CUTMASTER: TOWARDS AGENTIC HOUR-SCALE VIDEO EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Transforming an hour-scale source video into a short montage conditioned on background music, a user request, and a target duration requires balancing narrative alignment, emotional pacing, and visual quality. Existing agentic editing systems often organize decisions by processing stage or tool, leaving dependencies among these objectives implicit and resulting in suboptimal visual selection. We present **CutMaster**, an objective-factorized, stateful, and verification-driven agentic workflow built around a six-role team. A request-independent material analyst constructs reusable material memory, after which five request-conditioned roles coordinate their decisions through a shared editorial state. Music-aligned slots capture emotional pacing, story anchors provide narrative grounding, and a visually validated candidate space supports beam-search-based composition. Bounded replanning repairs intermediate states when earlier decisions become infeasible. We further introduce MontageX-Bench, a 40-task benchmark spanning four editing intents and three source domains. Compared with five recent state-of-the-art baselines, CutMaster achieves a Quality score of 81.40, outperforming the strongest baseline at 75.55, while providing the best overall balance across narrative, pacing, and visual-quality metrics.

1 INTRODUCTION

Producing a compelling short montage from an hour-scale video requires more than selecting representative moments. Given a source video, an editing prompt, background music, and a target duration, an editor must decide what to retain, how to arrange it, and how cuts should interact with the music. The result should express the requested subject or story, sustain an appropriate emotional progression, and remain visually coherent within a strict time budget. We call this setting *hour-scale source-to-short video montage editing*. Unlike generic video summarization, its output is jointly conditioned on editorial intent and a musical frame.

This is a multi-objective optimization problem involving **narrative coherence**, **emotional pacing**, and **visual quality**—three heterogeneous and mutually constraining metrics that demand immense human effort to balance effectively. However, existing automated video editing systems tend to focus on a single dimension (NarratoAI Contributors, 2026; Yan et al., 2026), or attempt to optimize multiple heterogeneous objectives within a step of iteration (Zhao et al., 2026; Li et al., 2026; Zhou et al., 2026), making it difficult to balance narrative, emotional, and visual factors simultaneously.

To address these challenges, we introduce **CutMaster**, the objective-factorized, stateful, verification-driven **MASTER** workflow shown in Figure 1. The *Material Analyst* builds request-independent Material Memory. The *Arrangement Architect* turns the request, music, and duration into music-aligned Slots; the *Story Editor* grounds pivotal dialogue; and the *Timeline Scout* retrieves and visually validates alternatives. The *Edit Composer* selects video clips under narrative and emotional constraints and finally generates an editing plan, after which the *Renderer* stitches the selected clips together according to the plan. The roles share a versioned *Shared Editorial State*, not free-form conversation. Material Memory and Story Anchors preserve narrative evidence, Slots externalize emotional pacing, and Candidate Space exposes visual alternatives. Validation triggers candidate-constrained repair or bounded replanning when earlier decisions become infeasible.

We also introduce **MontageX-Bench**, a cross-domain and cross-intent suite for this setting. It pairs four Editing Intents (event, character, emotion, narrative) with ten hour-scale videos from sports,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

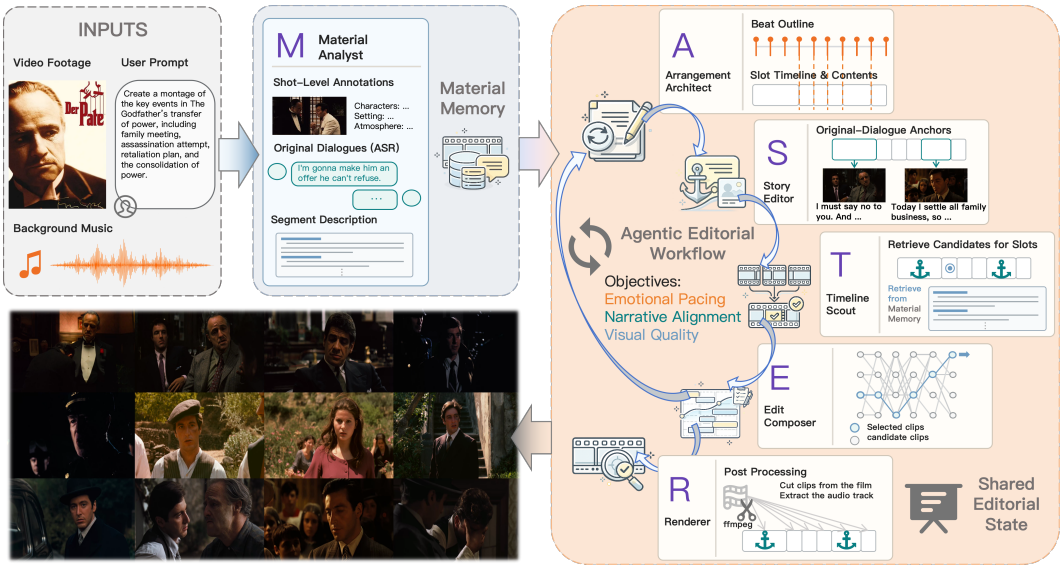


Figure 1: Overview of CutMaster. Given an hour-scale source video, a user prompt, background music, and a target duration, the Material Analyst (M) analyzes the video to construct reusable Material Memory. The remaining five agents then externalize Emotional Pacing, Narrative Alignment, and Visual Quality as structured artifacts in Shared Editorial State before constrained composition. The circular layout summarizes the staged workflow, and the interactions among agents are detailed in Section 3.2. A, S, T, E and R denote Arrangement Architect, Story Editor, Timeline Scout, Edit Composer, and Renderer; ASR denotes automatic speech recognition.

documentaries, and films, yielding 40 tasks. Six general metrics diagnose narrative-intent alignment, music-conditioned pacing, and visual quality. Across five recent baselines, CutMaster achieves the highest *Quality* score of 81.40 versus 75.55 for the runner-up. CutMaster ranks in the top two on almost every single metric, demonstrating broadly competitive performance.

Contributions. **(1) Framework:** We formulate the task as a multi-objective editorial decision problem and propose CutMaster, which externalizes narrative, emotional, and visual factors through a shared editorial state for constrained composition. **(2) Benchmark:** We introduce MontageX-Bench, a 40-task suite spanning four editing intents and three source domains with six general metrics. **(3) State-of-the-art performance:** Against five recent trending video editing systems, CutMaster attains the highest aggregate score and strongest cross-metric editorial balance.

2 Related Work

2.1 Source-to-Short Video Composition

Video summarization selects informative segments, as in SumMe (Gygli et al., 2014) and TVSum (Song et al., 2015). Query-Focused Video Summarization conditions selection on textual concepts, including for hour-scale recordings (Sharghi et al., 2017); long-form pretraining has also used timestamp-aligned pseudo-summaries (Argaw et al., 2024b). Computational editing extends beyond compression. Write-A-Video retrieves and arranges shots for themed text (Wang et al., 2019). QuickCut aligns narration with footage (Truong et al., 2016), while computational dialogue editing selects shots under editing idioms (Leake et al., 2017). Learning to Cut estimates cut plausibility from films (Pardo et al., 2021), and AutoTransition recommends audiovisual transitions (Shen et al., 2022). Domain-specific methods construct movie trailers (Argaw et al., 2024a) or documentary teasers (Xu et al., 2025). These formulations provide retrieval and composition mechanisms, but assume a summary or predefined output genre. Our task instead jointly conditions an hour-scale source on a user request, independent music, and target duration, requiring Narrative Alignment, Emotional Pacing, and Visual Quality in one edit.

2.2 AGENTIC VIDEO EDITING

Optimization-based home-video editing formulated selection and sequencing under duration and soundtrack constraints (Hua et al., 2004), and Audeosynth made the musical structure central to the montage construction (Liao et al., 2015). Emotion-Aware Montage later combined audiovisual affect, story completeness, and shot-level rhythm (Liu et al., 2023). Thus, narrative, affective, and rhythmic criteria predate agentic editors, informing earlier approaches to automated montage.

Agent frameworks supply reusable coordination patterns. AutoGen uses configurable multi-agent conversation (Wu et al., 2023). MetaGPT uses role-based procedural decomposition (Hong et al., 2024), while ChatDev uses communicative specialization (Qian et al., 2024). Multi-agent peer review further lets agents independently formulate solutions, critique their peers, and revise their outputs from structured feedback (Xu et al., 2023). ReAct interleaves reasoning and tool use (Yao et al., 2023). AniMaker introduces search-based composition for long-form animation, using MCTS-Gen to explore multi-candidate clip generation and story-aware selection (Shi et al., 2025). CutMaster instead performs Beam Search within the Candidate Space to identify a superior editing trace.

Editing systems adapt agentic coordination to existing footage differently. EditDuet employs an Editor–Critic interaction (Sandoval-Castaneda et al., 2025). NarratoAI structures commentary around dialogue scripts (NarratoAI Contributors, 2026), while OpenMontage serves as a coding-agent harness (OpenMontage Contributors, 2026). CutClaw combines music-anchored planning, retrieval, and review (Zhao et al., 2026); DIRECT decomposes mashup editing across global, local, and sequence levels (Li et al., 2026). GLANCE couples global planning with iterative local editing (Lin et al., 2026), whereas VideoAgent orchestrates a broad component library (Zhou et al., 2026).

These systems consider overlapping criteria but commonly organize responsibility by stage, temporal scale, or tool. CutMaster instead externalizes narrative evidence, music-conditioned Slots, and visually admissible candidates, then reconciles them through constrained composition and verified repair. This factorization exposes evidence and feasible choices for downstream inspection.

2.3 BENCHMARKS FOR VIDEO EDITING

Editing benchmarks evaluate different units. The Anatomy of Video Editing studies footage organization and assisted assembly (Argaw et al., 2022), while VISIOCITY assesses video summaries (Kaushal et al., 2021). VEU-Bench probes editing knowledge (Li et al., 2025); VEBench evaluates operational reasoning (Deng et al., 2026); and AgenticVBench targets tool-mediated post-production (Cao et al., 2026). These resources do not evaluate a complete source-to-short, music-conditioned montage. Generated-video evaluators such as VBench (Huang et al., 2024), EvalCrafter (Liu et al., 2024), and VideoScore (He et al., 2024) motivate separate, human-validated diagnostics, but address a different task.

System-specific suites also differ. CutClaw evaluates character- and narrative-centered edits (Zhao et al., 2026); DIRECT introduces Mashup-Bench for film mashups (Li et al., 2026); GLANCE introduces MVEBench (Lin et al., 2026); and VideoAgent uses VideoEdit for broad editing workflows (Zhou et al., 2026). MontageX-Bench instead crosses four Editing Intents (event, character, emotion, narrative) with sports, documentary, and film sources. Its six shared general output metrics support paired comparisons across request types and source domains.

3 CUTMASTER

CutMaster realizes objective-factorized editing through the six-role MASTER team. Each role externalizes a distinct class of editorial decisions, while Shared Editorial State preserves their dependencies for coordinated planning, targeted revision, and constrained composition.

3.1 PROBLEM FORMULATION

Define an editing request as $\mathbf{x} = (\mathcal{V}, \mathcal{B}, q, \tau)$, where $\mathcal{V}$ denotes an hour-scale Source Video, $\mathcal{B}$ the Background Music, q the User Prompt, and τ the Target Duration. The output is a Montage $\mathcal{Y} = (c_1, \ldots, c_K)$ assembled from source intervals, respecting both the original source chronology and the output-duration constraint.

We define the editorial objectives by

$$\mathbf{J}(\mathcal{Y}; \mathbf{x}) = [J_{\text{nar}}, J_{\text{emo}}, J_{\text{vis}}] \tag{1}$$

corresponding to Narrative Alignment, Emotional Pacing, and Visual Quality.

3.2 STATEFUL MASTER ARCHITECTURE

CutMaster separates reusable source understanding from request-conditioned editing:

$$\mathcal{M} = \Phi_{\text{M}}(\mathcal{V}),\ \widehat{\mathcal{Y}} = \Psi_{\text{ASTER}}(\mathcal{M}, \mathcal{B}, q, \tau) \tag{2}$$

The Material Analyst constructs reusable Material Memory $\mathcal{M}$, while the remaining five roles collaboratively produce an editing plan. The initials of these six roles together spell MASTER.

For each request, the workflow maintains a versioned Shared Editorial State

$$\mathcal{E} = (\mu, \mathbf{z}, \mathcal{H}, \{\mathcal{C}_i\}_{i=1}^{K}, \mathbf{f}, \mathcal{Y}^{(1:r)}) \tag{3}$$

where μ is the music profile, $\mathbf{z}$ the Slot plan, $\mathcal{H}$ the Story Anchors, $\mathcal{C}_i$ the candidate set for Slot i, $\mathbf{f}$ structured feedback, and $\mathcal{Y}^{(1:r)}$ the composed and revised script versions.

This shared state embodies the objective factorization principle. Material Memory and Story Anchors supply narrative evidence, music-conditioned Slots encode emotional pacing, and the Candidate Space exposes visual alternatives.

Material Analyst. The Material Analyst realizes the source-only mapping $\mathcal{M} = \Phi_{\text{M}}(\mathcal{V})$. It indexes each shot by a source interval $[s_j, e_j)$, $j = 1, \ldots, N$, and groups adjacent shots into segments. The resulting Material Memory $\mathcal{M}$ associates this timeline with shot-level visual evidence, time-aligned dialogue, segment descriptions, and a source-level story summary, bridging local retrieval evidence with long-range narrative context. Because $\mathcal{M}$ is independent of q, $\mathcal{B}$, and τ, it can be reused across editing tasks on the same source.

Arrangement Architect. After music analysis derives the music profile μ from $\mathcal{B}$, the Arrangement Architect maps $(\mathcal{M}, \mu, q, \tau)$ to an ordered Slot plan $\mathbf{z} = (z_1, \ldots, z_K)$. Each Slot z_i is defined by an editorial purpose, intended content, emotional and kinetic targets, required subjects Q_i, and permitted source regions $\mathcal{R}_i$. Boundary-alignment tools produce $0 = b_0 < \cdots < b_K = \tau$, assigning Slot i the output interval $[b_{i-1}, b_i)$ and fixed duration $d_i = b_i - b_{i-1}$.

The Arrangement Architect therefore establishes the Montage's macro structure and Emotional Pacing before exact visual selection. It establishes the narrative and emotional foundation, while delegating source-window selection to downstream roles.

Story Editor. Given $(\mathcal{M}, \mathbf{z}, q)$, the Story Editor produces a possibly empty set of Story Anchors $\mathcal{H} = \{h_\ell\}_{\ell=1}^{N_{\text{anc}}}$. Each anchor $h_\ell = (i_\ell, \gamma_\ell, \delta_\ell)$ binds a Slot index i_ℓ, a source interval $\gamma_\ell \subset \mathcal{V}$, and its original dialogue passage δ_ℓ. It thereby fixes an important event, decision, relationship, or character change to a particular Slot while preserving the source-synchronous picture and dialogue.

Story Anchors are optional rather than mandatory: $\mathcal{H}$ contains only high-confidence narrative evidence. Their purpose is to prevent such evidence from being lost when downstream selection is driven by visual appearance, motion, or music.

Timeline Scout. For each unanchored Slot z_i, the Timeline Scout constructs the candidate set $\mathcal{C}_i = \{\xi_{ij}\}_{j=1}^{n_i}$. Each candidate is a fixed-duration source window $\xi_{ij} = [s_{ij}, s_{ij} + d_i) \subseteq \mathcal{R}_i$, validated for temporal consistency, visible content, required subject, and basic visual usability. For an anchored Slot, the corresponding source interval γ_ℓ is retained as its fixed candidate.

The Timeline Scout does not choose the final clip c_i. Instead, the collection $\{\mathcal{C}_i\}_{i=1}^{K}$ exposes a grounded optimization space for the Edit Composer. This separation prevents an early retrieval decision from prematurely determining the final sequence.

Edit Composer. The Edit Composer selects a complete path $\mathbf{c} = (c_1, \ldots, c_K)$, where $c_i \in \mathcal{C}_i$, while preserving the Slot schedule, fixed Story Anchors, candidate membership, and source chronology. It evaluates both individual candidates and transitions between adjacent candidates.

For each candidate, the unary score $u_i(c)$ is a weighted sum of seven normalized features, where the features set $\mathcal{K}_u$ comprises semantic relevance, visual relevance, subject visibility, emotional fit, kinetic fit, salience, and duration compatibility:

$$u_i(c) = \sum_{k \in \mathcal{K}_u} \alpha_k \cdot f_k(i, c), \tag{4}$$

where f_k indexes the feature and α_k its weight.

For adjacent candidates, the pairwise score $p(c_{i-1}, c_i)$ is a weighted sum of four normalized features, where the features set $\mathcal{K}_p$ comprises visual continuity, emotional continuity, narrative bridging, and energy consistency and g_k indexes the feature and β_k is its weight:

$$p(c_{i-1}, c_i) = \sum_{k \in \mathcal{K}_p} \beta_k \cdot g_k(c_{i-1}, c_i), \tag{5}$$

The complete path is scored by

$$S(\mathbf{c}) = \alpha_u \sum_{i=1}^{K} u_i(c_i) + \alpha_p \sum_{i=2}^{K} p(c_{i-1}, c_i). \tag{6}$$

The Edit Composer employs beam search to optimize visual quality under narrative and emotional constraints established by upstream agents. Retaining only promising partial sequences reduces computational cost and runtime by limiting unary and pairwise score evaluations. Finally, the Edit Composer generates an editing plan that maps each clip in the final video to its source location and time interval in the original video and indicates whether it is an anchor clip.

Renderer. The Renderer acts as a lightweight post-processing pipeline that renders the mashup based on the editing plan generated by the Edit Composer. Specifically, it selects target clips from the source video, extracts the audio track, isolates the human voice, and finally concatenates the video clips, the original audio track, and the background music into a montage.

4 Experiments

4.1 Experimental Setup

Benchmark. MontageX-Bench comprises 40 music-conditioned editing tasks constructed from 10 approximately hour-scale source videos, totaling 18.3 hours of unique footage. Individual sources range from 47.9 to 177.2 minutes and comprise three sports broadcasts, three documentary episodes, and four films. For each source, we construct one request per Editing Intent (event, character, emotion, narrative) yielding a crossed design of 4 intents $\times$ 10 sources, for a total of 40 tasks. We also use 11 background music, each chosen to suit the corresponding task type. Appendix A compares the task composition and evaluation coverage of MontageX-Bench with related benchmarks.

Baselines. We compare CutMaster against five complementary editing systems: CutClaw (Zhao et al., 2026), DIRECT (Li et al., 2026), NarratoAI (NarratoAI Contributors, 2026), OpenMontage (OpenMontage Contributors, 2026), and VideoAgent (Zhou et al., 2026). Every system receives the same task inputs (source video, User Prompt, designated Background Music, 60-second target).

General metrics. We evaluate every output using six general metrics. Beat-Cut Synchronization, which uses PySceneDetect and a VLM to detect cuts, and Energy Correspondence, which correlates frame differences with audio RMS energy, assess music-conditioned pacing on a 0–100 scale. Instruction Following and Narrative Coherence assess intent alignment and narrative quality, respectively, while Visual Quality and Transition Continuity assess visual and sequential quality. These four metrics are rated on a 1–5 Likert scale by Qwen3.7-Plus. See Appendix F for details.

To compute the overall *Quality* score, we first normalize each 1–5 rating s to a 0–100 scale using $\widetilde{s} = (s-1)/4 \times 100$, then take the unweighted mean of all six metrics.

Table 1: General-metrics defined in Section 4.1 are reported as mean ± std. "w/o T&E" denotes the ablation without the Timeline Scout and Edit Composer, and "w/o S" denotes the ablation without the Story Editor. Best values are **bold** and second-best values are underlined.

System	Emotion		Narrative		Visual		Quality
	Beat-Cut Sync.	**Energy Corr.**	**Instruction Following**	**Narrative Coherence**	**Visual Quality**	**Transition Continuity**	
CutClaw	57.33	56.87	3.34 ± 0.04	3.28 ± 0.09	3.95 ± 0.05	3.55 ± 0.00	61.19 ± 0.54
DIRECT	73.93	**57.83**	3.30 ± 0.07	2.97 ± 0.05	3.48 ± 0.14	3.23 ± 0.05	59.32 ± 1.14
NarratoAI	57.24	52.92	4.38 ± 0.04	4.43 ± 0.07	4.63 ± 0.03	$\mathbf{4.29 \pm 0.06}$	75.55 ± 0.37
OpenMontage	56.12	49.50	3.49 ± 0.11	3.48 ± 0.09	4.11 ± 0.01	3.62 ± 0.06	62.19 ± 0.91
VideoAgent	65.61	54.39	3.16 ± 0.08	3.08 ± 0.11	4.15 ± 0.07	3.53 ± 0.04	61.28 ± 0.41
CutMaster	83.18	51.50	$\mathbf{4.63 \pm 0.05}$	$\mathbf{4.49 \pm 0.03}$	$\mathbf{4.77 \pm 0.01}$	4.26 ± 0.01	$\mathbf{81.40 \pm 0.10}$
w/o T&E	83.04	51.72	4.49 ± 0.07	4.33 ± 0.03	4.68 ± 0.04	4.16 ± 0.05	79.37 ± 0.06
w/o S	**86.71**	54.99	4.33 ± 0.07	4.27 ± 0.08	4.61 ± 0.06	4.07 ± 0.04	78.89 ± 0.70

Implementation Details. We implement CutMaster and access all the services of the foundation-model through Alibaba Cloud DashScope (Yang et al., 2025; Bai et al., 2025). Qwen3.7-Max serves as the LLM for editorial planning, dialogue-anchor selection, candidate retrieval, sequence review, and replanning, while Qwen3.7-Plus serves as the VLM for shot-level annotation, candidate verification, and pairwise visual assessment. Speech is transcribed using DashScope Fun-ASR (Gao et al., 2023). The source videos are segmented with PySceneDetect, while the musical structure, energy, and beats are extracted using librosa. We retain three validated candidates for each non-anchor slot and perform sequence selection using beam search with a width of 8. The montages are rendered with FFmpeg at 1920×1080 resolution and 30 fps.

4.2 General Performance

Main results. Table 1 reports the mean and standard deviation of task-averaged scores across three evaluation runs. CutMaster achieves the highest *Quality* score of 81.40, outperforming the strongest baseline, NarratoAI, by 5.85 points. CutMaster ranks among the top two methods on all four VLM-based metrics (Instruction Following, Narrative Coherence, Visual Quality, and Transition Continuity), indicating consistently strong instruction following, narrative coherence, and visual editing quality. Its advantage in Beat-Cut Synchronization further contributes to the overall Quality gain. After the Edit Composer selects the final clips, the Arrangement Architect refines their source windows through small temporal shifts, aligning internal shot boundaries with musical beats in the background music. This refinement improves synchronization beyond the boundaries between selected clips, explicitly accounting for cuts within each clip. The gains are nevertheless not uniform across all metrics: DIRECT achieves the highest Energy Correspondence, while CutMaster combines strong beat alignment with consistently high VLM-based scores.

Fine-grained results. Table 2 aggregates the same 40 tasks into overlapping Source-Domain and Editing-Intent marginals; these are consistency diagnostics, not independent samples. CutMaster has the highest descriptive *Quality* mean in every marginal, ranging from 77.60 on Sports to 83.93 on Film and from 79.90 on Character to 84.35 on Emotion. More results are in Appendix C.

Examining each baseline's design helps explain these patterns. NarratoAI uses a script-based (speech-transcription-driven) editing workflow that prioritizes narrative content selection without explicitly optimizing for emotional pacing or visual quality, which accounts for its higher Narrative Coherence but lower Beat-Cut Synchronization and Energy Correspondence. DIRECT, in contrast, uses optical flow to analyze motion intensity per shot for clip selection, hence its highest Energy Correspondence. These results indicate that CutMaster achieves strong instruction grounding, beat-aware cutting, and visual quality, while exposing Energy Correspondence as the weakness.

We additionally report results from GLM-5.3-Flash in Appendix B, using the same Likert rubric and input videos to examine sensitivity to evaluator choice. Both evaluators yield a consistent three-tier ordering in overall Quality: CutMaster and its ablated variants form the top tier, followed by NarratoAI, with the remaining four baselines forming the third tier.

Table 2: *Quality* means over all 40 tasks, the Sports (12), Documentary (12), and Film (16) subsets, and each ten-task Editing-Intent subset. Best values are **bold** and second-best values are underlined.

		Source Domain			Editing Intent			
System	Overall	Sports	Documentary	Film	Event	Character	Emotion	Narrative
CutClaw	61.19	60.47	56.90	64.94	62.03	62.02	63.58	57.12
DIRECT	59.32	52.95	55.14	67.23	57.70	56.36	70.60	52.63
NarratoAI	75.55	70.94	77.18	77.78	73.25	75.38	75.67	77.88
OpenMontage	62.19	54.92	65.56	65.11	64.55	60.07	64.08	60.04
VideoAgent	61.28	55.45	64.86	62.98	65.04	61.17	62.60	56.33
CutMaster	**81.40**	**77.60**	**81.84**	**83.93**	**80.99**	**79.90**	**84.35**	**80.38**
w/o T&E	79.37	76.01	78.20	82.77	79.54	78.26	80.75	78.94
w/o S	78.89	76.75	75.97	82.70	77.89	77.35	83.37	76.97

4.3 Ablation Study

To further verify the necessity of each agent, we compare CutMaster with two ablation versions: without the Timeline Scout and Edit Composer (**w/o T&E**), without the Story Editor (**w/o S**).

w/o T&E. The Timeline Scout and the Edit Composer primarily optimize visual quality under narrative and emotional constraints. As shown in Table 1, removing these two agents degrades all four VLM-judged metrics. Because clips sharing the same subject or scene have stronger visual continuity, visual quality and narrative quality are positively correlated. Removing the optimization step weakens both narrative and visual quality.

w/o S. The Story Editor is responsible for selecting anchor clips, which are chosen primarily for their narrative quality. Removing the Story Editor causes a clear drop in both narrative quality and visual quality, while the two music-conditioned pacing metrics improve. We attribute this trade-off to the role of anchors: since anchors are selected by narrative criteria, they may not fully match the current emotional requirement. Without these narrative anchors, every clip becomes free to be optimized independently, allowing the system to better satisfy emotional objectives.

4.4 Human Evaluation

Evaluation protocol. We collected 100 valid single-video evaluations from 10 participants. Each participant viewed an anonymized output together with its task instruction and rated instruction following, narrative coherence, visual quality, and transition continuity on five-point Likert scales. For comparability, *Subjective Quality* in this analysis is recomputed using only Instruction Following, Narrative Coherence, Visual Quality, and Transition Continuity on the raw scale of 1–5.

Table 3: Human Likert ratings.

	Narrative		Visual		
System	Instruction Following	Narrative Coherence	Visual Quality	Transition Continuity	Subjective Quality
CutClaw	3.47	3.24	3.71	3.47	3.47
DIRECT	3.75	3.50	3.69	3.81	3.69
NarratoAI	4.50	3.94	4.25	3.81	4.13
OpenMontage	3.42	3.11	3.63	3.32	3.37
VideoAgent	2.94	3.00	3.76	3.35	3.26
CutMaster	**4.67**	**4.47**	**4.60**	**4.13**	**4.47**

Human Likert ratings. Table 3 shows that CutMaster achieves a clear advantage across all four dimensions also assessed by the VLM evaluator, with a *Subjective Quality* score of 4.47 compared to 4.13 for the strongest baseline, NarratoAI. The greatest improvement occurs in narrative coherence, suggesting that the observed advantage extends beyond instruction satisfaction to the organization of the montage. The higher Visual Quality and Transition Continuity scores further indicate favorable human judgments of visual presentation and adjacent-clip transitions. These findings complement the overall performance trend observed in the automated evaluation, providing additional support from human viewers.

Agreement with VLM assessment. To examine whether automated scores reflect human judgments at the output level, we matched 100 valid human Likert evaluations with the corresponding

Table 4: Score correlations between human and VLM evaluations and among human annotators.

(a) Human–VLM score correlations.

Metric	Spearman ρ	Pearson r	Kendall τ_b
Instruction Following	0.754	0.756	0.655
Narrative Coherence	0.589	0.592	0.499
Visual Quality	0.464	0.446	0.402
Transition Continuity	0.460	0.426	0.410
Subjective Quality	0.664	0.683	0.549

(b) Inter-annotator score correlations.

Metric	Spearman ρ	Pearson r	Kendall τ_b
Instruction Following	0.778	0.769	0.750
Narrative Coherence	0.706	0.715	0.687
Visual Quality	0.689	0.690	0.659
Transition Continuity	0.690	0.709	0.677
Subjective Quality	0.843	0.819	0.795

system–task VLM scores averaged over three evaluation rounds. As shown in Table 4a, correlations are positive across all four dimensions, with the strongest alignment in Instruction Following. Subjective Quality achieves Spearman's $\rho = 0.664$, Pearson's $r = 0.683$, and Kendall's $\tau_b = 0.549$, supporting VLM evaluation as a complementary assessment.

Inter-annotator agreement. To assess the consistency of human judgments, annotators were first trained using the evaluation rubrics in Table 10 and then independently rated the same set of 12 videos. We computed Spearman's ρ, Pearson's r, and Kendall's τ_b to measure inter-annotator agreement. As shown in Table 4b, the four individual dimensions exhibit moderate-to-strong correlations. Subjective Quality, computed as the equal-weight mean of the four dimension scores, shows particularly strong agreement, with Spearman's $\rho = 0.843$, Pearson's $r = 0.819$, and Kendall's $\tau_b = 0.795$, indicating a high level of consistency among human annotators.

Pairwise human preferences. Figure 2 shows human comparisons of overall video quality. CutMaster achieves a preference rate greater than 50% against all baselines: 83.3% against NarratoAI, 100% against OpenMontage and CutClaw, 80.0% against VideoAgent, and 92.9% against DIRECT. Thus, its observed advantage is not limited to a single competing system or to the absolute rating scale used in the Likert evaluation. The preference over NarratoAI, the strongest baseline, further supports the consistency of CutMaster's advantage across both evaluation protocals. Together, the two evaluation protocols provide converging evidence that CutMaster's higher dimension-wise ratings are accompanied by a preference for its outputs as complete viewing experiences.

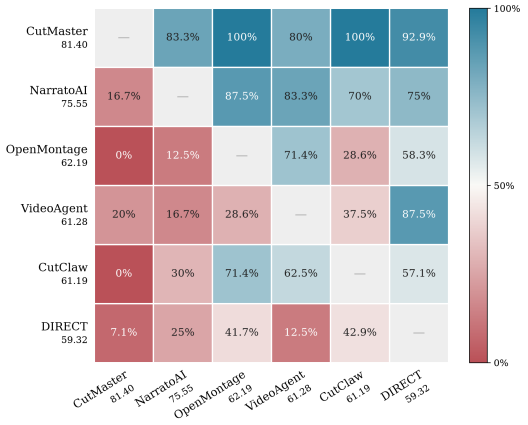


Figure 2: Pairwise human preferences.

4.5 Workflow Case Study: Objective-Factorized Editing

We inspect Film–Event `task_025`, compressing the key events in *The Godfather* into a 60-second montage. This trace yields a feasible slot plan through a single arrangement architect pass, resulting in an overall pipeline structure that is easy to visualize. The planning trace of `task_025` is shown in Figure 3. More case studies comparing CutMaster with baselines are provided in the Appendix D.

Planning and narrative grounding. Conditioned on Material Memory and the background music, the Arrangement Architect groups the final Slots into five narrative spans and produces 17 music-aligned Slots as shown in the A-part. In the S-part, the Story Editor fixes two dialogue anchors, Slot 5 and Slot 16, where the first begins with Don's refusal of the narcotics proposal and continues as an L-cut until 35.727 seconds, while the second grounds Michael's final settlement of family business. Therefore, emotional pacing and narrative evidence are externalized prior to actual visual selection.

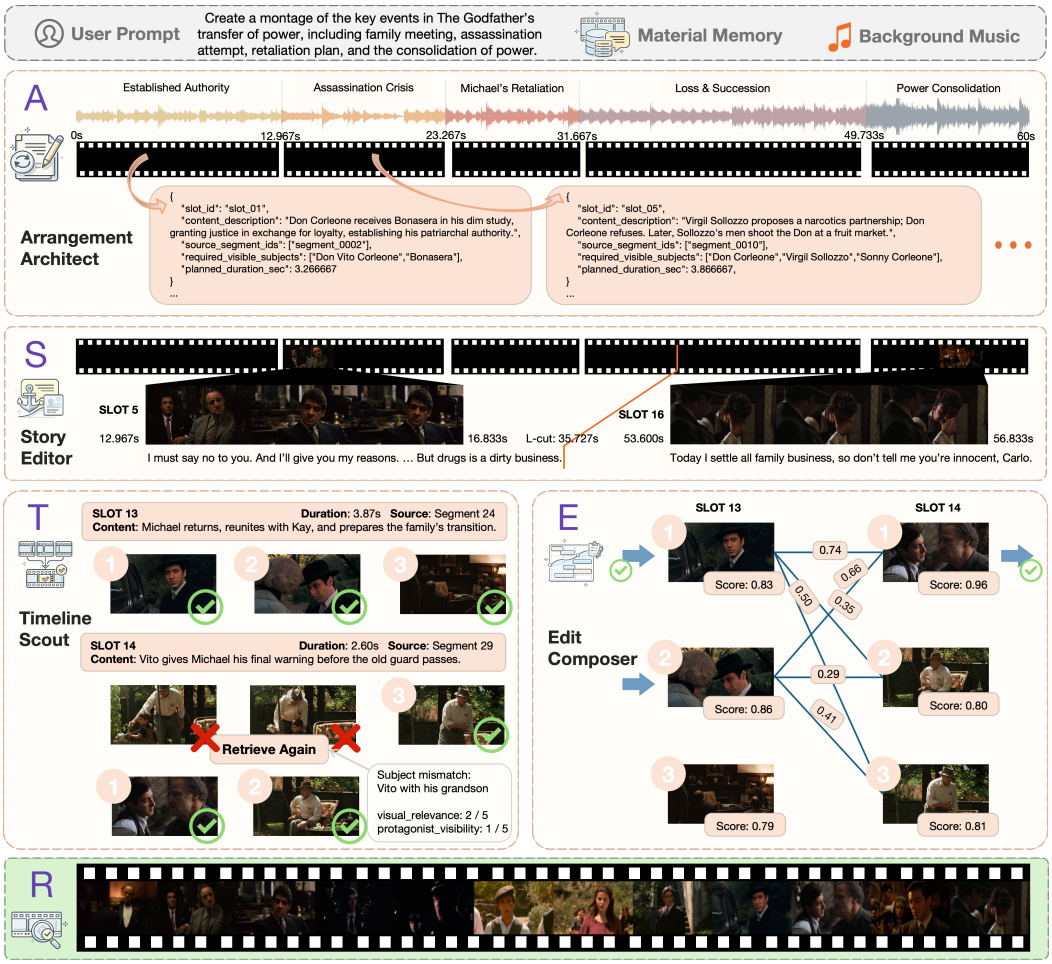


Figure 3: CutMaster workflow for Film–Event `task_025`. Details are described in Section 4.5.

Verified retrieval and constrained composition. The Timeline Scout selects three candidates for each of the 15 unanchored Slots, totaling 45 clips in Candidate Space. The T-Part illustrates the retrieval process of candidates for Slot 13 and Slot 14. Slot 13 secures three candidates in its initial retrieval pass, while the two first-round candidates for Slot 14 are both rejected due to the presence of Vito's grandson. A subsequent supplementary retrieval restores the candidate pool back to three.

The Edit Composer searches the full 17-layer Candidate Space with a beam width of eight, under fixed-anchor and candidate-membership constraints, optimizing visual quality subject to narrative and emotional objectives. In the E-part, Slot-13 candidate 2 has the highest unary score, but the retained path uses candidate 1 followed by Slot-14 candidate 1, due to its superior pairwise score.

5 Conclusion

We presented **CutMaster**, an objective-factorized, stateful, and verification-driven agentic workflow that transforms hour-scale source videos into short montages conditioned on background music, a user request, and a target duration. Through Shared Editorial State, MASTER externalizes narrative evidence, emotional pacing, and visual candidates, then combines them through Beam Search. We also introduced MontageX-Bench, a 40-task benchmark spanning four editing intents and three source domains. Compared with five recent state-of-the-art baselines, CutMaster achieves a Quality score of 81.40, outperforming the strongest baseline at 75.55. These results demonstrate the effectiveness of explicitly factorizing editing objectives and coordinating them through a persistent shared state, and suggest a promising direction for reliable multi-objective editing of hour-scale video.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the related-work survey, method implementation and benchmark tasks. We have not used generative AI tools for designing the CutMaster structure, analyzing the experimental results, or interpreting the data. Additionally, we used generative AI tools for code editing and debugging, literature and technical information retrieval, language polishing, and LaTeX table formatting. We have reviewed all AI-assisted work: the authors tested the implementation, and verified citations and factual claims. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

CutMaster edits existing footage rather than generating new visual content, but selecting, reordering, or shortening clips can still remove context and misrepresent events, people, or speech. Its outputs therefore require human review before publication, particularly for factual or sensitive material. The benchmark includes sports, documentary, and film footage, which may contain identifiable individuals, copyrighted content, or disturbing scenes. Use and redistribution of source videos, music, and derived edits should respect the applicable permissions and licenses; inclusion in the benchmark does not imply permission to redistribute these assets.

REPRODUCIBILITY STATEMENT

We document the task formulation and stateful editing workflow in Sections 3.1 and 3.2, and provide agent prompts and runtime request assembly in Appendix G. Section 4.1 describes the benchmark, evaluation protocol, and implementation settings; Appendix F specifies the VLM scoring rubric, automatic metrics, and Quality aggregation. Baseline configurations are detailed in Appendix E.

REFERENCES

Dawit Mureja Argaw, Fabian Caba, 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*, pp. 201–218, 2022. URL https://www.ecva.net/papers/eccv_2022/papers_ECCV/html/4736_ECCV_2022_paper.php.

Dawit Mureja Argaw, Mattia Soldan, Alejandro Pardo, Chen Zhao, Fabian Caba Heilbron, Joon Son Chung, and Bernard Ghanem. Towards automated movie trailer generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7445–7454, 2024a. doi: 10.1109/CVPR52733.2024.00711. URL https://openaccess.thecvf.com/content/CVPR2024/html/Argaw_Towards_Automated_Movie_Trailer_Generation_CVPR_2024_paper.html.

Dawit Mureja Argaw, Seunghyun Yoon, Fabian Caba Heilbron, Hanieh Deilamsalehy, Trung Bui, Zhaowen Wang, Franck Dernoncourt, and Joon Son Chung. Scaling up video summarization pretraining with large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8332–8341, 2024b. URL https://openaccess.thecvf.com/content/CVPR2024/html/Argaw_Scaling_Up_Video_Summarization_Pretraining_with_Large_Language_Models_CVPR_2024_paper.html.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Zongheng Cao, Yi Zheng, Rui Song, and Xinyu Hu. AgenticVBench: Can AI agents complete real-world post-production tasks?, 2026. URL https://arxiv.org/abs/2605.27705.

Brandon Castellano. PySceneDetect: Video cut detection and analysis tool. Computer software, May 2026. URL https://github.com/Breakthrough/PySceneDetect/releases/tag/v0.7-release.

Andong Deng, Dawei Du, Zhenfang Chen, Wen Zhong, Fan Chen, Guang Chen, Chia-Wen Kuo, Longyin Wen, Chen Chen, and Sijie Zhu. VEBench: Benchmarking large multimodal models for real-world video editing, 2026. URL https://arxiv.org/abs/2605.03276.

Zhifu Gao, Zerui Li, Jiaming Wang, Haoneng Luo, Xian Shi, Mengzhe Chen, Yabin Li, Lingyun Zuo, Zhihao Du, Zhangyu Xiao, and Shiliang Zhang. FunASR: A fundamental end-to-end speech recognition toolkit. *arXiv preprint arXiv:2305.11013*, 2023.

Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *European Conference on Computer Vision*, pp. 505–520, 2014. doi: 10.1007/978-3-319-10584-0_33. URL https://doi.org/10.1007/978-3-319-10584-0_33.

Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyan Jiang, Aaran Arulraj, Kai Wang, Quy Duc Do, Yuansheng Ni, Bohan Lyu, Yaswanth Narsupalli, Rongqi Fan, Zhiheng Lyu, Bill Yuchen Lin, and Wenhu Chen. VideoScore: Building automatic metrics to simulate fine-grained human feedback for video generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2105–2123. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.127. URL https://aclanthology.org/2024.emnlp-main.127/.

Sirui Hong, Mingchen Zhuge, Jiaqi Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for A multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VtmBAGCN7o.

Xian-Sheng Hua, Lie Lu, and Hong-Jiang Zhang. Optimization-based automated home video editing system. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(5):572–583, 2004. doi: 10.1109/TCSVT.2004.826750. URL https://doi.org/10.1109/TCSVT.2004.826750.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818, June 2024.

Vishal Kaushal, Suraj Kothawade, Anshul Tomar, Rishabh Iyer, and Ganesh Ramakrishnan. How good is a video summary? a new benchmarking dataset and evaluation framework towards realistic video summarization, 2021. URL https://arxiv.org/abs/2101.10514.

Mackenzie Leake, Abe Davis, Anh Truong, and Maneesh Agrawala. Computational video editing for dialogue-driven scenes. *ACM Trans. Graph.*, 36(4), jul 2017. ISSN 0730-0301. doi: 10.1145/3072959.3073653. URL https://doi.org/10.1145/3072959.3073653.

Bozheng Li, Yongliang Wu, Yi Lu, Jiashuo Yu, Licheng Tang, Jiawang Cao, Wenqing Zhu, Yuyang Sun, Jay Wu, and Wenbo Zhu. VEU-Bench: Towards comprehensive understanding of video editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13671–13680, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Li_VEU-Bench_Towards_Comprehensive_Understanding_of_Video_Editing_CVPR_2025_paper.html.

Ke Li, Maoliang Li, Jialiang Chen, Jiayu Chen, Zihao Zheng, Shaoqi Wang, and Xiang Chen. DIRECT: Video mashup creation via hierarchical multi-agent planning and intent-guided editing, 2026. URL https://arxiv.org/abs/2604.04875.

Zicheng Liao, Yizhou Yu, Bingchen Gong, and Lechao Cheng. AudeoSynth: Music-driven video montage. *ACM Transactions on Graphics*, 34(4):68:1–68:10, 2015. doi: 10.1145/2766966. URL https://doi.org/10.1145/2766966.

Zihao Lin, Haibo Wang, Zhiyang Xu, Siyao Dai, Huanjie Dong, Xiaohan Wang, Yolo Y. Tang, Yixin Wang, Qifan Wang, and Lifu Huang. GLANCE: A global–local coordination multi-agent framework for music-grounded non-linear video editing, 2026. URL https://arxiv.org/abs/2604.05076.

Wu-Qin Liu, Min-Xuan Lin, Hai-Bin Huang, Chong-Yang Ma, Yu Song, Wei-Ming Dong, and Chang-Sheng Xu. Emotion-aware music driven movie montage. *Journal of Computer Science and Technology*, 38(3):540–553, 2023. doi: 10.1007/s11390-023-3064-6. URL https://doi.org/10.1007/s11390-023-3064-6.

Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. EvalCrafter: Benchmarking and evaluating large video generation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22139–22149, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Liu_EvalCrafter_Benchmarking_and_Evaluating_Large_Video_Generation_Models_CVPR_2024_paper.html.

Brian McFee, Colin Raffel, Dawen Liang, Daniel P. W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th Python in Science Conference*, pp. 18–24. SciPy, 2015. doi: 10.25080/Majora-7b98e3ed-003. URL https://doi.org/10.25080/Majora-7b98e3ed-003.

Brian McFee et al. librosa/librosa: 0.11.0, March 2025. URL https://doi.org/10.5281/zenodo.15006942. Software release.

NarratoAI Contributors. NarratoAI: Ai-powered video commentary and editing. Software repository, 2026. URL https://github.com/linyqh/NarratoAI. version 0.8.4; accessed August 1, 2026.

OpenMontage Contributors. OpenMontage: An open-source agentic video production system. Software repository, 2026. URL https://github.com/calesthio/OpenMontage. accessed August 1, 2026.

Alejandro Pardo, Fabian Caba, Juan Léon Alcázar, Ali K. Thabet, and Bernard Ghanem. Learning to cut by watching movies. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6858–6868, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Pardo_Learning_To_Cut_by_Watching_Movies_ICCV_2021_paper.html.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. ChatDev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15174–15186. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.810. URL https://aclanthology.org/2024.acl-long.810/.

Marcelo Sandoval-Castaneda, Bryan Russell, Josef Sivic, Gregory Shakhnarovich, and Fabian Caba Heilbron. EditDuet: A multi-agent system for video non-linear editing. In *ACM SIGGRAPH 2025 Conference Papers*, pp. 1–11, 2025. doi: 10.1145/3721238.3730761. URL https://doi.org/10.1145/3721238.3730761.

Aidean Sharghi, Jacob S. Laurel, and Boqing Gong. Query-focused video summarization: Dataset, evaluation, and a memory network based approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2127–2136, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Sharghi_Query-Focused_Video_Summarization_CVPR_2017_paper.html.

Yaojie Shen, Libo Zhang, Kai Xu, and Xiaojie Jin. AutoTransition: Learning to recommend video transition effects. In *Computer Vision–ECCV 2022*, volume 13698 of *Lecture Notes in Computer Science*, pp. 285–300. Springer, 2022. doi: 10.1007/978-3-031-19839-7_17. URL https://doi.org/10.1007/978-3-031-19839-7_17.

Haoyuan Shi, Yunxin Li, Xinyu Chen, Longyue Wang, Baotian Hu, and Min Zhang. Animaker: Multi-agent animated storytelling with mcts-driven clip generation. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, SA Conference Papers '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400721373. doi: 10.1145/3757377.3764009. URL https://doi.org/10.1145/3757377.3764009.

Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. TVSum: Summarizing web videos using titles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5179–5187, 2015. doi: 10.1109/CVPR.2015.7299154. URL https://openaccess.thecvf.com/content_cvpr_2015/html/Song_TVSum_Summarizing_Web_2015_CVPR_paper.html.

Anh Truong, Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. 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. doi: 10.1145/2984511.2984569. URL https://doi.org/10.1145/2984511.2984569.

Miao Wang, Guo-Wei Yang, Shi-Min Hu, Shing-Tung Yau, and Ariel Shamir. Write-a-video: Computational video montage from themed text. *ACM Transactions on Graphics*, 38(6):177:1–177:13, 2019. doi: 10.1145/3355089.3356520. URL https://doi.org/10.1145/3355089.3356520.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Weihan Xu, Paul Pu Liang, Haven Kim, Julian McAuley, Taylor Berg-Kirkpatrick, and Hao-Wen Dong. TeaserGen: Generating teasers for long documentaries. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=G1n50BMqzm.

Zhenran Xu, Senbao Shi, Baotian Hu, Jindi Yu, Dongfang Li, Min Zhang, and Yuxiang Wu. Towards reasoning in large language models via multi-agent peer review collaboration, 2023. URL https://arxiv.org/abs/2311.08152.

Lecheng Yan, Yichong Zhang, Xiantao Xu, Jianze Lin, Ben Pan, Xiaoyu Zheng, Jiawei Qian, Anqi Wu, Jiahui Geng, Ruizhe Li, Fengyu Cai, Jingcheng Niu, Raymond Li, Wenxi Li, and Chenyang Lyu. Crayotter: Traceable multi-agent workflows for long-form video editing, 2026. URL https://arxiv.org/abs/2606.07636.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Shifang Zhao, Yihan Hu, Ying Shan, Yunchao Wei, and Xiaodong Cun. CutClaw: Agentic hours-long video editing via music synchronization, 2026. URL https://arxiv.org/abs/2603.29664.

Hengji Zhou, Lingxuan Huang, Jian Wang, Bing Zhou, Si Wu, Lianghao Xia, and Chao Huang. VideoAgent: All-in-one framework for video understanding and editing, 2026. URL https://arxiv.org/abs/2606.23327.

APPENDIX

A BENCHMARK DESIGN AND COMPARISON

MontageX-Bench combines a crossed task design with evaluation of narrative, music-conditioned pacing, and visual quality. Each of its ten source videos is paired with all four editing intents, allowing different editing objectives to be examined on the same source material.

Evaluation coverage. Table 5 compares the evaluation dimensions covered by MontageX-Bench and the benchmarks used in CutClaw (Zhao et al., 2026), DIRECT (Li et al., 2026), and Crayotter (Yan et al., 2026). MontageX-Bench includes all six dimensions in this comparison. The table summarizes coverage at the dimension level.

Table 5: Evaluation-dimension coverage across related editing benchmarks. A check mark indicates coverage of the corresponding dimension; a dash indicates that it is not covered.

Dimension	CutClaw	DIRECT	Crayotter	MontageX-Bench
Beat-Cut Sync.	✓	✓	–	✓
Energy Corr.	–	✓	–	✓
Instruction Following	✓	✓	✓	✓
Narrative Coherence	–	–	✓	✓
Visual Quality	✓	–	✓	✓
Transition Continuity	–	✓	✓	✓

Task composition. Table 6 summarizes the task counts, source-video counts, and domains of the compared benchmarks. MontageX-Bench contains 40 tasks from ten source videos, with a balanced task allocation to each editing intent.

Table 6: Task composition of related editing benchmarks. A dash indicates that it is not reported.

Benchmark	Tasks	Videos	Domains
CutClaw (Zhao et al., 2026)	20	10	Film, daily life
DIRECT (Li et al., 2026)	40	38	Film
Crayotter (Yan et al., 2026)	23	–	Pet, campus, travel, scenery, food
MontageX-Bench	40	10	Sports, documentary, film

B Evaluation with an Alternative VLM Judge

To examine sensitivity to the choice of VLM evaluator, we additionally evaluate the generated videos using GLM-5.3-Flash. We retain the same evaluation rubric for Instruction Following, Narrative Coherence, Visual Quality, and Transition Continuity. Quality is computed as the unweighted mean of the six metrics after mapping the four VLM ratings to the 0–100 scale. In addition, OpenMontage is averaged over 39 tasks because one video was rejected by the evaluator.

As shown in Table 7, CutMaster achieves the highest Quality score of 82.61, compared with 74.78 for the strongest baseline, NarratoAI. It also achieves the highest Instruction Following, Narrative Coherence, and Transition Continuity scores and ties for the highest Visual Quality score. These results are consistent with the Qwen-based evaluation in placing CutMaster ahead of the baselines in overall quality, although individual scores depend on the evaluator.

For the ablations, removing the Story Editor reduces Narrative Coherence from 4.58 to 4.35 and Quality from 82.61 to 80.33. Removing the Timeline Scout and Edit Composer reduces Transition Continuity from 4.53 to 4.40 and Quality to 81.25, while Visual Quality remains unchanged.

C Quality Breakdown across Domains and Intents

Table 8 reports all 12 combinations of the three Source Domains and four Editing Intents. Each Sports and Documentary cell averages three tasks, whereas each Film cell averages four.

Table 7: General-metric performance using GLM-5.3-Flash as the VLM evaluator. Metric definitions and scoring procedures follow those in Table 1.

System	Emotion		Narrative		Visual		Quality
	Beat-Cut Sync.	**Energy Corr.**	**Instruction Following**	**Narrative Coherence**	**Visual Quality**	**Transition Continuity**	
CutClaw	57.33	56.87	3.25	3.53	4.30	3.85	64.57
DIRECT	73.98	**57.83**	3.20	3.18	3.68	3.53	61.97
NarratoAI	57.24	52.92	4.03	4.30	**4.78**	4.43	74.78
OpenMontage†	56.12	49.50	3.51	3.64	4.15	3.85	64.06
VideoAgent	65.61	54.39	3.25	3.38	4.48	3.83	65.58
CutMaster	83.18	51.50	**4.55**	**4.58**	**4.78**	**4.53**	**82.61**
w/o T&E	83.04	51.72	4.40	4.53	**4.78**	4.40	81.25
w/o S	**86.71**	54.99	4.23	4.35	4.63	4.40	80.33

Table 8: Quality means for every Source-Domain–Editing-Intent cell. Best values within each cell are bold and second-best values are underlined. Human judgments are excluded from Quality.

	Sports				Documentary				Film			
System	Event	Character	Emotion	Narrative	Event	Character	Emotion	Narrative	Event	Character	Emotion	Narrative
CutClaw	61.40	58.32	65.92	56.23	63.90	55.65	56.39	51.68	61.10	69.57	67.21	61.88
DIRECT	49.16	51.08	65.60	45.97	58.88	44.20	75.87	41.61	63.21	69.44	70.40	65.89
NarratoAI	70.44	70.90	71.38	71.03	78.86	70.02	77.71	**82.10**	71.16	**82.77**	77.35	79.85
OpenMontage	53.59	46.56	61.70	57.83	71.78	66.54	64.40	59.51	67.36	65.34	65.62	62.11
VideoAgent	68.47	55.82	52.45	45.05	67.28	64.49	73.31	54.36	60.78	62.69	62.17	66.27
CutMaster	**75.30**	**79.50**	**83.37**	**72.25**	**80.46**	**77.68**	**87.45**	81.75	**85.64**	81.88	**82.77**	**85.44**

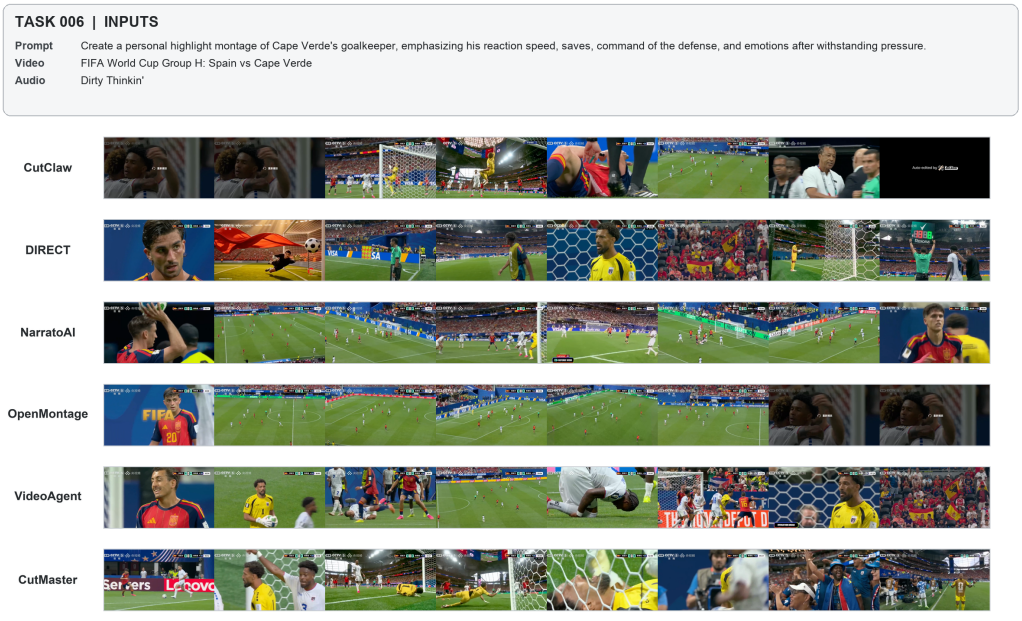


Figure 4: Six-system output comparison for Sports–Character `task_006`. Each row presents eight frames sampled at the midpoints of eight equal temporal bins over that method's complete rendered output.

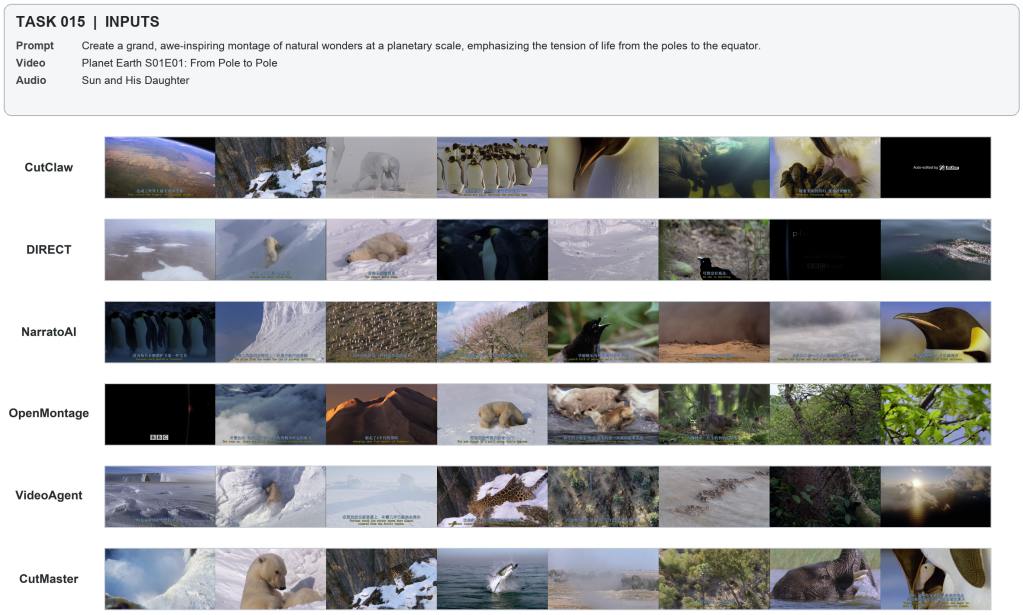


Figure 5: Six-system output comparison for Documentary–Emotion `task_015`. Each row presents eight frames sampled at the midpoints of eight equal temporal bins over that method's complete rendered output.

D Case Studies

D.1 Sports–Character Output Comparison

We inspect `Sport-Character task_006`, which asks CutMaster to extract the highlight reel of the Cape Verdean goalkeeper Vozinha. The results of CutMaster and the baselines are shown in Figure 4. These results reveal two recurring failure modes at the baselines. First, several outputs exhibit *subject drift*: NarratoAI and OpenMontage are dominated by generic wide play or opposing players, while DIRECT and VideoAgent show the requested goalkeeper only intermittently among shots of opponents, spectators, officials, or substitution activity. Second, relevant content is often weakly organized or repetitive; for example, CutClaw mixes a goalmouth action with repeated reaction frames and unrelated sideline content.

In contrast, CutMaster keeps a much stronger focus on the protagonist while still covering different parts of the request. Its sampled sequence moves from goalmouth pressure and interaction with defenders, through multiple save and recovery moments, to a close view of the goalkeeper's strain and broader post-match emotion. The progression also varies shot scale—from wide action to medium interaction and close-up reaction—without reducing the montage to generic match footage. Among the sampled outputs, CutMaster therefore presents the clearest personal-highlight arc, jointly improving protagonist fidelity, requested-action coverage, and visible emotional progression.

D.2 Documentary–Emotion Output Comparison

We inspect Documentary–Emotion `task_015`, which asks CutMaster to create a grand, awe-inspiring montage of natural wonders at a planetary scale, emphasizing the tension of life from the poles to the equator. The results of CutMaster and the baselines are shown in Figure 5.

All systems retrieve high-quality documentary footage and several broadly cover the requested range of biomes. Samples can reveal differences in editorial emphasis. DIRECT allocates most of its output to polar imagery and includes a credit frame; OpenMontage opens with a black title frame and relies heavily on landscape shots; CutClaw covers several ecosystems but repeats polar subjects; VideoAgent delivers the strongest baseline progression across biomes.

CutMaster uses every sampled position for relevant content and combines environmental force, predation, coordinated pursuit, aquatic adaptation, and renewal across polar, temperate, oceanic, and tropical settings. Its samples also vary scale, from a monumental waterfall and animal herds to close interactions involving a polar-bear cub and a penguin chick. The comparison therefore illustrates denser joint coverage of planetary scale and visible life tension, rather than superiority in the quality of the shared source footage.

E Baseline Adapters and Reproducibility

Comparison protocol. We enforce task-input and evaluator parity rather than identical model backbones, compute budgets, or rendering policies. Every system receives the same source video, designated Background Music, User Prompt, and 60-second Target Duration. We evaluate the original output produced by each system without additional trimming, padding, or duration adjustment.

CutClaw. We use native `local_run` planning and rendering in `film` mode, with the native hook and ending enabled. The designated background music (BGM) is used throughout; source dialogue is retained in the hook, montage-body source audio is muted, and the native ending is appended. Native automatic speech recognition (ASR) and dialogue analysis are used.

DIRECT. The execution path follows `main_preprocess`, Screenwriter, Director, Editor, and FFmpeg muxing. The designated BGM is trimmed to 60 seconds, and the native timeline follows its length. The task interface exposes no shot-duration control used by the adapter. The output uses BGM only, and the adapter adds neither an ASR stage nor a SubRip subtitle (SRT) stage.

NarratoAI. The adapter runs Bailian ASR, generates a short-mix script, applies clip caps, and renders in original-audio mode (`OST=1`). Fifteen clips are requested; each selected clip is capped at 4 seconds, and the accumulated sequence is capped at 60 seconds. Bailian ASR supplies an internal SRT file, and subtitle burn-in is disabled.

OpenMontage. We use a coding-agent harness with a constrained hybrid/source-footage-led profile, FFmpeg rendering, no support assets, and no human-review pauses. A 60-second target is supplied, and the native render is retained. The 4-second shot target is not included in the sanitized agent-visible payload. The recorded edit decisions use the designated BGM only and mute source audio. No systematic ASR use is observed.

VideoAgent. We use the fixed rhythm-edit chain consisting of VideoPreloader, RhythmDetector, RhythmContentGenerator, VideoSearcher, and VideoEditor. The designated BGM is trimmed to 60 seconds, and the native output is retained. The evaluated chain exposes no consumed shot-duration control. The output uses BGM only; local Whisper is used during source indexing.

CutMaster. We run the full MASTER workflow through `CutMaster(config)` and `run(request)`. Target Duration directly defines a frame-exact output timeline, while the 4-second shot target is used in music-aligned Slot planning. Ordinary source audio is muted; selected source-synchronous dialogue anchors are mixed over the BGM with music ducking. Bailian ASR is cached per source and used internally.

Versions and environments. Table 9 reports baseline configurations. Runs span heterogeneous hardware because several released systems require their original environments. Compared with DIRECT and VideoAgent, CutMaster has good cross-platform compatibility, and can run on macOS, Windows and Linux.

F Automatic Evaluator Details

VLM rubrics. The anchors in Table 10 are embedded directly in the benchmark's judge prompt.

Table 9: Versions, model configurations, and execution environments of baselines.

System	Evaluated version and model configuration	Platform
CutClaw	Commit `f02b2e43`; Qwen3.7-Max for language, Qwen3.7-Plus for vision, and Qwen3.5-Omni-Plus for speech understanding.	macOS arm64.
DIRECT	Commit `64abd68a` for all 40 tasks.	Ubuntu 22.04; NVIDIA RTX 4090 (24 GB).
NarratoAI	`0.8.4-adapted`, commit `f7be8aaf`; Qwen3.7-Max for text planning and Bailian Fun-ASR.	macOS arm64.
OpenMontage	`agent-harness-ffmpeg`, commit `e09afaa2`; Qwen3.7-Plus coding agent.	macOS arm64.
VideoAgent	`rhythm_video_edit`, commit `bb941c8a`; Qwen3.7-Plus visual model, Qwen3.7-Max language model, and local Whisper-large-v3-turbo.	Ubuntu 22.04; NVIDIA RTX 4090 (24 GB).
CutMaster	Qwen3.7-Plus for vision, Qwen3.7-Max for text, and Bailian Fun-ASR.	macOS arm64.

Table 10: 1–5 Likert rubric embedded in the general VLM-judge prompt: Instruction Following, Visual Quality, Transition Continuity, and Narrative Coherence.

Metric	1	2	3	4	5
Instruction Following	The central request is absent or contradicted.	The video is thematically relevant, but most core requirements lack direct evidence.	The request is meaningfully fulfilled, but a core requirement is missing or partial, or several secondary requirements are weak.	Every core requirement is directly supported; only minor secondary limitations remain.	All explicit requirements are fully and unambiguously supported with appropriate emphasis throughout.
Visual Quality	Pervasive editing-related technical problems prevent comfortable viewing or comprehension.	Repeated substantial defects impair several portions of the edit.	The edit is usable overall, but one substantial or several minor defects noticeably interrupt viewing.	The visual presentation is technically controlled throughout, with only isolated minor defects.	The visual presentation is consistently precise throughout, with no observed editing-induced defects.
Transition Continuity	Most shot boundaries disrupt comprehension or perceptual orientation.	Repeated disruptive boundaries make substantial portions difficult to follow.	The edit is understandable overall, but several weak boundaries or an important premature interruption remains.	Almost all boundaries are purposeful and readable, with only isolated minor weaknesses.	All observed boundaries preserve intended meaning and orientation with consistently precise timing.
Narrative Coherence	No discernible organizing relationship exists among the selected shots.	A theme is recognizable, but the sequence is largely disconnected or repetitive.	Basic organization is readable, but development, emphasis, or closure is substantially weak.	The sequence maintains clear progression, with only minor structural gaps or redundancy.	The edit forms a fully developed and economical structure with purposeful progression and a convincing task-appropriate ending.

Beat-Cut Synchronization. Beat-Cut Synchronization operates only on the final rendered video and does not access a system's edit-decision list. PySceneDetect 0.7's full-frame AdaptiveDetector (Castellano, 2026) first proposes candidate discontinuities using an adaptive threshold of 2.0, a minimum content value of 15.0, and a minimum scene length of five frames. For every candidate at time c, the frame immediately before the candidate and the frame at the candidate are resized to a width of 640 pixels and sent to the common VLM for a strict binary shot-boundary decision. The classifier accepts a discontinuous change in shot, viewpoint, framing, scale, scene, or moment in time, while rejecting within-shot subject or camera motion, blur, exposure changes, and overlay updates.

Beat times $\mathcal{B}$ are extracted from the montage's soundtrack with `librosa` 0.11.0 (McFee et al., 2015; 2025): mono audio is sampled at 22.05 kHz, onset strength is computed with a hop length of 512, and `beat_track` is run with `trim=False`. Let $\mathcal{C}$ denote the VLM-confirmed cuts and $d(c, \mathcal{B}) = \min_{b \in \mathcal{B}} |c - b|$. The benchmark computes

$$\text{Beat-Cut Synchronization} = \frac{100}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \exp\left(-\frac{d(c, \mathcal{B})}{\tau_{\text{beat}}}\right). \tag{7}$$

For all systems, the timing-tolerance constant is fixed to $\tau_{\text{beat}} = 0.196\,\text{s}$; consequently, a 100-ms cut–beat offset contributes approximately $100 \exp(-0.10/0.196) \approx 60.0$ points.

Energy Correspondence. Energy Correspondence is computed from the final rendered output, including its actual mixed soundtrack. Video frames are decoded as RGB images at 2 frames per second and resized to a width of 320 pixels. Visual energy m_t is the normalized mean absolute

```
Shared Assembly Runtime structure common to all tasks
Every task conditionally receives a whitelisted snapshot of maintained workflow state:
source-analysis artifacts for the Material Analyst, and relevant Shared Editorial State
artifacts for the five request-conditioned roles.  This is followed by the task-local body,
authoritative JSON Schema, and schema-derived template.  Retry diagnostics are appended
only after a failed attempt.
Runtime Context (template).
[optional prefix]
# Maintained workflow context
  {serialized artifacts selected for the current task}
{task instruction and task-local runtime context}
<response_contract>
  {authoritative JSON Schema for the current task}
</response_contract>
<response_template>
  {schema-derived JSON response template}
</response_template>
[retry only]
<previous_attempt_failures>
  {serialized diagnostics accumulated across failed attempts}
</previous_attempt_failures>
```

Figure 6: Runtime request assembly shared by all CutMaster's agents. Braced placeholders are instantiated from task parameters or maintained workflow context.

RGB difference between adjacent sampled frames. Audio is decoded as 16-kHz mono pulse-code modulation and divided into 0.5-second windows, whose root-mean-square values form the audio-energy sequence a_t. The two sequences are truncated to their common length and scored by

$$\text{Energy Correspondence} = 50\,(1 + \rho(\{m_t\}, \{a_t\}))\,, \tag{8}$$

clipped to $[0, 100]$, where ρ is Pearson correlation. The score is set to zero when fewer than three paired samples are available or either sequence has negligible variance. Because both Beat-Cut Synchronization and Energy Correspondence use the final mix rather than an isolated BGM track, each system's native source-audio policy remains part of the evaluated output.

G CutMaster Prompts and Runtime Request Assembly

CutMaster registers 11 model-mediated tasks; the two Slot-planning branches yield 12 task-instruction bodies. Figures 6– 11 reproduce the common request assembly and every fixed System Prompt verbatim, summarize each task instruction, and retain every task-local XML-like delimiter. Each request combines a fixed System Prompt, a task instruction, runtime-instantiated context, an authoritative JSON Schema, and a schema-derived response template. Braced text denotes values populated from task parameters or the maintained workflow context; for the five request-conditioned roles, it exposes selected artifacts from Shared Editorial State. XML-like tags are literal prompt delimiters. At inference time, the model receives concrete values rather than the displayed placeholders. The common request wrapper is shown separately in Figure 6.

M1 ASR Dialogue-Fragment Reconstruction

System Prompt (verbatim). You reconstruct complete spoken sentences from adjacent ASR subtitle fragments. Never rewrite dialogue and only return supplied candidate labels. Cue IDs are anchors, not candidate labels. Return strict JSON only.

Task Instruction (summary). Select supplied candidate labels whose adjacent same-speaker cues jointly form one complete sentence. Merge only ASR-split continuations; keep independent sentences separate, and never rewrite dialogue, timestamps, or return cue IDs.

Runtime Context (template).

<candidates>
{serialized ASR-fragment candidate groups}
</candidates>

M2 Full-Transcript Dialogue Segmentation

System Prompt (verbatim). You divide the complete source transcript into contiguous spoken-content Segments. Every dialogue line must be assigned exactly once and returned in source order. Segment boundaries must be compatible with supplied Shot memberships. Return strict JSON only.

Task Instruction (summary). Partition the complete transcript into source-ordered conversations or monologues without crossing major speaker, topic, location, narrative, or time breaks. Assign every dialogue ID exactly once; boundaries must fall between non-overlapping Shot memberships.

Runtime Context (template).

No additional task-local context block is appended. The complete transcript, source metadata, and Shot boundaries are selected into the maintained workflow context by the common request wrapper.

M3 Shot-Level Annotation

System Prompt (verbatim). You annotate exactly one source-video Shot from five uniformly sampled frames. The complete transcript is global narrative context, never visual evidence. Describe only visible people, actions, locations, lighting, colors, objects, and camera properties. Every categorical value must exactly match the response contract. Return strict JSON only.

Task Instruction (summary). Annotate one Shot from five frames, treating pixels rather than the transcript as visual evidence. Use generic identities when uncertain, apply the identity scale and dialogue-conditioned content type, and choose exactly one allowed camera scale, angle, and movement.

Runtime Context (template).

Shot has dialogue: {true or false}

{serialized Segment record containing segment identifier,
dialogue flag, speech mode, and aligned dialogue context}

<shot>
{serialized Shot record containing Shot identifier, timestamp,
aligned dialogue, and five sampled-frame timestamps}
</shot>

M4 Segment Summarization

System Prompt (verbatim). You create a concise, factual Segment-level summary from structured dialogue and Shot annotations. Return strict JSON only.

Task Instruction (summary). Summarize one Segment in one to three chronological sentences, integrating dialogue and Shot annotations. State grounded participants, actions, interactions, and narrative significance; never treat provider-rejected Shots as visual evidence.

Runtime Context (template).

{serialized Segment with dialogue and Shot-level annotations}

M5 Source-Level Story Summarization

System Prompt (verbatim). You synthesize reusable story understanding from Segment-level video descriptions and the complete ASR transcript. Ground every statement in the supplied structure and return strict JSON only.

Task Instruction (summary). Synthesize all Segment summaries and the chronological transcript into a grounded synopsis, source-ordered beats with exact Segment IDs, character arcs, themes, and ending. Do not reproduce the transcript, invent events, or prescribe edits.

Runtime Context (template).

No additional task-local context block is appended. Source metadata, Segment descriptions, and the chronological transcript are selected into the maintained workflow context by the common request wrapper.

Figure 7: Prompts for the Material Analyst. Its five model-mediated tasks build dialogue structure, Shot annotations, Segment descriptions, and the reusable story representation in Material Memory.

A Arrangement Architect: Slot Planning and Targeted Repair

System Prompt (verbatim). You are CutMaster's Arrangement Architect. Arrange the output timeline, pacing, emotional progression, and narrative structure, but do not select source timestamps. Return strict JSON only.

Full-Planning Instruction (summary).
Create request- and music-conditioned continuous-source Slots, choosing the Slot count from the narrative and available material. Specify grounded content, narrative role, emotion, motion, duration, required subjects, and permitted Segments. Meet the total, average, and per-Slot duration limits; keep Segment assignments unique and source-ordered, reserve chronological room for later Slots, defer beat alignment to the deterministic audio stage, and treat original dialogue as start-aligned L-cuts.

Runtime Context (template).
Target duration: {requested output duration in seconds}
Preferred Slot duration: {target clip duration in seconds}
Maximum Slot duration: {1.5 times the preferred Slot duration}
Retry note: {optional diagnostics from an earlier attempt}

Targeted-Repair Instruction (summary).
Repair only the failed Slots while preserving their IDs, output timing, exact desired and planned beat-aligned durations, and every unaffected Slot. Respect legal chronological intervals, allowed Segments, forbidden assignments, and structured rejection diagnostics. Redesign unsupported visible content, subjects, or source evidence; the workflow subsequently refreshes any Story Anchor invalidated by a changed Segment assignment.

Runtime Context (template).
<existing_slot_plan>
{serialized current Slot plan}
</existing_slot_plan>
<target_slot_constraints>
{serialized constraints for Slots requiring repair}
</target_slot_constraints>
<rejection_feedback>
{serialized retrieval or validation failure diagnostics}
</rejection_feedback>
Retry note: {optional diagnostics from an earlier attempt}

Figure 8: Prompts for the Arrangement Architect. Full Slot planning and targeted repair share one fixed System Prompt and instantiate separate task instructions and runtime contexts.

S Story Editor: Original-Dialogue Anchoring

System Prompt (verbatim). You are CutMaster's Story Editor. Select a few meaningful, coherent original-speech passages that directly anchor the user's requested story. Each passage may be one complete line or multiple consecutive lines. Every passage starts with its corresponding source picture as a start-aligned L-cut. Return strict JSON only.

Task Instruction (summary). Select a small set of request-relevant Story Anchors from original dialogue. Each Anchor uses consecutive items from one allowed Segment, satisfies source- and output-duration bounds, begins with synchronized source picture as a start-aligned L-cut, remains non-overlapping and source- chronological, and receives importance and coherence ratings of four or five.

Runtime Context (template).
Maximum Story Anchors: {task-specific upper bound}
Minimum Anchor duration: {duration in seconds}
<slots>
{serialized Slot plan}
</slots>
<video_summary>
{serialized reusable source-level story summary}
</video_summary>
<selected_source_segments>
{serialized source Segments eligible for dialogue selection}
</selected_source_segments>
<dialogue_constraints_by_slot>
{serialized per-Slot dialogue and duration constraints}
</dialogue_constraints_by_slot>

Figure 9: Prompt for the Story Editor, which selects request-relevant original dialogue as temporally grounded Story Anchors.

T1 Timeline Scout: Source-Candidate Retrieval

System Prompt (verbatim). You are CutMaster's Timeline Scout. Scout real source-video passages from a structured VideoDescription whose Segment timeline and Shot annotations are authoritative. Choose precise fixed-duration windows within that timeline; never invent timestamps, visuals, or dialogue. Return strict JSON only.

Task Instruction (summary). Return exactly the missing fixed-duration windows for every Slot. Each millisecond-precision window must lie within supplied Segments, overlap real Shots, and avoid other candidates for the same Slot as well as all confirmed or excluded ranges. Preserve confirmed candidates, use structured evidence without invention, and score semantic relevance, emotional intensity, and salience.

Runtime Context (template).

Candidates per Slot: {required candidate count}
<slots>
{serialized Slots requiring candidate retrieval}
</slots>
<confirmed_candidates>
{serialized candidates already validated and retained}
</confirmed_candidates>
<excluded_ranges>
{serialized source ranges that cannot be retrieved again}
</excluded_ranges>
<available_source_segments_by_slot>
{serialized eligible source Segments for each Slot}
</available_source_segments_by_slot>

T2 Timeline Scout: Candidate Visual Verification

System Prompt (verbatim). As CutMaster's Timeline Scout, inspect source-video contact sheets and resolve the requested subject from the maintained request and source title, then judge whether that subject and action are actually visible. Identity must come from pixels, never dialogue or assumptions. Return strict JSON only.

Task Instruction (summary). Inspect labeled contact sheets for the requested subject and Slot content. Pixels are primary evidence; annotations and aligned dialogue may only corroborate them. Score subject visibility and visual relevance on 1--5 scales, assign visibility 1 to a prominent confirmed different person, and use 3 or higher only when the combined visual evidence plausibly supports the requested identity.

Runtime Context (template).

<candidates>
{serialized candidates aligned with the attached contact sheets}
</candidates>

Figure 10: Prompts for the Timeline Scout. Source-candidate retrieval operates on structured Material Memory, while visual verification grounds the retrieved windows in attached contact sheets.

E Edit Composer: Adjacent Hard-Cut Compatibility Scoring

System Prompt (verbatim). As CutMaster's Edit Composer, evaluate whether two real source-video fragments form a coherent direct hard cut. Judge the visible tail of the first against the visible head of the second. Do not assume Slot text is visible fact and do not rely on transition effects. Return strict JSON only.

Task Instruction (summary). Score every supplied candidate pair across one adjacent-Slot boundary using contact sheets from the previous candidate's tail and the current candidate's head. Evaluate a direct hard cut only, treat text as context rather than proof, and return pixel-grounded visual-continuity, emotional-continuity, and narrative-bridge scores with supporting evidence.

Runtime Context (template).

Candidate pairs: {number of adjacent-Slot combinations}
<previous_slot>
{serialized previous-Slot specification}
</previous_slot>
<current_slot>
{serialized current-Slot specification}
</current_slot>
<candidate_pairs>
{serialized candidate-ID pairs and visible-content summaries}
</candidate_pairs>

Figure 11: Prompt for the Edit Composer, which scores adjacent candidate pairs for direct hard-cut compatibility before constrained sequence selection.