash-Preview	×	51.81	30.92	35.0	32.0	29.8
Gemini-3.1-Pro-Preview	×	49.05	30.10	44.0	37.2	34.9
MIL [27]	×	82.85	33.38	34.0	19.7	16.7
Ours	✓	**84.90**	**54.40**	**63.0**	**54.5**	**48.2**

in [27], so that first- and second-person mentions ("I'm Harry", "Hello, Harry") imply presence while third-person mentions ("Where is Harry?") imply absence. Each typed mention yields a positive or negative bag of face tracks from a surrounding window, here three shots centred on the mention's shot. A per-character linear MIL model (Normalized Set Kernel, the best-performing solver in [27]) is trained on InsightFace embeddings, and a second-stage classifier, trained on each model's most confident tracks, scores every track in the film and assigns the argmax name.

In Tab. 7, since the baseline shares our face tracking and clustering, it attains a comparable mTIoU_v (82.85 against 84.90), confirming that the two systems recover visual identities equally well. On the naming metrics, however, it falls far behind despite full-movie access, reaching 33.38 mTIoU_n against our 54.40 and 34.0 $\mathrm{PCR@5}$ against our 63.0. Dialogue cues under classical weak supervision are therefore insufficient to name characters reliably at feature-length scale, which is the gap our fine-tuned naming model closes.

G Details on Dataset Collection

MovieNet [28] provides body bounding-box annotations on sparse 240P keyframes indexed by (s,k), a shot index and within-shot image index. These indices do not identify frames in the native-resolution movie file, and MovieNet's shot segmentation does not coincide with ours. To build the MCN dataset, we must (i) recover the exact frame number $f^\star \in \{0,\dots,N-1\}$ in the full video for each annotated keyframe, and (ii) rescale every box from 240P to native resolution. We address this with a three-stage coarse-to-fine pipeline: CLIP retrieval for coarse localisation, longest-increasing-subsequence (LIS) filtering to remove inconsistent matches, and perceptual hashing for exact frame recovery.

CLIP Feature Extraction. For every MovieNet keyframe and for every 24th frame ($\approx$ 1 fps at 24 fps) of each full-resolution video, we extract a 512-D CLIP ViT-B/32 image embedding. The keyframes form a *query* pool $\mathcal{Q}$ and the video samples a *source* pool $\mathcal{S}$. Both pools are cached per movie so that the subsequent stages run independently per title.

Coarse Alignment via FAISS Retrieval and LIS Filtering. Within each movie we run exact L2 nearest-neighbor search (FAISS `IndexFlatL2`) of every query against $\mathcal{S}$ and take the top-1 source frame as a coarse anchor $\widehat{f}_i$ for keyframe i. CLIP retrieval is robust to the resolution gap and to codec and color differences between annotation crops and rendered video frames, but it occasionally returns a visually similar shot from elsewhere in the movie. We prune such errors by exploiting the fact that MovieNet keyframes are listed in temporal order: the sequence $(\widehat{f}_i)_i$ must be monotonically increasing. We compute its longest increasing subsequence in $O(n\log n)$ and mark every keyframe outside the LIS as unreliable ($\widehat{f}_i \leftarrow -1$). The criterion is strict but conservative: any rogue match must drop out of the LIS together with whichever neighbor it conflicts with, and the LIS retains the longer consistent chain.

Fine Alignment via Perceptual Hashing. The coarse stage anchors each keyframe to within ± 24 frames. To recover the exact frame we scan a window $[\widehat{f}_i - W, \widehat{f}_i + W]$ of the full video with $W = 150$. For each candidate frame and for the query we compute two 64-bit perceptual hashes on the CLAHE-normalized grayscale image: *pHash* (low-frequency DCT coefficients thresholded against their median) and *dHash* (horizontal gradient sign). Candidates are ranked by the combined Hamming distance $H_\mathrm{p} + H_\mathrm{d}$, and we accept the lowest-ranked frame satisfying both $H_\mathrm{p} \le 12$ and $H_\mathrm{d} \le 10$. If no candidate meets both thresholds we fall back to the lowest combined-distance frame in the window. An optional ORB+RANSAC inlier rerank on the top-M candidates disambiguates near-duplicates within the same shot. When a keyframe's coarse anchor was discarded by the LIS filter, we replace its window with one bracketed by the nearest two LIS-passing neighbors in the same movie, f_prev and f_next: we use $[\min(f_\mathrm{prev}, f_\mathrm{next}), \max(f_\mathrm{prev}, f_\mathrm{next})]$ when both exist, a $\pm W$ window around the single available neighbor when only one exists, and drop the keyframe entirely when neither exists.

Bounding Box Rescaling and Output. With the exact frame index $f_i^\star$ recovered, each MovieNet body box is rescaled from the per-row annotation resolution (W_q,H_q) to native video resolution (W_f,H_f) via axis-independent scale factors $s_x{=}W_f/W_q$ and $s_y{=}H_f/H_q$. The resulting aligned records, of the form $(s,k,f_i^\star,\mathbf{b}_i,\mathrm{pid})$ where $\mathbf{b}_i$ is the rescaled body box and pid is the MovieNet character identifier, serve as the automatic ground truth for the MCN dataset.

H Dataset Examples

Fig. 5 shows six representative annotations from the MCN dataset, drawn from films that span genres, eras, and visual styles — from the period crowds of *Gangs of New York* to the romantic comedy of *Sleepless in Seattle*, the political drama of *The Ides of March*, and the sports biopic *Moneyball*. This diversity is by design: a character-naming benchmark that generalizes beyond a single genre or production style must capture the full range of cinematographic conventions through which characters are introduced.

The examples also illustrate the variety of cues a model must exploit to resolve a name to the correct face. The most common case is *direct address*, in which one character names another in dialogue, but the visual realisation of this act is far from uniform. The named character may be identified through the canonical *shot–reverse-shot* structure, through an *eyeline match* when speaker and addressee share the frame, or through a *telephone conversation* where the named character is entirely off-screen at the moment of naming and must be linked to a face seen elsewhere in the clip. A separate class of examples requires reasoning from *on-screen text*: the character's name appears as a written cue — a name tag, a letter, a sign, or a screen — and the model must perform OCR and then ground the recovered string to the correct visual identity. Cases like these are unsolvable from dialogue alone and require multimodal reasoning over both spoken and written language alongside visual evidence.

I Qualitative Results for Character Naming

Qualitative Examples. Fig. 6 shows three representative examples covering the main cue types our model handles: *direct address*, where a speaker calls another character by name; *third-party reference*, where a name is mentioned about someone else in the scene; and *visual reasoning*, where the spoken cue is ambiguous and the mapping must be inferred from scene role. In each case, our model grounds the name to the correct colored box by jointly leveraging the spoken cue and the surrounding visual context.

Comparing with Baseline. We compare our output against a strong baseline on the same scene in Fig. 7. Our model produces a concise, well-grounded explanation and arrives at the correct identity, whereas the baseline generates a long chain of thought with repeated self-verifications yet commits to an incorrect mapping. Longer reasoning traces do not guarantee better grounding: our model achieves more accurate association with substantially fewer tokens.

Failure Cases. We analyse representative failure cases of our model and Gemini-3.1-Pro-Preview in Fig. 8 and Fig. 9. Fig. 8 shows cases where Gemini-3.1-Pro-Preview fails despite being the strongest proprietary baseline, suggesting that it can struggle to use subtle visual actions and interactions that are crucial for character naming. Fig. 9 further compares common failure modes across both models. Overall, our model appears better at leveraging fine-grained visual cues, speaker activity, and cross-turn dialogue context, while remaining errors are often caused by subtle actions, ambiguous speaker grounding, or noisy subtitles.

J Qualitative Results for Streaming Character Bank Construction

We present qualitative results for streaming character bank construction on two films, *Juno* and *Meet the Parents*. As shown in Fig. 10, our pipeline constructs clean character entries for the major characters in *Juno*, grouping representative face tracks and linking them to canonical names while also preserving discovered aliases. Fig. 11 shows a second example on *Meet the Parents*. The bank again identifies the most prominent characters, but also reveals typical failure cases in complex multi-person scenes, where ambiguous visual context and transcription errors can lead to incorrect or noisy name bindings. These examples demonstrate that the proposed bank construction procedure is effective across different films, while remaining errors are primarily due to challenging name evidence rather than face-track clustering. *Moreover, we refer readers to the supplementary video, which shows the full streaming character bank construction process for Juno.*

K Experiment Compute Requirements

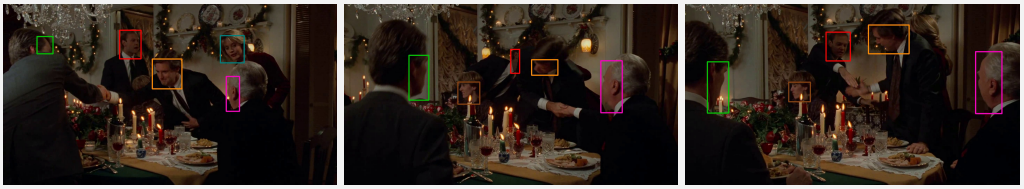
Fine-tuning of the MLLM is performed using four NVIDIA Ada 6000, A40, or L40S GPUs, each with 48 GB of memory. All inference on the MCN character-naming test set is conducted on a single NVIDIA Ada 6000, A40, or L40S GPU with 48 GB of memory. Construction of the streaming character bank also uses four NVIDIA Ada 6000, A40, or L40S GPUs.

L Broader Impacts

Our method identifies characters in movies using cinematic and dialogue cues. This technique can be used to generate accurate audio descriptions and speaker-attributed subtitles, improving accessibility for audiences with visual or hearing impairments. This method does not perform open-world face recognition or release any identity gallery from personal video sources. We therefore do not foresee significant negative societal impacts from this work.



Figure 5: **Examples from the MCN dataset.** Six character-naming annotations illustrating the range of problem types covered: direct address with shot–reverse-shot structure, direct address with eyeline match, direct address over the telephone, on-screen text, and OCR-based reasoning. Clips are drawn from *Moneyball*, *Gangs of New York*, *Sleepless in Seattle*, *The Help*, *Four Weddings and a Funeral*, and *The Ides of March*.



Subtitles:
<10.64 seconds> [unknown]: I don't know if we're gonna be able to do it. <11.28 seconds>
<12.64 seconds> [red]: Congratulations, Walter. <15.12 seconds>

Query: Walter; **GT Colour:** Orange.

Answer:
At 12.64s, the character in the red box is visibly speaking as he says 'Congratulations, Walter.' He is looking directly at the character in the orange box as they shake hands, identifying the orange box character as Walter. This is consistent with the orange box character's prominent role in the scene as the guest of honor. </think> "{"Walter": "orange"}"



Subtitles:
<0.08 seconds> [unknown]: Dear friends, what a joy it is to welcome you to our church on this wonderful day for Angus and Laura. <7.28 seconds>

Query: Laura the Bride, Angus the Groom; **GT Colour:** Blue, Green.

Answer:
At 0.08s, the off-camera speaker addresses guests saying 'what a joy it is to welcome you to our church on this wonderful day for Angus and Laura'. This establishes the man and woman in the center as the bride and groom. At 0.00s, the character in the blue box is the wedding woman walking down the aisle with a bouquet, confirming blue is Laura the Bride. At 0.00s, the man in the green box is the groom walking directly beside her, confirming green is Angus the Groom. Therefore, the groom is the green box, and the bride is the blue box. </think> "{"Laura the Bride": "blue", "Angus the Groom": "green"}"

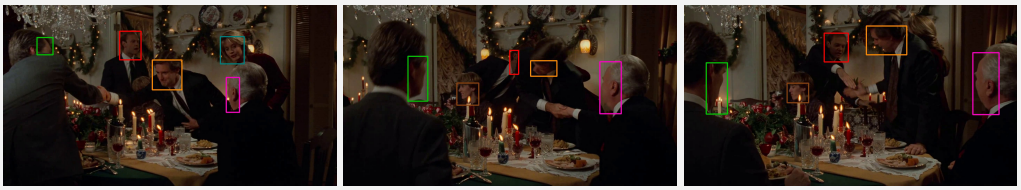


Subtitles:
<0.27 seconds> [pink]: You see when Jack had to retire for health reasons. <3.47 seconds>
<3.87 seconds> [brown]: That's a bunch of malarkey honey. <5.07 seconds>
<5.15 seconds> [brown]: I'm fine. <5.47 seconds>

Query: Jack Byrnes; **GT Colour:** Brown.

Answer:
At 0.27s, the character in the pink box is visibly speaking and refers to 'Jack' in the third person ('You see when Jack had to retire...'), which means pink is not Jack Byrnes. At 3.87s, the character in the brown box is visibly speaking in direct response, confirming he is the 'Jack' being referenced. Thus, Jack Byrnes corresponds to the brown box. </think> "{"Jack Byrnes": "brown"}"

Figure 6: **Qualitative results for character naming.** We associate character names with their facial appearances in movies, mirroring how audiences infer who is who while watching. Characters may be introduced through diverse cues, including *direct address by other characters* (top), *visual reasoning with common knowledge* (middle), and *third-party reference* (bottom). In each case, the model grounds the name to the correct colored box by combining the spoken cue with visual evidence such as gaze, gesture, and scene role. Correctly linking character identities is central to movie understanding, enabling coherent character tracking across scenes. The examples are taken from *Sleepless in Seattle*, *Four Weddings and a Funeral* and *Meet the Parents*.

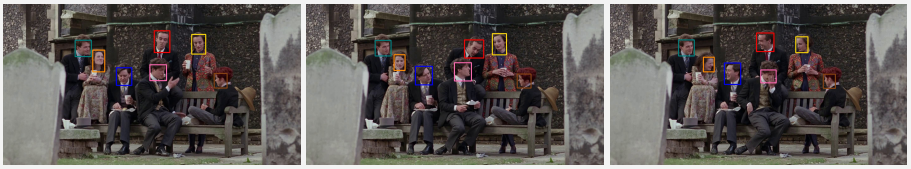


Subtitles:
<10.64 seconds> [unknown]: I don't know if we're gonna be able to do it. <11.28 seconds>
<12.64 seconds> [red]: Congratulations, Walter. <15.12 seconds>
Query: Walter; **GT Colour:** Orange.

Answer (Ours):
At 12.64s, the character in the red box is visibly speaking as he says 'Congratulations, Walter.' He is looking directly at the character in the orange box as they shake hands, identifying the orange box character as Walter. This is consistent with the orange box character's prominent role in the scene as the guest of honor. </think> "{"Walter": "orange"}"

Answer (Qwen3.5-9B):
The user wants me to identify the character names associated with specific colored bounding boxes in the video.
1. Analyze the subtitles: <10.64s> [unknown]: I don't know if we're gonna be able to do it. <12.64s> [red]: Congratulations, Walter. So, the speaker at 12.64 seconds is identified as "red" and says "Congratulations, Walter." This implies the person being congratulated is Walter.
2. Analyze the visual content: At 12.64 seconds (around frame 00:12), the person speaking is the man in the suit standing up, shaking hands. He is highlighted with a red box. The person being congratulated is sitting at the table. In the frames leading up to this (e.g., 00:00 - 00:10), there is a man sitting on the left side of the table (from the viewer's perspective) who is the focus of the handshake. Let's look at the handshake. The man in the red box (standing) shakes hands with the man in the green box (sitting on the left). Wait, let's re-examine the handshake sequence. At 00:03, a man in a white shirt (green box) extends his hand. At 00:04, a man in a dark suit (orange box) reaches out. (Similar Lines Ignored for Space.)
Wait, let's look at the subtitles again. <12.64s> [red]: Congratulations, Walter. The speaker is the man in the red box. He is standing. Who is he talking to? He is looking towards the man in the green box. The man in the green box is the one sitting at the head of the table (or near it). The man in the green box is the one being congratulated. Therefore, the man in the green box is Walter.
Let's check the other boxes. Red box: Man standing, speaking "Congratulations, Walter". Green box: Man sitting, being congratulated. So, Green = Walter. Orange box: Man sitting next to the woman in teal. He is shaking hands with the green box man. Teal box: Woman standing/sitting next to the orange box man. Brown box: Woman sitting next to the orange box man. Pink box: Man sitting on the right. Blue box: Woman sitting on the left.
The prompt asks to map character names to circle colors. The only name provided in the candidate list is "Walter". The subtitle says "[red]: Congratulations, Walter." This implies the person named Walter is the one being addressed. Visually, the red box man is looking at and shaking hands with the green box man. Therefore, the green box man is Walter.
Let's double check. At 12.64s, the red box man says "Congratulations, Walter." The red box man is looking at the green box man. So, Walter corresponds to the green box. Is there any other name? No, only "Walter" is in the candidate list. So I need to output {"Walter": "green"}.
Wait, let me re-read the prompt carefully. "Infer character identities ONLY from: 1) The spoken conversation in the subtitles 2) Visual consistency between speaking faces and their colored circles" "Return a JSON object mapping character names to circle colors." So, if the name is Walter, and the color is green, the output should be {"Walter": "green"}.
Let's verify the visual consistency. The red box man is the speaker. He says "Congratulations, Walter." He is looking at the green box man. The green box man is the one receiving the congratulations. So Green = Walter. Is it possible the red box man is Walter? "Congratulations, Walter." usually implies the speaker is congratulating someone else named Walter. If the speaker was Walter, he would say "Congratulations to me" or someone else would say it to him. So the speaker (Red) is not Walter. The listener (Green) is Walter.
Let's check if there are other names. The candidate list only has "Walter". So I only need to identify Walter.
Final check — Subtitle: [red]: Congratulations, Walter. Speaker: Red box. Addressee: Green box (visually). Name: Walter. Mapping: Walter → green. {"Walter": "green"}

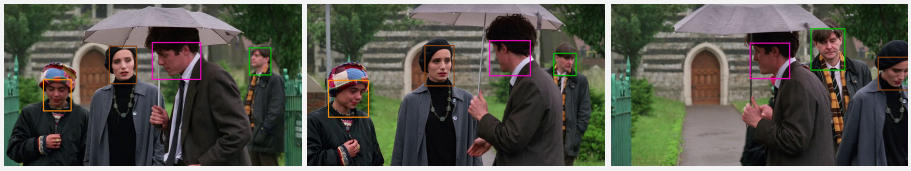
Figure 7: **Comparison of reasoning efficiency on character naming.** We compare our model against Qwen3.5-9B on the same scene. Our model produces a concise, well-grounded answer that directly links the spoken cue ("Congratulations, Walter") to visual evidence (gaze and handshake direction), correctly identifying the orange box as Walter. In contrast, the baseline generates a lengthy chain of thought with repeated re-examinations and self-verifications, yet arrives at an incorrect mapping (green). This illustrates that longer reasoning does not guarantee better grounding: our model achieves more accurate character naming with substantially fewer tokens.



Subtitles:
<0.32 seconds> [pink]: Well, I'd like to thank Fiona for those for those charming words about my future wife. <3.76 seconds>
<4.24 seconds> [pink]: And I'd like to take this opportunity before the day starts to read a little message from her to you all. <8.64 seconds>
<8.88 seconds> [blue]: Oh, exciting. <10.24 seconds>
Query: Fiona; **GT Colour:** Yellow.
Reason: The character pink points to the yellow character and then look directly at her when expressing gratitude to Fiona.
Answer (Ours):
At 0.32s, the character in the pink box is visibly speaking and addresses 'Fiona'. At 8.88s and 17.92s, the character in the blue box is visibly speaking, reacting to the message delivered by the pink speaker. In this scene, the character in the pink box reads a prepared message from Fiona, and the character in the blue box is being addressed. This confirms the character in the orange box is Fiona, who is reading her message aloud to the group in the flashback scene. By matching the central figure of the reading group, the orange box identifies the speaker Fiona. </think> "{"Fiona": "orange"}"
Answer (Gemini-3.1-Pro-Preview):
{"Fiona": "unknown"}



Subtitles:
<1.30 seconds> [orange]: I told you to put the camera low. <2.50 seconds>
<2.58 seconds> [orange]: Shoot me up from the bottom. <3.54 seconds>
Query: Rod Tidwell; **GT Colour:** None.
Reason: Insufficient evidence.
Answer (Ours):
At 1.30s, the character in the orange box is visibly speaking 'I told you to put the camera low.' and at 2.58s, he continues, 'Shoot me up from the bottom.' This dialogue matches the specific instruction given to the protagonist Rod Tidwell. The character in the teal box is not speaking or visible in this clip. Therefore, orange is Rod Tidwell. </think> "{"Rod Tidwell": "orange"}"
Answer (Gemini-3.1-Pro-Preview):
{"Rod Tidwell": "orange"}



Subtitles:
<0.00 seconds> [brown]: Charlie, I'll take the collar at home (I'll take Scarlett home), all right? <2.17 seconds>
<2.73 seconds> [none]: Yeah. <3.37 seconds>
Query: Scarlett; **GT Colour:** Orange.
Reason: Scarlett should be the character next to the speaker, as she says "I'll take Scarlet home".
Answer (Ours):
At 00:00, the character in the brown box is visibly speaking as she addresses her companion as 'Charlie' (subtitled as 'Charlie, I'll take the collar at home, all right?'). At 14:81, the character in the magenta box responds by saying 'Oh, Charlie' and continues a dialogue with the character in the green box, showing that magenta is Charlie. The character in the brown box is the female lead, the protagonist of the first sequence mentioned in the dialogue. By visual identification of the actress and her prominent role, she is mapped to brown. </think>"{"Scarlett": "brown"}"
Answer (Gemini-3.1-Pro-Preview*):
{"Scarlett": "brown"}

Figure 8: **Examples of failure cases for both our model and Gemini-3.1-Pro-Preview.** We show three examples in which Gemini-3.1-Pro-Preview, the strongest proprietary baseline, fails. These cases suggest that it struggles to capture subtle actions that are important for character naming. In the last example, Gemini is given the original audio rather than the erroneous subtitles for inference. The examples are taken from *Four Weddings and a Funeral* and *Jerry Maguire*.



Subtitles:
<0.34 seconds> [magenta]: We have met before. <1.46 seconds>
<1.78 seconds> [magenta]: Seriously? <2.42 seconds>
<4.18 seconds> [teal]: So I'd like to introduce you to my wife Emily. <6.90 seconds>
<7.30 seconds> [pink]: Hi, I'm Robbie's mom. <8.66 seconds>

Query: Emily; **GT Colour:** Pink.

Answer (Ours):
At 4.18s, the character in the teal box is visibly speaking and introduces the woman standing next to him as 'Emily'. This woman in the magenta box smiles and speaks during the introduction. At 7.30s, the character in the pink box responds by identifying themselves as 'Robbie's mom', implying the magenta box character is Emily. By 11.94s, the magenta character is visibly laughing 'How convenient' as the trio prepares to enter, confirming magenta is Emily. </think> "{"Emily": "magenta"}"

Answer (Gemini-3.1-Pro-Preview):
{"Emily": "pink"}

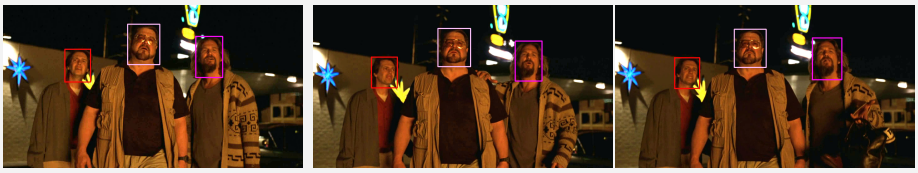


Subtitles:
<0.00 seconds> [orange]: Enough of this. Vader, release him. <1.83 seconds>
<1.92 seconds> [none]: As you wish. <3.38 seconds>

Query: Darth Vader; **GT Colour:** Cyan.

Answer (Ours):
At 0:00, the character in the orange box is visibly speaking, saying 'Enough of this, Vader, release him.' This is a direct address to the character in the black armor. At 1.92s, the character in the black armor (outlined in a cyan box) responds with 'As you wish,' indicating he is the one being addressed as Vader. Therefore, the character in the cyan box is Darth Vader. </think>"{\"Darth Vader\": \"cyan\"}"

Answer (Gemini-3.1-Pro-Preview):
{"Darth Vader": "unknown"}



Subtitles:
<1.26 seconds> [pink]: Who's the fucking novelist around here, you bunch of crybabies? <4.70 seconds>
<6.06 seconds> [magenta]: Hey. Cool, Walter. There never was any money. <9.18 seconds>

Query: Walter Sobchak; **GT Colour:** Pink.

Answer (Ours):
At 1.26s, the character in the pink box is visibly speaking while walking, as his jaw movement and the dialogue 'Who's the fucking novelist around here, you bunch of fucking crybabies?' match his line. At 6.06s, the character in the magenta box responds, 'Hey. Cool, Walter. There never was any money,' indicating he is addressing the pink speaker. This group interaction and dialogue confirm the character in the pink box is Walter Sobchak. </think> "{"Walter Sobchak": "pink"}"

Answer (Gemini-3.1-Pro-Preview):
{"Walter Sobchak": "unknown"}

Figure 9: **Examples of failure cases for either our model or Gemini-3.1-Pro-Preview.** We show three examples in which Gemini-3.1-Pro-Preview or our model fails under common failure modes. These cases suggest that our model is better at leveraging fine-grained visual cues, speaker activity, and cross-turn dialogue context to resolve character names, while remaining errors often arise from subtle actions, ambiguous speaker grounding, or noisy subtitles. The examples are taken from *Crazy, Stupid, Love.*, *Star Wars: Episode IV - A New Hope*, and *The Big Lebowski*.

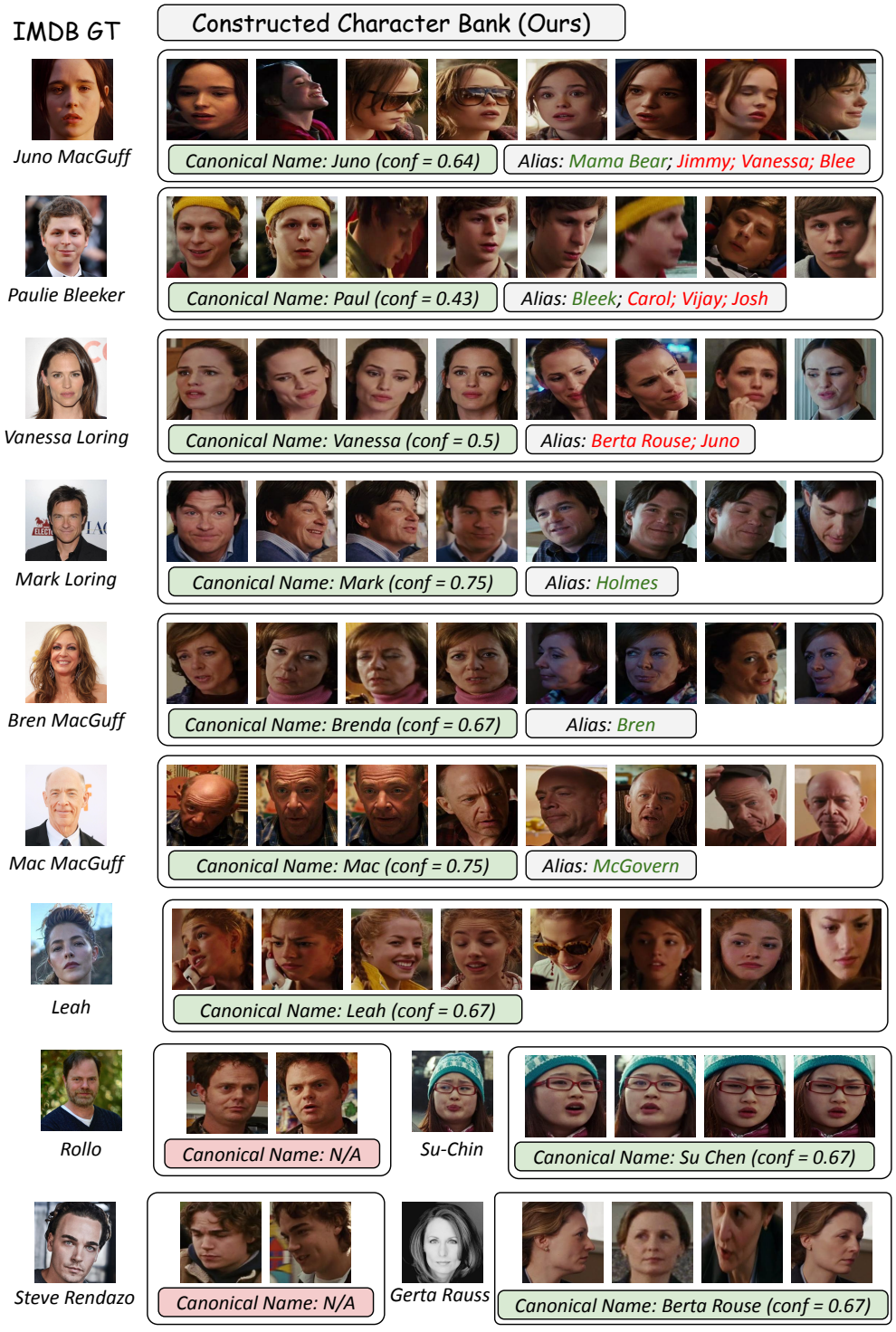


Figure 10: **Example of the constructed character bank for *Juno*.** We show the character bank constructed by our streaming pipeline. Each row (apart from the last two) corresponds to one character entry and includes representative face tracks, the predicted canonical name with confidence, and discovered aliases. The example illustrates that the bank captures major characters with clean visual clusters and resolves name variations, such as mapping *Mama Bear* to *Juno*, *Bren* to *Brenda*, and *McGovern* to *Mac*. Some naming errors remain, typically due to noisy or ambiguous character-name associations.

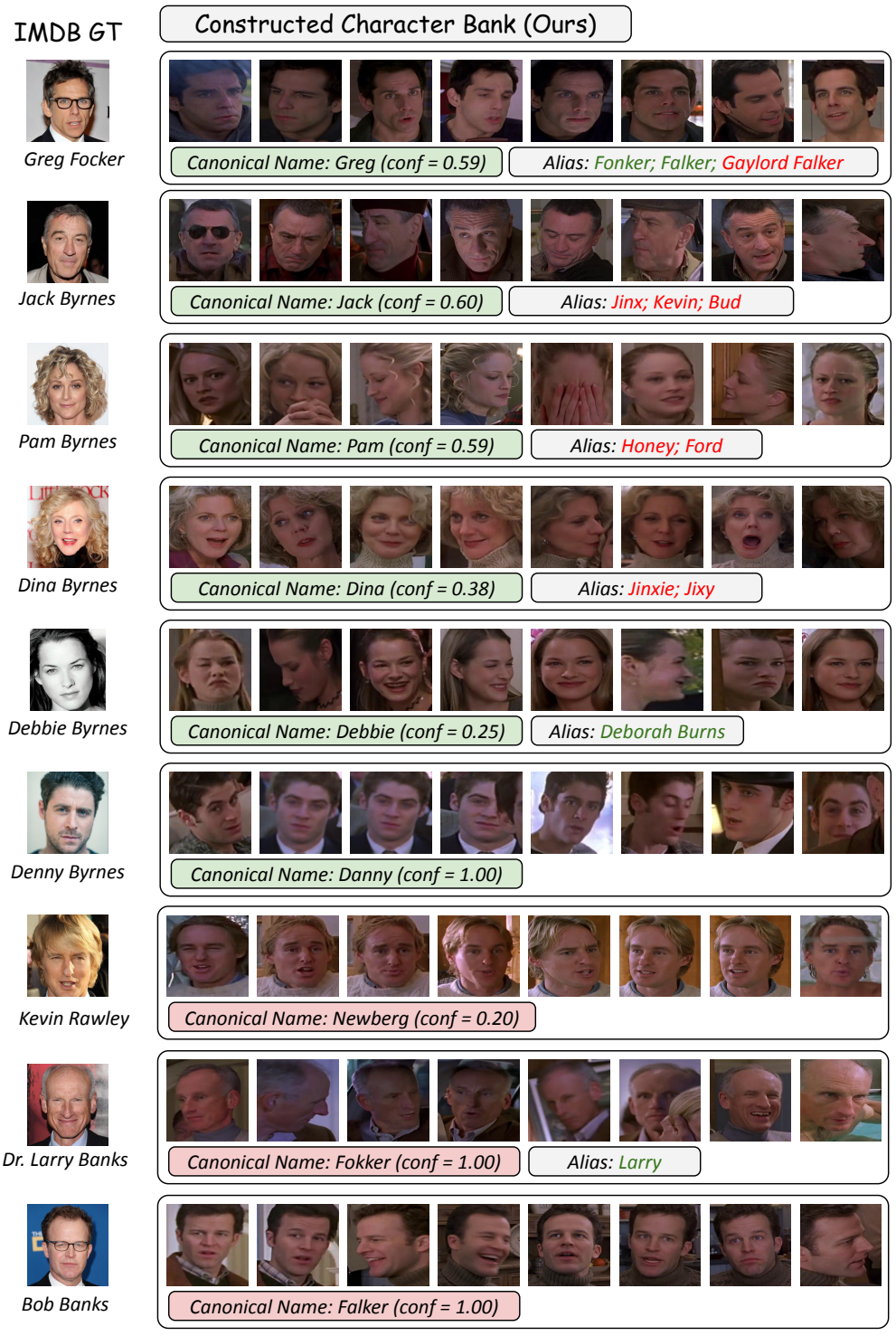


Figure 11: **Example of the constructed character bank for *Meet the Parents*.** We show another example of a character bank constructed by our streaming pipeline. Our method correctly identifies the most prominent characters, but errors remain for the bottom three entries. These characters often appear in complex multi-person scenes, making reliable name binding more difficult. In addition, transcription errors can introduce noisy name variants; for example, *Fonker* and *Falker* should both be resolved to *Focker* for *Greg Focker*.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction enumerate four contributions (the Character Naming task, the MCN dataset, a fine-tuned video-and-dialogue model, and a streaming character bank pipeline), each substantiated by Secs. 3–5.3 and Tabs. 1–4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Sec. 5.2 (sensitivity to upstream transcription and active-speaker detection errors; failure modes on subtle cues, motion-heavy scenes, and off-screen named characters), Sec. 5.3 (degraded performance on movies with many minor or background characters), and qualitative failure cases in Appendices I and J.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper presents an empirical contribution (a dataset, a fine-tuned model, and a streaming pipeline) and contains no theoretical results requiring proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Sec. 3 describes the prompting and two-stage SFT+GRPO recipe, Sec. 4.1 specifies the streaming bank update and error-correction operators, and Appendices A–D provide preprocessing details, the full GRPO formulation, and implementation details (base model, thresholds, batch sizes, learning rates, and schedules). All annotations and code will be released.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the

case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We do not include code or annotations with this submission, but commit to making all code and annotations publicly available upon publication. The MCN dataset is built on top of publicly available MovieNet [28] and Movie101 [73], and all base models used (Qwen3-VL-4B-Thinking, Qwen3-VL-8B-Instruct, Qwen3.5-9B) are open-source.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Tab. 1 reports the train/test splits, and Sec. 3.2 together with Appendices B and D detail the SFT and GRPO stages, including the composite reward, group size, KL coefficient, learning rates, batch size, schedule, and the SFT trace construction protocol.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Results in Tabs. 2–6 are reported as single-run accuracies, mTIoU, and PCR@K values without error bars. The MCN test set is large (1,582 questions across 30 movies for CN and 20 movies for FO), but multi-seed runs were not performed due to the cost of fine-tuning multimodal LLMs and querying proprietary baselines.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resources for fine-tuning, inference, and streaming character bank construction are described in Appendix K.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses publicly available film datasets (MovieNet [28], Movie101 [73]), does not involve human subjects or crowdsourcing, and preserves anonymity in the submission. We have reviewed the NeurIPS Code of Ethics and the work conforms to it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impacts are discussed in Appendix L.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The released assets are character-naming annotations and a fine-tuned MLLM specialised to associate dialogue-mentioned names with on-screen face tracks within films from existing public movie datasets. The model does not perform open-world face identification of real people, so we judge the misuse risk to be low and comparable to the underlying public datasets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All third-party assets are cited at first use, including MovieNet [28], Movie101 [73], Qwen3-VL and Qwen3.5 [2], DeepSeek-R1/GRPO [22], Gemini [19], TalkNet [57], Parakeet [47], DiariZen [23], ECAPA-TDNN [14], RetinaFace/InsightFace [13], SAM 3 [8], PySceneDetect [9], and spaCy. We use these assets in accordance with their licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The MCN dataset is documented in Sec. 2.3 (construction, statistics, and preprocessing), Appendix G (coarse-to-fine alignment for native-resolution annotations), and Appendix H (qualitative examples). Documentation will accompany the dataset and code release.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. Manual annotation and quality control of the test set were performed by the authors on publicly available movie data.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects research and therefore did not require IRB approval.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were used only for minor writing assistance (grammar checking and phrasing) and do not impact the core methodology, scientific rigor, or originality of the research, so declaration is not required per the NeurIPS policy.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.