A Benchmark and Multi-Agent System for Instruction-driven Cinematic Video Compilation

Peixuan Zhang[#1], Chang Zhou[†2], Ziyuan Zhang[3], Hualuo Liu[2], Chunjie Zhang[2], Jingqi Liu[4,7], Xiaohui Zhou[2], Xi Chen[2], Shuchen Weng[*4,5], Si Li[*1,8], and Boxin Shi[5,6]

[1] School of Artificial Intelligence, Beijing University of Posts and Telecommunications, China
[2] AI Technology Center, Online Video Business Unit, Tencent PCG, China
[3] Tsinghua University, China [4] Beijing Academy of Artificial Intelligence, China
[5] State Key Lab of Multimedia Info. Processing, Computer Science, Peking University, China
[6] Nat'l Eng. Research Ctr. of Visual Technology, Computer Science, Peking University, China
[7] School of Software and Microelectronics, Peking University, China
[8] Beijing Key Laboratory of Multimodal Data Intelligent Perception and Governance, China

{pxzhang, lisi}@bupt.edu.cn, shuchenweng@pku.edu.cn, chanzhou@tencent.com

Abstract. The surging demand for adapting long-form cinematic content into short videos has motivated the need for versatile automatic video compilation systems. However, existing compilation methods are limited to predefined tasks, and the community lacks a comprehensive benchmark to evaluate the cinematic compilation. To address this, we introduce CineBench, the first benchmark for instruction-driven cinematic video compilation, featuring diverse user instructions and high-quality ground-truth compilations annotated by professional editors. To overcome contextual collapse and temporal fragmentation, we present CineAgents, a multi-agent system that reformulates cinematic video compilation into "design-and-compose" paradigm. CineAgents performs script reverse-engineering to construct a hierarchical narrative memory to provide multi-level context and employs an iterative narrative planning process that refines a creative blueprint into a final compiled script. Extensive experiments demonstrate that CineAgents significantly outperforms existing methods, generating compilations with superior narrative coherence and logical coherence.

Keywords: Video Compilation · Multi-Agent System · Secondary Creation · Cinematic Production

1 Introduction

The explosive growth of short video platforms (*e.g.*, Tencent Video [3], TikTok [4], and YouTube [5]) has fundamentally reshaped how people consume entertainment [31]. Audiences are increasingly drawn to fast-paced short videos

[#] Work done during an internship at Tencent PCG.
[†] Project leader. [*] Corresponding authors.

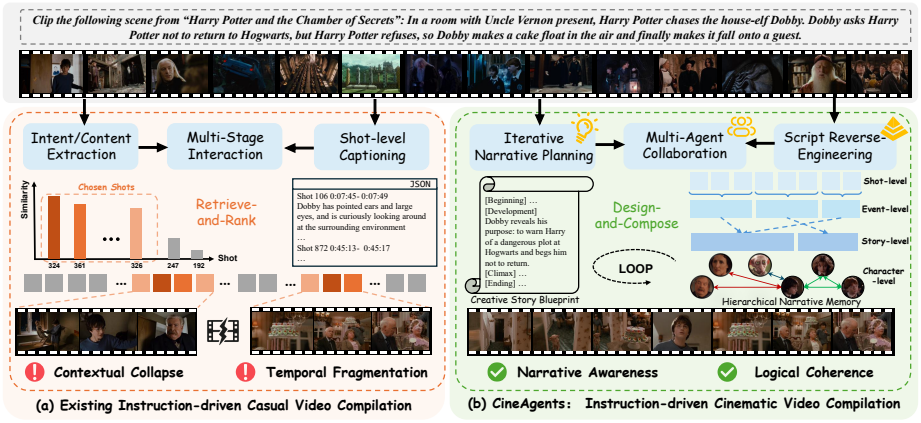

Fig. 1: Comparison of video compilation paradigms. (a) Existing video compilation methods typically rely on a "retrieve-and-rank" paradigm. When applied to complex cinematic videos, this isolated approach inevitably leads to contextual collapse and temporal fragmentation, leading to narratively incoherent results. (b) In contrast, our proposed **CineAgents** introduces a novel "design-and-compose" paradigm. Through multi-agent collaboration, the system first reverse-engineers the source video into a hierarchical narrative memory, and then performs iterative narrative planning to produce a logically coherent and compelling final compilation.

over traditional long-form cinematic content. This trend has motivated a massive community of secondary creators to adapt existing films and TV series into condensed clips, driving an urgent demand for versatile automatic video compilation approaches to streamline this process [34, 38, 41].

Early instruction-driven compilation works [21, 26, 35, 45, 58] predominantly focused on casual videos characterized by linear narratives. While recent works have explored the cinematic domain, they remain constrained to predefined tasks (*e.g.*, B-roll insertion [24], trailer generation [49], and highlight detection [17, 51]). Consequently, the community lacks a comprehensive benchmark to evaluate cinematic compilation under versatile user instructions, limiting the development of generalized compilation systems.

In this paper, we first propose **CineBench**, a comprehensive benchmark for the instruction-driven cinematic video compilation task. Unlike casual videos that typically feature linear narratives (*e.g.*, vlogs and scenic recordings), cinematic content is built upon intricate plot structures, character arcs, and thematic undertones. This benchmark includes in-the-wild user instructions that range from high-level creative intents to specific editing operations, and evaluates models' abilities to perform cinematic compilation by selecting, reordering, and assembling video shots into a coherent final sequence. To ensure a rigorous evaluation of these complex elements, we recruited professional video editors to perform narrative re-authoring, creatively fulfilling each instruction to construct high-quality ground-truth compilations. As a result, CineBench contains over

500 instruction-video pairs from 70 diverse films and series, accompanied by 11 evaluation metrics organized into three core aspects.

Building upon this benchmark, we observe that existing casual video compilation methods face two key challenges: *(i) Contextual collapse*: For long cinematic videos, shot-level captions often suffer from informational redundancy and narrative inaccuracy. Consequently, models fail to grasp the broader context and retrieve shots that align with the user's instruction (*e.g.*, Fig. 1 (a) left, the model incorrectly includes a shot of Harry and Dobby struggling over a lamp, as it over-relies on matching the characters from the prompt while ignoring the narrative context). *(ii) Temporal fragmentation*: Current methods over-rely on matching isolated content from the user's instruction, preventing them from constructing a coherent narrative flow. As a result, the generated video becomes a temporally fragmented sequence of clips lacking narrative coherence (*e.g.*, Fig. 1 (a) right, the video abruptly cuts from Harry and Dobby's struggle to a cake falling on a guest, omitting the connecting plots). Fundamentally, these challenges stem from their "retrieve-and-rank" formulation, which concatenates isolated shots based on local similarities while neglecting global narrative dependencies.

To address these limitations, we present **CineAgents**, a multi-agent system inspired by the collaboration of a professional editing studio. Specifically, we reformulate cinematic video compilation as a "design-and-compose" paradigm that orchestrates a collaborative workflow, as illustrated in Fig. 1 (b). To overcome contextual collapse, script reverse-engineering converts the long cinematic video into a structured script. This script constructs a hierarchical narrative memory, providing narrative context by organizing the cinematic video at multiple levels (*i.e.*, shot, event, story, and character). To overcome temporal fragmentation, we translate the user's instruction into an initial creative story blueprint. Through an iterative narrative planning process, the system logically refines this blueprint and adjusts the shot sequence to produce the final compiled script. Once the script meets all editing requirements, the system directly leverages external video tools to seamlessly assemble the final clips.

In summary, our contributions are listed as follows:

- We construct CineBench, the first benchmark for instruction-driven cinematic video compilation, with professional editors to annotate ground truths.
- We present the CineAgents, a multi-agent system that reformulates instruction-driven cinematic video compilation into a design-and-compose paradigm.
- We introduce script reverse-engineering and hierarchical narrative memory, which analyze videos to provide narrative context across multiple levels.
- We propose an iterative narrative planning process that progressively refines the blueprint into compiled script, ensuring logical coherence.

2 Related Works

2.1 Automated Video Compilation

Early research in automatic video compilation primarily relied on the similarity analysis of extracted features, leveraging audio [36], textual [45,50], or visual [38]

cues. Notably, CMVE [26] introduced the first automatic pipeline for compilation based on a pre-defined story, which inspired subsequent works [9, 21, 35] to enhance editing precision by training models to ensure sequence coherence. However, these foundational methods are inherently designed for casual videos with linear narratives, lacking the representational capacity for complex cinematic content. To effectively analyze and reorganize such cinematic videos, researchers have proposed a wide range of approaches, though they remain heavily confined to predefined tasks. For instance, B-Script [24] focuses exclusively on B-roll insertion by analyzing primary footage for semantic cutaways; similarly, CCANet [49] and TGT [7] are tailored for trailer generation by scoring segments based on attention intensity, while HIVE [51] and CLC [17] are limited to highlight detection via high-impact moment sequencing. Ultimately, since these models are constrained to specific objectives, they cannot support compilation guided by in-the-wild user instructions, preventing users from realizing their diverse creative concepts.

2.2 Multi-Agent Cinematic Generation

While foundational visual processing techniques have made significant strides in specific domains such as image enhancement and modality decoupling [14–16,53, 61,63,64], the rapid advancement of Large Language Models (LLMs) [6,10,43,44] and Vision-Language Models (VLMs) [28–30, 56, 57, 65] has equipped AI systems with powerful reasoning and planning capabilities for complex visual tasks. Based on these foundation models [6, 42], recent studies have introduced LLM-driven planning and advanced context modeling to cinematic and long video creation [11, 18, 25, 60]. For instance, CineVerse [37], VideoGen-of-Thought [67] and STAGE [62] utilize LLMs to generate structured scripts, imposing a logical narrative on the synthesized videos. Concurrently, pioneering works such as Captain Cinema [55], Holocine [32], and CineScene [22] have further pushed the boundaries of multi-shot narrative generation and implicit 3D scene representation for cinematic outputs. This paradigm has further evolved into multi-agent collaboration [20,66], with frameworks like FilmAgent [59] and MovieAgent [54] proposing interactive systems for end-to-end cinematic video generation. However, despite their success in generating new content, these methods are ill-equipped for cinematic compilation, which fundamentally relies on logically reorganizing existing video shots into a coherent new narrative.

3 Benchmark

While cinematic video compilation approaches are rapidly emerging, existing benchmarks remain confined to predefined tasks (*e.g.*, B-roll insertion [24], trailer generation [7,49], and highlight detection [17,51]). This fundamentally limits the development of generalized compilation systems. To address this gap, we introduce **CineBench**, the first benchmark for instruction-driven cinematic video compilation, as illustrated by two representative examples in Fig. 2 (a).

3.1 Data Curation and Task Formulation

To construct a comprehensive and diverse benchmark, we first curate cinematic videos from both English- and Chinese-language films and television series[9]. These sources span a wide range of release years (1930-2020) and genres (*e.g.*, action, comedy, and romance) to ensure a broad representation of narrative styles. To facilitate a rigorous evaluation of in-the-wild instructions, we deconstruct user intentions into the following fundamental components.

Source selection. This component defines whether the compilation draws from a single source (*e.g.*, *The Shawshank Redemption*) or integrates footage from multiple sources (*e.g.*, combining *The Shawshank Redemption* and *Green Book*).

Target content. This component specifies the narrative focus of the compilation. This can range from a single subject, such as a character (*e.g.*, Andy Dufresne) or an event (*e.g.*, the prison escape), to multiple interwoven subjects, such as several interacting characters (*e.g.*, Tony Lip and Dr. Don Shirley) or parallel plotlines (*e.g.*, contrasting family life with solitude).

Temporal requirements. This component defines the temporal arrangement of clips, including preserving chronological order (*e.g.*, a 5-minute limit within the sequence), enforcing non-linear storytelling (*e.g.*, a highlight-first structure), and performing extractive edits (*e.g.*, trailer generation or story summarization).

Editing operations. This component specifies audio-visual enhancements applied to the final compilation, such as text overlays (*e.g.*, adding cinematic titles), background music (*e.g.*, applying a score matching the emotional tone), cover images (*e.g.*, using a movie poster as a title card), or transition effects (*e.g.*, inserting fade transitions for flashback sequences).

3.2 Ground Truth Annotation and Verification

Guided by these four fundamental components, we recruit five professional video editors to manually craft user instructions as ground truths after watching the entirety of each source film. To ensure the quality and consistency of these human-annotated pairs, we conduct an inter-annotator agreement analysis on a randomly selected 20% of the data, yielding a Cohen's Kappa [27] of 0.61, indicating substantial agreement among the experts. Detailed statistics regarding the distribution of cinematic videos and compilation instructions are provided in Fig. 2 (b) and (c), encompassing release years, genres, instruction lengths, and keyword frequencies.

To further assess model robustness and evaluate hallucination in compilation frameworks, we design a challenging subset of negative instructions. These are systematically crafted by introducing factual inconsistencies among the source videos, characters, and plot events (*e.g.*, requesting scenes of a character in a film in which they never appeared). In total, CineBench includes over 500 instruction-compilation pairs sourced from more than 70 films and series. Notably, 30% samples serve as an adversarial set specifically designed to evaluate a model's ability to recognize and appropriately refuse factually impossible requests.

[9] These materials are used strictly for non-commercial academic research purposes.

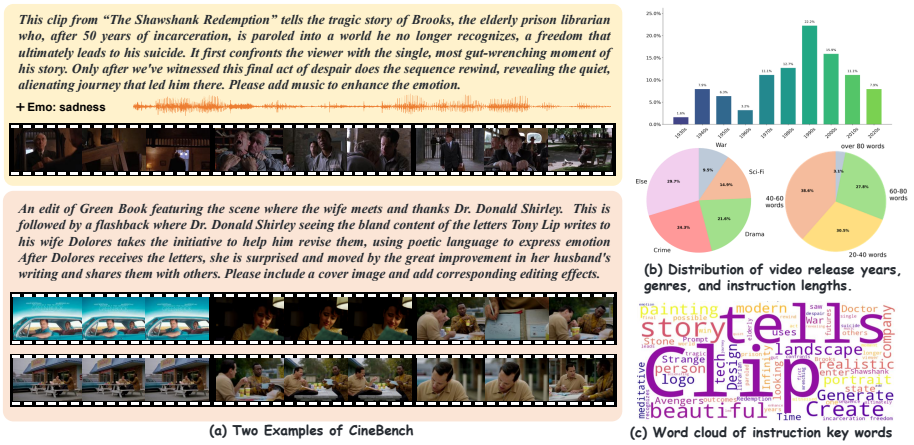

Fig. 2: Overview and statistic of CineBench. (a) Examples of instructions and compiled video clips. (b) Distribution of video release years, types and instruction lengths. (c) Word cloud of instruction keyword frequencies.

3.3 Evaluation Metrics

To enable a comprehensive evaluation of the instruction-driven cinematic video compilation, we introduce a protocol that assesses a model's performance across three key dimensions:

Narrative grounding. This dimension evaluates the model's comprehension of the source video's narrative. It is quantified using two metrics: inspired by Video-bench [23], we compute *Script-Video Consistency (SVC)* by calculating the semantic similarity between the generated script annotations and the original video at the shot level; and we employ a VLM [42] to rate the *Shot Coherence (SC)*, assessing the logical flow between consecutive shots in the generated script.

Retrieval precision. This dimension assesses the accuracy and relevance of the retrieved shots. The evaluation comprises several metrics: shot-level *Precision*, *Recall*, and *F1-score* measure selection accuracy against ground-truth annotations; the *Temporal Correctness Score (TCS)* evaluates the ratio of the correctly ordered shot sequence's total duration [46] to the generated video's total duration; and VLM-based scores calculates *Narrative Logic (NL)* and *Prompt Adherence (PA)*.

Overall quality. This dimension captures the holistic performance and reliability of the system. It is measured through three key indicators: the *Execution Success Rate (ESR)* tracks the percentage of instructions processed without system errors; the *Adversarial Rejection Rate (ARR)* evaluates the model's ability to correctly refuse factually impossible requests from our adversarial set; and a VLM-based *Compiled Quality (CQ)* score for the final compiled video.

All VLM-based evaluations are rated on a scale of 1 to 10 for readability.

4 Method

4.1 Overview

We design the **CineAgents** system to generate a final compiled video $\mathcal{V}_{\text{out}}$ from a user's text instruction $\mathcal{I}$ and a set of source videos $\mathcal{V}_{\text{src}}$. This process is initiated by the manager agent ($\mathcal{A}_{\text{magr}}$), which interprets the user's instruction to recruit the necessary specialist agents. As shown in Fig. 3, this task is accomplished through a two-stage process:

Script reverse-engineering. The script agent ($\mathcal{A}_{\text{scr}}$) first parses the source videos $\mathcal{V}_{\text{src}}$ into a shot-level script by integrating their visual, audio, and text modalities. It then constructs a hierarchical narrative memory $\mathcal{M}$ to provide a multi-level narrative context, addressing the challenge of contextual collapse. The process can be formulated as:

$$\mathcal{M} = \mathcal{A}_{\text{scr}}(\mathcal{V}_{\text{src}}) \tag{1}$$

Cinematic sequence production. This stage embodies "design-and-compose" paradigm to overcome temporal fragmentation. It begins with the "design" phase, where the director agent ($\mathcal{A}_{\text{dir}}$) and orchestrator agent ($\mathcal{A}_{\text{orch}}$) collaboratively perform iterative narrative planning. They transform the user's instruction $\mathcal{I}$ into a logically coherent compiled script $\mathcal{L}_{\text{final}}$ by leveraging the narrative memory $\mathcal{M}$. This is followed by the "compose" phase, where the editor agent ($\mathcal{A}_{\text{edit}}$) executes the script $\mathcal{L}_{\text{final}}$, assembling shots from $\mathcal{V}_{\text{src}}$ and applying external video tools to generate the final video $\mathcal{V}_{\text{out}}$. The entire process is formulated as:

$$\mathcal{V}_{\text{out}} = \mathcal{A}_{\text{edit}}\big((\mathcal{A}_{\text{orch}} \leftrightarrow \mathcal{A}_{\text{dir}})(\mathcal{I}, \mathcal{M}), \mathcal{V}_{\text{src}}\big) \tag{2}$$

4.2 Script Reverse-Engineering

The extensive length and multimodal information (text, visual, audio) of cinematic videos pose a significant challenge for existing video compilation methods, which will cause contextual collapse. To overcome this, we introduce Script Reverse-Engineering to transform the source videos into a hierarchical narrative memory, as shown in Fig. 3 (a).

Character detection and tracing. We employ an "anchor and propagate" strategy for robust character identification. Specifically, we establish identity anchors $\mathcal{A}_{\text{c}}$ by creating a character dataset $\mathcal{D}_{\text{cinfo}}$ from metadata (*e.g.*, Wikipedia). We perform character matching using InsightFace [2] for frontal views, while the SOLIDER [12] model handles non-frontal views where face recognition is unreliable. We then propagate these anchor identities along temporal trajectories, which are formed by grouping shots with similar appearance features. This process can be formulated as:

$$\text{ID}(T_k) = \arg\max_{c \in \mathcal{C}} \sum_{p_i \in T_k} \text{Cosine}(F(p_i), e_c), \tag{3}$$

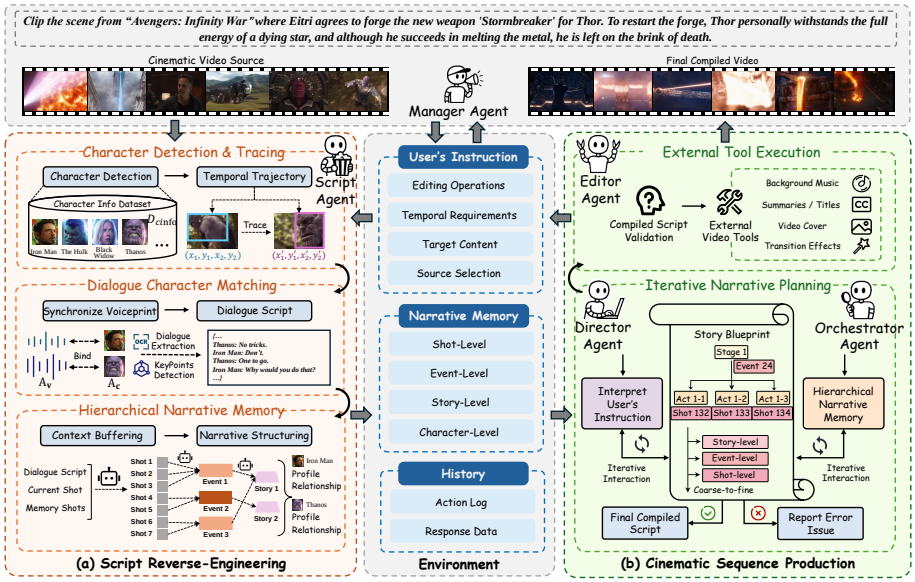

Fig. 3: Overview of the proposed CineAgents. Given a user instruction and a set of source videos, our system generates a final compiled video through: **(a) Script reverse-engineering:** The script agent integrates the multimodal information from the source videos to construct a hierarchical narrative memory, which provides a multi-level context to overcome contextual collapse (Sec. 4.2). **(b) Cinematic sequence production:** The director agent and orchestrator agent collaborate through iterative narrative planning to generate a compiled script to overcome temporal fragmentation. Subsequently, the editor agent applies external tools to assemble and enhance the final video (Sec. 4.3). This entire collaborative pipeline is orchestrated within cinematic compilation environment, where a manager agent supervises the workflow and coordinates the specialized agents to achieve the final narrative goal (Sec. 4.4).

where T_k is the k-th character trajectory composed of a set of person detections $\{p_i\}$. $F(p_i)$ is the feature embedding of a detection p_i extracted by either InsightFace or SOLIDER. $\mathcal{C}$ is the set of all candidate characters, and $e_c \in \mathcal{A}_c$ is the anchor embedding for character c. This strategy allows us to maintain consistent character identities even through occlusions and non-frontal poses.

Dialogue character matching. With character identities established, we link them to their corresponding dialogues. Since conventional ASR methods [8, 39] struggle with large casts and off-screen speakers, we create voiceprint anchors $\mathcal{A}_v$ from synchronized audio-visual cues. We extract dialogue from on-screen subtitles via OCR [13] and verify the on-screen speaker using lip activity detection with InsightFace [2] keypoints and a ResNet [19] classifier. We then use WeSpeaker [48] to extract audio features from all of a character's confirmed anchors and apply K-means clustering to these features to bind a unique voiceprint $\mathcal{V}_c$ to each character. Finally, using the established speaker-dialogue mapping, we

can associate each character in the cinematic source with their corresponding dialogue to generate the dialogue script.

Hierarchical narrative memory. With characters and dialogues identified, we construct the hierarchical narrative memory to distill the complex multimodal information into a multi-level structured representation. At the shot level, we first segment the video into individual shots using AutoShot [69]. To ensure these summaries maintain narrative flow, we buffer preceding 10 shots as memory to store key multimodal features (*e.g.*, visual embeddings, character IDs, and dialogue) from the history context. An LLM [42] is then provided with the current shot's information along with this historical context to produce a detailed summary. This context-aware summarization S_t for the t-th shot can be formulated as a conditional generation process:

$$S_t = \arg\max_S P(S \mid I_t, M_t), \tag{4}$$

where S_t is the generated summary, I_t is the current shot's information, and $M_t = \{I_{t-k}, \ldots, I_{t-1}\}$ is the memory buffer. Next, we group these summarized shots into coherent events using BaSSL [33], abstract them into story, and concurrently generate character profiles. This entire hierarchical structure is encapsulated within the final narrative memory $\mathcal{M}_{\text{narr}}$, which is defined as:

$$\mathcal{M}_{\text{narr}} = (\mathcal{S}_{\text{sum}}, S_{\text{event}}, f_{\text{story}}(S_{\text{event}}), f_{\text{profile}}(S_{\text{event}})), \tag{5}$$

where $S_{\text{event}} = f_{\text{event}}(\mathcal{S}_{\text{sum}})$ and $\mathcal{S}_{\text{sum}} = \{S_t\}$ is the set of all shot summaries from Eq. (4). The functions f_{event}, f_{story}, and f_{profile} represent the processes of event grouping, story abstraction, and profile generation, respectively.

4.3 Cinematic Sequence Production

While the hierarchical narrative memory provides structured access to video content, naively assembling shots based on user instructions can lead to temporally fragmented and logically incoherent results. To overcome this challenge, we introduce the "design-and-compose" paradigm to plan a story blueprint and then assemble the final video, as illustrated in Fig. 3 (b).

Iterative narrative planning. Our iterative planning module operates as a collaborative dialogue between two specialized agents. The director agent acts as the creative planner, translating the user's instruction into a story blueprint structured across distinct narrative stages (*e.g.*, beginning, rising action, climax). Conversely, the orchestrator agent serves as the validator, tasked with grounding each proposal in concrete evidence from the hierarchical narrative memory. Specifically, this unfolds as a top-down grounding process. The director agent initiates a high-level narrative intent, which the orchestrator attempts to ground at the story and character levels. If a consensus is reached, the blueprint is updated with the confirmed context, and the dialogue descends to the next level. Conversely, if supporting evidence is lacking, the orchestrator prompts the director to revise its proposal. This collaborative cycle of proposal, validation,

and refinement can be formally expressed as an iterative update to the story blueprint B:

$$B_{t+1} = B_t \oplus f_{\text{orch}}(f_{\text{dir}}(U, B_t), \mathcal{M}_{\text{narr}}), \tag{6}$$

where B_t is the blueprint at iteration t, and U is the initial user instruction. The function f_{dir} represents the director agent's proposal, while f_{orch} is the orchestrator agent's grounding function, which validates the proposal against the narrative memory $\mathcal{M}_{\text{narr}}$. The $\oplus$ operator signifies the update process: it integrates the successfully grounded evidence from f_{orch} into the blueprint. If grounding fails, the result of the right-hand side is an empty set, prompting a revision by f_{dir} in the next iteration. This cycle terminates when precise shots are identified to realize every request, finalizing the blueprint into a compiled script of specific shot IDs.

External tool execution. Once the manager agent confirms an error-free workflow and validates the compiled script against user requirements, the editor agent initiates the final editing stage. To fulfill the user's specified editing operations, our system provides a suite of four common external tools: adding background music, inserting text summaries or titles, generating a video cover, and applying transition effects. By comprehensively analyzing the user's intent and the final compiled script, the editor agent selects and applies the appropriate tools, ultimately generating the final edited video sequence.

4.4 Cinematic Compilation Environment

We define the cinematic compilation environment as an interactive workspace to facilitate our multi-agent system. Within this workspace, agents operate by referring to two foundational components: the parsed user's instruction, which defines the ultimate goal of the task, and the hierarchical narrative memory, which serves as the agent's internal knowledge base of the pre-analyzed cinematic video content. Interactions within this environment are governed by a structured message-passing protocol. Under this protocol, every agent action and its corresponding response are recorded in a shared history. This history serves a dual purpose: it not only provides dynamic context for subsequent decisions but also acts as a version-controlled log. Crucially, this log allows the system to reload a previous compilation state for direct modification, eliminating the need to repeat foundational steps for secondary edits. This mechanism also allows the manager agent to monitor the output of each action and make targeted interventions or responses as needed. This design ensures that all agent actions are traceable and consistently aligned with both the user's instructions and the source materials.

5 Experiments

5.1 Experiment Setup

We evaluate our method on the proposed CineBench, a benchmark tailored for instruction-driven cinematic video compilation. Our CineAgents is a training-free system that leverages the powerful foundation model Gemini-2.5-pro [42] to

Table 1: Quantitative experiment results of comparison and ablation. ↑ (↓) means higher (lower) is better. The best performances are highlighted in **bold**.

Method	Narrative grounding		Retrieval precision						Overall quality		
	SVC ↑	SC ↑	Precision ↑	Recall ↑	F1-score ↑	TCS ↑	NL ↑	PA ↑	ESR ↑	ARR ↑	CQ ↑
Comparison with state-of-the-art methods											
Claude [1]	0.112	7.94	16.24	56.44	20.29	18.61%	7.62	7.95	63.84%	12.31%	8.21
Gemini [42]	0.131	8.57	30.88	76.21	41.59	37.41%	8.08	8.47	74.11%	23.85%	8.67
MetaGPT [20]	*N/A*	*N/A*	43.71	72.92	55.73	46.21%	8.23	8.71	85.93%	52.31%	8.85
LAVE [47]	*N/A*	*N/A*	38.22	63.21	44.60	35.37%	8.60	8.43	70.34%	40.28%	8.53
VideoAgent [68]	*N/A*	*N/A*	17.86	40.05	18.84	21.35%	7.55	7.37	47.60%	63.31%	7.34
Ours (CineAgents)	**0.154**	**9.02**	**62.43**	**76.49**	**64.13**	**52.09%**	**8.76**	**9.06**	**92.76%**	**87.23%**	**9.01**
Ablation study											
W/o DCM	0.148	8.89	60.09	74.58	61.17	50.76%	8.71	8.85	89.91%	85.34%	8.94
W/o HNM	*N/A*	*N/A*	49.12	70.19	58.04	46.87%	8.63	8.69	84.60%	72.96%	8.73
W/o INP	*N/A*	*N/A*	54.70	71.67	59.52	49.27%	8.64	8.70	85.07%	78.81%	8.81

orchestrate a multi-agent system. This system interprets a user's textual instruction and synthesizes the final compiled clips from source videos. All experiments are conducted on a server with four NVIDIA A100 GPUs.

5.2 Comparison with State-of-the-Art Methods

Given the absence of prior work on instruction-driven cinematic video compilation, we establish several baselines from related domains. Our comparison experiments include: *(i)* the state-of-the-art Vision Large Language Models (VLLMs) (Claude-3.7-sonnet [1] and Gemini-2.5-pro [42]); *(ii)* the multi-agent framework (MetaGPT [20]); *(iii)* the casual video compilation method (LAVE [47]); and *(iv)* the all-in-one video editing framework (VideoAgent [68]). To ensure a fair comparison, the VLLMs are prompted with a two-stage Chain-of-Thought (CoT) [52] strategy that first retrieves and then reorders shots, mirroring the retrieve-and-rank paradigm. Moreover, both the VLLMs and MetaGPT leverage the script generated by our CineAgents, ensuring the evaluation focuses on compilation ability rather than script quality.

Qualitative comparisons. We provide a visual comparison of the compiled results in Fig. 4. The results highlight distinct failure modes for each baseline. Single VLLMs like Claude [1] and Gemini [42] struggle to maintain long-range narrative context, leading to shot inaccuracy. MetaGPT [20] suffers from poor precision, retrieving numerous irrelevant shots that disrupt the narrative flow. LAVE [47] is designed for casual video editing rather than cinematic video compilation, resulting in chronologically incoherent compilations. VideoAgent [68] as an all-in-one model, lacks the specialized robustness required for the nuanced task of video compilation, making it less reliable. In contrast, our CineAgents produces a compilation that closely mirrors the human ground truth. It successfully identifies the precise semantic events and arranges them in the exact sequence requested, generating predominantly correct segments. This visual ev-

Fig. 4: Qualitative comparison with state-of-the-art methods. Segments in **blue** are correct in both content and temporal position, aligning with the human reference (GT); **green** segments are semantically correct but temporally misplaced; **orange** segments are semantically irrelevant to the instruction; and **dashed regions** represent areas where the model fails to retrieve the corresponding content.

idence compellingly validates our system's superior capability in fine-grained narrative grounding and complex temporal reasoning.

Quantitative comparisons. We present quantitative comparisons in Tab. 1, demonstrating that CineAgents outperforms all state-of-the-art methods across all metrics. Specifically, CineAgent demonstrates a more accurate understanding of cinematic content (SVC), correctly interprets user instructions to retrieve corresponding clips (Precision, Recall, and F1-score), excels at arranging them in a coherent compilation sequence (TCS), and operates with high system reliability and robustness (ESR and ARR).

User study. Given the subjective nature of cinematic video compilation, quantitative metrics and LLM-based evaluations are insufficient for a comprehensive assessment. Therefore, we conduct four user studies to evaluate human preference for our results: *(i)* **Text-Prompt Alignment (TPA):** Participants are asked to select the video that most faithfully represents the content of the user's instruction. *(ii)* **Sequence Alignment Evaluation (SAE):** Participants are tasked with identifying the compilation whose shot sequence most accurately follows the order specified in the instruction. *(iii)* **Narrative Coherence Evaluation (NCE):** This study assesses internal consistency, asking participants to choose the video that presents the most logical and coherent story. *(iv)* **Overall**

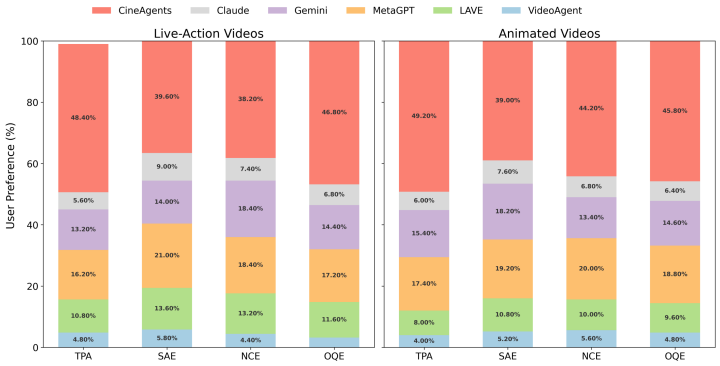

Fig. 5: User study results on Live-Action and Animated videos.

Quality Evaluation (OQE): Participants select the best video, considering all factors including content relevance, narrative flow, and overall viewing experience. For each study, we randomly selected 50 samples from CineBench and recruited 25 volunteers to provide independent evaluations. As shown in Fig. 5, our model achieves the highest preference scores across all four evaluations.

5.3 Ablation study

We discard various modules and create three variants to study the impact of our proposed modules. The scores of the ablation study are shown in Tab. 1.
W/o DCM (Dialogue Character Matching). We discard the dialogue character matching module, which prevents the model from associating dialogue with specific characters. As a result, the model lacks the character-dialogue cues to accurately annotate the script (lower SVC score).
W/o HNM (Hierarchical Narrative Memory). We discard the hierarchical narrative memory module, forcing the model to rely solely on shot-level information. This prevents the system from building high-level understanding of the narrative, causing inaccurate and irrelevant retrieval results (lower F1-score).
W/o INP (Iterative Narrative Planning). We replace the iterative narrative planing with a retrieve-and-rank method. By prioritizing superficial keyword similarity, this variant loses the ability to distinguish between feasible and unfeasible requests. As a result, when challenged with an adversarial set, it fails to refuse instructions (lower ARR score).

6 Discussion

6.1 Different Task Clarification

We clarify the differences between video compilation and related tasks. These tasks operate at different levels:

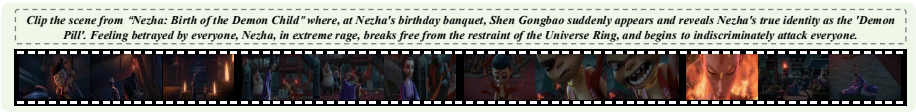

Fig. 6: Visual example of CineAgents compilation performance on animated content. This highlights the model's generalization capability beyond live-action video.

- Video generation and editing are concerned with pixel-level manipulation. They focus on synthesizing visual content or altering the pixels of frames.
- Video understanding involves the passive analysis of video content. Its goal is to interpret semantics for tasks like question answering, without any capability to modify or rearrange the video structure.
- Video compilation operates at the structural level which selects and sequences existing video clips to construct a new narrative. It is worth noting that some works (*e.g.*, EditDuet [40] and HIVE [51]) are not open-source.

6.2 Generalization Across Genres

To quantify this generalization, we conduct four user studies on 10 animated films and 50 instructions. As detailed in Fig. 5, participants evaluate the results using the same four criteria (TPA, SAE, NCE, and OQE), with the results confirming our model's superior versatility across genres.

6.3 Efficiency and Cost Analysis

Transforming a video into a hierarchical narrative memory for a film or TV series takes an average of 1 hour and costs around $14 in API fees via the Gemini-2.5-pro [42]. This comprehensive memory is fully reusable for all subsequent instructions. Consequently, Executing a single instruction takes an average of 6 minutes and costs approximately $0.62 in API fees. This highlights the system's practicality for iterative, real-world creative workflows.

7 Conclusion

In this paper, we present CineBench, the first comprehensive benchmark for instruction-driven cinematic video compilation, featuring high-quality ground truths crafted by professional video editors. To tackle this challenging task, we introduce CineAgents, a multi-agent system that reformulates the traditional retrieve-and-rank approach into a design-and-compose paradigm. Specifically, our system employs script reverse-engineering to construct a hierarchical narrative memory, effectively overcoming contextual collapse. Furthermore, it utilizes an iterative narrative planning process to ensure the logical coherence, resolving temporal fragmentation. Extensive experiments demonstrate that CineAgents significantly outperforms existing methods in narrative grounding, temporal correctness, and alignment with complex user instructions. We believe CineBench

and CineAgents represent a significant step toward generalized compilation systems, empowering creators to efficiently compile cinematic content.

Limitations. While CineAgents demonstrates superior performance in cinematic video compilation, its current implementation relies on off-the-shelf computer vision models for character detection and temporal tracing. In complex cinematic scenes, these models may fail, and such tracking inaccuracies can cascade into errors. We leave the integration of more advanced perceptual modules as future work to further improve the compilation accuracy of CineAgents.

8 Acknowledgements

This work is supported by National Natural Science Foundation of China (Grant No. 62136001), Beijing Major Science and Technology Project (Grant No. Z2511-00008125009), BUPT Innovation and Entrepreneurship Support Program (2025-YC-T029), and Beijing Key Laboratory of Multimodal Data Intelligent Perception and Governance.

References

1. Claude-3.7-sonnet. Accessed February 25, 2025 [Online] https://claude.ai/ 11
2. Insightface. [Online] https://github.com/deepinsight/insightface/ 7, 8
3. Tencent video. [Online] https://v.qq.com/ 1
4. Tiktok short. [Online] https://www.tiktok.com/ 1
5. Youtube short. [Online] https://www.youtube.com/shorts/ 1
6. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023) 4
7. Argaw, D.M., Soldan, M., Pardo, A., Zhao, C., Heilbron, F.C., Chung, J.S., Ghanem, B.: Towards automated movie trailer generation. In: CVPR (2024) 4
8. Bain, M., Huh, J., Han, T., Zisserman, A.: Whisperx: Time-accurate speech transcription of long-form audio. arXiv preprint arXiv:2303.00747 (2023) 8
9. Barua, A., Benharrak, K., Chen, M., Huh, M., Pavel, A.: Lotus: Creating short videos from long videos with abstractive and extractive summarization. arXiv preprint arXiv:2502.07096 (2025) 4
10. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. NeurIPS (2020) 4
11. Cai, S., Yang, C., Zhang, L., Guo, Y., Xiao, J., Yang, Z., Xu, Y., Yang, Z., Yuille, A., Guibas, L., et al.: Mixture of contexts for long video generation. arXiv preprint arXiv:2508.21058 (2025) 4
12. Chen, W., Xu, X., Jia, J., Luo, H., Wang, Y., Wang, F., Jin, R., Sun, X.: Beyond appearance: a semantic controllable self-supervised learning framework for human-centric visual tasks. In: CVPR (2023) 7
13. Cui, C., Sun, T., Lin, M., Gao, T., Zhang, Y., Liu, J., Wang, X., Zhang, Z., Zhou, C., Liu, H., Zhang, Y., Lv, W., Huang, K., Zhang, Y., Zhang, J., Zhang, J., Liu, Y., Yu, D., Ma, Y.: Paddleocr 3.0 technical report (2025), https://arxiv.org/abs/2507.05595 8

14. Du, S., Bai, Y., Ma, J., Liu, M., Li, Y.: Frequency-decoupled learning for joint thin-cloud removal and pansharpening. In: ICASSP (2026) 4
15. Du, S., Zou, Y., Li, J., Liu, M., Li, Y., Shang, C., Shen, Q.: Pansharpening for thin-cloud contaminated remote sensing images: a unified framework and benchmark dataset. In: AAAI (2026) 4
16. Du, S., Zou, Y., Wang, Z., Li, X., Li, Y., Shang, C., Shen, Q.: Unsupervised hyperspectral image super-resolution via self-supervised modality decoupling. IJCV (2026) 4
17. Gan, B., Shu, X., Qiao, R., Wu, H., Chen, K., Li, H., Ren, B.: Collaborative noisy label cleaner: Learning scene-aware trailers for multi-modal highlight detection in movies. In: CVPR (2023) 2, 4
18. Guo, Y., Yang, C., Yang, Z., Ma, Z., Lin, Z., Yang, Z., Lin, D., Jiang, L.: Long context tuning for video generation. In: ICCV (2025) 4
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 8
20. Hong, S., Zhuge, M., Chen, J., Zheng, X., Cheng, Y., Wang, J., Zhang, C., Wang, Z., Yau, S.K.S., Lin, Z., et al.: Metagpt: Meta programming for a multi-agent collaborative framework. In: ICLR (2023) 4, 11
21. Hu, P., Xiao, N., Li, F., Chen, Y., Huang, R.: A reinforcement learning-based automatic video editing method using pre-trained vision-language model. In: ACMMM (2023) 2, 4
22. Huang, K., Huang, Y., Li, Y., Bai, J., Wang, X., Lin, Z., Ning, X., Yu, J., Wang, Y., Liu, X.: Cinescene: Implicit 3d as effective scene representation for cinematic video generation. In: CVPR (2026) 4
23. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., et al.: Vbench: Comprehensive benchmark suite for video generative models. In: CVPR (2024) 6
24. Huber, B., Shin, H.V., Russell, B., Wang, O., Mysore, G.J.: B-script: Transcript-based b-roll video editing with recommendations. In: CHI (2019) 2, 4
25. Jia, W., Lu, Y., Huang, M., Wang, H., Huang, B., Chen, N., Liu, M., Jiang, J., Mao, Z.: Moga: Mixture-of-groups attention for end-to-end long video generation. arXiv preprint arXiv:2510.18692 (2025) 4
26. Koorathota, S., Adelman, P., Cotton, K., Sajda, P.: Editing like humans: a contextual, multimodal framework for automated video editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1701–1709 (2021) 2, 4
27. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. biometrics (1977) 5
28. Li, Y., Li, J., Yu, K., Xiao, X., Liu, D., Wang, T., Tang, R.: Personalize your large vision-language models with in-context prompt tuning. arXiv preprint arXiv:2605.31513 (2026) 4
29. Li, Y., Yang, J., Shen, Z., Han, L., Xu, H., Tang, R.: Catp: Contextually adaptive token pruning for efficient and enhanced multimodal in-context learning. In: AAAI (2026) 4
30. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. In: CVPR (2024) 4
31. Lu, J., Xiao, M., Wang, W., Du, Y., Wu, Z., Hua, C.: Multi-modal and metadata capture model for micro video popularity prediction. arXiv preprint arXiv:2502.17038 (2025) 1

32. Meng, Y., Ouyang, H., Yu, Y., Wang, Q., Wang, W., Cheng, K.L., Wang, H., Ma, S., Li, Y., Chen, C., et al.: Holocine: Holistic generation of cinematic multi-shot long video narratives. In: CVPR (2026) 4
33. Mun, J., Shin, M., Han, G., Lee, S., Ha, S., Lee, J., Kim, E.S.: Boundary-aware self-supervised learning for video scene segmentation. arXiv preprint arXiv:2201.05277 (2022) 9
34. Pardo, A., Caba, F., Alcázar, J.L., Thabet, A.K., Ghanem, B.: Learning to cut by watching movies. In: ICCV (2021) 2
35. Pardo, A., Wang, J.H., Ghanem, B., Sivic, J., Russell, B., Heilbron, F.C.: Generative timelines for instructed visual assembly. arXiv preprint arXiv:2411.12293 (2024) 2, 4
36. Pavel, A., Reyes, G., Bigham, J.P.: Rescribe: Authoring and automatically editing audio descriptions. In: Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology. pp. 747–759 (2020) 3
37. Phung, Q., Mai, L., Heilbron, F.D.C., Liu, F., Huang, J.B., Ham, C.: Cineverse: Consistent keyframe synthesis for cinematic scene composition. arXiv preprint arXiv:2504.19894 (2025) 4
38. Podlesnyy, S.: Towards data-driven automatic video editing. In: Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery: Volume 1. pp. 361–368. Springer (2020) 2, 3
39. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision. In: ICML (2023) 8
40. Sandoval-Castaneda, M., Russell, B., Sivic, J., Shakhnarovich, G., Caba Heilbron, F.: Editduet: A multi-agent system for video non-linear editing. In: Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (2025) 14
41. Soe, T.H., Slavkovik, M.: Ai video editing tools. want and how far is AI from delivering them (2021) 2
42. Team, G., Anil, R., Borgeaud, S., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., Millican, K., et al.: Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023) 4, 6, 9, 10, 11, 14
43. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023) 4
44. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023) 4
45. Truong, A., Berthouzoz, F., Li, W., Agrawala, M.: Quickcut: An interactive tool for editing narrated video. In: Proceedings of the 29th Annual Symposium on User Interface Software and Technology. pp. 497–507 (2016) 2, 3
46. Wagner, R.A., Fischer, M.J.: The string-to-string correction problem. JACM (1974) 6
47. Wang, B., Li, Y., Lv, Z., Xia, H., Xu, Y., Sodhi, R.: Lave: Llm-powered agent assistance and language augmentation for video editing. In: Proceedings of the 29th International Conference on Intelligent User Interfaces (2024) 11
48. Wang, H., Liang, C., Wang, S., Chen, Z., Zhang, B., Xiang, X., Deng, Y., Qian, Y.: Wespeaker: A research and production oriented speaker embedding learning toolkit. In: ICASSP (2023) 8
49. Wang, L., Liu, D., Puri, R., Metaxas, D.N.: Learning trailer moments in full-length movies with co-contrastive attention. In: ECCV (2020) 2, 4

50. Wang, M., Yang, G.W., Hu, S.M., Yau, S.T., Shamir, A., et al.: Write-a-video: computational video montage from themed text. ACM Trans. Graph. (2019) 3
51. Wang, X., Li, X., Wei, Y., Song, X., Song, Y., Xia, X., Zeng, F., Chen, Z., Liu, L., Xu, G., et al.: From long videos to engaging clips: A human-inspired video editing framework with multimodal narrative understanding. arXiv preprint arXiv:2507.02790 (2025) 2, 4, 14
52. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. NeurIPS (2022) 11
53. Weng, S., Zhang, P., Chang, Z., Wang, X., Li, S., Shi, B.: Affective image filter: Reflecting emotions from text to images. In: ICCV (2023) 4
54. Wu, W., Zhu, Z., Shou, M.Z.: Automated movie generation via multi-agent cot planning. arXiv preprint arXiv:2503.07314 (2025) 4
55. Xiao, J., Yang, C., Zhang, L., Cai, S., Zhao, Y., Guo, Y., Wetzstein, G., Agrawala, M., Yuille, A., Jiang, L.: Captain cinema: Towards short movie generation. In: ICLR (2025) 4
56. Xiao, X., Liu, C., Liao, C.T., Zhang, Y., Lan, Q., Wei, Y., Zhao, L., Wang, J., Gu, J., Ye, M., Wang, T., Xu, H.: Staying vigilant: Mitigating visual laziness via counterfactual visual alignment in mllms. In: ECCV (2026) 4
57. Xiao, X., Zhang, Y., Li, X., Wang, T., Wang, X., Wei, Y., Hamm, J., Xu, M.: Visual instance-aware prompt tuning. In: ACMMM (2025) 4
58. Xiong, Y., Heilbron, F.C., Lin, D.: Transcript to video: Efficient clip sequencing from texts. In: ACMMM (2022) 2
59. Xu, Z., Wang, J., Wang, L., Li, Z., Shi, S., Hu, B., Zhang, M.: Filmagent: Automating virtual film production through a multi-agent collaborative framework. In: SIGGRAPH Asia (2024) 4
60. Zhang, L., Cai, S., Li, M., Zeng, C., Lu, B., Rao, A., Han, S., Wetzstein, G., Agrawala, M.: Pretraining frame preservation in autoregressive video memory compression. arXiv preprint arXiv:2512.23851 (2025) 4
61. Zhang, P., Jia, Z., Cai, Z., Weng, S., Li, S., Shi, B.: Recontraster: Making your posters stand out with regional contrast. In: ACL (2026) 4
62. Zhang, P., Jia, Z., Liu, K., Weng, S., Li, S., Shi, B.: Stage: Storyboard-anchored generation for cinematic multi-shot narrative. In: CVPR (2026) 4
63. Zhang, P., Weng, S., Tang, J., Li, S., Shi, B.: Towards deeper emotional reflection: Crafting affective image filters with generative priors. TPAMI (2025) 4
64. Zhang, P., Weng, S., Zhu, C., Tang, B., Jia, Z., Li, S., Shi, B.: Affective image editing: Shaping emotional factors via text descriptions. IJCV (2026) 4
65. Zhang, X., Wei, J., Wang, Y., Tan, J., Li, Y., Zhang, Y., Chen, Z., Zhang, D., Yu, D., Xu, W., et al.: How far are video models from true multimodal reasoning? arXiv preprint arXiv:2604.19193 (2026) 4
66. Zhang, Y., Sun, R., Chen, Y., Pfister, T., Zhang, R., Arik, S.: Chain of agents: Large language models collaborating on long-context tasks. In: NeurIPS (2024) 4
67. Zheng, M., Xu, Y., Huang, H., Ma, X., Liu, Y., Shu, W., Pang, Y., Tang, F., Chen, Q., Yang, H., Sernam, L.: Videogen-of-thought: A collaborative framework for multi-shot video generation. In: NeurIPS NextVid Workshop Oral (2025) 4
68. Zhou, H., Huang, L., Wu, S., Xia, L., Huang, C., et al.: Videoagent: All-in-one agentic framework for video understanding and editing 11
69. Zhu, W., Huang, Y., Xie, X., Liu, W., Deng, J., Zhang, D., Wang, Z., Liu, J.: Autoshot: A short video dataset and state-of-the-art shot boundary detection. In: CVPR (2023) 9