Reframe Anything: LLM Agent for Open World Video Reframing*

Jiawang Cao[1], **Weiheng Chi**[3], **Wenbo Zhu**[1] , **Lirian Su**[1], **Yuyang Sun**[1,2], **Jay Wu**[1]

[1] Opus AI Research
[2] Southeast University
[3] National University of Singapore

Abstract

The rapid proliferation of mobile devices and social media has fundamentally transformed content dissemination, with short-form video emerging as a dominant medium. To adapt original video content to this format, manual reframing is often required to meet constraints on duration and device screen size. This process is not only labor-intensive and time-consuming but also demands significant professional expertise. While machine learning models—such as video salient object detection—offer promising avenues for automation, existing approaches typically lack human-in-the-loop interaction, making it difficult to accommodate personalized user preferences. To address these limitations, AI systems must be capable of fully understanding user intent and dynamically tailoring video reframing strategies in response to evolving requirements. The powerful capabilities of large language models (LLMs) make them particularly well-suited for handling such complex multimodal interaction scenarios. Building on this insight, we introduce **R**eframe **A**ny **V**ideo **A**gent (RAVA), an LLM-based agent that integrates visual foundation models with human instructions to intelligently restructure visual content for video reframing. RAVA operates in three stages: *perception*, where it interprets user instructions and video content; *planning*, where it determines suitable aspect ratios and reframing strategies; and *execution*, where it invokes editing tools to produce the final video. Our experiments demonstrate the effectiveness of RAVA in both video salient object detection and real-world reframing tasks, showcasing its potential as a powerful tool for AI-powered video editing.

1 Introduction

Short-form video has rapidly emerged as a dominant medium for content dissemination, driven by the widespread adoption of social media platforms and handheld mobile devices [5]. The prevalence of vertically-oriented displays, especially on mobile devices, has led to a fundamental shift in how video content is produced, consumed, and optimized. However, traditional videos are often captured in landscape orientation and may not be readily compatible with the varying aspect ratios used across different platforms. This mismatch creates a growing demand for automatically adapting or reconstructing original videos to suit diverse screen formats without compromising visual quality or narrative coherence.

This transformation process, referred to as *video reframing*, entails dynamically selecting and emphasizing the most semantically meaningful or visually compelling elements within a video. From a creative and editorial standpoint, reframing often involves applying operations such as

*Published as a workshop paper at NeurIPS 2025 Workshop 'What Makes a Good Video: Next Practices in Video Generation and Evaluation'

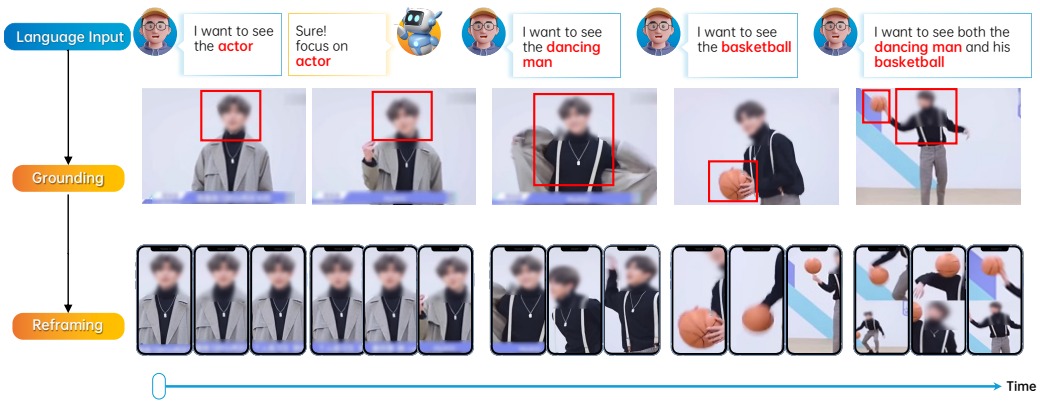


Figure 1: Overview of the open-world video reframing task. Even for the same video, different users may focus on different subjects of interest. Thus, it is essential to implement video reframing based on user instructions to fulfill personalized objectives.

cropping, panning, zooming, and compositing to ensure that the reframed video preserves viewer engagement and conveys the intended message effectively.

Manual video reframing is both time-consuming and labor-intensive. It requires considerable domain knowledge and aesthetic sensitivity on the part of professional editors, and often entails scene-by-scene adjustment to maintain temporal consistency and visual balance. Consequently, manual workflows significantly increase production costs and hinder scalability. To alleviate this burden, researchers have turned to automatic reframing methods, especially those leveraging advances in machine learning and computer vision.

A key direction in automated reframing is *video saliency detection* [40, 14], which aims to identify spatial regions that are likely to attract human attention. For example, Christel et al. [4] propose the use of bottom-up visual cues to compute saliency maps, which in turn guide cropping decisions. However, such methods are typically driven by low-level features and may fall short in capturing high-level semantics or task-specific preferences, resulting in suboptimal reframing outcomes.

To enhance semantic awareness, the field has evolved toward *video salient object detection*, which focuses on segmenting temporally coherent and visually dominant objects across frames. These approaches improve the preservation of meaningful content during reframing. Nevertheless, existing models often exhibit limited generalization capabilities due to dataset bias and domain dependency, and their performance may degrade in diverse or open-world settings. Furthermore, viewers' subjective preferences introduce additional complexity. As illustrated in Figure 1, different users may focus on different subjects within the same video. Therefore, it is crucial to develop a reframing framework that is not only content-aware but also user-controllable through explicit instructions.

Recent advances in large language models (LLMs), such as ChatGPT [24] and GPT-4 [25], have revolutionized the landscape of artificial intelligence. These models exhibit remarkable capabilities in understanding and generating natural language, and have been shown to possess emergent reasoning abilities across diverse tasks. More notably, the advent of multimodal LLMs like GPT-4V [26] has enabled models to process and interpret visual content through text-based interactions. Such systems can reason about visual scenes, describe spatial layouts, and align language with perceptual cues, opening new possibilities for flexible, instruction-driven media generation.

Building on this paradigm, LLM-based agents have gained traction for their ability to orchestrate complex workflows through high-level planning and natural language understanding. Examples such as TaskMatrix [20], AutoGPT [43], and MetaGPT [11] demonstrate how LLMs can perform perception, decision-making, and control in multi-step environments. Furthermore, systems like AppAgent [44] and MobileAgent [39] show the feasibility of using LLMs to operate mobile applications and perform UI-level actions guided by user instructions.

Motivated by these developments, we propose **R**eframe **A**ny **V**ideo **A**gent (RAVA), an LLM-based agent designed to perform flexible and personalized video reframing based on natural language instructions. RAVA is structured into three core stages: *perception*, *planning*, and *execution*. In the

perception stage, RAVA parses user directives and analyzes video content to extract salient objects and generate descriptive metadata. The planning stage uses this contextual information to formulate reframing strategies, such as selecting aspect ratios, determining object priorities, arranging layouts, and specifying visual effects that align with user intent. Finally, in the execution stage, RAVA translates these strategies into concrete editing operations and applies them to the video, with support for iterative refinement through user feedback.

To evaluate RAVA, we conduct experiments from two perspectives. First, we examine its capacity to understand and follow user instructions in the context of video salient object detection, serving as a proxy for semantic grounding. Second, we apply RAVA to real-world reframing scenarios, including social media adaptation and vertical cropping tasks. Both quantitative results and user studies demonstrate the utility and effectiveness of RAVA in enhancing AI-driven video editing workflows.

Our main contributions are summarized as follows:

- We introduce **RAVA**, a novel LLM-based agent capable of performing flexible, personalized video reframing guided by natural language instructions.
- We propose a structured framework comprising perception, planning, and execution stages, enabling RAVA to interpret video content and user intent in a unified pipeline.
- We validate RAVA through extensive experiments on both benchmark and real-world tasks, showing its potential to improve automation and personalization in video editing.

2 Related Work

Video Editing. Recent advances in movie analysis have notably progressed, particularly in the area of Audio-Visual Event (AVE) Localization, which involves identifying and precisely localizing events within a video [37, 9]. These advancements can aid video editors by streamlining the editing workflow [33], although they do not directly enable automated video editing. Beyond such analysis, several works have integrated machine learning techniques into the video editing process itself. For example, Argaw et al. [3] introduce a benchmark suite targeting various video editing tasks, including visual effects, automated footage organization, and assistance in video assembly. Similarly, Rao et al. [31] propose a benchmark for selecting optimal camera angles from multiple inputs—an essential component in television production. Despite these efforts, existing methods generally overlook the task of video reframing, which focuses on emphasizing the most compelling segments of a video. To address this, research on *video salient object detection* [12, 45, 36] has emerged as a promising solution. However, these methods often rely on domain-specific training datasets, which limits their generalization capabilities across diverse real-world scenarios and undermines their interpretability. Overcoming these limitations remains a critical challenge for fully automated and flexible video reframing.

Open Vocabulary Segmentation. Open Vocabulary Segmentation aims to partition images into semantically meaningful regions without being restricted to a fixed set of predefined categories. This represents a significant departure from traditional segmentation approaches [29, 18, 41], which are typically constrained by limited label vocabularies and struggle to generalize to unseen objects. Foundational models such as CLIP [30] and ALIGN [13] have paved the way for segmenting novel object categories using natural language supervision. For instance, LSeg [19] trains an image encoder to produce pixel-level embeddings, employing CLIP-derived text embeddings as per-pixel classifiers. To exploit inexpensive image-level supervision, OpenSeg [10] introduces weakly-supervised grounding losses and random word dropout to enhance alignment between textual and visual modalities. Although substantial progress has been made, the field continues to face challenges such as limited annotated data and difficulty scaling to diverse domains. To address this, SAM [16] proposes a promptable foundation model for segmentation that delivers strong zero-shot performance. Building on this, HQ-SAM [15] leverages the architecture and prior knowledge of SAM to generate higher-quality masks. Similarly, MedSAM [21] demonstrates the potential of adapting SAM for medical image segmentation, underscoring the broad applicability of these foundation models.

LLM Agent. The emergence of agentic frameworks such as AutoGPT [43], MetaGPT [11], and HuggingGPT [34] illustrates the rapid integration of Large Language Models (LLMs) for performing

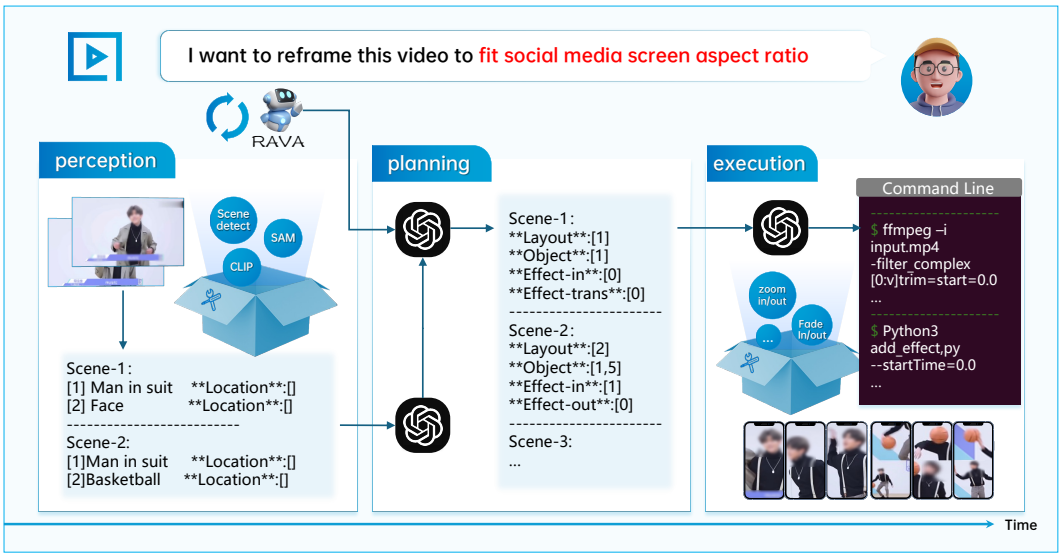


Figure 2: The overall workflow of our proposed **R**eframe **A**ny **V**ideo **A**gent (RAVA). RAVA receives user input through a language user interface (LUI) tailored for reframing tasks, invokes an object grounding function to retrieve relevant visual information from the video, and then automatically performs reframing based on the user's request.

complex and autonomous tasks. With the development of multimodal LLMs, including Flamingo [2], Multimodal [8], and AudioLM [35], these models have evolved to handle diverse input modalities—text, images, audio, and video—directly. This shift contrasts with earlier systems such as TaskMatrix [20], which rely on auxiliary models to translate visual inputs into linguistic representations through image captioning or object recognition. Leveraging these enhanced perceptual capabilities, recent efforts such as AppAgent [44], MobileAgent [39], and VisualWebArena [17] have developed agents capable of interacting with mobile applications and executing web-based tasks. Despite the growing body of research on LLM agents, their application in video editing remains relatively underexplored. One recent work, LAVE [38], introduces an agent capable of performing user-goal-driven video editing. However, its functionalities are limited in scope. In contrast, our proposed research delves deeper into leveraging LLMs for *automated video reframing*, aiming to enhance the precision, flexibility, and human-alignment of video editing systems.

3 Reframe Any Video Agent

We introduce the **R**eframe **A**ny **V**ideo **A**gent (RAVA), a novel framework for real-world video reframing that leverages the power of large language models and supports a Language User Interface (LUI) for intuitive user interaction. RAVA is designed to operate in open-world scenarios, where video content may include previously unseen objects. The system robustly identifies all objects within a scene, determines their relevance, and reframes video content into various aspect ratios tailored to the requirements of different social media platforms.

In addition, RAVA enhances the visual experience by supporting two types of visual effects: intra-scene (within a single scene) and inter-scene (between scenes). The entire video reframing process is fully automated and consists of the following three key stages:

Object Grounding. Given an original video composed of M scenes, denoted as $\{\mathbf{S}_1, \ldots, \mathbf{S}_M\}$, each scene $\mathbf{S}_k$ contains N visual elements represented by segmentation masks $\{\mathbf{O}_1, \ldots, \mathbf{O}_N\}$. The objective is to identify the most salient object(s), i.e., $\{\mathbf{O}_i, \ldots, \mathbf{O}_j\} \subseteq \{\mathbf{O}_1, \ldots, \mathbf{O}_N\}$, within each scene $\mathbf{S}_k$.

Layout Setting. In scenarios involving multiple important objects—such as conversational scenes—it is necessary to determine a layout $\mathbf{L}_k \in \{1, 2, 3, \ldots, N\}$ that specifies how these objects should be arranged in sub-windows for simultaneous display. For each scene $\mathbf{S}_k$, the final layout is expressed

Figure 3: Video editing tools in RAVA: The first row shows *Layout Settings*, where `L` indicates the number of selected objects. The second row, *Effect In-scene*, represents visual effects applied within a scene, including `Zoom in` and `Zoom out`. The third row, *Effect Trans-scene*, illustrates transition effects across scenes, including `Fade in` and `Fade out`.

as $\mathbf{L}_k = n$, where $n = \text{count}\{i, \ldots, j\}$ corresponds to the number of selected salient objects $\{\mathbf{O}_i, \ldots, \mathbf{O}_j\}$.

Effect Adding. Once layout configuration is complete, RAVA applies appropriate visual effects based on scene content. Intra-scene effects include operations such as zooming in and out to emphasize focus, while inter-scene effects—such as fade-ins and fade-outs—are used during transitions between scenes $\mathbf{S}_k$ and $\mathbf{S}_{k+1}$.

As illustrated in Figure 2, RAVA's workflow automates the video reframing pipeline, enabling adaptation to various aspect ratios and platform-specific standards. Through intelligent object prioritization, layout adjustment, and visual effect integration, RAVA enhances user engagement and streamlines content optimization for different audiences and distribution platforms.

3.1 Perception

The perception phase of RAVA is divided into two core components: *language learning* and *video understanding*. Language learning aims to capture the user's focus and intent, while video understanding interprets the visual content present in video frames.

LLMs are particularly adept at dialogue comprehension. By designing tailored prompts, the agent can effectively interpret user goals—essentially structuring the input information received through the Language User Interface (LUI). Specifically, we initiate the process by providing RAVA with a video and a user-defined interest. This context includes both natural language input and additional information retrieved from auxiliary tools, as detailed below. The LLM then produces a video topic along with structured target information, which serves as input to the subsequent planning phase.

Inspired by cinematic scripts and production workflows, we employ shot detection to segment the video into meaningful scenes. This is achieved using `scenedetect`[2], which divides the original video into M scenes, denoted as $\{\mathbf{S}_1, \ldots, \mathbf{S}_M\}$.

To understand these scenes, several tools are incorporated. RAM [46] is used to identify all visible objects, followed by SAM [16] and Grounded-SAM [32] for extracting segmentation masks and spatial locations. CLIP [30] is utilized to generate captions for each object. Consequently, each

[2] `https://www.scenedetect.com/`

visual element $\mathbf{O}_i$ is represented by a triplet of *caption*, *mask*, and *bounding box coordinates* $\{x_1, y_1, x_2, y_2\}$.

Ultimately, each scene's visual semantics are converted into a structured textual description. For example, as illustrated in Figure 2, a scene may be described as: "Scene-1: Object-1: a boy standing in...". This integration of natural language understanding and video analysis enables RAVA to comprehensively perceive video content, laying the foundation for context-aware reframing.

3.2 Planning

Following the perception phase, which generates structured semantic understanding of video content, the planning phase is responsible for devising a comprehensive strategy to guide video reframing. This plan must accommodate varying aspect ratios, emphasize key objects, and apply suitable visual effects to optimize both aesthetic quality and viewer engagement.

The planning phase in RAVA comprises the following components:

Aspect Ratio Determination. Determining the appropriate aspect ratio is fundamental. This decision takes into account user preferences, platform constraints, and scene composition. The system dynamically selects an optimal aspect ratio for each scene to ensure effective visual communication.

Object Importance Hierarchy. Among the objects identified in each scene, a prioritization mechanism is needed. Leveraging the reasoning capabilities of the LLM, RAVA constructs a hierarchy of object importance based on contextual relevance and user interest, which informs both selection and layout decisions.

Dynamic Layout Configuration. Based on object importance and spatial positioning, RAVA generates a layout configuration that enhances narrative coherence. As illustrated in Figure 3, layout choices account for dialog interactions, scene dynamics, and the number of focal objects, arranging them into sub-windows if necessary.

Visual Effect Strategy. RAVA formulates a plan for applying both in-scene and trans-scene effects, guided by user input or automatically inferred intent. This includes decisions about the type, intensity, and timing of effects such as zoom or fade, ensuring they support the storytelling rather than distract from it.

Execution Blueprint. All planning results are compiled into a structured execution blueprint, encompassing scene-specific information such as aspect ratios, layout configurations, object selections, and visual effect instructions. This blueprint is designed for direct parsing in the execution phase.

Agent Feedback Loop. Optionally, the agent can validate preliminary outputs by generating low-resolution previews. These previews are reviewed by the LLM, which compares the result against user goals and refines the blueprint if necessary.

This planning mechanism functions like a storyboard, offering a scene-by-scene breakdown of actions to be executed. It transforms high-level directives into a detailed plan, ensuring smooth and goal-aligned execution.

3.3 Execution

In the final execution phase, RAVA converts the planned blueprint into concrete actions. Using regular expression parsing, the system extracts structured directives from the plan, each of which corresponds to an executable function. For every scene $\mathbf{S}_k$ in the video $\{\mathbf{S}_1, \ldots, \mathbf{S}_M\}$, the execution settings include layout $\mathbf{L}_k \in \{1, 2, \ldots, N\}$, selected object set $\{\mathbf{O}_i, \ldots, \mathbf{O}_j\}$, and visual effects for in-scene $\mathbf{E}_{in} \in \{\text{zoom in}, \text{zoom out}\}$ and trans-scene $\mathbf{E}_{trans} \in \{\text{fade in}, \text{fade out}\}$.

The system generates a JSON file encoding all execution settings for each scene. For example, if $\mathbf{L}_k = 2$, the system selects two primary objects and arranges them vertically. If $\mathbf{E}_{in} = \{\text{zoom in}\}$, the corresponding API is called to magnify the objects. Similarly, if $\mathbf{E}_{trans} = \{\text{fade out}\}$, the system inserts a fade-out transition at the end of the scene.

While this work presents relatively simple implementations of visual effects, the underlying framework is modular and extensible. Additional effects or editing functionalities can be seamlessly

Table 1: Quantitative results for the Video Salient Object Detection (VSOD) task evaluated on the DAVIS_{16} and FBMS datasets. We compare our proposed method against two baselines: UPL and A2S-v2, under different scene detection (SD) configurations, where 'SD' denotes the number of scenes segmented from the input video using scene boundary detection algorithms. The performance is assessed using four commonly used metrics: Mean Absolute Error (MAE), maximum F-measure (max-F_β), maximum Enhanced-alignment metric (max-E_m), and Structure-measure (S_m). Higher values of F_β, E_m, and S_m and lower values of MAE indicate better performance.

Method	SD		DAVIS_{16}				FBMS			
	α_1	α_2	MAE	max-F_β	max-E_m	S_m	MAE	max-F_β	max-E_m	S_m
UPL	5	5	.0390	.8025	.9183	.8426	.0850	.6651	.8513	.7439
A2S-v2			.0663	.4858	.5786	.5817	.0851	.6444	.8366	.7004
Ours			.0501	.7025	.8219	.7795	.1015	.5721	.7532	.6643
UPL	5	30	.0367	.8127	.9275	.8481	.0844	.6673	.8458	.7527
A2S-v2			.0638	.5046	.5929	.5926	.0832	.6406	.8288	.7054
Ours			.0419	.6727	.8177	.7680	.1148	.5446	.7131	.6422
UPL	10	5	.0381	.8009	.9210	.8361	.0848	.6670	.8615	.7373
A2S-v2			.0640	.4907	.5723	.5836	.0900	.6299	.8446	.6836
Ours			.0506	.7126	.8256	.7804	.1313	.5128	.6776	.6089

integrated by extending the action space and API interface. This design ensures that RAVA is not only effective for current reframing needs but also adaptable to future demands in video editing.

4 Experiments

To evaluate the effectiveness of the proposed **R**eframe **A**ny **V**ideo **A**gent (RAVA), we conduct experiments on two key tasks that are central to video understanding and editing.

Video Salient Object Detection. In the first task, we apply RAVA to the well-established challenge of video salient object detection, which involves segmenting the most visually prominent objects as perceived by human viewers. This task serves as a proxy for evaluating the agent's ability to comprehend and prioritize visual content in alignment with human perception.

Video Reframing. In the second task, we evaluate RAVA on the video reframing task, where the goal is to adjust the framing of video scenes to emphasize the most important elements. This not only improves the visual composition but also enhances the narrative quality and user engagement of the content.

These two tasks collectively demonstrate RAVA's capabilities in both low-level visual understanding and high-level editing decision-making, validating its potential as a general-purpose agent for AI-driven video editing.

4.1 Video Salient Object Detection

Datasets & Metrics. We evaluate RAVA on two widely-used benchmarks: DAVIS_{16} [28] and FBMS [23]. The DAVIS_{16} dataset comprises 50 videos with a total of 3,455 annotated frames, while FBMS includes 33 videos and 720 annotated frames. To assess performance, we employ four commonly adopted metrics: Mean Absolute Error (MAE) [27], F-measure (F_β) [1], E-measure (E_m) [7], and S-measure (S_m) [6].

Settings. All videos are composed at 30 frames per second and then segmented into scenes using shot detection. To ensure effective scene segmentation, we select a low threshold value α_1 and a relatively high minimum scene length α_2—corresponding to the parameters `threshold` and `min scene length` in `scenedetect`—to reduce the likelihood of detecting overly short or fragmented scenes. Following segmentation, each scene is individually processed to generate salient object masks. It is worth noting that in the perception phase, each object $\mathbf{O}_i$ is represented only by a caption and a mask. This design choice is made to avoid excessive reframing, which could lead to jittery transitions and negatively impact the viewing experience.

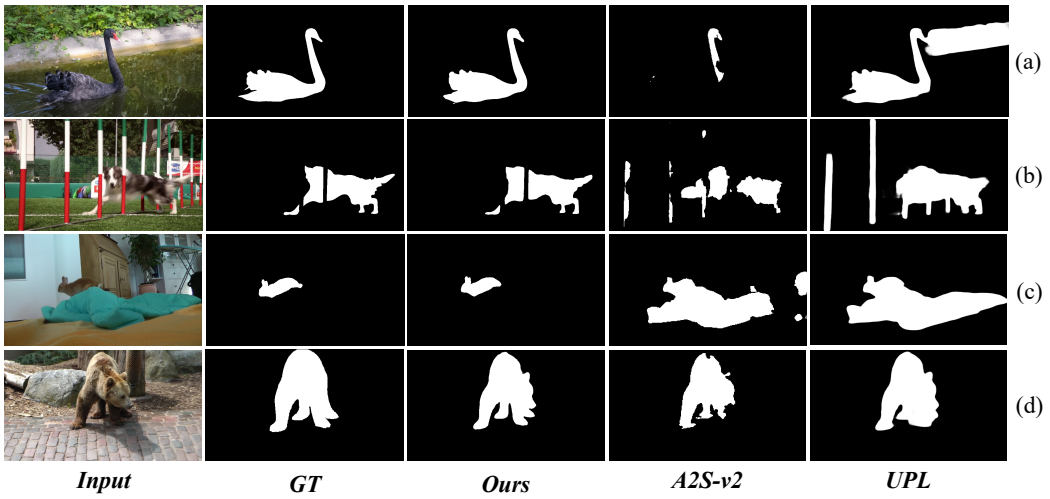


Figure 4: Qualitative comparisons on two video salient object detection datasets against two state-of-the-art methods. RAVA demonstrates robustness in challenging cases involving occlusion and distractions, and occasionally even surpasses human annotations.

Results. We compare RAVA against two state-of-the-art video salient object detection methods: UPL [42] and A2S-v2 [47]. As shown in Table 1, across different scene detection settings, RAVA consistently achieves competitive performance. Although RAVA is not specifically designed for video salient object detection, its strong performance validates the effectiveness and generality of the framework.

We present qualitative results in Figure 4 for further analysis:

- **(a)** RAVA successfully segments the full instance of the *Blackswan*, while A2S-v2 yields an incomplete mask, and UPL incorrectly includes background elements.
- **(b)** Under occlusion, RAVA maintains accurate segmentation; A2S-v2 incorrectly captures parts of the occluding object, and UPL fails to account for the occlusion entirely.
- **(c)** When faced with distractors in the scene, RAVA correctly isolates the salient object, whereas A2S-v2 and UPL are misled by irrelevant elements.
- **(d)** In certain challenging scenes, RAVA achieves results that are visually more accurate than the provided human annotations, highlighting the strength of its segmentation capabilities.

Ablation Study. To evaluate the importance of visual perception, we conduct an ablation study by replacing the multimodal LLM in RAVA with GPT-4 [25], which lacks direct visual processing capabilities. While the visual inputs are omitted during perception, all other components and settings are kept unchanged.

The results, presented in Table 2, show that even without direct visual input, GPT-4—guided by textual scene descriptions—achieves reasonably good performance on both datasets. This highlights the robustness and transferability of RAVA's architecture. Nonetheless, the performance gap confirms the necessity of incorporating vision-capable LLMs to achieve optimal results in multimodal tasks such as video salient object detection.

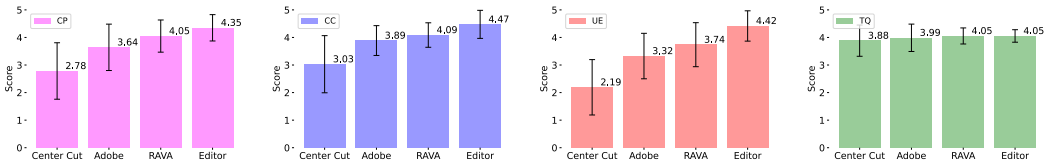
4.2 Video Reframing

Settings. To assess the video editing capabilities of RAVA in the wild, a user study is conducted with 12 participants. Edited versions of 20 videos are created using three reframe methodologies in addition to RAVA:

- **Editor**: This method involves a professional video editor (experience more than 3 years) who manually reframe the videos.

Table 2: Performance of our framework on the Video Salient Object Detection (VSOD) task when equipped with a single-modality Large Language Model (LLM), specifically GPT-4. The results demonstrate that even when restricted to a single-modality LLM without direct visual input, the system can achieve reasonable performance on both datasets, highlighting the strong generalization ability of language-based perception and planning.

LLM	SD		$DAVIS_{16}$				FBMS			
	α_1	α_2	MAE	max-F_β	max-E_m	S_m	MAE	max-F_β	max-E_m	S_m
GPT-4	5	5	.0831	.6497	.7885	.7395	.1294	.4953	.6943	.6125
	5	30	.0548	.6163	.7930	.7422	.1642	.4856	.6670	.5915
	10	5	.0690	.6432	.7913	.7454	.1307	.4931	.6684	.6069



(a) CP (b) CC (c) UE (d) TQ

Figure 5: Overall scores of the individual attributes: Content Preservation (CP), Continuity and Consistency (CC), User Experience (UE), and Technical Quality (TQ).

- **Adobe**: This method is based on the results obtained by ordinary users utilizing the reframe tool in Adobe Premiere Pro to adjust the videos, following the instructions[3].
- **Center Cut**: This method selects the center point of the video, maintaining a 9:16 aspect ratio, with the width unchanged.

To minimize the impact of the video itself and to maintain an element of unbiased evaluation by users, the open caption tool[4] is employed to add captions for each video. After watching the original video, each participant views the 4 edited versions in a random sequence. The participants review all reframed videos, and they are unaware of the editing methods employed for each video. This arrangement led to a comprehensive experimental design involving 20 (number of videos) × 12 (users) × 4 (editing strategies). Users are required to compare the reframed version of a video with the original and provide a rating on a scale from 0 to 5 for each of the attributes. These attributes were inspired by studies on video re-positioning[22], and it's important to note that, although video reframing and video repositioning differ technically, both aim to direct the attention of the viewer to the focal scene events within given rendering constraints. Thus, some of the questions used to assess methods of video re-positioning are also applicable to video editing. Our attributes of interest include: Content Preservation, Continuity and Consistency, User Experience, and Technical Quality.

Results. As seen in the results presented in Figure 5 and outlined in our experimental data, the traditional editing method, referred to as 'Editor', received the highest overall mean score of 4.32, indicating a strong ability to maintain the relevance and completeness of the original content. This could be attributed to the manual effort and expertise that video editors bring to the reframing endeavor, ensuring that significant elements are not lost. Our proposed method, RAVA, achieves an avarage score of 3.98 from four aspects, suggesting that while RAVA performs reasonably well in terms of relevancy and scene completeness, there is room for improvement when compared to professional editing. The performance of RAVA surpasses the automated 'Adobe' tool, with 'Adobe' scoring an mean score of 3.71. This close competition hints that RAVA is on par with other available semi-automated video editing tools in terms of content preservation. The 'Center Cut' received the lowest mean score of 2.78 in Content Preservation, reflecting its limited ability to identify and maintain critical video elements.

The variability in scores, as indicated by the interquartile range in the boxplot, further substantiates the need for an intelligent and context-aware reframing technique. Future work could explore enhancing

[3]https://helpx.adobe.com/premiere-pro/using/auto-reframe.html

[4]https://www.opus.pro/tools/opusclip-captions

the object identification and importance determination algorithms of RAVA to further close the gap between automated and professional video editing tools.

Besides, we strongly recommend that readers view the edited video included in the supplementary materials for a more intuitive understanding.

5 Conclusion

In this work, we introduce **R**eframe **A**ny **V**ideo **A**gent (RAVA), a novel LLM-based agent designed to perform video reframing tasks guided by human instructions. Leveraging the powerful capabilities of large language models, RAVA follows a structured three-stage pipeline—*perception*, *planning*, and *execution*—to accurately interpret user directives, analyze video content, prioritize salient objects, determine optimal layouts, and apply appropriate visual effects. This design ensures that the reframed output aligns closely with both narrative intent and user preferences. Our extensive experiments, encompassing both classic computer vision tasks and real-world video reframing scenarios, validate the effectiveness of RAVA and highlight its potential in enabling AI-assisted video editing. Through quantitative evaluations and user studies, we demonstrate that RAVA significantly enhances the efficiency and personalization of video content creation, thereby offering a powerful tool for content producers across diverse platforms.

Limitations. Despite its promising performance, the current system inherits limitations from its reliance on foundational models. While these models provide strong general-purpose capabilities, their performance can become a bottleneck depending on task complexity or domain specificity. Future work can explore the integration of more advanced or specialized visual models to further boost performance. Additionally, extending the agent's capabilities to perform temporal video editing—such as condensing long-form content into concise highlights—represents a compelling direction for enhancing the flexibility and applicability of LLM-based video editing agents.

References

[1] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In *2009 IEEE conference on computer vision and pattern recognition*, pages 1597–1604. IEEE, 2009.

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

[3] Dawit Mureja Argaw, Fabian Caba Heilbron, Joon-Young Lee, Markus Woodson, and In So Kweon. The anatomy of video editing: a dataset and benchmark suite for ai-assisted video editing. In *European Conference on Computer Vision*, pages 201–218. Springer, 2022.

[4] Christel Chamaret and Olivier Le Meur. Attention-based video reframing: Validation using eye-tracking. In *2008 19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008.

[5] Thomas Cochrane. Mobile social media as a catalyst for pedagogical change. In *EdMedia+ innovate learning*, pages 2187–2200. Association for the Advancement of Computing in Education (AACE), 2014.

[6] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *Proceedings of the IEEE international conference on computer vision*, pages 4548–4557, 2017.

[7] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment measure for binary foreground map evaluation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2018.

[8] Hiroki Furuta, Ofir Nachum, Kuang-Huei Lee, Yutaka Matsuo, Shixiang Shane Gu, and Izzeddin Gur. Multimodal web navigation with instruction-finetuned foundation models. *arXiv preprint arXiv:2305.11854*, 2023.

[9] Tiantian Geng, Teng Wang, Jinming Duan, Runmin Cong, and Feng Zheng. Dense-localizing audio-visual events in untrimmed videos: A large-scale benchmark and baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22942–22951, 2023.

[10] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *European Conference on Computer Vision*, pages 540–557. Springer, 2022.

[11] Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 2023.

[12] Feiyan Hu, Simone Palazzo, Federica Proietto Salanitri, Giovanni Bellitto, Morteza Moradi, Concetto Spampinato, and Kevin McGuinness. Tinyhd: Efficient video saliency prediction with heterogeneous decoders using hierarchical maps distillation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2051–2060, 2023.

[13] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021.

[14] Lai Jiang, Mai Xu, Tie Liu, Minglang Qiao, and Zulin Wang. Deepvs: A deep learning based video saliency prediction approach. In *Proceedings of the european conference on computer vision (eccv)*, pages 602–617, 2018.

[15] Lei Ke, Mingqiao Ye, Martin Danelljan, Yu-Wing Tai, Chi-Keung Tang, Fisher Yu, et al. Segment anything in high quality. *Advances in Neural Information Processing Systems*, 36, 2024.

[16] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023.

[17] Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*, 2024.

[18] John Lambert, Zhuang Liu, Ozan Sener, James Hays, and Vladlen Koltun. Mseg: A composite dataset for multi-domain semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2879–2888, 2020.

[19] Boyi Li, Kilian Q. Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation, 2022.

[20] Yaobo Liang, Chenfei Wu, Ting Song, Wenshan Wu, Yan Xia, Yu Liu, Yang Ou, Shuai Lu, Lei Ji, Shaoguang Mao, et al. Taskmatrix. ai: Completing tasks by connecting foundation models with millions of apis. *arXiv preprint arXiv:2303.16434*, 2023.

[21] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. *Nature Communications*, 15(1):654, 2024.

[22] KL Bhanu Moorthy, Moneish Kumar, Ramanathan Subramanian, and Vineet Gandhi. Gazed–gaze-guided cinematic editing of wide-angle monocular video recordings. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–11, 2020.

[23] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1187–1200, 2013.

[24] OpenAI. Chatgpt. `https://openai.com/research/chatgpt`, 2021.

[25] OpenAI. Gpt-4. `https://openai.com/research/gpt-4`, 2023.

[26] OpenAI. Gpt-4v. `https://openai.com/research/gpt-4v-system-card`, 2023.

[27] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *2012 IEEE conference on computer vision and pattern recognition*, pages 733–740. IEEE, 2012.

[28] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016.

[29] Pedro O Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1713–1721, 2015.

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

[31] Anyi Rao, Xuekun Jiang, Sichen Wang, Yuwei Guo, Zihao Liu, Bo Dai, Long Pang, Xiaoyu Wu, Dahua Lin, and Libiao Jin. Temporal and contextual transformer for multi-camera editing of tv shows. *arXiv preprint arXiv:2210.08737*, 2022.

[32] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

[33] Ana Serrano, Vincent Sitzmann, Jaime Ruiz-Borau, Gordon Wetzstein, Diego Gutierrez, and Belen Masia. Movie editing and cognitive event segmentation in virtual reality video. *ACM Transactions on Graphics (TOG)*, 36(4):1–12, 2017.

[34] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36, 2024.

[35] Fangxun Shu, Lei Zhang, Hao Jiang, and Cihang Xie. Audio-visual llm for video understanding. *arXiv preprint arXiv:2312.06720*, 2023.

[36] Yukun Su, Jingliang Deng, Ruizhou Sun, Guosheng Lin, Hanjing Su, and Qingyao Wu. A unified transformer framework for group-based segmentation: Co-segmentation, co-saliency detection and video salient object detection. *IEEE Transactions on Multimedia*, 2023.

[37] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *Proceedings of the European conference on computer vision (ECCV)*, pages 247–263, 2018.

[38] Bryan Wang, Yuliang Li, Zhaoyang Lv, Haijun Xia, Yan Xu, and Raj Sodhi. Lave: Llm-powered agent assistance and language augmentation for video editing. *arXiv preprint arXiv:2402.10294*, 2024.

[39] Junyang Wang, Haiyang Xu, Jiabo Ye, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. Mobile-agent: Autonomous multi-modal mobile device agent with visual perception. *arXiv preprint arXiv:2401.16158*, 2024.

[40] Wenguan Wang, Jianbing Shen, Jianwen Xie, Ming-Ming Cheng, Haibin Ling, and Ali Borji. Revisiting video saliency prediction in the deep learning era. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):220–237, 2019.

[41] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero-and few-label semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8256–8265, 2019.

[42] Pengxiang Yan, Ziyi Wu, Mengmeng Liu, Kun Zeng, Liang Lin, and Guanbin Li. Unsupervised domain adaptive salient object detection through uncertainty-aware pseudo-label learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3000–3008, 2022.

[43] Hui Yang, Sifu Yue, and Yunzhong He. Auto-gpt for online decision making: Benchmarks and additional opinions. *arXiv preprint arXiv:2306.02224*, 2023.

[44] Zhao Yang, Jiaxuan Liu, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. Appagent: Multimodal agents as smartphone users. *arXiv preprint arXiv:2312.13771*, 2023.

[45] Yichen Yuan, Yifan Wang, Lijun Wang, Xiaoqi Zhao, Huchuan Lu, Yu Wang, Weibo Su, and Lei Zhang. Isomer: Isomerous transformer for zero-shot video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 966–976, 2023.

[46] Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, et al. Recognize anything: A strong image tagging model. *arXiv preprint arXiv:2306.03514*, 2023.

[47] Huajun Zhou, Bo Qiao, Lingxiao Yang, Jianhuang Lai, and Xiaohua Xie. Texture-guided saliency distilling for unsupervised salient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7257–7267, 2023.

A Rationale

Having the supplementary compiled together with the main paper means that:

- The supplementary can back-reference sections of the main paper, for example, we can refer to sec:intro;
- The main paper can forward reference sub-sections within the supplementary explicitly (e.g. referring to a particular experiment);
- When submitted to arXiv, the supplementary will already included at the end of the paper.

To split the supplementary pages from the main paper, you can use Preview (on macOS), Adobe Acrobat (on all OSs), as well as command line tools.