# EditIQ: Automated Cinematic Editing of Static Wide-Angle Videos via Dialogue Interpretation and Saliency Cues

Rohit Girmaji*
CVIT
International Institute of Information Technology, Hyderabad
Hyderabad, Telangana, India
rohit.girmaji@research.iiit.ac.in

Bhav Beri*
CVIT
International Institute of Information Technology, Hyderabad
Hyderabad, Telangana, India
bhav.beri@research.iiit.ac.in

Ramanathan Subramanian
University of Canberra
Canberra, ACT, Australia
ramanathan.subramanian@ieee.org

Vineet Gandhi
CVIT
International Institute of Information Technology, Hyderabad
Hyderabad, Telangana, India
vgandhi@iiit.ac.in

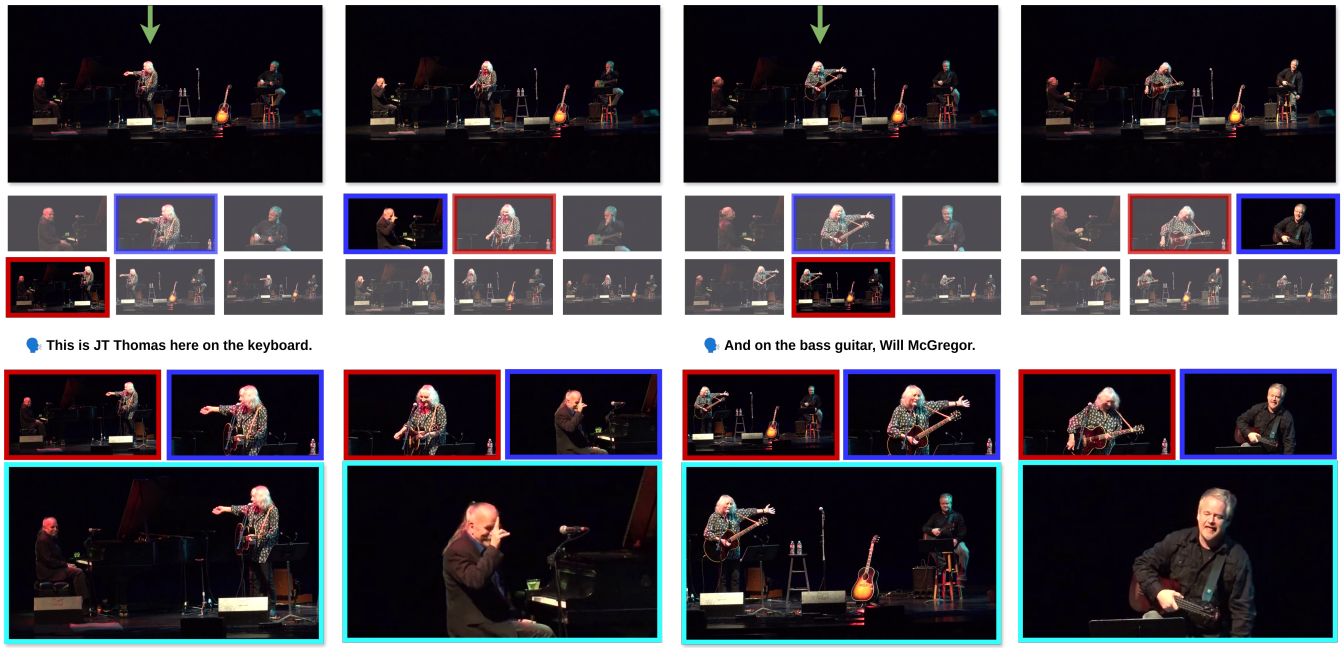


**Figure 1: We present *EditIQ*, an automated video editing pipeline based on dialogue understanding using LLMs and visual understanding via video saliency. First row presents original video frames input to the pipeline, which generates multiple *rushes* (depicted in the next two rows). The speaker is denoted by a green arrow in the original frame and the transcript below 3rd row. LLMs are employed to analyze the scene's narrative, guiding shot selections (highlighted in red on the left for each frame in 4th row). Simultaneously, saliency analysis captures prominent visual scene content, giving alternate shot selections (shown in blue on 4th row right for each frame). Combining the language and visual-based scene understanding results generates optical video shots captured in the 5th row.**

*Both authors contributed equally to this research.


*IUI '25, Cagliari, Italy*

ACM ISBN 979-8-4007-1306-4/25/03
https://doi.org/10.1145/3708359.3712113

## Abstract

We present *EditIQ*, a completely automated framework for cinematically editing scenes captured via a stationary, large field-of-view and high-resolution camera. From the static camera feed, *EditIQ* initially generates multiple virtual feeds, emulating a team of cameramen. These virtual camera shots termed *rushes* are subsequently assembled using an automated editing algorithm, whose objective is to present the viewer with the most vivid scene content. To understand key scene elements and guide the editing process, we employ a two-pronged approach: (1) a large language model (LLM)-based dialogue understanding module to analyze conversational flow, coupled with (2) visual saliency prediction to identify meaningful scene elements and camera shots therefrom. We then formulate cinematic video editing as an energy minimization problem over shot selection, where cinematic constraints determine shot choices, transitions, and continuity. *EditIQ* synthesizes an aesthetically and visually compelling representation of the original narrative while maintaining cinematic coherence and a smooth viewing experience. Efficacy of *EditIQ* against competing baselines is demonstrated via a psychophysical study involving twenty participants on the *BBC Old School* dataset plus eleven theatre performance videos. Video samples from *EditIQ* can be found at https://editiq-ave.github.io/.

## CCS Concepts

• **Information systems** → **Multimedia content creation**; • **Mathematics of computing** → *Combinatorial optimization*; • **Computing methodologies** → *Computational photography*; • **Human-centered computing** → User studies.

## Keywords

Automated Editing, Dialogue Interpretation, Large Language Models, Visual Saliency, Cinematic Video Editing, Shot Selection, Dynamic Programming

**ACM Reference Format:**

Rohit Girmaji, Bhav Beri, Ramanathan Subramanian, and Vineet Gandhi. 2025. EditIQ: Automated Cinematic Editing of Static Wide-Angle Videos via Dialogue Interpretation and Saliency Cues. In *30th International Conference on Intelligent User Interfaces (IUI '25), March 24–27, 2025, Cagliari, Italy*. ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3708359.3712113

## 1 INTRODUCTION

Professional video production of an event like a theatre performance or a quiz show usually requires a team of experienced camera operators to film the scene from multiple angles. These multi-camera recordings, termed *rushes*, are later compiled through careful manual editing to craft a coherent narrative designed to enhance audience engagement and viewing experience. The editing of these performances is typically performed in chronological order, with the process primarily focused on selecting the most appropriate rush at each moment. However, filming in confined spaces, such as live theatre performances, presents unique challenges, including limited vantage points, the impossibility of performing retakes, and the impracticality of maneuvering bulky equipment, making the task both difficult and demanding. Consequently, the need for (i) a skilled camera crew, (ii) multiple cameras and supporting equipment, and (iii) expert editors significantly increases the complexity and cost of the video production process.

Consequently, production houses use a wide-field-of-view static camera positioned at a distance suitable to capture the entire stage. This method is common due to its ease of implementation and ability to capture the entire scene. While effective for archival purposes, wide-angle visuals are ineffective in engaging the audience. The distant camera feed conveys the entire scene but fails to capture close-up details like facial expressions and emotions, which are central to cinematic storytelling. As renowned film editor Thelma Schoonmaker once said, "Close-ups reveal the soul of the character, engaging the audience in a profound way."

Prior automated video editing efforts have sought to transform static wide-angle footage into more engaging content using machine learning and optimization techniques. The goal is to reduce costs and complexity of production, while still delivering quality content. To reduce the reliance on multiple cameramen, Gandhi *et al.* [19, 20] proposed a framework for automatically generating multiple clips suitable for video editing by simulating pan-tilt-zoom camera movements within the frame of a single static camera. Moorthy *et al.* [41] demonstrated that efficient camera selection can be achieved by leveraging eye-gaze data from users. Their work assumes that humans inherently focus on salient scene aspects, and that gaze can serve as a proxy to localize key scene elements. Though their method produces impressive results, its usefulness is restricted due to reliance on gaze data, which may not always be available.

To eliminate the need for auxiliary user data to perform video editing, we present *EditIQ*, a fully automated multi-camera video production pipeline for staged events, using footage from one or more wide-angle, stationary static cameras. We leverage prior efforts [20] for simulating multiple virtual cameras, and our focus is on automating the camera selection process. Optimal camera selection demands a nuanced understanding of the scene, capturing key elements such as dialogue, the speaker, and the actors' actions and reactions. Reaction shots, in particular, are vital in editing as they convey the emotional tone of the scene, enhancing the viewer's engagement with the scene and their perceptual understanding. While the human gaze may naturally track key scene elements, automating this intricate process poses a significant challenge.

*EditIQ* primarily seeks to leverage advancements in large language models (LLMs) for automated video editing. Recent studies have highlighted strong LLM capabilities for understanding key scene elements such as emotions [56], entailment [64], and coreference resolution [51]. Our work is the first to demonstrate that these models can be effectively utilized to guide camera selection based on narratives and conversations, and determining which scene elements should be visually emphasized. *E.g.*, in the exemplar scene illustrated in Figure 1, a camera following the speaker (or an audio source) would focus solely on the lead singer, missing the non-verbal reactions of other musicians as they are introduced. An LLM instead recognizes that when the lead singer says "J T Thomas on the keyboard," the visual attentional focus should shift to the addressed person.

LLMs nevertheless struggle to capture scene actions not explicitly referenced in dialogue, and multimodal LLMs typically perform

frame-level processing, making them less effective at understanding temporal actions. To overcome this limitation, we additionally utilize a saliency prediction architecture trained to model human gaze, identifying key areas of importance in a scene. Specifically, we extend a spatio-temporal action localization backbone [42] for video saliency prediction [58]. Again referring to Figure 1, the visual saliency network accurately captures key actions, like the keyboard player's salute or the lead singer's hand movement, while referring to the bass guitarist.

Once the LLM and visual saliency pipelines enable the identification of key scene elements (and corresponding video shots), we formulate camera shot selection as a discrete optimization problem, where one among the rushes is selected for viewer presentation at each time-frame. Speaker information (obtained from off-the-shelf detection models), LLM predictions, and saliency outputs serve as three unary potential terms in the cost matrix. Akin to [41], these potentials are combined with additional constraints that model cinematic editing principles, such as avoiding jump cuts (causing jarring transitions), maintaining rhythm (consistent pacing of transitions), avoiding transient shots, and ensuring proper framing (to prevent cutting off actors). This optimization is then solved using dynamic programming.

To validate *EditIQ*, we performed a psychophysical study with 20 participants, comparing multiple edited versions of performance recordings from the BBC Old School Dataset (BBC-OSD) [29], which captures a quiz show plus 11 theatre sequences. Our editing strategy surpasses several competing approaches, including random editing, wide-shot framing, and speaker detection-based editing. On BBC-OSD, *EditIQ* achieves an output that is considerably close to the expert human edit. Our research contributions include:

**(1) Semantic shot potentials:** We use novel LLM and visual saliency-based predictions as potentials to quantify the importance of multiple rushes generated from the original recording. LLMs provide dialogue-based visual cues, while saliency augments information regarding scene actions.

**(2) Fully automated editing pipeline:** The potentials derived from active speaker detection, LLM predictions, and visual saliency outputs are combined with cinematic constraints, framing camera shot selection as a discrete optimization problem. This process is fully automated, requiring only the wide-angle video and scene information to synthesize the edit. *EditIQ* can edit a 2-minute video featuring five performers in just 2 minutes on a PC equipped with Nvidia 4090Ti GPU. In contrast, manual editing is significantly more onerous and time-consuming.

**(3) Comprehensive Evaluation:** We evaluate our method on the professionally curated BBC-OSD dataset [29], which is specifically designed to assess the automated editing of wide-angle recordings. We compiled an additional set of 11 high-quality 4K theatre sequences to add variety to our evaluation. Results from a comprehensive user study indicate that *EditIQ* outputs are preferred by users in terms of narrative effectiveness, preservation of emotions and actions, and overall viewing experience.

# 2 RELATED WORK

## 2.1 Editing through automated crops

Working with high-resolution footage like 4K or 8K opens up a variety of creative possibilities for editing, especially for tracking and zooming in on specific video segments. Dynamic cropping has demonstrated notable success in the domains of video retargeting and video stabilization. Automated video retargeting focuses on adjusting content to a specified aspect ratio by dynamically selecting cropping windows. Previous attempts to address this issue have drawn on user annotations [34], motion or saliency information [37, 59], and gaze tracking [47]. Grundmann *et al.* [21] proposed an algorithm for automatically applying constrainable, L1-optimal camera paths to generate stabilized videos by removing undesired motions.

In automated production systems based on cropping, early efforts primarily focused on lectures and presentations [24, 62]. These approaches typically rely on rule-based editing processes confined to controlled settings, featuring a single presenter in front of a chalkboard or slide screen. In the context of sports, Schäfer *et al.* [49] introduced a system that enables the visualization of user-specific cropping windows within an ultra-high-resolution feed. Carr *et al.* [8] used virtual pan-tilt-zoom (PTZ) cropping over a robotic player following a camera for editing of Basketball games. Additionally, virtual PTZ movements have been explored for cinematographic editing in panoramic and 360-degree videos [17, 50, 52].

In contrast to traditional pan-and-scan-like content editing, our work focuses on simulating a multicamera workflow with automated camera selection. Gandhi *et al.* [20] proposed a method for simulating virtual PTZ cameras by shifting a cropping window within wide-angle recordings, aiming to replicate cameraman-like movements using L1-norm-based optimization [21]. Building on this approach, our work introduces automatic selection among these simulated virtual cameras.

The aforementioned methods also relate and benefit from advancements in computer vision and machine learning techniques, such as object detection [28], action recognition and localization [16, 43], person tracking [2, 7], pose estimation [55], video saliency prediction [27, 58], and head pose estimation [30].

## 2.2 Automated Camera Selection

Automated camera selection has been studied in 3D environments, particularly for applications in pre-visualization (previz) and computer games. Early research [13, 23] utilized film idioms and conventional formulas for capturing scenes through sequences of shots [5]. Another stream of research treats camera selection as a discrete optimization problem, addressing it through dynamic programming approaches [18, 36, 39]. Meratbi *et al.* [39] limits to dialogue-driven scenes and utilizes Hidden Markov Models (HMM) for camera selection. Other important aspects, such as editing rhythm, the avoidance of jump cuts, and continuity editing, are addressed in [18]. Although our work draws from these efforts, stage performances present unique limitations, including restricted camera placement and a lack of access to scene geometry, character localization, and event data available in 3D environments.

The problem of camera selection has been thoroughly studied in the context of sports events [10–12, 44, 57]. Wang *et al.* [57] employs

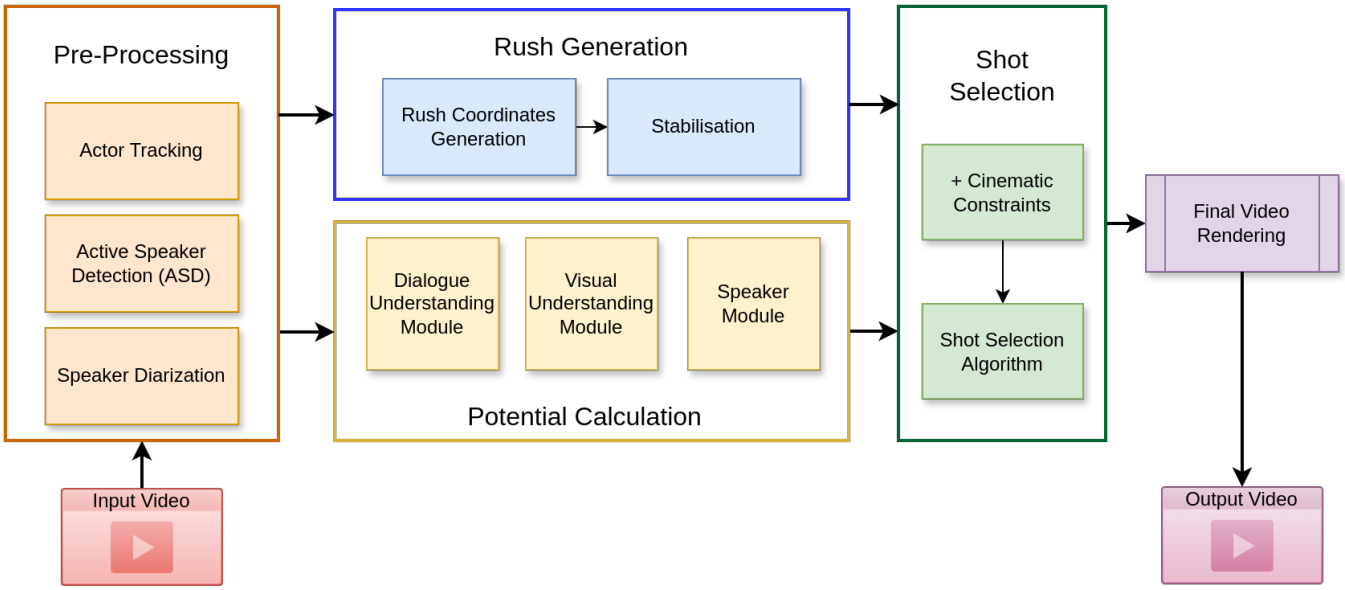


**Figure 2: *EditIQ* Pipeline: This fully automated pipeline takes input in the form of video and face crops + IDs and outputs the completely edited video. The various parts of the pipeline are shown in the figure, with each step operating on the outputs of the previous ones.**

HMMs for the task, and salient action coverage is maximized in [11]. Other studies [10, 12] adopt a data-driven methodology, training regressors to evaluate the significance of each camera angle at any given moment. Pan *et al.* [44] employs an event-based approach, initially identifying events of interest before selecting the most appealing views for those events.

Arev *et al.* [4] propose a method for the automatic editing of multiple social camera feeds. Their approach uses a trellis graph representation to optimize an objective function, which seeks to maximize coverage of the key content in a scene while maintaining adherence to cinematographic principles, such as avoiding jump cuts. The importance of the content is quantified based on joint attention across multiple cameras [45]. Work by Leake *et al.* [35] introduces an idiom-based method for editing dialogue-driven scenes. Their system accepts multiple camera angles, various takes, and the film script as inputs, generating the most informative set of shots for each line of dialogue.

A key distinction of our work from the aforementioned methods is the lack of multiple manually operated camera feeds or takes; instead, we utilize a single wide-angle recording and virtually simulate cameras for editing. The most similar work to ours is GAZED [41], which uses the human gaze to identify salient scene elements, assuming that actor tracks are available. In contrast, our approach eliminates reliance on gaze data and introduces a fully automated editing pipeline that does not require actor tracks or rule-based idioms. Human gaze provides a strong direct proxy for scene importance; in contrast, our method relies on trained machine learning models, tackling several predictive uncertainties faced while automation in real-world applications.

## 3 EditIQ Overview

This paper introduces a comprehensive, end-to-end video editing system - *EditIQ* - designed to automate the cinematic editing and video production process. The system is designed to process static, high-definition recordings of the scene intended for editing as its input and transform them into visually appealing edited videos adhering to the established cinematic principles. The overall architecture of the system mirrors traditional video production pipelines and is structured into four key stages: (i) *Pre-processing* of the input video (ii) *Rush Generation* to generate cinematically valid shots for different actors or elements in the scene. (iii) *Potential Calculation* to calculate each shot importance based on factors like dialogue understanding and spatial saliency cues. (iv) *Shot Selection* to choose the most appropriate shot at each moment to ensure engaging storytelling.

The various steps in the pipeline are shown in the Figure 2. We'll talk about each of the pipeline steps in detail in the following subsections. The inputs to the pipeline include: (i) High-definition video recording captured from stationary camera(s) covering the entire scene (ii) Brief scene description and cast information, including the names of the actors and a single photo of each.

### 3.1 Pre-Processing

Hereafter, frames from the original wide-angle input video are also referred to as "*master shots*". Given the master shot, several key features are derived to support the upcoming stages of the pipeline:

(1) *Actor Tracks*: For the purposes of this work, we utilize the BoT-SORT [2] model for person detection and tracking. This model provides a list of bounding box coordinates $[x, y, h, w]$ for each identified person in every frame, maintaining identity preserving tracks for each actor. Tracking errors, if any,

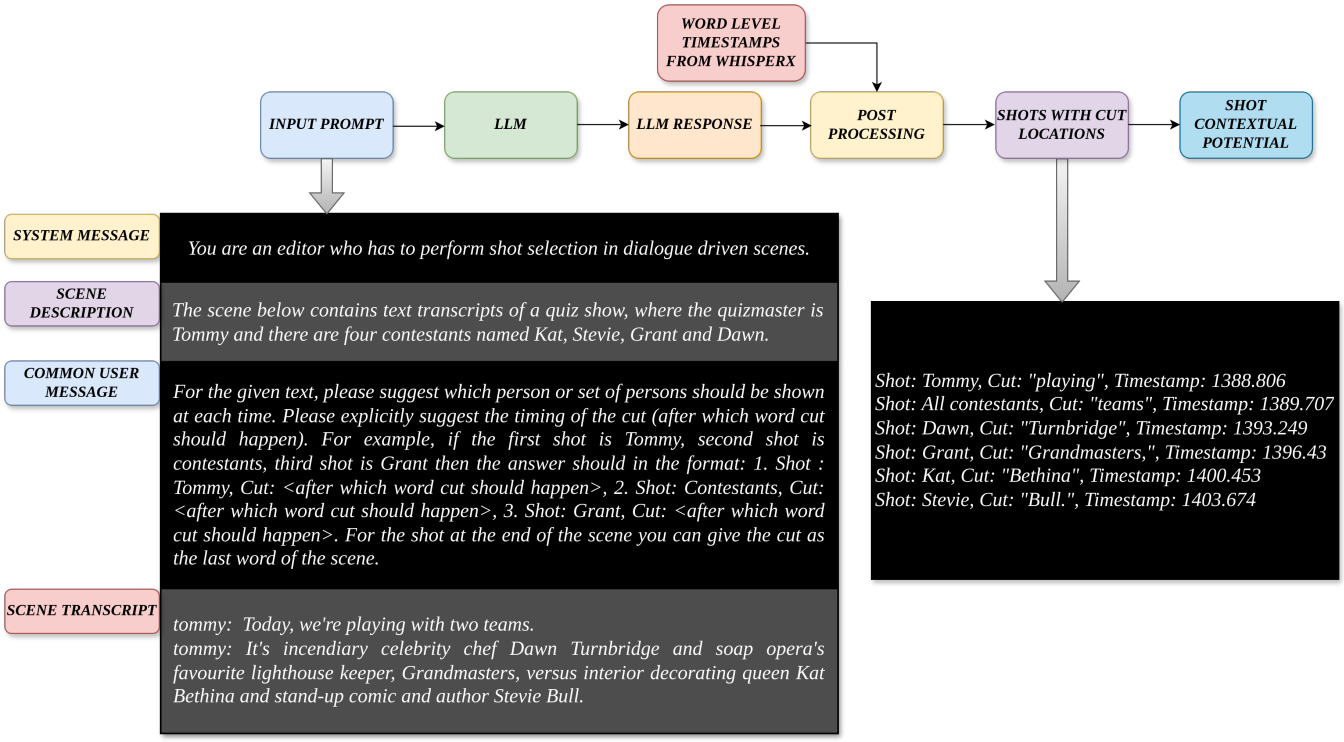


**Figure 3: Dialogue understanding module to get Contextual Potential from LLM for different shots based on the transcript of a scene. The post-processing in the above figure performs mapping between the LLM response and word level timestamps (from pre-processing) to get the cut locations.**

were corrected before sending them forward. These tracks are essential for Shot Generation and Video Editing tasks. In practice, this step is algorithm-agnostic, enabling the use of any alternative algorithms.

(2) *Character Aware Subtitling*: We utilize the WhisperX [6] model on the audio stream of our dataset to detect speech regions with word-level timestamps. The detected words are concatenated to generate a full transcription for each video, which is then segmented into sentences using a sentence tokenizer. Next, we employ the method from [31] to generate character-aware subtitles, producing a complete dialogue transcript with accurate speech timestamps and speaker identification. In brief, we first pick high-quality exemplars for each character using TalkNet [53] and then leverage these exemplars to classify all speech segments by speaker identity [32].

## 3.2 Rush Generation

The second stage of the *EditIQ* pipeline consists of the shot generation task. Herein, a simulation approach [20] was employed to automatically produce virtual PTZ cameras by maneuvering multiple cropping windows following a particular actor or a group of actors within the master shot. We predominantly use medium shots (framing the actor from head to waist) for individual subjects and full shots (depicting the subject from head to toe) to capture two or more actors.

For a master shot with $n$ actors, we generate $2^n - 1$ virtual shots. This includes ${}^nC_1$ 1-shots (individual actors), ${}^nC_2$ 2-shots (two actors), ${}^nC_3$ 3-shots, etc., all in a 16:9 aspect ratio. These shots, along with the master shot, are referred to as "rushes" for further selection and editing. We define $S_t$ as the set of rushes at time $t$ as:

$$S_t = \{A \mid A \subseteq \{x_1, x_2, \ldots, x_n\} \text{ and } A \neq \emptyset\} \cup \{\text{Master Shot}\} \quad \text{where } x_i \text{ is the } i^{th} \text{ actor} \tag{1}$$

Smaller framings like Medium Shots (MS) highlight an actor's actions and expressions in detail, while larger framings like Full Shots (FS) capture the whole actor(s), emphasizing their presence and context within the scene. An example of the generated rushes is presented in Figure 1, featuring three actors. The rushes include the master shot, three 1-shots, two 2-shots, and one 3-shot.

In contrast to the previous works [20], which use an upper-body detector, we use a person pose estimator, which gives us more control over the shot framing and provides a more detailed understanding of the subject's orientation and body posture. For this study, we use YOLOv8-Pose [28, 55], for its high precision and real-time performance.

Following [20], we first obtain per frame shot estimations, which are then optimized to ensure well-composed shots mimicking the movement of professional cameramen. The virtual PTZ simulation is formulated as an optimization problem, with the objective of minimizing a sum-of-squares term that measures closeness to the

original per frame estimations, combined with $L_1$-norm regularization on velocity and jerk [20, 21].

### 3.3 Dialogue Understanding Module — Contextual Potential

In video editing, dialogue is a core component that shapes how the story is told, characters are developed, and emotions are conveyed. An editor must not only understand the literal meaning of the words but also interpret the various attributes conveyed through dialogue, such as tone, emotion etc. By controlling when to cut, how to pace conversations, and which reactions to emphasize, editors can transform a simple dialogue scene into a powerful, emotionally engaging moment that drives the narrative forward and deepens the audience's connection to the characters.

As part of the Dialogue Understanding Module, we utilize Large Language Models (LLMs) to interpret dialogues and assist with shot sequence suggestions. The LLM receives a scene transcript and a concise prompt with instructions for generating a shot sequence that visually narrates the scene. We also provide a brief scene description to enrich the context. The transcript includes speaker details (from the pre-processing step), allowing the LLM to identify the individual or group to be shown at each moment in the scene. Additionally, we request explicit cut points, specifying the exact word after which each cut should occur. The LLM response, containing shot suggestions and cut information, is post-processed with word-level timestamps to determine precise cut locations (as timestamps). Shot suggestions from LLMs are used to compute the contextual potential for each shot at any time instant.

Figure 3 illustrates this process using an example from the BBC-OSD, showing the user prompt, a sample transcript, and the final shot sequence suggestions with cut locations derived from word-level timestamps. A key principle followed for the contextual potential is that other shots that either include or overlap with the selected shot should not have a zero cost, but rather a small one, as they still partially capture the context used by the LLM during its recommendations. The calculation of the contextual potential depends on the type of shot selected by the model:

(1) If a single-order shot is selected, it is assigned a cost of $\lambda_c$. Higher-order shots (p-order shots that contain the actor selected) are assigned a cost of $\frac{\lambda_c}{p}$, as they can also convey the same context as the single-order shot but less effectively than a close-up. All other shots receive a cost of 0.

$$C(s_t^X) = \begin{cases} \lambda_c & s_t^X = s_t^{X_s} \\ \frac{\lambda_c}{p} & x_s \in \mathrm{X},\ |X| > 1 \\ 0 & \text{for remaining 1-shots} \end{cases} \tag{2}$$

where $s_t^X$ refers to a shot $s$ at time $t$ that contains a set of actors $X$ and $s_t^{X_s}$ is the single-order shot selection from LLM with actor $x_s$.

(2) If a p-order shot ($p > 1$) is selected, single-order shots involving actors from the selected shot are given a cost of $\frac{\lambda_c}{2^{p-1}}$. As calculated through Equation (4), the selected shot receives a cost of $\lambda_c$, while all other shots receive a cost of less than $\lambda_c$.

$$C(s_t^X) = \begin{cases} \frac{\lambda_c}{2^{p-1}} & \mathrm{X} \in X_s,\ |X| = 1 \\ 0 & \text{for remaining 1-shots} \end{cases} \tag{3}$$

where $s_t^X$ refers to a shot $s$ at time $t$ that contains a set of actors $X$ and $X_s$ is the higher-order shot selection from LLM with actors $X_s = \{x_i \mid i = 1, 2, \ldots, p\}$

For higher-order shots, the contextual potential is calculated as described in [41]. For example, consider a 2-shot $s_t^X$ where $X = \{x_1, x_2\}$ containing actors $x_1$ and $x_2$. Contextual potential for this higher-order shot is defined in terms of the contextual potentials of the single-order shots $C(s_t^{x_1})$ and $C(s_t^{x_2})$ as follows:

$$C(s_t^{\{x_1,x_2\}}) = C(s_t^{x_1}) + C(s_t^{x_2}) - \left|C(s_t^{x_1}) - C(s_t^{x_2})\right| \tag{4}$$

Similarly, contextual potentials of two 2-shots $C(s_t^{\{x_1,x_2\}})$ and $C(s_t^{\{x_2,x_3\}})$ can be used to compute the contextual potential of a 3-shot $C(s_t^{\{x_1,x_2,x_3\}})$, when the actors appear on screen in the order $x_1$, $x_2$, $x_3$ from left to right.

### 3.4 Visual Understanding Module — Saliency Potential

As previously discussed, dialogue understanding is crucial in the video editing process. However, it faces a limitation in that contextual potential alone cannot fully capture the visual information present in a scene. To address this, we incorporate saliency prediction, which allows us to extract essential visual elements beyond the dialogue, emphasizing actions, reactions, and overall scene dynamics. This approach ensures a more comprehensive understanding of both verbal and non-verbal elements within the video.

The *EditIQ* pipeline employs a modified 3D convolutional, visual-only *Video Saliency Prediction* (VSP) model, building on the ViNet [27] architecture. ViNet was chosen because of its compact size as compared to other VSP models yet having competitive results among other factors - code availability and reproducibility of results. We observe that the modified ViNet model more effectively captures the overall essence of the scene, beyond merely detecting motion cues or primary semantic features such as faces. Further details about the model and results are explained in the Appendix B.

The above VSP model outputs a saliency map for each frame of the video, equivalent to the size of the original video frame. We then apply a threshold of $\tau_{sal}$ on the map, to eliminate values below this threshold, enhancing the clarity of salient features while reducing noise.

Subsequently, the saliency score for each actor is computed by taking the mean of the thresholded saliency values within its bounding box. To ensure comparability between actors, we normalize these saliency scores across all actors. The actor with the highest saliency score is assigned a value of $\lambda_{Sal}$, while the second most salient actor receives a score of $\lambda_{Sal}/2$:

$$V(s_t^x) = \begin{cases} \lambda_{Sal} & x \text{ is salient actor} \\ \frac{\lambda_{Sal}}{2} & x \text{ is second-most salient actor} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

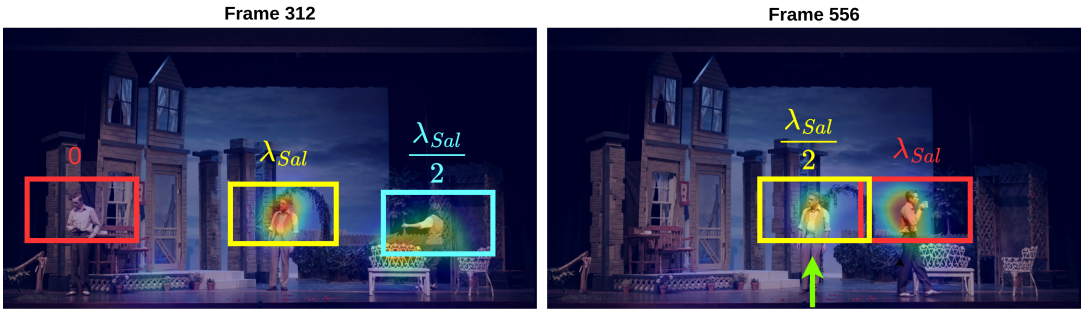


**Figure 4: Saliency potential of different single-order shots for two frames in a theatre video (potential value is shown along with the actor shot). Green arrow indicates the speaker, if any.**

where $s_t^x$ refers to a shot $s$ at time $t$ that contains a single actor $x$. An example of the same is illustrated in Figure 4, we can observe that saliency potential picks actions and actor movements, rather than focusing solely on the speaker, and is especially effective when there are no dialogues. Saliency potential for higher-order shots is then computed from the saliency potentials of constituent lower-order shots similar to Equation (4).

## 3.5 Speaker Module — Speaker Potential

Speaker potential is designed to assign importance to the shot corresponding to the active speaker. Given the speaker-aware subtitles and the corresponding timestamps, the speaker potential ($S$) is defined as follows:

$$S(s_t^x) = \begin{cases} \lambda_{Sp} & x \text{ is speaker} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $s_t^x$ refers to a 1-shot containing actor $x$ at time $t$.

## 3.6 Cinematic Constraints

While the aforementioned potentials provide costs based on the importance of various shots, editing decisions made solely on these costs may lack cinematic validity. The contextual potential derived from LLMs does not incorporate visual information, which can lead to issues like overlapping cuts or jump cuts. Additionally, LLMs lack awareness of the duration of the shot, as this aspect is addressed later during the post-processing phase in the Dialogue Understanding Module (Section 3.3). Similarly, saliency in its raw form is not designed for a delicate task like video editing and does not account for cinematic rules such as minimum shot duration or the continuity of flow. Therefore, it is crucial for any video editing pipeline to consider cinematic principles separately alongside these potentials or costs.

In this study, we adopt an approach similar to GAZED [41], where penalties are applied to shots that violate various cinematic principles. These penalties are then fed into the Shot Selection Algorithm, which selects the final shots based on both the shot potentials and the associated penalties. We define four types of penalty terms, which are detailed below. The total penalty for any given shot is the cumulative sum of these individual penalties at any given timestamp.

*3.6.1 Overlap Penalty:* During the transition between two consecutive shots, excessive overlap can lead to a jump cut, which disrupts the continuity and can be visually jarring. To avoid such visually jarring transitions we use overlap penalty, designed to minimize overlap between consecutive shots. This penalty is applied only when two distinct shots, $s_t^i$ and $s_{t+1}^j$, are involved and a cut occurs between them.

$$O(s_t^i, s_{t+1}^j, \gamma) = \begin{cases} 0 & \text{if } \gamma \leq \alpha \\ \frac{\mu\gamma}{\alpha} & \text{if } \alpha < \gamma \leq \beta \\ \nu & \text{if } \gamma > \beta \end{cases}$$

Here, $\gamma$ represents the overlap ratio, calculated as the intersection-over-union (IoU) between two consecutive shots $s_t^i$ and $s_{t+1}^j$. The penalty is piecewise, i.e. no penalty is applied if the IoU is below threshold $\alpha$, a linear penalty is applied for IoU values between $\alpha$ and $\beta$, and a high penalty $\nu$ is applied when the overlap ratio exceeds $\beta$, showcasing a significant overlap that violates cinematic principles.

*3.6.2 Misframing Penalty:* Poorly framed shots occur when another actor is partially visible in the current frame, which can disrupt the composition. For example, the shot suggestions provided by LLMs may not account for the spatial arrangement of actors, potentially recommending a cut to an actor sitting in close proximity to another. To avoid such shots, we define misframing penalty as follows:

$$M(s_t^i) = \begin{cases} \lambda_{\text{mis}} & \text{if the framing is poor} \\ 0 & \text{otherwise} \end{cases}$$

If a shot $s_t^i$ is found to be poorly framed, the penalty $\lambda_{\text{mis}}$ is added to its cost. A framing is defined to be poor if it overlaps with actors beyond the shot definition.

*3.6.3 Rhythm Penalty:* The pacing of cuts plays an important part in determining the overall feel of a scene in video editing. Shot duration directly influences how some audiences perceive the mood and energy of a sequence. For example, longer shots create a slower rhythm, bringing in calmness or emotional depth, often used in romantic or contemplative scenes. While, on the other hand, shorter shots create a faster rhythm, heightening tension or energy, a technique commonly applied in action scenes in editing. To manage the rhythm of cuts, we use the rhythm penalty, which regulates

shot duration to maintain cinematic flow. The rhythm penalty is applied based on the duration of the current shot ( $\tau$ ), calculated as follows:

$$R(s_t^i, s_{t-1}^j, \tau) = \begin{cases} \gamma_1 \left(1 - \frac{1}{1+\exp(l-\tau)}\right) & \text{if } i \neq j \\ \gamma_2 \left(1 - \frac{1}{1+\exp(-m+\tau)}\right) & \text{if } i = j \end{cases}$$

In this equation, $\tau$ is the time the current shot has been held, and $l$ and $m$ are parameters that control the rhythm timings. The constants $\gamma_1$ and $\gamma_2$ are scaling factors for rhythm penalty. When transitioning to a new shot ($i \neq j$), the penalty increases if the new shot is cut too quickly, i.e., before $\tau = l$ seconds, to prevent rapid cutting. While on the other hand, if a shot is held for too long ($i = j$), a penalty builds up as $\tau$ exceeds $m$ seconds, encouraging a cut to introduce new visual information. Together, these two conditions help control the rhythm of cuts, ensuring that the scene is neither too rushed nor overly static.

*3.6.4 Transition Penalty:* Extremely fast cuts in editing can confuse or disorient the audience and can undermine the emotional weight of a scene. Fast cuts can obscure important details and weaken storytelling clarity, and might compromise aesthetic quality. To prevent this, we apply a transition penalty to promote a minimum shot duration. Given two consecutive shots, $s_t^i$ and $s_{t+1}^j$, the penalty is defined as:

$$T(s_t^i, s_{t+1}^j) = \begin{cases} 0 & \text{if } i = j \\ \lambda_{trans} & \text{if } i \neq j \end{cases}$$

Here, $\lambda_{trans}$ is the transition penalty parameter.

## 3.7 Shot Selection

Given the multiple types of shots and rushes, our next step in the *EditIQ* pipeline involves selecting the shot that best fits the storytelling at each moment in time. We frame shot selection as a discrete optimization problem, evaluating the importance of each shot per frame while adhering to cinematic principles like avoiding jump cuts, rapid transitions, and irregular cutting rhythms. Shot importance at each time is determined by the potentials as explained in Section 3.3, Section 3.4, and Section 3.5, while cinematic principles are incorporated as penalty terms. The final solution is derived by finding the optimal path in an editing graph, which, for a scene with $n$ actors, consists of $2^n - 1$ nodes per frame. Each node represents a rush, with edges indicating transitions (cuts) or continuity (no cut) between shots.

Given a sequence of frames $t = [1, 2, \ldots, T]$ and the set of generated shots (rushes) $S_t$ (Equation (1)), our method selects a sequence of shots $\epsilon = \{r_t \mid i = 1, 2, \ldots, T\}$, $r_t \in S_t$ by minimizing the following objective function:

$$E(\epsilon) = \sum_{t=1}^{T} -\ln\Big(U(r_t)\Big) + \sum_{t=2}^{T} \Big[O(r_{t-1}, r_t, \gamma) + R(r_t, r_{t-1}, \tau) + T(r_{t-1}, r_t)\Big] + \sum_{t=1}^{T} M(r_t) \tag{7}$$

where $U(r_t)$ is the unary cost for a shot, representing the shot's importance. This unary cost is the cumulative sum of contextual potential, saliency potential, and speaker potential (Equation (8)). The second and third terms represent different penalties described in Section 3.6.

$$U(r_t) = C(r_t) + V(r_t) + S(r_t) \tag{8}$$

We solve Equation (7) using dynamic programming. Our method outputs a sequence of shots for each frame $t$ selected from a series of shots generated over time $\{S_t \mid i = 1, 2, \ldots, T\}$. We build a cost matrix $CM(r_t, t)$, $r_t \in S_t, t = [1, 2, \ldots, T]$ whose elements are computed recursively as follows:

$$CM(r_t, t) = \begin{cases} -\ln(U(r_t)) + M(r_t) & t = 1 \\ \min_k \Big[CM(r_k, t-1) - \ln(U(r_t)) & \\ +O(r_k, r_t, \gamma) + R(r_t, r_k, \tau) & \\ +T(r_k, r_t) + M(r_t)\Big] & \text{otherwise} \end{cases}$$

The cost matrix is constructed during a forward pass across the time dimension. For each element in the matrix, we calculate and store the minimum cost required to reach it. Once the matrix is completed, we backtrack to determine a sequence of optimal shots. For the edited video, we use the wide shot or master shot from the original footage as the establishing shot, setting its duration to 2 seconds and then optimizing only over the remaining frames.

# 4 Experiments

To assess whether our method - *EditIQ*, which leverages a dialogue understanding module and saliency prediction to guide shot selection, results in a visually compelling and coherent cinematic representation of scenes, we conducted a psychophysical user study involving twenty participants, with the details as follows.

## 4.1 Dataset

The study utilizes the BBC Old School Dataset (BBC-OSD) [29]. BBC-OSD, curated by BBC R&D, is a comprehensive resource for advancing research into AI-driven automated video editing. The dataset includes comedy fiction (sitcom), drama, and game show elements and is set during the filming of a fictional game show called "Old School". It includes raw footage of multiple takes from the short TV program, along with behind-the-scenes content and rich metadata. Unlike conventional TV shoots, this production was tailored specifically to create data for automated editing, offering static wide-angle views and multi-participant interactions. The dataset provides insights into the entire production process, from planning to metadata generation, enabling the development of sophisticated AI editing systems for various use cases. They also provide a human-edited programme as a benchmark for automated editing systems. We use the Edit Decision List (EDL) corresponding to the human-edited programme to extract videos from the raw footage, with a total duration of approximately 30 minutes.

In addition to the above-defined dataset, we selected eleven segments from three stage and theatre performances recorded in 4K resolution ($3840 \times 2160$). These videos include a mix of music concerts and various theatre acts, all captured using a wide-angle static camera, with no pan, cut, or zoom operations. The purpose of

selecting these recordings is to evaluate the effectiveness of our *EditIQ* editing pipeline on more diverse scenarios. The chosen videos present a range of challenging cases, including rapid dialogues, actor co-referencing, abrupt story transitions, and critical background actions and emotions. Each video requires precise editing to ensure the narrative flows smoothly without missing key elements or important shots.

## 4.2 LLM Configuration & Details

For our LLM-based inferences, we utilized the Claude 3.5 Sonnet model developed by Anthropic, specifically the "claude-3-5-sonnet-20240620" checkpoint [3]. The maximum context window length for the Claude model was 200K tokens during the time of building the system. For reproducibility purposes, we set the temperature parameter to 0.

## 4.3 Parameter Selection

Cinematic constraint parameters play a crucial role in shaping the output of the pipeline and can be seen as the personalization of the edits. Most parameters in *EditIQ* are either drawn from established literature or set empirically. For instance, the rhythm penalty parameter $m$, which governs the maximum sequence length, is set to 7, reflecting the average shot length in films over the past two decades [15]. The minimum shot duration, controlled by parameter $l$, is set to 1, with $\gamma_1$ assigned a high value (100), as cuts shorter than 1 second tend to disrupt continuity, and very fast cuts are generally undesirable. Similarly, the penalty for overlap, $\nu$ is kept at a very high value ($10^6$), to strictly avoid jump cuts. For overlap cost parameters, we set $\alpha = 0.15$, as cuts with less than 15% overlap typically pose no visual issues, while $\beta = 0.3$ ensures that cuts with over 30% overlap are flagged as abrupt.

These parameters can be adjusted to customize the editing style, such as creating faster or slower-paced edits. The algorithm's computational efficiency enables interactive content exploration, allowing for real-time adjustments to personalize the final output.

## 4.4 Baselines

We compared the videos generated using the *EditIQ* pipeline against various competing video editing baselines, including Random, Wide, and Speaker. These baselines were selected because they do not require any manual data collection for the editing process. For a fair comparison, all baseline videos were shown to users at the same resolution and audio quality, with each video retaining the exact timeline of the original footage.

In addition to these baselines, we included and conducted ablation studies using only the LLM-based Contextual Potential and Saliency-based Visual Potential to demonstrate that relying on these methods alone is insufficient for effective video editing. Furthermore, we included Human Edits as a baseline for the BBC-OSD Dataset, allowing us to compare our results with professional editing standards.

*4.4.1 Random Baseline:* The Random baseline (Ran) is a simple method where shots are chosen randomly from the available footage at different time intervals, without considering what is happening in the scene (the context). After selecting the shots, we apply cinematic rules and penalties to try and improve the video, since random selections often break these rules and can lead to awkward or unappealing results. This lack of coherence makes it the weakest baseline in terms of narrative flow and visual consistency.

*4.4.2 Wide Baseline:* The Wide baseline is inspired by video retargeting techniques and is similar to the letterboxing method described in prior research [26]. This approach focuses on selecting the widest possible shot that includes all performers on stage. This shot is essentially a zoomed-in version of the master shot, capturing the entire scene without excluding any actors. The goal is to ensure that no one is left out of the frame, prioritizing coverage over more focused or dynamic shots. This method is simple and effective for keeping all performers in view at all times, but it lacks the flexibility to adapt to changes in action or focus within the scene.

*4.4.3 Speaker Baseline:* Speaker cues are valuable for editing dialogue driven scenes, as highlighted in previous studies by Ranjan *et al.* [48] and Leake *et al.* [35], who advocate for selecting shots that clearly showcase the speaker. Our speaker-based (Sp) editing baseline follows a similar approach by choosing the shot that best highlights the speaker from the available footage. This selection process relies on information obtained from the character-aware subtitling (Section 3.1). The current shot selection remains unchanged until a different speaker takes the floor. To minimize abrupt transitions, a minimum shot duration is enforced. If there is a period of silence lasting more than 10 seconds, the algorithm will switch to a wide shot for the subsequent time interval.

*4.4.4 LLM-Only baseline:* The LLM-based baseline leverages insights from large language models to select shots that align with the narrative context. By analyzing the dialogues within the video, this approach aims to choose shots that enhance storytelling and maintain coherence. The LLM Potential is computed as outlined in Section 3.3, ensuring that the selected shots contribute meaningfully to the narrative. However, since LLMs lack information about long or short cut times (based on how contextual potential is calculated) as well as lack of visual information (overlapping shots, jump cuts), corrections based on cinematic principles are applied afterward to ensure a smooth visual flow and adherence to established editing standards. This process helps mitigate potential issues that may arise from the initial shot selection, enhancing the overall quality of the edited video.

*4.4.5 Saliency-Only baseline:* The Saliency (Sal) baseline uses visual saliency detection to find the most important parts of a video frame, focusing on shots that highlight these key elements. By emphasizing what draws the viewer's attention, this method aims to create a more engaging experience. The Saliency Potential is calculated as outlined in Section 3.4, which measures how well each shot showcases these important visuals. However, since saliency detection doesn't consider rules like minimum shot duration or storytelling flow, we apply corrections based on cinematic principles afterwards.

*4.4.6 Human Edits (Only for BBC-OSD):.* These edits were performed by professional video editors at the BBC, providing a valuable point of comparison for our approach against an actual edit (or an established ground truth). Users rated these edits alongside others, unaware that they were created by humans. The videos used

for this baseline were taken directly from the BBC-OSD without any modifications, ensuring an accurate representation of professional editing standards.

# 5 Evaluation & User Study

## 5.1 Materials & Methods:

To evaluate the efficacy of *EditIQ* against the aforementioned video editing baselines, we conducted a psychophysical study involving 20 users (aged 20–25 and including 2 females). Original and edited versions of 11 videos from the BBC-OSD and 11 theatre performance videos generated by all baselines plus *EditIQ* were viewed by users. The maximum video length over these 22 videos was 94s. For a fair comparison, identical *EditIQ* parameters were used for generating all video edits. Upon viewing the original video, each user viewed the edited[1] versions at the same pixel resolution in a random sequence to eliminate order-specific effects.

The study design was such that each user viewed the original and edited versions of 2-3 videos so that the 20 users cumulatively viewed all 22 recordings, and the experiment lasted around 20 minutes per user. We ensured that the original and edited versions of each recording received exactly two user ratings resulting in a 11 (video types) × 2 (user ratings/video) × 7 (editing strategies) factor design for BBC-OSD, and a 11 (video types) × 2 (user ratings/video) × 6 (editing methods) factor design for theatre recording videos.

Users were naive to the strategies employed for generating the edited versions. Users had to *compare* each edited version against the original and provide a Likert rating on a [-5,5] scale for each of the attributes described below. These attributes were adopted from [26, 47], and are designed to evaluate how effectively the edited versions capture *focal scene events* given event recording constraints. The attributes of interest included:

(1) **Narrational Effectiveness (NE):** *How effectively did the edited video convey the original narrative?*
(2) **Scene actions (SA):** *How well did the edited video capture actor movements and actions?*
(3) **Actor Emotions and Reactions (ER):** *How well did the edited video capture actor emotions and reactions?*
(4) **Viewing experience (VX):** *How would you rate the edited video for aesthetic quality?*

Users were familiarized regarding these attributes, and about cinematic video editing conventions prior to the study. Users had to rate for questions (1)–(4), *relative* to a reference score of '0' for the original video. A *positive* score would therefore imply that the edited version was *better* than the original for the target attribute, while a *negative* score conveyed that the edited version was *worse* than the original with respect to the criterion. User responses were collated, and mean scores were computed per criterion and editing strategy over all videos (see Figure 5). Statistics and inferences from the user study are presented below.

[1]Corresponding to the six baselines including human-edited plus *EditIQ* for BBC-OSD, and five baselines plus *EditIQ* for theatre videos.

## 5.2 Results and Discussion

*5.2.1 BBC Old School (BBC-OSD):.* Bar plots depicting mean user scores across attributes and editing methods are presented in Figure 5. A two-way balanced analysis of variance (ANOVA) on the compiled NE, SA, ER, and VX user scores across methods revealed the main effects of *editing strategy* on user opinions ($p < 0.000001$ for all four attributes), and *video type* ($p < 0.005$ for all four attributes). No interaction effects were noted. We hypothesized that combining LLM and visual saliency cues via *EditIQ* would result in an engaging, vivid, and aesthetic edit, which is generally validated by Figure 5. Across the four attributes, *EditIQ* expectedly outperforms the Random, Speaker, Wide, and Saliency baselines, but performs comparably to LLM and inferior to professional human edits.

Investigating specific attributes, *post-hoc* independent $t$-tests on NE scores revealed a significant difference between *EditIQ* vs. Sp ($p < 0.05$), *EditIQ* vs. Ran ($p < 0.001$), *EditIQ* vs. Wide ($p < 0.000001$) and *EditIQ* vs. Sal ($p < 0.05$). *EditIQ* vs. LLM NE values were very comparable ($p = 0.6728$), while NE scores for human edits were significantly higher than for *EditIQ* ($p < 0.05$). These results cumulatively convey that carefully compositing shots which provide a closer view of the key scene actor(s) and action(s) is crucial for effective scene narration. The Random baseline, which selects shots independent of scene content, performs worst, followed by the wide baseline, which can only present the entire scene context without a focus on scene details. The Speaker and Saliency-based editing methods, which respectively employ speech and visual scene cues for shot selections, perform comparably ($p = 0.5968$), but LLM-based editing, which is guided by the scene narrative, significantly outperforms visual saliency ($p < 0.05$). conveying that visual cues only supplement LLM capabilities for automated video editing.

For conveying scene actions, user score trends are very similar to NE scores. *EditIQ* significantly outperforms Ran ($p < 0.0005$), Wide ($p < 0.000001$), Sp ($p < 0.001$) and Sal ($p < 0.05$), while performing similar to LLM ($p = 0.9177$) and under-performing compared to human editing ($p < 0.005$). Visual saliency and speech-based editing are deemed comparable by users ($p = 0.7306$), while LLM-based editing outperforms both methods ($p < 0.05$). Slightly different trends are, however, noted with respect to conveying actor emotions and reactions. *EditIQ* is rated significantly higher than Ran ($p < 0.0005$), Wide ($p < 0.000001$), Sp ($p < 0.001$) and Sal ($p < 0.05$), very comparable to LLM ($p = 0.9177$) and inferior to human editing ($p < 0.005$). Saliency-based editing scores better for this attribute, performing better than Sp ($p < 0.01$), but still lower than LLM-based editing ($p < 0.05$). These trends convey that saliency is more effective at capturing visually prominent facial expression changes and reactions, even if they cannot effectively capture the general scene narrative or critical scene actions.

Finally, familiar trends repeat with respect to the viewing experience. *EditIQ* outperforms all other competing automated approaches, but scores significantly lower than professional human editing ($p < 0.001$). Saliency and speaker-based editing again perform very comparably ($p = 0.9326$), with LLM-based editing outperforming these two approaches ($p < 0.05$).

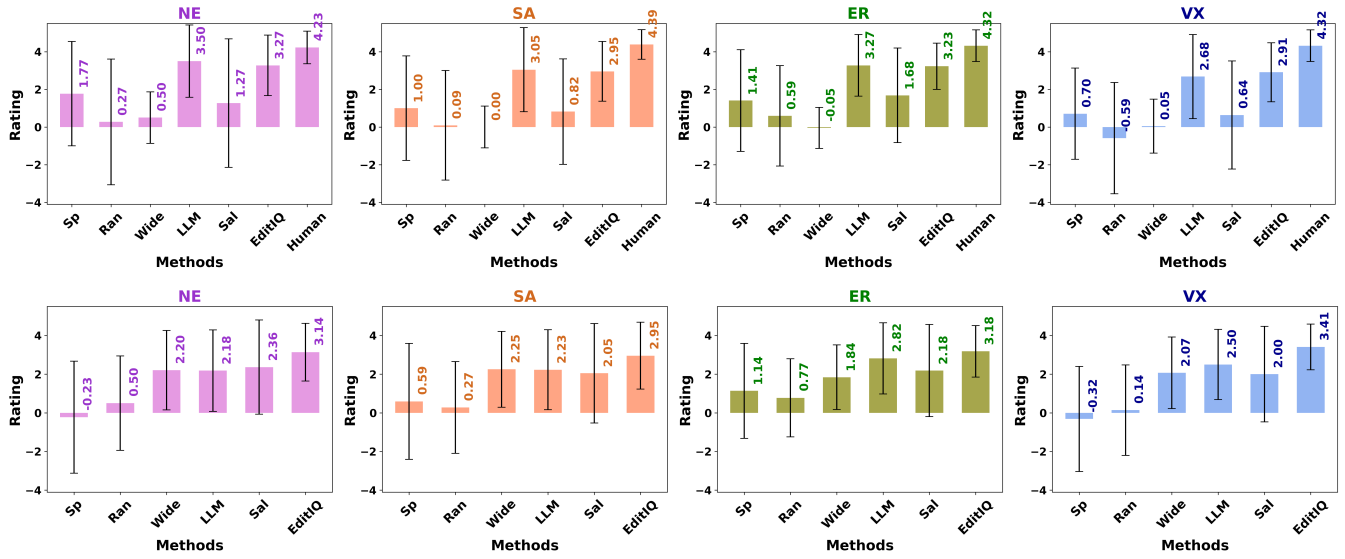


**Figure 5: User Study Evaluation: Bar plots denoting mean user ratings for the different editing methodologies across four evaluation attributes for the (top row) BBC-OSD and (bottom row) Theatre recordings. Error bars denote unit standard deviation. Best viewed in color and under zoom.**

*5.2.2 Theatre recordings:* Scores very different to the BBC-OSD, which captures a quiz event involving four participants in a smallish venue, are obtained for the theatre recordings capturing a larger venue, and where actions and events can possibly happen at the stage periphery as well. Human-edited outputs are not available for the theatre sequences, and therefore we will only compare the automated editing approaches.

Repeating a two-way ANOVA for theatre sequence user scores conveyed the main effect of editing methodology ($p < 0.0005$ for all attributes), but no effect of video type or interaction effects. Given the large spatial context in theatre performances, the Wide baseline capturing the entire scene scores relatively higher as compared to the BBC-OSD.

For narrational effectiveness, the Sp and Ran baselines score similarly poorly ($p = 0.3731$), while the Wide, LLM and Sal editing approaches perform superiorly and comparably. *EditIQ* achieves the highest scores, which are significantly higher than for Sp ($p < 0.0001$), Ran ($p < 0.0005$), marginally higher than Wide ($p = 0.0927$) and LLM ($p = 0.0909$) and insignificantly higher than the Sal baseline. With respect to scene actions, trends are generally similar, with *EditIQ* scoring significantly higher than Sp ($p < 0.005$) and Ran ($p < 0.0005$) but only insignificantly higher than the Wide, LLM, and Sal baselines. For facial expressions and reactions, *EditIQ* again scores significantly higher than Ran ($p < 0.00005$), Sp ($p < 0.005$) and Wide ($p < 0.01$), marginally higher than Sal ($p = 0.095$) but only insignificantly higher than LLM-based editing. Finally, with respect to viewing experience, our *EditIQ* approach scores significantly higher than Sp ($p < 0.00001$), Ran ($p < 0.000005$), Wide ($p < 0.001$), and Sal ($p < 0.05$) baselines, and marginally higher than the LLM ($p = 0.0571$) baseline. This conveys that our proposed editing approach can effectively combine LLM and visual-based cues to engagingly and vividly convey static camera recordings to viewers.

*5.2.3 Past-Experience of Participants.* Among the 20 participants, five had prior experience in video editing, while the remaining individuals lacked familiarity with video editing tools and techniques. Experienced participants exhibited a more critical perspective, consistently assigning significantly lower ratings to the random baseline compared to non-experienced participants, whose scores were closer to neutral. This highlights their heightened ability to detect and penalize editing flaws. Additionally, experienced participants demonstrated greater appreciation for high-quality edits, rating human edits higher—4.4 (NE), 4.8 (SA), 4.6 (ER), 5.0 (VX)—compared to non-experienced participants—4.17 (NE), 4.26 (SA), 4.23 (ER), 4.11 (VX). This indicates that their expertise allowed them to recognize the nuances and limitations inherent in the editing process, leading to a more informed evaluation.

*5.2.4 Discussion Summary.* We evaluated *EditIQ* against competing baselines under two varied settings: (1) a quiz event captured by the BBC-OSD, and (2) theatre recordings that involve a very diverse context and dynamics compared to quizzes. Although the combination of the saliency and LLM cues is not very beneficial in the quiz context, where speech cues essentially guide visual attention, the benefit is more apparent for theatre performance edits where actions from actors other than the speaker could be regarded as salient. In both settings, however, *EditIQ* is found to generally outperform other automated approaches while only scoring inferior with respect to professional human editing for BBC-OSD.

## 6 Conclusion

This work introduces *EditIQ*, a framework for the automatic editing of stage performance videos captured by unmanned, static, wide-angle, high-resolution cameras. We employ LLMs for dialogue understanding and prompt it to suggest which person or set of persons should be shown at each word timestamp of the character-aware subtitles. Additionally, we employ video saliency prediction methods to capture actions and other visual elements that are not conveyed through the dialogue. The LLM suggestions, saliency predictions, and speaker information are combined together to quantify the importance of each shot in the generated rushes at each time. These unary shot potentials are then combined with cinematic penalties like avoiding jump cuts and fast cuts, avoiding improper framings, and maintaining rhythm. The result is a meticulously edited sequence that not only preserves key content but also adheres to cinematic principles, resulting in a visually compelling video. The effectiveness of *EditIQ* compared to competing baselines is demonstrated through a psychophysical study involving twenty participants using the BBC-OSD and eleven theatre performance videos. *EditIQ* generally outperforms other baselines, scoring lower only in comparison to professional human editing for the BBC-OSD.

## 7 Limitations and Ethical Considerations

A key limitation of the current system is its inability to perform real-time editing, a critical feature for live events. Previous work has established the feasibility of online rush generation, stabilization, and camera selection [1, 20]. Future endeavors will seek to integrate these with a streaming LLM variant, enabling dialogues to be processed incrementally rather than as a complete script.

It is also important to emphasize that this project is not intended to replace human editors but rather to serve as an assistive tool. The results from the human evaluation clearly demonstrate that human edits consistently received higher scores than those produced by automated methods, underscoring the irreplaceable expertise of professional editors. Instead, this system aims to support editors by generating novel ideas or reducing their workload, particularly in tasks such as selecting shots from extensive footage. Nevertheless, the system provides a cost-effective solution for low-budget theaters, allowing them to create visually appealing edits of performances without the need for expensive multi-camera setups or professional editors.

## Acknowledgments

We express our sincere gratitude to BBC R&D and the authors of the Old School Dataset (OSD), which includes raw rushes and expertly crafted human edits. This meticulously curated dataset has been pivotal in shaping our work. We further extend special thanks to Stephen Jolly and Graeme Phillipson for their dedicated efforts in formally providing us with this resource. We would also like to thank The Dorset Players for their performance of *The 39 Steps* theatre play and to Martin HS Theatre for their presentation of *All My Sons* play. We also thank all participants who contributed to the User Study.

## References

[1] Sudheer Achary, Rohit Girmaji, Adhiraj Anil Deshmukh, and Vineet Gandhi. 2024. Real Time GAZED: Online Shot Selection and Editing of Virtual Cameras from Wide-Angle Monocular Video Recordings. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 4108–4116.

[2] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. 2022. BoT-SORT: Robust Associations Multi-Pedestrian Tracking. *arXiv preprint arXiv:2206.14651* (2022).

[3] Anthropic. 2024. The Claude 3 Model Family: Opus, Sonnet, Haiku. https://assets.anthropic.com/m/61e7d27f8c8f5919/original/Claude-3-Model-Card.pdf

[4] Ido Arev, Hyun Soo Park, Yaser Sheikh, Jessica Hodgins, and Ariel Shamir. 2014. Automatic editing of footage from multiple social cameras. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 81.

[5] Daniel Arijon. 1976. Grammar of the film language. (1976).

[6] Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. 2023. WhisperX: Time-Accurate Speech Transcription of Long-Form Audio. *INTERSPEECH 2023* (2023).

[7] Jinkun Cao, Jiangmiao Pang, Xinshuo Weng, Rawal Khirodkar, and Kris Kitani. 2023. Observation-centric sort: Rethinking sort for robust multi-object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9686–9696.

[8] Peter Carr, Michael Mistry, and Iain Matthews. 2013. Hybrid robotic/virtual pan-tilt-zom cameras for autonomous event recording. In *Proceedings of the 21st ACM international conference on Multimedia*. 193–202.

[9] Qinyao Chang and Shiping Zhu. 2021. Temporal-spatial feature pyramid for video saliency detection. *arXiv preprint arXiv:2105.04213* (2021).

[10] Christine Chen, Oliver Wang, Simon Heinzle, Peter Carr, Aljoscha Smolic, and Markus H. Gross. 2013. Computational sports broadcasting: Automated director assistance for live sports. In *Proceedings of the 2013 IEEE International Conference on Multimedia and Expo, ICME 2013, San Jose, CA, USA, July 15-19, 2013*.

[11] Fan Chen and Christophe De Vleeschouwer. 2010. Personalized production of basketball videos from multi-sensored data under limited display resolution. *Computer Vision and Image Understanding* 114, 6 (2010), 667–680.

[12] Jianhui Chen, Lili Meng, and James J Little. 2018. Camera Selection for Broadcasting Soccer Games. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 427–435.

[13] David B Christianson, Sean E Anderson, Li-wei He, David H Salesin, Daniel S Weld, and Michael F Cohen. 1996. Declarative camera control for automatic cinematography. In *AAAI/IAAI, Vol. 1*. 148–155.

[14] Antoine Coutrot and Nathalie Guyader. 2015. An efficient audiovisual saliency model to predict eye positions when looking at conversations. In *2015 23rd European Signal Processing Conference (EUSIPCO)*. IEEE, 1531–1535.

[15] James Cutting and Ayse Candan Simsek. 2015. Shot Durations, Shot Classes, and the Increased Pace of Popular Movies. *Projections* 9 (12 2015), 40–52. doi:10.3167/proj.2015.090204

[16] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. Slowfast networks for video recognition. In *International Conference on Computer Vision (ICCV)*. 6202–6211.

[17] Vamsidhar Reddy Gaddam, Ragnhild Eg, Ragnar Langseth, Carsten Griwodz, and Pål Halvorsen. 2015. The Cameraman Operating My Virtual Camera is Artificial: Can the Machine Be as Good as a Human? *ACM Trans. Multimedia Comput. Commun. Appl.* 11, 4, Article 56 (June 2015), 20 pages.

[18] Quentin Galvane, Rémi Ronfard, Christophe Lino, and Marc Christie. 2015. Continuity editing for 3D animation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[19] Vineet Gandhi and Rémi Ronfard. 2015. A computational framework for vertical video editing. In *4th Workshop on Intelligent Camera Control, Cinematography and Editing*. Eurographics Association, 31–37.

[20] Vineet Gandhi, Remi Ronfard, and Michael Gleicher. 2014. Multi-clip video editing from a single viewpoint. In *Proceedings of the 11th European Conference on Visual Media Production* (London, United Kingdom) *(CVMP '14)*. Association for Computing Machinery, New York, NY, USA, Article 9, 10 pages. doi:10.1145/2668904.2668936

[21] Matthias Grundmann, Vivek Kwatra, and Irfan Essa. 2011. Auto-directed video stabilization with robust L1 optimal camera paths. In *CVPR 2011*. 225–232. doi:10.1109/CVPR.2011.5995525

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

[23] Li-wei He, Michael F Cohen, and David H Salesin. 1996. The virtual cinematographer: a paradigm for automatic real-time camera control and directing. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 217–224.

[24] Rachel Heck, Michael Wallick, and Michael Gleicher. 2007. Virtual videography. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 3, 1 (2007), 4–es.

[25] Yani Ioannou, Duncan Robertson, Roberto Cipolla, and Antonio Criminisi. 2017. Deep Roots: Improving CNN Efficiency with Hierarchical Filter Groups. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[26] Eakta Jain, Yaser Sheikh, Ariel Shamir, and Jessica Hodgins. 2015. Gaze-Driven Video Re-Editing. *ACM Trans. Graph.* 34, 2, Article 21 (March 2015), 12 pages. doi:10.1145/2699644

[27] Samyak Jain, Pradeep Yarlagadda, Shreyank Jyoti, Shyamgopal Karthik, Ramanathan Subramanian, and Vineet Gandhi. 2021. Vinet: Pushing the limits of visual modality for audio-visual saliency prediction. In *IROS*. IEEE, 3520–3527.

[28] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. 2023. *Ultralytics YOLO*. https://github.com/ultralytics/ultralytics

[29] Stephen Jolly, Graeme Phillipson, and Michael Evans. 2023. Old School: An 8K Multicamera Shoot to Create a Dataset for Computational Cinematography. In *Proceedings of the 2023 ACM International Conference on Interactive Media Experiences Workshops* (Nantes, France) *(IMXw '23)*. Association for Computing Machinery, New York, NY, USA, 76. doi:10.1145/3604321.3604374

[30] Yueying Kao, Bowen Pan, Miao Xu, Jiangjing Lyu, Xiangyu Zhu, Yuanzhang Chang, Xiaobo Li, and Zhen Lei. 2023. Toward 3d face reconstruction in perspective projection: Estimating 6dof face pose from monocular image. *IEEE Transactions on Image Processing* 32 (2023), 3080–3091.

[31] Bruno Korbar, Jaesung Huh, and Andrew Zisserman. 2024. Look, Listen and Recognise: character-aware audio-visual subtitling. (2024).

[32] Bruno Korbar and Andrew Zisserman. 2022. Personalised CLIP or: how to find your vacation videos. In *British Machine Vision Conference*.

[33] Petros Koutras, Athanasios Katsamanis, and Petros Maragos. 2014. Predicting Eyes' Fixations in Movie Videos: Visual Saliency Experiments on a New Eye-Tracking Database. In *Engineering Psychology and Cognitive Ergonomics*, Don Harris (Ed.). 183–194.

[34] Philipp Krähenbühl, Manuel Lang, Alexander Hornung, and Markus Gross. 2009. A system for retargeting of streaming video. In *ACM SIGGRAPH Asia 2009 Papers* (Yokohama, Japan) *(SIGGRAPH Asia '09)*. Association for Computing Machinery, Article 126, 10 pages.

[35] Mackenzie Leake, Abe Davis, Anh Truong, and Maneesh Agrawala. 2017. Computational video editing for dialogue-driven scenes. *ACM Trans. Graph.* 36, 4, Article 130 (July 2017), 14 pages. doi:10.1145/3072959.3073653

[36] Christophe Lino, Mathieu Chollet, Marc Christie, and Rémi Ronfard. 2011. Computational model of film editing for interactive storytelling. In *International Conference on Interactive Digital Storytelling*. Springer, 305–308.

[37] Feng Liu and Michael Gleicher. 2006. Video retargeting: automating pan and scan. In *Proceedings of the 14th ACM international conference on Multimedia*. 241–250.

[38] Yufan Liu, Minglang Qiao, Mai Xu, Bing Li, Weiming Hu, and Ali Borji. 2020. Learning to predict salient faces: A novel visual-audio saliency model. In *European Conference on Computer Vision (ECCV)*. 413–429.

[39] Billal Merabti, Marc Christie, and Kadi Bouatouch. 2016. A Virtual Director Using Hidden Markov Models. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 51–67.

[40] Kyle Min and Jason J Corso. 2019. Tased-net: Temporally-aggregating spatial encoder-decoder network for video saliency detection. In *International Conference on Computer Vision (ICCV)*. 2394–2403.

[41] K. L. Bhanu Moorthy, Moneish Kumar, Ramanathan Subramanian, and Vineet Gandhi. 2020. GAZED- Gaze-guided Cinematic Editing of Wide-Angle Monocular Video Recordings. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. ACM. doi:10.1145/3313831.3376544

[42] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. 2021. Actor-context-actor relation network for spatio-temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 464–474.

[43] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. 2021. Actor-context-actor relation network for spatio-temporal action localization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. 464–474.

[44] Yingwei Pan, Yue Chen, Qian Bao, Ning Zhang, Ting Yao, Jingen Liu, and Tao Mei. 2021. Smart director: An event-driven directing system for live broadcasting. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 17, 4 (2021), 1–18.

[45] Hyun Park, Eakta Jain, and Yaser Sheikh. 2012. 3D Social Saliency from Head-mounted Cameras. In *Advances in Neural Information Processing Systems*, F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (Eds.), Vol. 25. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2012/file/1bf2efbbe0c49b9f567c2e40f645279a-Paper.pdf

[46] Minglang Qiao, Yufan Liu, Mai Xu, Xin Deng, Bing Li, Weiming Hu, and Ali Borji. 2023. Joint Learning of Audio-Visual Saliency Prediction and Sound Source Localization on Multi-face Videos. 132 (2023), 2003–2025.

[47] Kranthi Kumar Rachavarapu, Moneish Kumar, Vineet Gandhi, and Ramanathan Subramanian. 2018. Watch to edit: Video retargeting using gaze. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 205–215.

[48] Abhishek Ranjan, Jeremy Birnholtz, and Ravin Balakrishnan. 2008. Improving meeting capture by applying television production principles with audio and motion detection. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 227–236. doi:10.1145/1357054.1357095

[49] Ralf Schäfer, Peter Kauff, and Christian Weissig. 2010. Ultra high resolution video production and display as basis of a format agnostic production system. In *Proceedings of International Broadcast Conference (IBC 2010)*, Vol. 1.

[50] Yu-Chuan Su, Dinesh Jayaraman, and Kristen Grauman. 2016. Pano2vid: Automatic cinematography for watching 360 videos. In *Asian Conference on Computer Vision*. Springer, 154–171.

[51] Kawshik Sundar, Shubham Toshniwal, Makarand Tapaswi, and Vineet Gandhi. 2024. Major Entity Identification: A Generalizable Alternative to Coreference Resolution. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 11679–11695.

[52] Chengzhou Tang, Oliver Wang, Feng Liu, and Ping Tan. 2019. Joint stabilization and direction of 360 videos. *ACM Transactions on Graphics (TOG)* 38, 2 (2019), 1–13.

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

[54] Antigoni Tsiami, Petros Koutras, and Petros Maragos. 2020. Stavis: Spatio-temporal audiovisual saliency network. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. 4766–4776.

[55] Rejin Varghese and Sambath M. 2024. YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness. In *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*. 1–6. doi:10.1109/ADICS58448.2024.10533619

[56] Krishnapriya Vishnubhotla, Adam Hammond, Graeme Hirst, and Saif M Mohammad. 2024. The Emotion Dynamics of Literary Novels. *ACL* (2024).

[57] Jinjun Wang, Changsheng Xu, Engsiong Chng, Hanqing Lu, and Qi Tian. 2008. Automatic composition of broadcast sports video. *Multimedia Systems* 14, 4 (2008), 179–193.

[58] Wenguan Wang, Jianbing Shen, Jianwen Xie, Ming-Ming Cheng, Haibin Ling, and Ali Borji. 2019. Revisiting video saliency prediction in the deep learning era. *TPAMI* 43, 1 (2019), 220–237.

[59] Yu-Shuen Wang, Hui-Chih Lin, Olga Sorkine, and Tong-Yee Lee. 2010. Motion-based video retargeting with optimized crop-and-warp. In *ACM SIGGRAPH 2010 papers*. 1–9.

[60] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. 2018. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *European Conference on Computer Vision (ECCV)*.

[61] Junwen Xiong, Ganglai Wang, Peng Zhang, Wei Huang, Yufei Zha, and Guangtao Zhai. 2023. CASP-Net: Rethinking video saliency prediction from an audio-visual consistency perceptual perspective. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. 6441–6450.

[62] Cha Zhang, Yong Rui, Jim Crawford, and Li-Wei He. 2008. An automated end-to-end lecture capture and broadcasting system. *ACM Transactions on multimedia computing, communications, and applications (TOMM)* 4, 1 (2008), 1–23.

[63] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. 2018. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[64] Zhihan Zhou, Xue Gu, Yujie Zhao, and Hao Xu. 2024. POP-CEE: Position-oriented Prompt-tuning Model for Causal Emotion Entailment. In *Findings of the Association for Computational Linguistics ACL 2024*. 4199–4210.

# A Prompts

## A.1 System Message

`You are an editor who has to perform shot selection in dialogue driven scenes.`

## A.2 Scene Description

*A.2.1 BBC-OSD.* `The scene below contains text transcripts of a quiz show, where the quizmaster is Tommy and there are four contestants named Kat, Stevie, Grant and Dawn.`

*A.2.2 Theatre.* `The scene below contains text transcripts of a scene from a theatre play.`

## A.3 Common Prompt

*A.3.1 BBC-OSD.* `For the given text, please suggest which person or set of persons should be shown at each time. Please explicitly suggest the timing of the cut (after which word cut should happen). For example, if the first shot is Tommy, second shot is contestants, third shot is Grant then the answer should in the format: 1. Shot: Tommy, Cut: <after which word cut should happen>, 2. Shot: Contestants, Cut: <after which word cut should happen>, 3. Shot: Grant, Cut: <after which word cut should happen>. For the shot at the end of the scene you can give the cut as the last word of the scene.`

*A.3.2 Theatre.* `For the given text, please suggest which person or set of actors should be shown at each time. Please explicitly suggest the timing of the cut (after which word cut should happen). For example, if the first shot is actorX, second shot is (actorX and actorY), third shot is actorZ then the answer should in the format: 1. Shot: actorX, Cut: <after which word cut should happen>, 2. Shot: (actorX and actorY), Cut: <after which word cut should happen>, 3. Shot: actorZ, Cut: <after which word cut should happen>. For the shot at the end of the scene you can give the cut as the last word of the scene.`

# B Video Saliency Prediction (VSP) Model

We use a modified VSP model based on the 3D convolutional ViNet [27] model, with two key modifications made to the original ViNet [27] model:

(1) We addressed limitations found in action classification backbones like S3D [60], which tend to overlook background actions by focusing on primary motion in human-centric videos. Instead, we integrated a Spatio-Temporal Action Localization (STAL) backbone [16, 43], pre-trained on the AVA actions dataset [22], alongside our custom decoder. This combination improves the ability to localize and classify actions, thereby better capturing the essence of the scene.
(2) The ViNet decoder was restructured to enhance computational efficiency, by incorporating filter groups [25] and channel shuffle layers [63]. This method reduces the original model's size and parameter count by threefold, while simultaneously improving Saliency Prediction performance.

We observe that the modified ViNet model captures the overall essence of the scene more effectively, beyond merely detecting motion cues or primary semantic features such as faces. For instance, Figure 6 illustrates a frame from a video in the MVVA [38] dataset, where a group of people are being interviewed. While other existing SOTA models, like the original ViNet [27], limit to head movements and end up mistakenly highlighting all faces as salient, our model accurately identifies the most relevant face, such as the person speaking or the one receiving attention from others in the scene. Results for the model on a few human-centric datasets are shown in Table 1 & Table 2.

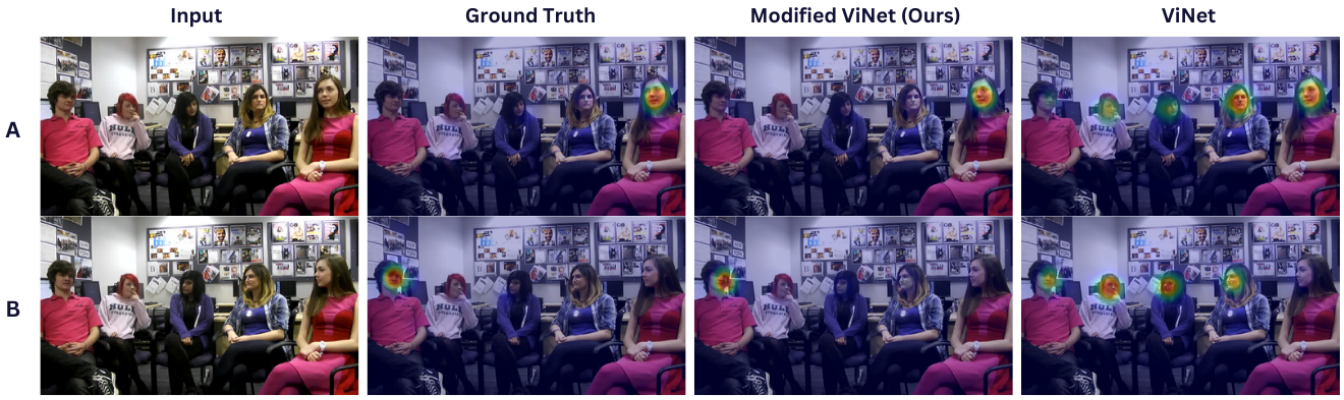


**Figure 6: Here, we compare our modified Saliency Prediction model with state-of-the-art ViNet Model [27]. Our model captures the essence of the whole scene and performs joint attention to capture interactions. It focuses on the key actor, whereas ViNet limits to head movements and captures all the faces as salient.**

**Table 1: Results on MVVA [38] Dataset**

| METHOD | MVVA | | | |
|---|---|---|---|---|
| | CC↑ | NSS↑ | AUC↑ | KLDiv↓ |
| **TASED-Net [40]** | 0.653 | 3.319 | 0.905 | 0.970 |
| **STAViS [54]** | 0.77 | 3.060 | 0.91 | 0.80 |
| **ViNet [27]** | 0.81 | 4.470 | 0.93 | 0.75 |
| **VAM-Net [46]** | 0.741 | 4.002 | 0.912 | 0.783 |
| **Ours** | 0.821 | 4.792 | 0.93 | 0.689 |

**Table 2: Results on Coutrot2 [14] and ETMD [33] Datasets**

| METHOD | Coutrot2 | | | | ETMD | | | |
|---|---|---|---|---|---|---|---|---|
| | CC↑ | NSS↑ | AUC↑ | SIM↑ | CC↑ | NSS↑ | AUC↑ | SIM↑ |
| **TASED-Net [40]** | 0.437 | 3.17 | 0.921 | 0.314 | 0.509 | 2.63 | 0.916 | 0.366 |
| **STAViS [54]** | 0.652 | 4.19 | 0.940 | 0.447 | 0.560 | 2.84 | 0.929 | 0.412 |
| **ViNet [27]** | 0.724 | 5.61 | 0.95 | 0.466 | 0.569 | 3.06 | 0.928 | 0.409 |
| **TSFP-Net [9]** | 0.718 | 5.30 | 0.957 | 0.516 | 0.576 | 3.09 | 0.932 | 0.433 |
| **CASP-Net [61]** | 0.756 | 6.07 | 0.963 | 0.567 | 0.616 | 3.31 | 0.938 | 0.471 |
| **Ours** | 0.860 | 6.563 | 0.963 | 0.610 | 0.632 | 3.519 | 0.943 | 0.493 |