SHAPE THE NARRATIVE: MESSAGE-DRIVEN NARRATIVE VIDEO EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Video editing is fundamentally message-driven: even from the same source footage, the selected shots change depending on the narrative derived from messages the editor wishes to convey. Benchmarks for a closely related task, video summarization, reduce editorial intent to a single, message-agnostic notion of saliency and thus do not account for this diversity. For evaluating message-driven video editing, we present **MEDit-Bench**, a dataset and benchmark, which pairs long-form videos with multiple editing messages and multiple professionally produced edits per message, demonstrating that different messages yield substantially different edits from the same source. We define an automatic evaluation protocol based on temporal alignment metrics, and find that an LLM-as-a-judge preference, a natural proxy for narrative quality, is unreliable for this task due to severe position bias. We additionally annotate each message with ambiguity and contextfulness scores, and show that both dimensions negatively correlate with model performance, establishing video editing difficulty as a meaningful stratification factor. Experiments with state-of-the-art MLLMs and reinforcement fine-tuned baselines show that while strong models approach human temporal alignment at lenient thresholds, all models fall behind humans at stricter criteria. A human perceptual study further shows that professional human edits remain preferred over model outputs on every evaluated aspect. The project page is available at `https://anonymous.4open.science/w/editnet/`.

1 INTRODUCTION

Online videos have become a primary medium for information, education, and entertainment. As a result, there is a growing demand for automating video editing, which transforms raw footage into watchable content. Video summarization has been extensively studied as a step toward automation, supported by datasets such as SumMe (Gygli et al., 2014), TVSum (Song et al., 2015), and VSUMM (de Avila et al., 2011), tailored to evaluate models' ability to find salient segments. Representative methods include supervised importance prediction (Wang et al., 2020; Apostolidis et al., 2021; Son et al., 2024a), vision-language approaches leveraging CLIP-like models (Narasimhan et al., 2021; He et al., 2023; Li et al., 2023b;a), and LLM-based frameworks (Lee et al., 2025).

However, video editing is not merely about selecting salient segments. Unlike saliency-based summarization, where the goal is to identify important moments, professional editing is inherently purposeful: it requires deep comprehension of long-form video content and the creative judgment to compose shots into a sequence that tells a story. The resulting edit is therefore not a compression of the footage but a *deliberate creative act*: it demands both an understanding of what is happening in the video and a deliberate choice of how to present it to communicate something meaningful to viewers.

The key ingredient that drives this process is the *message*, a natural language statement of what the editor wishes to communicate to viewers. To construct a narrative from this message, the editor must determine not just *which* moments to include but how they should be composed into a coherent story: a visually salient moment may be entirely irrelevant to a given message, while an apparently unremarkable shot may be indispensable for the narrative setup. Crucially, the same source footage can yield radically different edits under different messages, and even for the same message, multiple valid interpretations exist that reflect individual editorial style. This diversity (across messages and

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.

across editors) is absent from existing video summarization benchmarks, which primarily focus on summaries of visual highlights (Gygli et al., 2014; Song et al., 2015) and thus conflate the richness of editorial intent into a single notion of saliency. This subjectivity makes message-driven editing difficult not only to solve but also to evaluate: valid edits can convey the same message through different shots and narrative structures, so existing quantitative metrics do not directly measure communicative success. We therefore design MEDit-Bench primarily as a benchmark for evaluating models under multiple valid human references, rather than assuming a single objectively correct edit. As an initial step beyond temporal overlap, we test the widely used LLM-as-a-judge paradigm (Zheng et al., 2023; Gu et al., 2024; Zhang et al., 2023); the severe position bias and order inconsistency we observe show that reliable automatic evaluation of narrative editing remains open.

In this work, we focus on *message-driven video editing* and make the following contributions. **(1)** We formulate the task and introduce **MEDit-Bench** (**ME**ssage-**D**riven video ed**it**ing **Bench**mark), an evaluation benchmark pairing 60 long-form videos with 540 professional edits across multiple messages and editors. This design explicitly represents the task's subjectivity and supports evaluation without imposing a single correct edit. **(2)** We establish an evaluation protocol centered on temporal alignment metrics (R@θ, F1@θ, mIoU), stratified by message quality (ambiguity and contextfulness) and complemented by a 3,240-judgment human perceptual study. As a first automatic probe of narrative quality, we adopt the commonly used LLM-as-a-judge paradigm, but uncover severe position bias and order inconsistency, cautioning against treating it as a substitute for human evaluation. **(3)** We find that even strong MLLMs (Gemini-3-pro, GPT-5, Qwen3-VL) fall well short of the human performance at strict cut-precision thresholds, a gap corroborated by our human perceptual study in which professional edits remain consistently preferred over model outputs.

2 Related Work

2.1 Video Editing and Summarization Datasets

Video editing and summarization datasets have been developed for a variety of objectives and modalities. Classic video summarization benchmarks include SumMe (Gygli et al., 2014), TVSum (Song et al., 2015), VSUMM (de Avila et al., 2011), and the large-scale Mr. HiSum (Sul et al., 2023). They score clip-level importance based on visual saliency or viewer interest. More recently, multimodal datasets have emerged that pair video with text descriptions for semantic summarization (Lin et al., 2024; Qiu et al., 2024), broadening the scope of textual signals in evaluation. Query-focused (Sharghi et al., 2017) and script-driven (Mylonas et al., 2025) summarization instead take textual queries, so a single video has a different target summary for a different query. However, they require the summary to contain the *exact* visual cues specified in the query, whereas we emphasize the storyline depicted by the messages, permitting segments providing only contextual information.

Datasets focusing on movie analysis and editing technique (Huang et al., 2020; Pardo et al., 2022; Shou et al., 2021) have advanced shot boundary detection and structural analysis of professional film editing. However, these datasets address individual editing components in isolation and do not model the end-to-end editing process conditioned on a communicative message. In contrast, MEDit-Bench is centered on the message that an editor wishes to convey. Each source video is paired with multiple natural-language messages, and professional editors independently produce edits for each, enabling evaluation of message-conditioned diversity, which is a dimension absent from existing benchmarks.

2.2 Video Summarization and Editing Methods

Video summarization has been the predominant target of automated video editing. Early work used CNNs and RNNs to assign importance scores to frames or shots (Fajtl et al., 2019; Zhu et al., 2020; Wang et al., 2020; Apostolidis et al., 2021; Ghauri et al., 2021; Jiang & Mu, 2022; Son et al., 2024a). More recent methods leverage vision-language models, such as CLIP, to incorporate textual cues and improve multimodal alignment (Narasimhan et al., 2021; He et al., 2023; Li et al., 2023b;a). An LLM's chain-of-thought reasoning has been used for frame-wise importance scoring, further pushing the state of the art (Lee et al., 2025). Although these methods achieve strong performance on importance-based benchmarks, they optimize for a fixed notion of "important" content and do not support message-conditioned narrative construction.

Video editing has been studied from different perspectives. Methods for aligning narration or dialogue scripts with footage temporally (Truong et al., 2016; Leake et al., 2017), text-based clip retrieval for montage generation (Wang et al., 2019; Xiong et al., 2022), and systems that transfer professional editing styles or generate full timelines from natural-language instructions (Frey et al., 2021; Pardo et al., 2024) each address specific aspects of the editing pipeline. Unlike these existing works, our task requires the model to holistically interpret a long video and autonomously construct a narrative aligned with the given message.

2.3 Video Temporal Localization with MLLMs

Video temporal localization involves a suite of related tasks such as temporal grounding (Lei et al., 2021), action localization and video summarization (Gygli et al., 2014; Song et al., 2015; Pang et al., 2023; Zhou et al., 2018), which invariably require the model to find the segments that best correspond to the input query. Recently, MLLMs have been leveraged to tackle such tasks, with most of them focusing on developing MLLM-friendly representations of time, including time-aware positional embedding (Ren et al., 2024; Guo et al., 2025), semantic-based embedding matching (Liu et al., 2024; Pang et al., 2026), and interleaved textual timestamps (Meinardus et al., 2024), where models are mostly trained with supervised fine-tuning. As the time representation starts to converge to the interleaved timestamps (Bai et al., 2025; Zhang et al., 2026a), recent works have begun to study the optimization strategy, *i.e.*, the shift from supervised fine-tuning to reinforcement learning (Wang et al., 2025b; Zhang et al., 2026a).

Message-driven video editing requires the same temporal localization ability but is harder in two ways. First, rather than retrieving segments that explicitly match the query, it must infer those that are contextually relevant to the message, *i.e.*, that set up, explain, or advance it, not merely depict its surface content. Second, while existing multi-segment tasks output segments that are either near-duplicates of one semantic concept (*e.g.*, action localization) or mutually independent (*e.g.*, dense captioning (Krishna et al., 2017; Yang et al., 2023), video summarization), ours are contextually connected into a coherent storyline. For instance, an edit for "The revenge of the white rabbit" may include segments establishing why the rabbit seeks revenge, not only the revenge scenes.

3 MEDit-Bench

We design MEDit-Bench to evaluate a model's ability to generate video narratives from a high-level message, which requires deep comprehension of the video content and the creation of a convincing storyline that conveys the message.

3.1 Data Collection

MEDit-Bench is constructed in three stages: (i) source video selection, (ii) editing message creation, and (iii) professional editing and annotation.

Source videos and messages. We select 60 videos of length 7–18 minutes from LongVideoBench (Wu et al., 2024), VideoMME (Fu et al., 2025), and Blender Studio open movies (Blender Studio, 2026). For each source video, we write three distinct editing messages that state what the editor should communicate to viewers, resulting in 180 video-message pairs. Messages are designed to be (i) visually communicable without relying on audio or dialogue, (ii) diverse in intent so that they encourage substantially different edits from the same footage, and (iii) varied in difficulty, spanning clear and concrete messages to ambiguous or context-dependent ones. We annotate every message for ambiguity and contextfulness, and report the protocol and its effect on model performance in Sec. A.8.

Professional annotation. During the annotation process, each video-message pair is given to three professional video editors, who independently produce a collection of temporal cuts, each of which is represented by the start and end timestamps of a selected clip, resulting in 540 edited videos in total; further details on dataset construction, statistics, and representative examples are provided in the supplementary material. Editors are instructed to (i) trim the source to a 1–3 minute summary using only cut operations, (ii) reorder shots freely, (iii) ignore the audio and base their decisions solely on the visual content, and (iv) provide a rationale for every cut to discourage careless cuts.

Table 1: Comparison with representative video summarization and compilation datasets. Footage counts source videos or cinematic works; n/r denotes values not reported. †Derived from the standard 15% length budget. ‡The 91.4% of the VideoXum test split for which script annotations are available, with durations measured over the 3,121 of them we could obtain. §Oracle summaries over the 180 query-video pairs of the standard split.

Dataset	Text input	Footage	Avg. footage dur.	Avg. summary dur.
SumMe (Gygli et al., 2014)	None	25	2m40s	24s†
VSUMM (de Avila et al., 2011)	None	50	1m30s	14s†
TVSum (Song et al., 2015)	None	50	4m12s	38s†
VideoXum (Lin et al., 2024)	None	4,000	2m4s	20.5s
QFVS (Sharghi et al., 2017)	Object	4	4h14m	5m4s§
S-VideoXum (Mylonas et al., 2025)‡	Script	3,419	2m5s	20.6s
CineBench (Zhang et al., 2026b)	Instruction	> 70	n/r	n/r
MEDit-Bench	Message	60	**13m44s**	**2m12s**

We preserve the submitted shot order for sequence-aware evaluation and release both the ordering and rationales to support future research.

3.2 Comparison with Existing Datasets

Table 1 compares MEDit-Bench with representative summarization datasets along three axes, namely the presence of a communicative message, the length of the source footage, and the length of the target summary.

Vision-only benchmarks such as SumMe (Gygli et al., 2014), VSUMM (de Avila et al., 2011) and TVSum (Song et al., 2015) do not provide text-conditioned summaries. VideoXum (Lin et al., 2024) provides an additional video-to-text summarization benchmark. S-VideoXum (Mylonas et al., 2025) generates video summaries based on a script which is, however, generated from each existing summary so that the script describes the very content the summary already contains, reversing the direction of the annotation pipeline with respect to ours. Query-focused summarization (Sharghi et al., 2017) conditions summarization on concrete object labels (*e.g.*, {HAT, PHONE}) without considering high-level narratives. CineBench (Zhang et al., 2026b) is closest to our setting in conditioning the compilation of long-form footage on open-ended text, but its instructions can explicitly specify target content, temporal structure, and post-production operations. MEDit-Bench starts one level earlier: its message states what the edit should communicate, leaving the model to construct a narrative, choose visual evidence, and determine the shot order. More comparisons are in Sec. B.

Table 2: Agreement measured by F1. [1]Different editors, same video and message. [2]Different messages, same video and editor. *Quoted from Otani et al. (2019).

	Avg	Max
SumMe*	0.31	0.54
TVSum*	0.54	0.78
MEDit-Bench (ours)		
Cross-Ann.[1]	0.42	0.55
Cross-Msg.[2]	**0.15**	**0.27**

We analyze agreement across editors (cross-annotator) and across messages (cross-message), following the existing protocol (Otani et al., 2019). Table 2 shows that cross-annotator F1-Avg is 0.42 (higher than SumMe at 0.31 and lower than TVSum at 0.54), reflecting editors' different styles. Cross-message F1-Avg is only 0.15, indicating that messages substantially diversify the edits even for the same editor.

3.3 Task Definition

Given a video V and a message M in natural language text, we propose to predict the edited video as a sequence of temporal cuts

$$\hat{\mathbf{C}} = (c_1, c_2, \ldots, c_K), \tag{1}$$

where $c_k = (s_k, e_k)$, with s_k and e_k ($0 \leq s_k < e_k \leq T$, with T the video duration) denoting the start and end timestamps of the k-th cut. Unlike conventional summarization, which targets a single general synopsis per video, our task is explicitly conditioned on M, which is arbitrarily specified by the user, enabling $\hat{\mathbf{C}}$ to convey diverse narratives.

3.4 Evaluation Metrics

We propose to evaluate a predicted edit $\hat{\mathbf{C}}$ on the MEDit-Bench's dataset $\mathcal{D} = \{(V, M, \mathbf{C})\}$ using the following three metrics, where V, M, and $\mathbf{C} = (c_1, \ldots, c_{K'})$ represent the footage, message, and reference cuts, respectively. Overlap is measured against professional edits made for the same message, so a model that misreads the message selects the wrong moments and scores low, and the metric inherits its validity from what the references are rather than from the overlap formula alone.

Recall at Temporal IoU. Since each edit consists of multiple cuts, we adopt a *union-based* temporal IoU formulation that treats a cut sequence as a single temporal region. Let $\mathrm{tIoU}(c, c') = |c \cap c'|/|c \cup c'|$ be the temporal IoU of two cuts, where $|\cdot|$ denotes the duration, we define a binary hit at threshold θ as

$$h_\theta(\hat{\mathbf{C}},\, \mathbf{C}) = \mathbb{1}\Big[\mathrm{tIoU}(\hat{\mathbf{C}},\, \mathbf{C}) \geq \theta\Big], \tag{2}$$

where $\mathrm{tIoU}(\hat{\mathbf{C}},\, \mathbf{C}) = \mathrm{tIoU}\Big(\bigcup_m \hat{c}_m,\, \bigcup_n c_n\Big)$. R@$\theta$ is the hit rate averaged over $\mathcal{D}$:

$$\mathrm{R@}\theta = \frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} h_\theta(\hat{\mathbf{C}},\, \mathbf{C}), \tag{3}$$

which we report at $\theta \in \{0.3, 0.5, 0.7\}$ along with the mean temporal IoU (mIoU).

F1 Score with Order-Preserving Cut Matching. We collect all candidate pairs (i, j) with $\mathrm{tIoU}(\hat{c}_i, c_j) \geq \theta$ subject to an order-preserving constraint: for any $(i, j), (i', j') \in \mathcal{M}_\theta$, $i < i'$ implies $j < j'$. We find a maximum-cardinality matching $\mathcal{M}_\theta \subseteq [K] \times [K']$ satisfying this constraint via dynamic programming, equivalent to solving the longest common subsequence problem. The F1 at threshold θ is defined as

$$\mathrm{F1@}\theta = \frac{2\, P@\theta \cdot R@\theta}{P@\theta + R@\theta}, \tag{4}$$

where $\mathrm{P@}\theta = |\mathcal{M}_\theta|/K$, $\mathrm{R@}\theta = |\mathcal{M}_\theta|/K'$, and $\theta \in \{0.3, 0.5, 0.7\}$. Unlike the union-based recall above, this F1 score evaluates cut localization, cut quantity consistency, and sequence order: reversing otherwise matching cuts reduces the matched size $|\mathcal{M}_\theta|$.

Human Perceptual Evaluation. Temporal overlap does not fully capture whether an edit communicates its message or forms a coherent narrative. We therefore also conduct human perceptual evaluation. Evaluators compare a model edit with a professional edit and rate both on a five-point Likert scale for five aspects. These are *(A) message transmission*, which measures how well the intended narrative is conveyed, *(B) cut order*, which measures whether the shot order serves the message, *(C) shot continuity*, which measures the smoothness of transitions, *(D) contextual depth*, which measures whether sufficient context is provided, and *(E) narrative pacing*, which measures whether cut timing and rhythm are appropriate. Each evaluator also gives an initial preference before rating the aspects and a final preference afterward. We report the mean score for each aspect and both preference rates. Further protocol details are provided in Sec. E.

LLM-based Preference. Temporal metrics alone are insufficient: an edit $\hat{\mathbf{C}}$ may convey the message well without temporally matching $\mathbf{C}$, or match it closely yet miss key context (Liu et al., 2023b). Following CineBench (Zhang et al., 2026b), we use Gemini-3-pro in an LLM-as-a-judge protocol (Zheng et al., 2023; Gu et al., 2024) to probe *semantic* quality. Because MEDit-Bench specifies a high-level communicative goal rather than concrete content or operations, its judge must determine whether shot selection and order construct a plausible narrative that conveys the message, making evaluation harder than checking explicit requests. The judge performs a blind A/B comparison of $\hat{\mathbf{C}}$ and $\mathbf{C}$ given M, considering shot selection and storytelling flow. Let $\mathcal{J}_M(\mathbf{C}, \hat{\mathbf{C}}) \in \{0, 1\}$ return 1 if the first-presented edit is preferred. Since LLM judges exhibit *position bias* Shi et al. (2025), we evaluate each pair in both presentation orders. Let N_M and N_H denote the numbers of comparisons in which the model edit and the human edit, respectively, are preferred in both orders:

$$\begin{aligned} N_M &= \sum_{\mathcal{D}} \mathcal{J}_M(\hat{\mathbf{C}}, \mathbf{C})\big(1 - \mathcal{J}_M(\mathbf{C}, \hat{\mathbf{C}})\big), \\ N_H &= \sum_{\mathcal{D}} \big(1 - \mathcal{J}_M(\hat{\mathbf{C}}, \mathbf{C})\big)\mathcal{J}_M(\mathbf{C}, \hat{\mathbf{C}}). \end{aligned} \tag{5}$$

Comparisons with different winners across the two orders contribute to neither count. We report the model win rate among the order-consistent comparisons:

$$\text{WinRate} = \frac{N_M}{N_M + N_H}. \tag{6}$$

4 METHODS

To tackle the proposed message-driven video editing task, we propose a training-free baseline method using frozen MLLMs (Sec. 4.1) and a reinforcement learning method trained on pseudo-labels distilled from a frontier MLLM (Sec. 4.2).

4.1 EDITING WITH FROZEN MLLMS

MLLMs such as LLaVA (Liu et al., 2023a) and the Qwen-VL family (Bai et al., 2025) have been shown to excel in video tasks such as video QA and captioning. Recent works have also revealed their potential in video temporal localization tasks such as temporal grounding and video summarization (Ren et al., 2024; Pang et al., 2026; Liu et al., 2024). To probe the message-driven video editing capability of such MLLMs, we treat an MLLM as a frozen function f that directly maps the video and message to a sequence of cuts:

$$\hat{\mathbf{C}} = f(V, M). \tag{7}$$

The MLLM is prompted with a fixed instruction (see the supplementary material for the full prompt and configuration) that introduces the task and specifies the output format.

4.2 GRPO WITH EDITING REWARDS

Group relative policy optimization (GRPO) (Shao et al., 2024) is widely used to train MLLMs on video tasks (Feng et al., 2025; Li et al., 2025; Wang et al., 2025b; Zhang et al., 2026a); for temporal grounding, Time-R1 (Wang et al., 2025b) and TimeLens (Zhang et al., 2026a) reward the temporal intersection-over-union (tIoU) between the predicted and reference span. We adopt the same cut format but find a tIoU-only reward suboptimal for editing: grounding localizes a *single* reference span, whereas an edit is a *sequence* of cuts. Extending tIoU to the union of the predicted and reference cuts rewards only *which* portion of the timeline is selected, not *how* it is partitioned, leaving the number and boundaries of cuts unsupervised; a few coarse cuts can then match the reference's coverage but not its composition. We therefore complement this coverage term with a *cut-level F1* reward that matches predicted and reference cuts one-to-one, rewarding the correct number and placement of cuts.

Reward. Following Sec. 3.4, let $\hat{\mathbf{C}} = (\hat{c}_1, \ldots, \hat{c}_K)$ and $\mathbf{C} = (c_1, \ldots, c_{K'})$ be the predicted and reference cuts of a training triplet $(V, M, \mathbf{C}) \in \mathcal{D}_{\text{GRPO}}$. Our *coverage* reward is the temporal IoU between the unions of the two cut sets,

$$\mathcal{R}_{\text{cov}} = \text{tIoU}(\hat{\mathbf{C}}, \mathbf{C}), \tag{8}$$

which scores temporal overlap but is invariant to how the union is partitioned into cuts.

The *cut-level F1* reward instead operates on individual cuts: we form a greedy matching set $\mathcal{P} \subseteq [K] \times [K']$ from the prediction and reference cut pairs as described in Sec. 3.4 but without the tIoU-based thresholding, and then accumulate soft true positives $\text{TP} = \sum_{(m,n)\in\mathcal{P}} \text{tIoU}(\hat{c}_m, c_n)$, yielding

$$\mathcal{R}_{\text{F1}} = \frac{2\,\text{P} \cdot \text{R}}{\text{P} + \text{R}}, \tag{9}$$

where $\text{P} = \text{TP}/K$ and $\text{R} = \text{TP}/K'$. This soft, tIoU-weighted TP, rather than a thresholded count, keeps the reward continuous in the cut boundaries.

The final reward combines these two terms with equal weight, *i.e.*,

$$\mathcal{R} = \mathcal{R}_{\text{cov}} + \mathcal{R}_{\text{F1}}, \tag{10}$$

so that $\mathcal{R}_{\text{cov}}$ enforces a decent global coverage while $\mathcal{R}_{\text{F1}}$ encourages the correct number and placement of cuts. We ablate the effect of these two rewards in the supplementary material.

Table 3: Main results on MEDit-Bench. Temporal alignment is measured by R@θ, order-preserving F1@θ, and mIoU. LLM Pref. is computed over presentation-order-consistent comparisons: green denotes cases in which the model is preferred in both orders, and red denotes cases in which the human reference is preferred in both orders; order-inconsistent cases are excluded. Because this metric remains affected by judge limitations, we report it only as a reference. The Human row shows inter-annotator performance (gray).

Model	R@0.3	R@0.5	R@0.7	F1@0.3	F1@0.5	F1@0.7	mIoU	LLM Pref.
Random	0.030	0.002	0.000	0.118	0.053	0.017	0.112	16.7%
Commercial models.								
GPT-5	**0.348**	0.065	0.009	**0.262**	**0.148**	0.063	**0.253**	77.8%
GPT-5-mini	0.157	0.033	0.000	0.221	0.109	0.040	0.186	31.1%
Gemini-3-pro (Team et al., 2023)	0.263	0.054	0.004	0.242	0.145	**0.080**	0.235	72.8%
Gemini-2.5-pro (Team et al., 2023)	0.144	0.033	0.006	0.173	0.097	0.049	0.152	28.3%
Open-source models.								
Qwen3-VL-4B (Bai et al., 2025)	0.032	0.009	0.002	0.070	0.034	0.013	0.119	25.2%
Gemma-4-E4B	0.111	0.017	0.000	0.155	0.071	0.024	0.169	30.6%
Qwen3-VL-8B (Bai et al., 2025)	0.144	0.030	0.004	0.168	0.096	0.041	0.165	26.1%
InternVL3.5-8B (Wang et al., 2025a)	0.020	0.000	0.000	0.103	0.042	0.011	0.080	7.4%
Qwen3-VL-32B (Bai et al., 2025)	0.248	0.070	0.013	0.224	0.122	0.060	0.227	40.7%
MEDitor-4B (Ours)	0.244	0.069	0.013	0.245	0.145	0.070	0.229	52.9%
MEDitor-8B (Ours)	0.304	**0.100**	**0.026**	0.250	0.147	0.061	0.241	54.9%
Human	0.432	0.133	0.033	0.265	0.177	0.106	0.290	—

Policy Optimization. We optimize the policy π_Φ with GRPO (Shao et al., 2024), which directly maximizes the reward $\mathcal{R}$ rather than imitating the reference cuts by maximum likelihood. For each training sample $(V, M, \mathbf{C}) \in \mathcal{D}_{\text{GRPO}}$, we sample a group $\mathcal{G} = \{\hat{\mathbf{C}}\}$ of $|\mathcal{G}|$ candidate edits from the current policy and score each with the reward,

$$r = \mathcal{R}(\hat{\mathbf{C}}, \mathbf{C}). \tag{11}$$

The group-normalized advantage of $\hat{\mathbf{C}}$ is given by

$$A(\hat{\mathbf{C}}) = \frac{r - \mu_{\mathcal{G}}}{\sigma_{\mathcal{G}}}, \tag{12}$$

where $\mu_{\mathcal{G}}$ and $\sigma_{\mathcal{G}}$ are the mean and standard deviation of r over $\mathcal{G}$. The GRPO objective is

$$\mathcal{L}_{\text{GRPO}}(\Phi) = -\mathbb{E}\left[\sum_{\hat{\mathbf{C}} \in \mathcal{G}} A(\hat{\mathbf{C}}) \log \pi_\Phi(\hat{\mathbf{C}} \mid V, M)\right], \tag{13}$$

where we drop the KL divergence term as in (Wang et al., 2025b; Zhang et al., 2026a).

Training Data. To build a training set without expensive professional annotation, we construct the dataset $\mathcal{D}_{\text{GRPO}}$ by distilling pseudo-labels from a frontier MLLM $\mathcal{M}_{\text{teacher}}$ over 977 videos from MLVU (Zhou et al., 2025), with durations ranging from 5 to 15 minutes. Formally, for each source video V in MLVU, $\mathcal{M}_{\text{teacher}}$ directly generates three message-edit pairs, which yields $977 \times 3 = 2{,}931$ distillation samples in total; details of the construction procedure and dataset statistics are given in Sec. C.3. GRPO uses the teacher cuts as a scalar reward on the model's own rollouts rather than as targets to imitate, so the supervision need not reach reference quality and the student is never fitted to teacher-specific behavior.

5 Experiments

5.1 Evaluation Setup

We evaluate a range of commercial and open-source MLLMs, with each model's default frame sampling strategy (more details in Sec. C), on MEDit-Bench in zero-shot settings, as well as MEDitor, our GRPO-based fine-tuned models (Sec. 4.2). We use the temporal, human perceptual, and LLM-based measures defined in Sec. 3.4. We also evaluate MEDitor on classic video summarization benchmarks, *i.e.*, SumMe (Gygli et al., 2014) and TVSum (Song et al., 2015), and on the script-driven S-VideoXum (Mylonas et al., 2025), with the setup and results reported in Sec. 5.4. We additionally compare against direct text-to-video retrieval baselines and apply SD-VSum to MEDit-Bench for the reverse transfer direction, with the setup and results reported in Sec. B.3 and Sec. B.4 of the supplementary material.

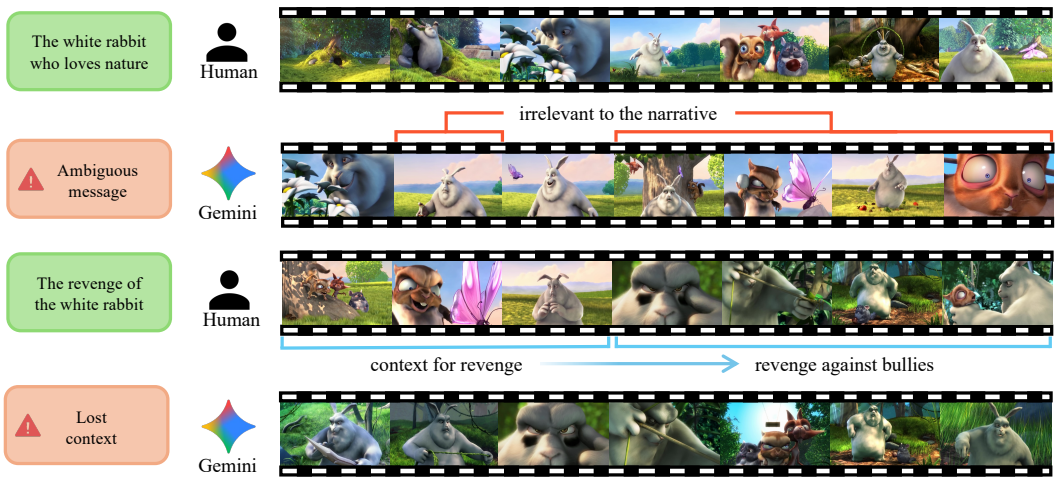

Figure 1: Qualitative comparison of human and Gemini-3-pro edits on two cases where most evaluators preferred the human edit. Each pair shows the human edit (top strip) and the Gemini edit (bottom strip) alongside the diagnosed failure mode. *Top* ("The white rabbit who loves nature"): the abstract message causes Gemini to select clips irrelevant to the intended narrative. *Bottom* ("The revenge of the white rabbit"): Gemini omits the contextual buildup needed to make the reversal meaningful.

5.2 MEDIT-BENCH RESULTS

Temporal alignment. Most models outperform the Random baseline, but fall short of the human editing. Among zero-shot models, larger commercial models generally outperform open-source ones, suggesting that general-purpose video reasoning capacity is an important factor. Performance degrades sharply at stricter thresholds; even the best R@0.7 is only 0.026, indicating that precise cut-boundary alignment remains very challenging. MEDitor substantially improves performance: MEDitor-8B achieves the best or joint-best scores among all zero-shot open-source models and rivals commercial models, achieving the highest R@0.5 (0.100) and R@0.7 (0.026) among all evaluated models. These gains remain under the order constraint: F1@0.3 rises from 0.070 to 0.245 for the 4B model and from 0.168 to 0.250 for the 8B model.

LLM-based preference. While the win-rate metric has known reliability concerns (Sec. 6), its ranking is broadly consistent with model capability: larger models score markedly higher than smaller ones, and MEDitor raises win rate substantially, mirroring the trend in temporal metrics. One exception is Gemini-3-pro, whose win rate (72.8%) appears disproportionately high relative to its temporal scores, possibly reflecting self-enhancement bias.

Human perceptual evaluation. Six evaluators compared professional human edits with Gemini-3-pro outputs over all 540 pairs; the protocol and the full per-aspect results are in Sec. E of the supplementary material. Human edits are preferred in 51.7% of direct choices and in 53.3% of comparisons by summed aspect scores. They also score higher than the model on all five aspects, with the largest margin on *cut order*, indicating that models particularly struggle to order shots so that they convey the narrative. The margins are nonetheless narrow, and individual judgments disagree heavily, so the two are separated only in aggregate (Sec. E.3 of the supplementary material).

5.3 QUALITATIVE ANALYSIS

To illustrate the nature of the gap between humans and models, we present qualitative comparisons on two video-message pairs for which most evaluators preferred the human edit and the score gap was especially pronounced (Fig. 1).

Sample 1 (message: *"The white rabbit who loves nature"*). The message lacks references to concrete objects, making the cut selection criterion inherently ambiguous. The model (Gemini-3-pro) selects clips mostly irrelevant to the nuance of the message, failing to ground the abstract message in visual content. The widest gaps are on message transmission and cut order (Human/model 4.17/3.17 and 4.17/3.00, respectively). We observe that models tend to handle messages containing explicit object names more reliably, suggesting that message concreteness is a key factor for good edits.

Sample 2 (message: *"The revenge of the white rabbit"*). This message implies a causal arc from victimization to retaliation. The human editor includes contextual buildup, *i.e.*, scenes establishing the bullying dynamic, before the counterattack, making the eventual reversal comprehensible and emotionally resonant. The model cuts directly to the revenge without sufficient setup, losing the narrative context that gives the revenge its meaning. The largest gap appears in cut order (Human/model 4.33/2.67), whereas shot continuity is rated *higher* for the model edit (3.83 vs. 3.50), demonstrating that temporal smoothness alone does not imply narrative coherence.

5.4 Generalization to Video Summarization

Setup. We probe whether the capabilities acquired through GRPO training transfer to classic video summarization, without any additional fine-tuning. The prompt format and model weights are kept identical to the message-driven editing task, replacing only the message M. Because the model's output is sensitive to the phrasing of M (different messages select different cuts), we do not rely on a single message but *ensemble* a small set of generic highlight messages ("critical", "exciting", and "memorable moments"), running the model once per message and averaging their per-frame selection masks into a graded, phrasing-robust per-frame importance. We then compute Kendall's τ and Spearman's ρ between this importance and the human ground truth by following Otani et al. (2019). Following Son et al. (2024b), for SumMe, the reference is the consensus selection frequency (the fraction of annotators selecting each frame); for TVSum, the coefficients are computed for each annotator separately and then averaged.

Results. Although neither benchmark is seen during training, the editing capability transfers zero-shot. MEDitor-8B attains the best SumMe scores (τ/ρ 0.218/0.254) and both MEDitor models lead on TVSum (0.193/0.219), surpassing prior unsupervised and vision-language methods on both benchmarks. The improvement over the untrained Qwen3-VL backbones is largest on TVSum, where τ climbs from 0.081 (4B) and 0.089 (8B) to 0.193, confirming that the transfer is driven by GRPO training rather than the base model. On TVSum our models even exceed the leave-one-out human agreement (0.177/0.204): they match the annotator consensus more closely than individual annotators match one another, reflecting the well-known annotator disagreement on this benchmark. On SumMe they remain below the higher human agreement (0.296/0.329), and model scale helps there (8B $>$ 4B) while making little difference on TVSum.

Transfer to script-driven summarization. The same transfer holds for S-VideoXum (Mylonas et al., 2025), whose input is a script rather than a message: MEDitor-8B reaches 28.38 F1 against 24.71 for SD-VSum, which is trained on this benchmark, while the reverse direction fails (full setup, table, and analysis in Sec. B.4).

6 Limitations

While professional production also involves titles and captions, color grading, and motion effects, MEDit-Bench focuses on cut-only editing. Cut selection, however, is the foundational step that drives these later decisions, making it the natural starting point.

Scoring a message-driven edit remains an open question. Temporal alignment is reproducible but misses narrative coherence. Our LLM judge is insensitive to narrative craft, exhibits position and self-enhancement biases, and depends on commercial models subject to silent updates. Human evaluation is direct but costly and noisy: only 6.7% of our 540 pairs drew a unanimous verdict from six evaluators, while score variation falls from 1.9 points with one rater to 0.4 with five. This difficulty motivates the benchmark: its 3,240 released judgments let candidate metrics be tested against independent human verdicts. Rubric-based evaluation is a promising path toward calibrated measures of narrative coherence, message transmission, and pacing (Wei et al., 2026).

7 Conclusion

We introduced MEDit-Bench, a benchmark pairing long-form videos and multiple natural-language editing messages with professional edits. Results show that precise cut placement and narrative coherence remain open challenges, and human evaluation prefers professional edits across all evaluated aspects. We hope MEDit-Bench supports progress in message-aware video editing.

AI USE STATEMENT

For paper preparation, generative AI tools were used only to polish language and presentation. They were not used to draft the substantive scientific content, formulate the research questions or hypotheses, develop the conceptual framework or methodology, or interpret the experimental results; these were written and determined by the human authors. Separately, LLMs were used as components of the experiments: GPT-5 generated the pseudo-labeled message–edit pairs used for GRPO training, an LLM-assisted protocol supported the ambiguity and contextfulness annotations, and Gemini-3-pro was evaluated as an LLM-based preference judge. These uses, including the models, prompts, aggregation procedures, and limitations, are documented in the main paper and supplementary material. The authors manually reviewed every AI-generated artifact and output used in this work and validated all experimental results. The authors take full responsibility for all claims, results, citations, and artifacts in this submission.

ETHICS STATEMENT

This work involves human contributors as professional video editors and as evaluators in the perceptual study. Editors were recruited using the objective screening criteria described in the supplementary material and were paid $67–$100 USD per source video; crowd evaluators were paid approximately $2 USD per evaluated pair. We report judgments only in aggregate and do not use demographic attributes in our analysis. MEDit-Bench releases source-video identifiers, editing messages, and derived cut timestamps rather than redistributing third-party source footage. The source videos are drawn from publicly available datasets and Blender Studio films; users remain responsible for complying with the applicable licenses and terms of the original providers. As message-conditioned editing can also be used to create misleading narratives, the dataset and models should not be used to misrepresent people or events. Our benchmark is intended for research on understanding and evaluating narrative video editing.

REPRODUCIBILITY STATEMENT

The main paper specifies the task, model, reward, and evaluation metrics. The supplementary material provides the dataset construction and quality-control procedure (Section A), retrieval and model configurations and GRPO training hyperparameters (Section C), ablations (Section D), the human-study protocol, and the complete inference and evaluation prompts (Section G). We will publicly release all code needed for data processing, training, inference, and evaluation; the trained MEDitor model weights; the complete MEDit-Bench annotations; and the MEDitor GRPO training data. Because the source videos are hosted by their original providers, we will release their identifiers and our derived messages and cut annotations. Experiments involving commercial models may be affected by provider-side model updates; we therefore identify the evaluated models and document their prompts and inference procedures.

REFERENCES

Mehryar Abbasi and Parvaneh Saeedi. Adopting self-supervised learning into unsupervised video summarization through restorative score. In *ICIP*, pp. 425–429, 2023.

Evlampios Apostolidis, Georgios Balaouras, Vasileios Mezaris, and Ioannis Patras. Combining global and local attention with positional encoding for video summarization. In *IEEE Int. Symp. Multimedia*, pp. 226–234, 2021. doi: 10.1109/ISM52913.2021.00045.

Evlampios Apostolidis, Georgios Balaouras, Vasileios Mezaris, and Ioannis Patras. Summarizing videos using concentrated attention and considering the uniqueness and diversity of the video frames. In *Int. Conf. Multimedia Retrieval*, pp. 407–415, 2022.

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 *CVPR*, pp. 8332–8341, 2024.

Shuai Bai et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Blender Studio. Blender studio films. https://studio.blender.org/films/, 2026. Collection of open movies by Blender Studio.

Sandra Eliza Fontes de Avila, Ana Paula Brandão Lopes, Antonio da Luz, and Arnaldo de Albuquerque Araújo. VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognit. Lett.*, 32(1):56–68, 2011.

Jiri Fajtl, Hajar Sadeghi Sokeh, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. Summarizing videos with attention. In *Asian Conf. Comput. Vis. Workshops*, pp. 39–54, 2019.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-R1: Reinforcing video reasoning in MLLMs. *NeurIPS*, 38:99114–99137, 2025.

Nathan Frey, Peggy Chi, Weilong Yang, and Irfan Essa. Automatic non-linear video editing transfer. In *CVPRW*, 2021. URL https://arxiv.org/abs/2105.06988.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *CVPR*, 2025.

Wei Ge, Yongwei Nie, Fei Ma, Keke Tang, Fei Richard Yu, Hongmin Cai, and Ping Li. Training-free language-guided video summarization via multi-grained saliency scoring. In *Computational Visual Media*, pp. 89–104, 2025.

Junaid Ahmed Ghauri, Sherzod Hakimov, and Ralph Ewerth. Supervised video summarization via multiple feature sets with parallel attention. In *Int. Conf. Multimedia and Expo*, pp. 1–6, 2021.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. A survey on LLM-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024.

Yongxin Guo, Jingyu Liu, Mingda Li, Dingxin Cheng, Xiaoying Tang, Dianbo Sui, Qingbin Liu, Xi Chen, and Kevin Zhao. VTG-LLM: Integrating timestamp knowledge into video LLMs for enhanced video temporal grounding. In *AAAI*, volume 39, pp. 3302–3310, 2025.

Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *ECCV*, pp. 505–520, 2014.

Bo He, Jun Wang, Jielin Qiu, Trung Bui, Abhinav Shrivastava, and Zhaowen Wang. Align and attend: Multimodal summarization with dual contrastive losses. In *CVPR*, 2023.

Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. MovieNet: A holistic dataset for movie understanding. In *ECCV*, 2020.

Hao Jiang and Yadong Mu. Joint video summarization and moment localization by cross-task sample transfer. In *CVPR*, pp. 16367–16377, 2022.

Yunjae Jung, Donghyeon Cho, Dahun Kim, Sanghyun Woo, and In So Kweon. Discriminative feature learning for unsupervised video summarization. In *AAAI*, 2019.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, pp. 706–715, 2017.

Mackenzie Leake, Abe Davis, Anh Truong, and Maneesh Agrawala. Computational video editing for dialogue-driven scenes. *ACM Trans. Graph.*, 36(4), 2017.

Min Jung Lee, Dayoung Gong, and Minsu Cho. Video summarization with large language models. In *CVPR*, pp. 18981–18991, 2025.

Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. *NeurIPS*, 34:11846–11858, 2021.

Haopeng Li, Qiuhong Ke, Mingming Gong, and Tom Drummond. Progressive video summarization via multimodal self-supervised learning. In *WACV*, pp. 5573–5582, 2023a.

Haopeng Li, Qiuhong Ke, Mingming Gong, and Tom Drummond. Progressive video summarization via multimodal self-supervised learning. In *WACV*, pp. 5584–5593, 2023b.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Qwen3-VL-Embedding and Qwen3-VL-Reranker: A unified framework for state-of-the-art multimodal retrieval and ranking, 2026a.

Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. VideoChat-R1: Enhancing spatio-temporal perception via reinforcement fine-tuning. *NeurIPS*, 2025.

Xinhao Li, Yuhan Zhu, Xiangyu Zeng, et al. Videochat3: Fully open video mllm for efficient and generalist video understanding, 2026b. URL https://arxiv.org/abs/2607.14935.

Jingyang Lin, Hang Hua, Ming Chen, Yikang Li, Jenhao Hsiao, Chiuman Ho, and Jiebo Luo. VideoXum: cross-modal visual and textural summarization of videos. *IEEE Trans. Multimedia*, 26:5548–5560, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 36:34892–34916, 2023a.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using gpt-4 with better human alignment. In *EMNLP*, pp. 2511–2522, 2023b. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153/.

Ye Liu, Zongyang Ma, Zhongang Qi, Yang Wu, Ying Shan, and Chang W Chen. E.T. Bench: Towards open-ended event-level video-language understanding. *NeurIPS*, 37:32076–32110, 2024.

Boris Meinardus, Hector Rodriguez, Anil Batra, Anna Rohrbach, and Marcus Rohrbach. Chrono: A simple blueprint for representing time in MLLMs. *arXiv preprint arXiv:2406.18113*, 2024.

Manolis Mylonas, Evlampios Apostolidis, and Vasileios Mezaris. SD-VSum: A method and dataset for script-driven video summarization. In *ACM Int. Conf. Multimedia*, 2025.

Medhini Narasimhan, Anna Rohrbach, and Trevor Darrell. CLIP-It! language-guided video summarization. In *NeurIPS*, volume 34, pp. 13988–14000, 2021.

Mayu Otani, Yuta Nakashima, Esa Rahtu, and Janne Heikkilä. Rethinking the evaluation of video summaries. In *CVPR*, pp. 7588–7596, 2019. doi: 10.1109/CVPR.2019.00778.

Zongshang Pang, Yuta Nakashima, Mayu Otani, and Hajime Nagahara. Contrastive losses are natural criteria for unsupervised video summarization. In *WACV*, pp. 2010–2019, 2023.

Zongshang Pang, Mayu Otani, and Yuta Nakashima. Measure Twice, Cut Once: A semantic-oriented approach to video temporal localization with video LLMs. In *ICLR*, 2026.

Alejandro Pardo, Fabian Caba Heilbron, Juan León Alcázar, Ali Thabet, and Bernard Ghanem. MovieCuts: A new dataset and benchmark for cut type recognition. In *ECCV*, pp. 668–685, 2022.

Alejandro Pardo, Jui-Hsien Wang, Bernard Ghanem, Josef Sivic, Bryan Russell, and Fabian Caba Heilbron. Generative timelines for instructed visual assembly. In *NeurIPS Workshop*, 2024.

Jielin Qiu, Jiacheng Zhu, William Han, Aditesh Kumar, Karthik Mittal, Claire Jin, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Ding Zhao, Bo Li, and Lijuan Wang. MMSum: A dataset for multimodal summarization and thumbnail generation of videos. In *CVPR*, pp. 21909–21921, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pp. 8748–8763, 2021.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. TimeChat: A time-sensitive multimodal large language model for long video understanding. In *CVPR*, pp. 14313–14323, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.

Aidean Sharghi, Jacob S. Laurel, and Boqing Gong. Query-focused video summarization: Dataset, evaluation, and a memory network based approach. In *CVPR*, 2017.

Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. Judging the judges: A systematic study of position bias in LLM-as-a-Judge. In *Proc. Int. Joint Conf. Nat. Lang. Process.*, pp. 292–314, 2025.

Mike Zheng Shou, Stan Weixian Lei, Weiyao Wang, Deepti Ghadiyaram, and Matt Feiszli. Generic event boundary detection: A benchmark for event segmentation. In *ICCV*, pp. 8055–8064, 2021. doi: 10.1109/ICCV48922.2021.00797.

Jaewon Son, Jaehun Park, and Kwangsu Kim. CSTA: Cnn-based spatiotemporal attention for video summarization. In *CVPR*, pp. 18847–18856, 2024a.

Jaewon Son, Jaehun Park, and Kwangsu Kim. Csta: Cnn-based spatiotemporal attention for video summarization. In *CVPR*, pp. 18847–18856, 2024b.

Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. TVSum: Summarizing web videos using titles. In *CVPR*, pp. 5179–5187, 2015. doi: 10.1109/CVPR.2015.7299154.

Tomoya Sugihara, Shuntaro Masuda, Ling Xiao, and Toshihiko Yamasaki. Language-guided self-supervised video summarization using text semantic matching considering the diversity of the video. In *ACM Int. Conf. Multimedia Asia*, 2024.

Jinhwan Sul, Jihoon Han, and Joonseok Lee. Mr. HiSum: a large-scale dataset for video highlight detection and summarization. *NeurIPS*, 36, 2023.

Gemini Team et al. Gemini: A family of highly capable multimodal models, 2023.

Anh Truong, Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. QuickCut: An interactive tool for editing narrated video. In *ACM Symp. User Interface Softw. Technol.*, pp. 497–507, 2016.

Junyan Wang, Yang Bai, Yang Long, Bingzhang Hu, Zhenhua Chai, Yu Guan, and Xiaolin Wei. Query Twice: Dual mixture attention meta learning for video summarization. In *ACM Int. Conf. Multimedia*, pp. 4023–4031, 2020.

Miao Wang, Guo-Wei Yang, Shi-Min Hu, Shing-Tung Yau, and Ariel Shamir. Write-A-Video: Computational video montage from themed text. *ACM Trans. Graph.*, 38(6):177:1–177:13, 2019.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025a.

Ye Wang, Ziheng Wang, Boshen Xu, Yang Du, Kejun Lin, Zihan Xiao, Zihao Yue, Jianzhong Ju, Liang Zhang, Dingyi Yang, et al. Time-R1: Post-training large vision language model for temporal video grounding. *NeurIPS*, 38:83330–83364, 2025b.

Yana Wei, Hongbo Peng, Yanlin Lai, Liang Zhao, Kangheng Lin, En Yu, Keyu Lv, Han Zhou, Yin Tang, Haodong Li, Mitt Huang, Hangyu Guo, Jianjian Sun, Zheng Ge, Xiangyu Zhang, Daxin Jiang, and Vishal M. Patel. PerceptionRubrics: Calibrating multimodal evaluation to human perception. *ICML*, 2026.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: a benchmark for long-context interleaved video-language understanding. In *NeurIPS*, 2024.

Yu Xiong, Fabian Caba Heilbron, and Dahua Lin. Transcript to Video: Efficient clip sequencing from texts. In *ACM Int. Conf. Multimedia*, pp. 5407–5416, 2022.

Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *CVPR*, pp. 10714–10726, 2023.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, pp. 11975–11986, 2023.

Jun Zhang, Teng Wang, Yuying Ge, Yixiao Ge, Xinhao Li, and Limin Wang. TimeLens: Rethinking video temporal grounding with multimodal LLMs. In *CVPR*, pp. 10419–10429, 2026a.

Peixuan Zhang, Chang Zhou, Ziyuan Zhang, Hualuo Liu, Chunjie Zhang, Jingqi Liu, Xiaohui Zhou, Xi Chen, Shuchen Weng, Si Li, and Boxin Shi. A benchmark and multi-agent system for instruction-driven cinematic video compilation. In *ECCV*, 2026b.

Xinlu Zhang, Yujie Lu, Weizhi Wang, An Yan, Jun Yan, Lianke Qin, Heng Wang, Xifeng Yan, William Yang Wang, and Linda Ruth Petzold. Gpt-4v(ision) as a generalist evaluator for vision-language tasks. *CoRR*, abs/2311.01361, 2023.

Bin Zhao, Haopeng Li, Xiaoqiang Lu, and Xuelong Li. Reconstructive sequence-graph network for video summarization. *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2021.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and Chatbot Arena. In *NeurIPS*, 2023.

Sheng-Hua Zhong, Jingxu Lin, Jianglin Lu, Ahmed Fares, and Tongwei Ren. Deep semantic and attentive network for unsupervised video summarization. *ACM Trans. Multimedia Comput. Commun. Appl.*, 18(2), 2022.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. MLVU: Benchmarking multi-task long video understanding. In *CVPR*, pp. 13691–13701, 2025.

Kaiyang Zhou, Yu Qiao, and Tao Xiang. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *AAAI*, 2018.

Wencheng Zhu, Jiwen Lu, Jiahao Li, and Jie Zhou. DSNet: A flexible detect-to-summarize network for video summarization. *IEEE Trans. Image Process.*, 30:948–962, 2020.

SUPPLEMENTARY MATERIAL

This supplementary material provides (A) dataset collection details including statistics, editor recruitment, message creation, and message quality annotations, (B) further comparison with existing datasets including retrieval baselines and cross-dataset transfer, (C) implementation details including GRPO training data construction, (D) ablation studies, (E) human perceptual study protocol and reliability analysis, (F) LLM-as-judge limitations, (G) full inference prompts, and (H) representative dataset examples.

A DATASET COLLECTION DETAILS

A.1 VIDEO SELECTION

We selected 60 videos from LongVideoBench (Wu et al., 2024), VideoMME (Fu et al., 2025) and Blender Studio open movies (Blender Studio, 2026) applying the following criteria.

Domain diversity. We aimed for a balanced distribution across eight categories to ensure the benchmark covers a wide range of visual styles and content types. Table 4 shows the resulting distribution.

Table 4: Domain distribution of MEDit-Bench source videos.

Category	**# Videos**
Vlog	14
Travel	13
Movie	9
Animation	6
Sports	5
Documentary	5
Food	4
Cooking	4
Total	**60**

Message potential. We selected videos with visually rich and varied content, enabling the creation of multiple distinct editing messages per video.

Visual editability. We excluded videos in which the primary information is conveyed through speech rather than visual content (*e.g.*, news broadcasts where spoken text is essential for understanding), ensuring that all editing decisions can be made based on visual content alone.

A.2 DATASET STATISTICS

Figure 2 summarizes the statistics of MEDit-Bench. Source videos range from 461 to 1,076 seconds (824 seconds on average), while edited videos range from 28 to 329 seconds (132 seconds on average), with an average compression ratio of 16.5%. Cuts are distributed almost uniformly over time, indicating minimal temporal bias. MEDit-Bench contains 6,231 cuts in total; the average cut length is 11.40 seconds (median 5.09 seconds).

Shot reordering. To characterize the extent to which editors deviate from chronological ordering, we count cuts whose start timestamp precedes that of the immediately preceding cut in the edit sequence. Table 5 shows the reordering rates for all evaluated models and the human reference. Among professional edits, 26.1% of edits and 5.3% of cuts involve at least one temporal reversal, whereas current AI models produce near-zero reordering rates. This gap highlights that intentional shot reordering, an important tool in professional narrative editing, is largely absent from model outputs, representing a qualitative gap beyond what temporal alignment metrics alone can capture.

A.3 EDITOR RECRUITMENT AND COMPENSATION

Editors were recruited through a crowdsourcing platform via an open call. We screened applicants on two quantitative criteria that the platform records and publishes for every account, namely a lifetime client rating of at least **4 out of 5** and at least **50 completed jobs**, and we additionally

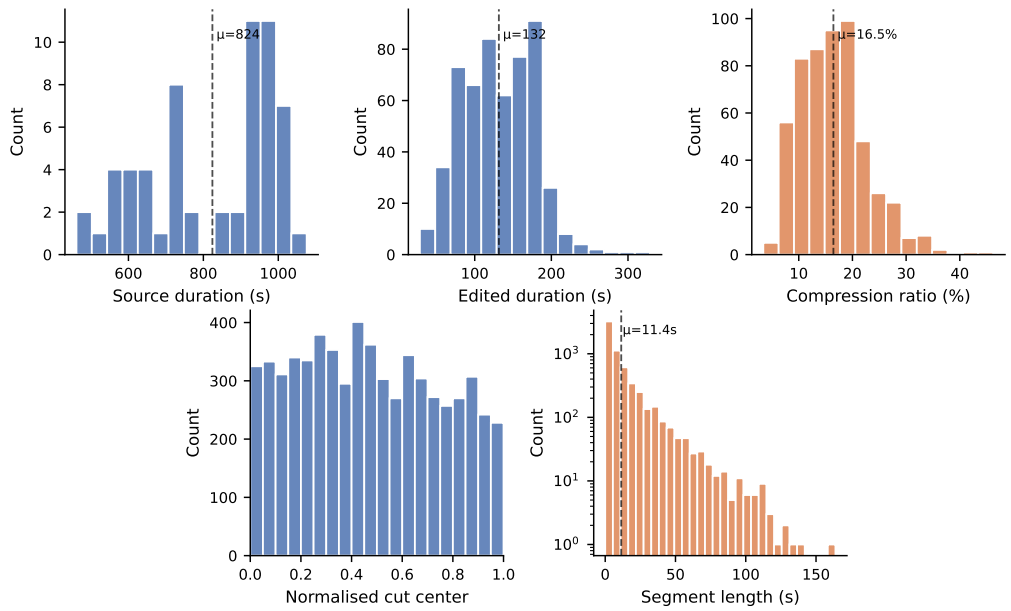

Figure 2: Statistics of the MEDit-Bench dataset.

Table 5: Shot reordering rates for AI models and human reference editors. "Edits" counts edits with at least one temporal reversal; "Cuts" counts individual cuts placed before the preceding cut.

Model	Edits w/ reorder	Cuts (reorder)
GPT-5	0/180 (0.0%)	0/1542 (0.0%)
GPT-5-mini	3/180 (1.7%)	3/1106 (0.3%)
Gemini-3-pro	0/180 (0.0%)	0/2025 (0.0%)
Gemini-2.5-pro	0/180 (0.0%)	0/1582 (0.0%)
Qwen3-VL-4B	14/180 (8.0%)	124/15939 (0.8%)
Gemma-4-E4B	1/180 (0.6%)	1/934 (0.1%)
Qwen3-VL-8B	0/180 (0.0%)	0/2304 (0.0%)
InternVL3.5-8B	0/180 (0.0%)	0/1472 (0.0%)
Qwen3-VL-32B	0/180 (0.0%)	0/2730 (0.0%)
MEDitor-4B	2/180 (1.1%)	2/2030 (0.1%)
MEDitor-8B	4/180 (2.2%)	5/1503 (0.3%)
Human (reference)	**141/540 (26.1%)**	**329/6231 (5.3%)**

required demonstrated prior video editing experience. Both thresholds are objective and verifiable by anyone on the platform, so the resulting pool does not depend on our own impression of an applicant. The rating aggregates the assessments of dozens of past clients rather than ours, and the job count establishes that the editor has sustained paid work rather than passed a single audition. Each editor was compensated at **$67–$100 USD per source video**, which covers producing three edited videos (one per message) along with the required deliverables.

A.4 Editor Instructions

Each editor received a source video (7–18 minutes) and three editing messages, and was asked to produce one edited video per message. Editors are instructed to (i) trim the source to a 1–3 minute summary using only cut operations, (ii) reorder shots freely, (iii) ignore the audio and base their decisions solely on the visual content, and (iv) provide a rationale for every cut to discourage careless cuts. We preserve the submitted shot order for sequence-aware evaluation and release both the ordering and rationales to support future research.

For each edited video, editors were required to submit three deliverables: (i) an exported video file (mp4), (ii) the editing project file (Adobe Premiere Pro or equivalent), and (iii) a rationale memo (one line per cut) explaining why each scene was selected.

A.5 Data Availability

The released dataset consists of (i) an identifier for each source video, (ii) the editing messages in text form, and (iii) the cut timestamps for each professional edit, extracted from the editors' project files (Adobe Premiere Pro or equivalent) to ensure frame-level precision. The source videos originate from LongVideoBench (Wu et al., 2024), VideoMME (Fu et al., 2025), and the Blender Studio open movies (Blender Studio, 2026), all of which are publicly available, so the underlying footage can be freely accessed.

A.6 Quality Control

We place the quality control on the editor rather than on the edit, and this is a deliberate choice. Narrative quality has no accepted objective measure, which is the premise of this benchmark, so screening submissions by how good we judged the resulting story would write our own taste into the reference set. The references would then record what we find convincing rather than what professionals actually do, which is the opposite of what the benchmark is for. We therefore front-load the control into recruitment (Sec. A.3) and keep the review of submissions to effort and compliance.

All submitted edits were manually reviewed by the authors against the instructions in Sec. A.4. For edits with an unusually low number of cuts or those that appeared superficial, the rationale memo was cross-checked against the actual cut sequence to verify that each cut was intentional and consistent with the stated intent. Submissions that failed this check, for instance by omitting per-cut rationales or by falling outside the requested length range, were returned to the editor for resubmission. We did not rank or filter submissions by how convincing we personally found the resulting narrative.

A.7 Message Creation Process

After watching each source video in full, the authors created three editing messages per video. Messages were designed to satisfy the following properties.

Visual communicability. Each message describes a narrative that can be conveyed through visual content alone, without relying on audio or spoken dialogue, consistent with the visual-only editing constraint imposed on editors.

Diversity of intent. The three messages per video were designed to evoke distinctly different impressions or narratives from the same footage, ensuring that different messages yield substantially different edits.

Varied difficulty. The message set deliberately spans a range of difficulty levels. This includes straightforward messages with clear visual anchors, ambiguous messages open to multiple equally valid interpretations, and context-dependent messages that require introductory shots to make the narrative meaningful to viewers. This design is reflected in the ambiguity and contextfulness annotations described in Sec. A.8.

A.8 Message Quality Annotations

Not every message is equally easy to turn into an edit. Some messages can be conveyed by many different sets of shots, and some require background that the message never names. We annotate all 180 messages along these two axes using an LLM-assisted, human-corrected protocol. The complete prompt is provided in Sec. G.5.

Ambiguity (1–5) is how many valid readings a message admits. *"A story of resilience"* offers no concrete visual anchor and can be realized in many ways.

Contextfulness (1–5) is how much context a message presupposes. *"The long-awaited reunion"* needs shots establishing who separated and why before the reunion itself carries any weight.

The two axes pull in opposite directions. Ambiguity widens the set of edits that could count as correct, while contextfulness narrows it by demanding shots the message does not mention. Figure 3 shows their distribution, which is moderate-to-high for ambiguity (scores 3–4) and low-to-moderate for contextfulness (scores 1–2).

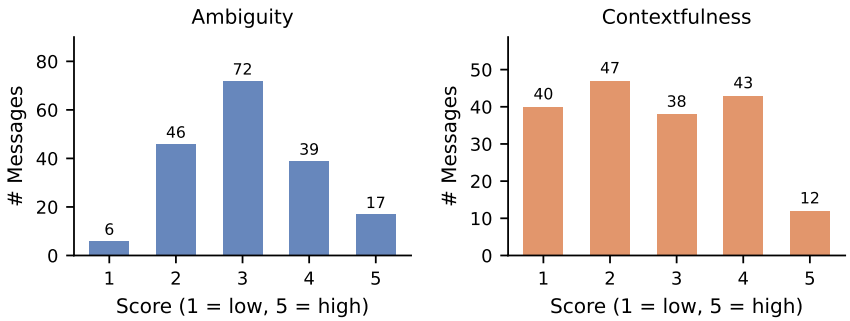

Figure 3: Distribution of message quality annotations across the 180 MEDit-Bench messages. Ambiguity measures how open the message is to interpretation. Contextfulness measures the extent to which external context is required.

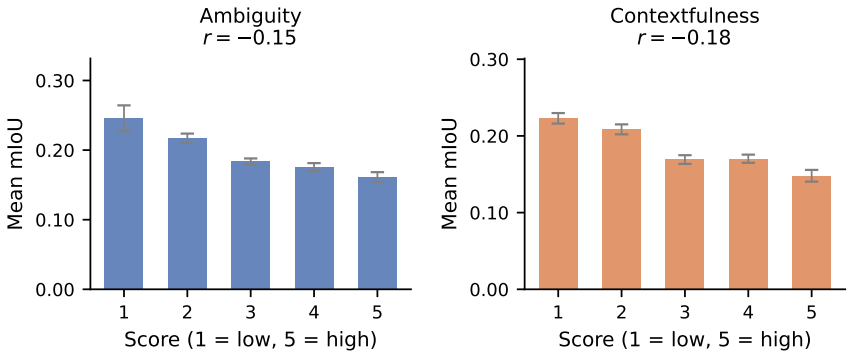

Figure 4: Mean mIoU as a function of message ambiguity (left) and contextfulness (right), aggregated across all models. Error bars show standard error.

Both axes predict how well models do. Figure 4 shows that ambiguity and contextfulness each correlate negatively with mIoU ($r = -0.15$ and $r = -0.18$) when aggregated over all evaluated models. The most ambiguous messages (score 5) yield mIoU roughly 34% lower than the least ambiguous ones (score 1), and contextful messages pose a separate difficulty by requiring models to supply background that the message does not name. Message quality is therefore a meaningful stratification factor for evaluation.

B Further Comparison with Existing Datasets

B.1 Compression Ratio

Sec. 3.2 of the main paper compares source and target durations in absolute terms. Table 6 instead reports the fraction of the source footage that each target retains, computed as the mean of the per-pair ratios rather than as the ratio of the means.

Table 6: Fraction of the source footage retained by the target summary, computed over the videos used for evaluation. †Derived from the 15% length budget imposed by the evaluation protocol rather than measured from the annotations. ‡Restricted to the portion of the VideoXum test split for which script annotations are available.

Dataset	Avg. source	Avg. target	Retained
SumMe (Gygli et al., 2014)	2m40s	24s	15.0%†
VSUMM (de Avila et al., 2011)	1m30s	14s	15.0%†
TVSum (Song et al., 2015)	4m12s	38s	15.0%†
QFVS (Sharghi et al., 2017)	4h14m	5m4s	**2.1%**
VideoXum (Lin et al., 2024)	2m4s	20.5s	16.5%
S-VideoXum (Mylonas et al., 2025)‡	2m5s	20.6s	16.5%
MEDit-Bench	13m44s	2m12s	16.5%

SumMe, VSUMM, and TVSum sit at exactly 15% because their protocols impose a 15% length budget, so these entries are budget-derived rather than measured. MEDit-Bench retains 16.5%, the same as the VideoXum family, even though its footage is an order of magnitude longer.

QFVS is the only outlier at 2.1%, which we measured from the 180 oracle summaries of the standard split against the frame counts of the four source videos. At 2% of a four-hour recording, most shots are irrelevant to any query and the task is to find the few that are not. At 16% of a fourteen-minute video, most shots are plausible and the task is to choose among them and order them. The two benchmarks therefore ask for different things rather than for different amounts of the same thing.

B.2 Qualitative Comparison of Input Text

Table 7 compares representative inputs from four text-conditioned datasets.

Table 7: Representative text inputs and their relation to the target edit.

Dataset	Example input	Relation to the target
QFVS (Sharghi et al., 2017)	*Hat and phone*	Names visible concepts that the selected shots should contain
S-VideoXum (Mylonas et al., 2025)	*A woman and a young girl in a boat … both wearing life jackets*	Generated from an existing summary and describes its selected visual content
CineBench (Zhang et al., 2026b)	*It first confronts the viewer with the single, most gut-wrenching moment of his story. Only after we've witnessed this final act of despair does the sequence rewind … Please add music to enhance the emotion.*	Specifies a narrative ordering and a concrete post-production operation
MEDit-Bench	*The revenge of the white rabbit*	Written before editing and states the intended meaning without prescribing shots

QFVS specifies objects to retrieve, while the S-VideoXum script describes content already chosen for its target summary. The CineBench example is taken from an instruction shown in the paper. It specifies how the story should unfold: the ending comes first and is followed by a rewind. It also requests an editing operation, adding music. MEDit-Bench instead fixes only the communicative goal before editing, leaving the model to construct the narrative and choose the evidence, context, and shot order needed to express it.

B.3 Retrieval Baselines

We test the claim underlying the comparisons above, namely that our messages do not name the content to be selected. If a message named its content the way a query or a script does, embedding it and matching it against the shots would recover much of the reference edit.

We first split each source video into shots by content-aware detection on HSV differences, with a threshold of 27 and a minimum shot duration of 0.5 seconds. Each shot is represented by four evenly spaced frames, which we embed with CLIP ViT-L/14 (Radford et al., 2021), SigLIP So400m (Zhai et al., 2023), or Qwen3-VL-Embedding-8B (Li et al., 2026a), then average within the shot and normalize. Embedding the message with the same encoder gives one similarity score per shot, and we keep the highest-scoring shots until 15% of the video is filled.

The two variants differ only in the unit on which this budget is spent. The scene variant keeps whole shots, which is what an editor would do. The frame variant copies each shot's score to every second it contains and keeps the top 15% of seconds, which can cut a shot in half but matches how per-frame scoring methods such as SD-VSum select, making that comparison direct.

The best retrieval result reaches 0.131 F1@0.3 and 0.162 mIoU, compared with 0.250 and 0.241 for MEDitor-8B, and the gap widens at stricter thresholds where retrieval reaches 0.019 R@0.5 against 0.100. Retrieval therefore recovers little of the reference, which is what the ambiguity and con-

Table 8: Direct retrieval baselines on MEDit-Bench. Bold marks the best retrieval baseline per metric. The last three rows are quoted from the main results table for reference and are not retrieval baselines.

Embedder	Selection	R@0.3	R@0.5	R@0.7	F1@0.3	F1@0.5	F1@0.7	mIoU
CLIP (Radford et al., 2021)	Scene	0.046	0.004	0.002	0.068	0.039	0.025	0.116
CLIP (Radford et al., 2021)	Frame	0.056	0.002	0.000	0.091	0.046	0.022	0.116
Qwen3-VL (Li et al., 2026a)	Scene	**0.120**	0.017	**0.007**	0.093	0.057	0.036	0.160
Qwen3-VL (Li et al., 2026a)	Frame	0.119	**0.019**	0.002	**0.131**	**0.073**	**0.037**	**0.162**
SigLIP (Zhai et al., 2023)	Scene	0.041	0.000	0.000	0.066	0.042	0.026	0.108
SigLIP (Zhai et al., 2023)	Frame	0.037	0.000	0.000	0.087	0.047	0.022	0.108
Qwen3-VL-8B		0.144	0.030	0.004	0.168	0.096	0.041	0.165
MEDitor-4B		0.244	0.069	0.013	0.245	0.145	0.070	0.229
MEDitor-8B		0.304	0.100	0.026	0.250	0.147	0.061	0.241

Table 9: Zero-shot transfer to the S-VideoXum test split, where each script is given as the input message.

Model	F1	τ	ρ
SD-VSum (Mylonas et al., 2025)	24.71	0.101	0.123
Qwen3-VL-4B (Bai et al., 2025)	24.73	0.082	0.085
Qwen3-VL-8B (Bai et al., 2025)	25.90	0.099	0.101
MEDitor-4B (Ours)	26.95	0.113	0.119
MEDitor-8B (Ours)	**28.38**	**0.128**	**0.133**

textfulness annotations in Sec. A.8 lead us to expect, since an ambiguous message has no concrete anchor to embed and a contextful one calls for shots it never mentions. Choosing the right shots instead requires inferring which of them set up or advance the message, which is a reasoning step rather than a similarity computation, and this is why we build the task around MLLMs.

B.4 Cross-Dataset Evaluation

Sec. 5.4 of the main paper probes whether the message-driven editing capability transfers to classic video summarization without additional fine-tuning; here we give the full setup and results for both directions of that transfer with S-VideoXum (Mylonas et al., 2025), and the reverse-direction transfer of SD-VSum onto MEDit-Bench.

Forward direction: MEDitor on S-VideoXum. We repeat the summarization probe of Sec. 5.4 on S-VideoXum, feeding each script in place of the message M and converting the predicted segments to a 1 FPS binary mask.

Table 9 shows that MEDitor-8B reaches 28.38 F1 against 24.71 for SD-VSum, which is trained on this benchmark, and both MEDitor models improve over their untrained backbones. Message-driven editing therefore transfers to script-driven summarization even though the two text inputs differ.

Reverse direction: SD-VSum on MEDit-Bench. Here we report the reverse direction, applying the released SD-VSum checkpoint to MEDit-Bench with CLIP video and text embeddings sampled at 1 FPS. SD-VSum emits a per-frame score rather than a cut list, so we convert it either by taking the top 15% of frames or by thresholding the score after min-max normalization within each video and message pair.

On MEDit-Bench, SD-VSum reaches at best 0.096 mIoU and 0.022 R@0.3, against 0.241 and 0.304 for MEDitor-8B. Its scores localize the amount of material to keep more closely than the particular parts, with a best mIoU of 0.096 near the 0.112 of the random baseline in Tab. 3 of the main paper and a best F1@0.5 of 0.028 against 0.147 for MEDitor-8B. Taken with the forward direction above, the transfer is asymmetric, since MEDitor carries over to script-driven summarization while SD-VSum does not carry over in the reverse direction. This follows from the input each method is built for, since an S-VideoXum script names the content its summary contains whereas a message expresses a communicative intent rather than describing the content to select.

Table 10: SD-VSum transfer to MEDit-Bench. Selection is the rule that turns the per-frame score of SD-VSum into a cut list. Bold marks the best SD-VSum setting per metric. The last two rows are quoted from the main results table for reference, and MEDitor needs no conversion because it emits a cut list directly.

Method	Selection	mIoU	R@0.3	R@0.5	R@0.7	F1@0.5
SD-VSum (Mylonas et al., 2025)	Top 15%	0.088	0.020	**0.002**	0.000	**0.028**
	Threshold 0.9	0.012	0.000	0.000	0.000	0.008
	Threshold 0.8	0.021	0.002	0.000	0.000	0.012
	Threshold 0.7	0.029	0.007	0.000	0.000	0.014
	Threshold 0.6	0.053	0.009	**0.002**	0.000	0.015
	Threshold 0.5	**0.096**	**0.022**	0.000	0.000	0.027
MEDitor-4B	—	0.229	0.244	0.069	0.013	0.145
MEDitor-8B	—	0.241	0.304	0.100	0.026	0.147

C Implementation Details

Every model is evaluated under its standard video-processing configuration. We apply no task-specific or relevance-based frame selection anywhere in the pipeline. Frames are drawn uniformly in time by each model's own decoder, never as a function of the message, so no model is shown a message-conditioned subset of the footage. The frame budget nonetheless differs between models, and we state it below for each one.

C.1 Open-Source Models

Qwen3-VL. We evaluate Qwen3-VL in 4B, 8B, and 32B parameter variants. Videos are sampled at 2 FPS with a maximum of 2,048 frames. Spatial resolution is controlled by a per-frame lower bound of `min_pixels = 65,536` and a cross-frame upper bound of `total_pixels = 20,971,520`, which allows the model to adaptively trade frame count against per-frame resolution based on video length. The maximum number of generated tokens is set to 1024.

Gemma-4. We evaluate Gemma-4-E4B using its native video processing pipeline with its default sampling rate of 1 FPS. Thinking mode is disabled (`enable_thinking = False`) and the maximum number of generated tokens is set to 1,024.

InternVL3.5. We evaluate InternVL3.5-8B with its default fixed-frame setting, which uniformly samples 32 frames from each video regardless of its duration rather than sampling at a fixed rate. Each frame is resized to 448×448 and treated as a single tile (`max_num = 1`, no dynamic sub-image splitting). The maximum number of generated tokens is set to 1024.

C.2 Commercial Models

GPT-5. Following (Zhang et al., 2026a), we sample frames at 1 FPS and prepend a textual timestamp label (*e.g.*, `Frame at 2.5s:`) to each frame. For videos shorter than 50 seconds all frames are passed directly; for videos of 50–80 seconds we uniformly downsample to 50 frames; for videos longer than 80 seconds we sample at 1 FPS and arrange every 4 consecutive frames into a 2×2 grid image. For GPT-5, which is a reasoning model, we use the default value for the `reasoning_effort` parameter. GPT-5-mini uses the same pipeline.

Gemini. Source videos are uploaded to the Google File API and passed directly to the model via the `file_data` interface, delegating all frame sampling and resolution control to the model's native video processing. We evaluate both Gemini-2.5-pro and Gemini-3-pro using this pipeline with its default sampling rate of 1 FPS.

C.3 GRPO Training Data Construction

We use GPT-5 as the teacher model $\mathcal{M}_{\text{teacher}}$ to generate pseudo-labeled (message, cut sequence) pairs over 977 source videos drawn from the MLVU benchmark (Zhou et al., 2025), with durations ranging from 5 to 15 minutes. For each video, the teacher generates three message-edit pairs, yielding $977 \times 3 = 2{,}931$ training samples in total.

Dataset statistics. Table 11 and Figure 5 summarize the key statistics of the resulting training set. Source footage averages 543 seconds (9.1 min), and the generated edits average 74 seconds with a compression ratio of approximately 14.3%. Each generated edit selects on average 4.2 cuts, with an average segment length of 17.7 seconds.

Table 11: Statistics of MEDit-Bench and MEDitor (GRPO) training data.

		MEDit-Bench	MEDitor (GRPO)
Source	Mean	824 s	543 s
	Median	905 s	490 s
Edit duration	Mean	132 s	74 s
	Median	130 s	69 s
Cuts per edit	Mean	11.5	4.2
	Median	6.0	4.0
Compression ratio	Mean	16.5%	14.3%
	Median	16.2%	13.7%
Segment length	Mean	11.4 s	17.7 s
	Median	5.1 s	14.0 s

Figure 5 shows the distributions of source duration, edited duration, compression ratio, normalised cut center, and segment length across the 977 training videos and 2,931 generated edits.

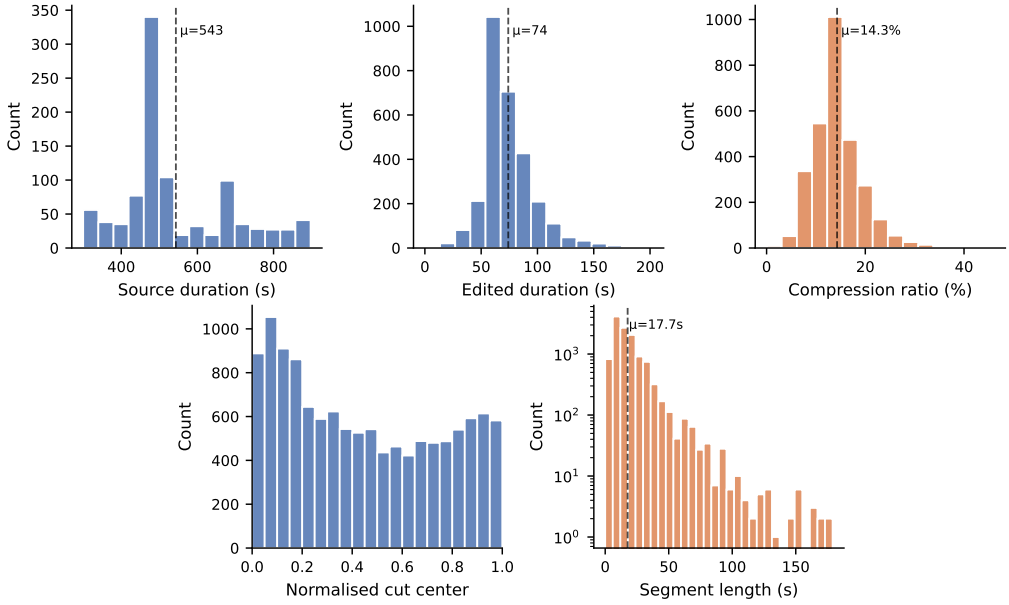

Figure 5: Distribution statistics of the MEDitor (GRPO) training data. From left to right: source video duration (s), generated edit duration (s), compression ratio (%), normalised cut center, and segment length (s, log scale). Dashed lines mark the mean.

Generation prompt. The prompt used to generate pseudo-labels is shown in Section G.2.

C.4 MEDitor Training and Inference

Training. We initialize MEDitor directly from Qwen3-VL-4B/8B-Instruct and train it with GRPO on the 2,931 distilled (message, cut sequence) pairs. For every sample we draw $G = 4$ rollouts at temperature 1.0, normalize the advantages within the group, and drop the KL term ($\beta = 0$). The vision encoder is frozen while the language model and the vision-language merger are trained, with a constant learning rate of 1×10^{-6}, global batch size 64 for half an epoch, beyond which the training reward starts to plateau. Videos are decoded at 2 fps and capped at 448 frames with a total visual-token budget of 14,336 (per-frame floor 64).

Inference. At test time MEDitor uses greedy decoding (no sampling) with the same video sampling budget as training. Given a video and message, it emits the plain seconds list specified by the prompt, which we parse into the predicted cut set; no post-processing or re-ranking is applied.

C.5 Human Performance Evaluation

The *Human* row in Table 3 of the main paper reports inter-annotator performance under a leave-one-out protocol. For each (video, message) pair, one of the three editors is designated as the predictor and the remaining two as references. All temporal alignment metrics (R@θ, F1@θ, mIoU) are computed between the predictor's edit and each reference edit separately, then averaged over the two references. This procedure is repeated for all three choices of predictor editor, and the final score is the average over all three combinations and all (video, message) pairs in MEDit-Bench.

D Ablation Studies

D.1 Reward Function Ablation

We ablate the GRPO reward on the MEDit-Bench to isolate the contribution of each term in our combined reward $\mathcal{R} = \mathcal{R}_{\text{cov}} + \mathcal{R}_{\text{F1}}$. All three variants train the base Qwen3-VL-4B under identical settings (GPT-5 pseudo-labels, half an epoch, global batch 64) and differ only in the reward: coverage only ($\mathcal{R}_{\text{cov}}$), cut-matching only ($\mathcal{R}_{\text{F1}}$), and their equal-weight combination.

Table 12: Reward ablation on MEDit-Bench. mIoU is union-mask coverage; F1@τ uses the order-preserving cut matching defined in the main paper. **Bold** denotes the best of the three rewards.

Reward	mIoU	F1@.3	F1@.5	F1@.7	Cuts/edit
Raw (no training)	11.90	6.99	3.42	1.32	88.5
$\mathcal{R}_{\text{cov}}$ only	24.01	12.18	5.68	3.72	1.8
$\mathcal{R}_{\text{F1}}$ only	22.80	24.67	13.43	5.84	8.5
Combined (ours)	**24.73**	**24.84**	**15.12**	**7.43**	8.2
Human	28.97	26.45	17.74	10.60	11.5

Findings. The combined reward is best on all four alignment metrics, and the two terms are complementary rather than redundant. The raw model severely over-segments the video (88.5 cuts per edit vs. the human 11.5). Coverage alone ($\mathcal{R}_{\text{cov}}$) overcorrects in the opposite direction because it is blind to how the timeline is partitioned: the policy collapses to a single fat span (1.8 cuts), matches at most a couple of the human moments, and substantially reduces F1 (F1@.5 $= 5.68$). Cut-matching alone brings the cut count much closer to the human reference (8.5 cuts) and substantially improves F1, but is slightly weaker on coverage. Combining the two produces a similarly realistic cut count (8.2) together with the highest coverage and F1.

D.2 Transfer to Other Backbones

To test whether our GRPO recipe is specific to Qwen3-VL, we apply it unchanged to VideoChat3-4B (Li et al., 2026b), a video-LLM with a different vision encoder and temporal tokenization: identical reward, training data, prompt, video sampling budget, and training budget.

Table 13: GRPO transfer to a different backbone on MEDit-Bench. Both models are trained with the identical recipe for half an epoch.

Model	R@0.3	R@0.5	R@0.7	F1@0.3	F1@0.5	F1@0.7	mIoU
Qwen3-VL-4B	0.032	0.009	0.002	0.070	0.034	0.013	0.119
+ GRPO	0.244	0.069	0.013	0.245	0.145	0.070	0.229
VideoChat3-4B	0.133	0.024	0.006	0.106	0.056	0.024	0.149
+ GRPO	0.220	0.056	0.017	0.214	0.121	0.056	0.209
Human	0.432	0.133	0.033	0.265	0.177	0.106	0.290

As shown in Table 13, GRPO roughly doubles VideoChat3's F1 at every threshold (F1@0.3: $0.106 \rightarrow 0.214$) and lifts mIoU from 0.149 to 0.209, mirroring the gains on Qwen3-VL. The recipe

thus transfers to architectures beyond Qwen3-VL without any per-backbone tuning, indicating that the improvement stems from the reward design rather than from properties of a particular base model.

D.3 Message Ensembling for Video Summarization

Table 14: Zero-shot performance on video summarization benchmarks. **Bold** and underline denote the best and second-best scores per metric. *Human*: per video, each annotator is scored against the consensus of the remaining annotators (SumMe) or against every other annotator individually (TVSum), then averaged over annotators. Methods are cited by year; see the bibliography for full references.

Method	SumMe		TVSum		Method	SumMe		TVSum	
	τ	ρ	τ	ρ		τ	ρ	τ	ρ
Vision-only methods.					*Vision-language methods.*				
CSNet (2019)	—	—	0.025	0.034	LGSSVS (2024)	0.102	0.148	0.133	0.174
CASUM (2022)	0.063	0.084	0.160	0.210	LfVS (zero-shot) (2024)	0.125	0.148	0.151	0.182
DSAVS (2022)	—	—	0.080	0.087	Ge et al. (2025)	0.157	0.182	<u>0.184</u>	0.213
DRDSN (2018)	0.047	0.048	0.020	0.026	SSPVS (2023b)	0.154	<u>0.207</u>	0.151	0.199
RSGN$_u$ (2021)	0.071	0.073	0.048	0.052	Qwen3-VL-4B (2025)	0.138	0.161	0.081	0.092
RSSUM (2023)	0.007	0.015	0.080	0.106	Qwen3-VL-8B (2025)	<u>0.160</u>	0.186	0.089	0.101
CTVSum (2023)	0.036	0.044	0.161	0.212	MEDitor-4B (Ours)	0.149	0.172	**0.193**	**0.219**
					MEDitor-8B (Ours)	**0.218**	**0.254**	**0.193**	<u>0.218</u>
Random	0.000	0.000	0.000	0.000	Human	0.296	0.329	0.177	0.204

Our video-summarization evaluation runs the model with three generic messages ("critical", "exciting", and "memorable" moments) and averages their per-frame selection masks into a single soft importance, which is then scored once. We emphasize that AVG3 averages the model's *predictions*, not the per-message metrics: it forms one averaged prediction and computes a single τ/ρ from it, rather than scoring each message separately and averaging the three coefficients. Table 15 decomposes this ensemble against each constituent message, where every single-message column is scored on that message's own prediction.

Table 15: Message-ensembling ablation for video summarization (Kendall τ). Single-message columns score each message's prediction on its own; AVG3 averages the three predictions (per-frame masks) into one importance and scores that single averaged prediction.

Model	Data	critical	exciting	memorable	AVG3
Qwen3-VL-4B	SumMe	0.121	0.136	0.107	**0.138**
	TVSum	0.056	**0.092**	0.064	0.081
Qwen3-VL-8B	SumMe	0.106	**0.163**	0.146	0.160
	TVSum	**0.092**	0.076	0.052	0.089
MEDitor-4B	SumMe	**0.159**	0.132	0.108	0.149
	TVSum	0.181	0.189	0.168	**0.193**
MEDitor-8B	SumMe	0.198	0.197	0.217	**0.218**
	TVSum	0.176	0.162	0.150	**0.193**

Findings. No single message is consistently best: the top-scoring message changes across models and datasets (e.g., "critical" for MEDitor-4B on SumMe but "exciting" on TVSum). Averaging the three avoids this brittle choice while matching or exceeding the best single message; for both MEDitor models on TVSum, AVG3 strictly beats every individual message (e.g., MEDitor-8B: 0.193 vs. at most 0.176).

E Human Perceptual Study Details

E.1 Study Design

We conducted a human perceptual study comparing professional human edits with Gemini-3-pro outputs on **540 edit pairs** from MEDit-Bench. Each pair was independently assessed by **six evaluators**, comprising three in-house evaluators and three crowd workers. Evaluators watched both

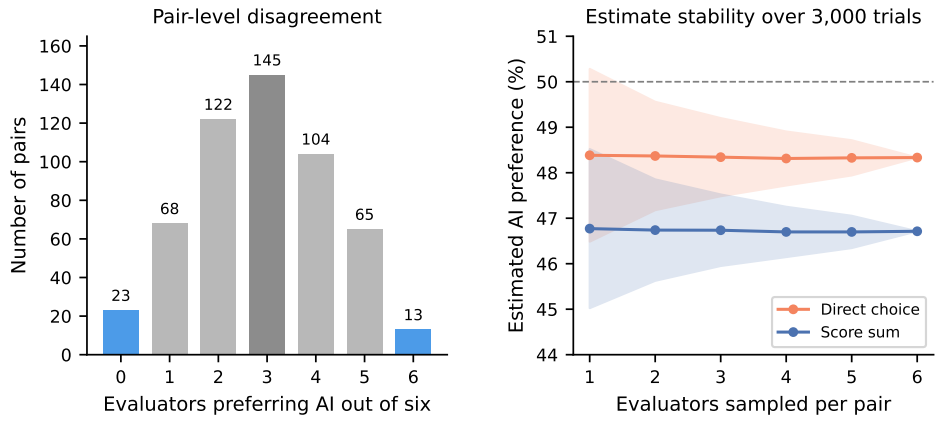

Figure 6: Evaluator disagreement and estimate stability. The left panel shows the number of pairs grouped by how many of six evaluators preferred the model. The right panel shows the estimated model preference as the number of sampled evaluators increases. Shaded regions denote one standard deviation over 3,000 trials.

edits side by side and first provided an immediate preference. They then scored both edits on five aspects using a 5-point Likert scale and gave a considered preference after reviewing their scores. This protocol yields **3,240 evaluator-pair judgments**.

Participants. The crowd workers were recruited without restrictions on gender or age and were drawn randomly from the available pool. Each crowd worker was compensated approximately **$2 USD per evaluated pair**.

E.2 Evaluation Aspects

Evaluators rated each edit on five aspects adapted from film editing practice, with higher scores indicating better quality.

(A) Message Transmission. Does the edit successfully communicate the intended message?
(B) Cut Order. Is the ordering of shots logical for conveying the narrative?
(C) Shot Continuity. Do transitions between cuts feel natural and visually coherent?
(D) Contextual Depth. Does the edit provide sufficient context for the narrative?
(E) Editing Pace. Is the pacing appropriate for the intended message?

E.3 Evaluator Subjectivity and Estimate Stability

Evaluator disagreement. Of the 540 pairs, 145 pairs had an even three-to-three split in the final direct choice, accounting for **26.9%** of the data. Only 36 pairs received unanimous choices, accounting for **6.7%**. Moreover, 371 pairs had between two and four votes for the model, accounting for **68.7%**. These results show that perceived editing quality is highly subjective and that a judgment from one evaluator can depend strongly on who is assigned.

Stability with more evaluators. For each number of evaluators from one to six, we sampled that many evaluators independently for every pair without replacement and recomputed the overall model preference. We repeated this procedure 3,000 times. Direct choice uses the final preference, while score sum compares the sums of the five aspect scores. Score ties contribute one half to each edit. The standard deviation decreases from 1.9 percentage points with one evaluator to 0.4 points with five evaluators for direct choice, and from approximately 1.8 to 0.4 points for score sum. With all six available evaluators, the subsampling variance is zero because every trial uses the same finite pool. This endpoint does not imply zero uncertainty beyond these six evaluators.

E.4 Aggregate Results

Figure 7 reports the aggregate outcome of the study. Human edits win 51.7% of the direct choices and 53.3% of the comparisons by summed aspect scores, and they score higher than Gemini-3-pro on every one of the five aspects, with the largest margin on *cut order*.

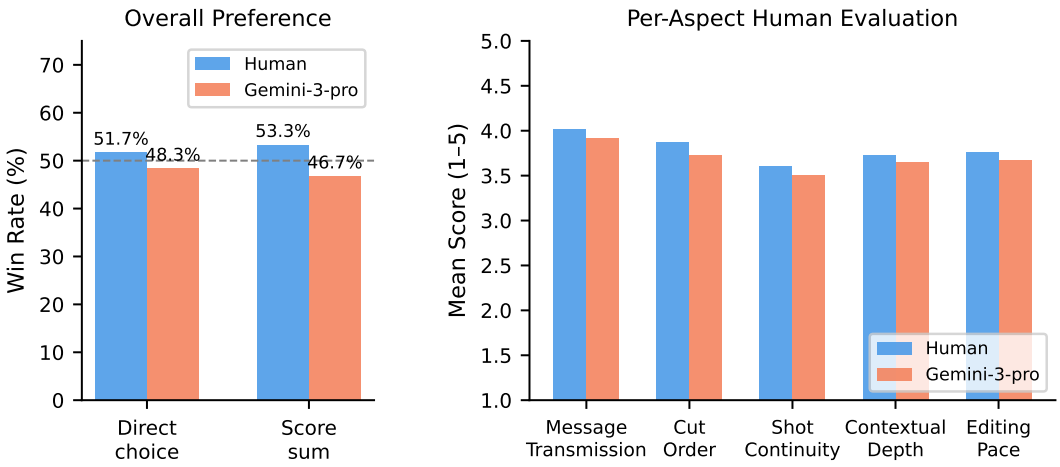

Figure 7: Human perceptual study comparing professional human edits with Gemini-3-pro outputs over 540 pairs rated by six evaluators. *Left*: win rates under direct choice and under summed aspect scores. *Right*: mean scores (1–5) per evaluation aspect, where higher is better.

F LLM-AS-A-JUDGE LIMITATIONS

Position bias. LLM judges tend to favor whichever video appears in position A, independent of content quality. We quantify this across three diagnostic runs of our Gemini-3-pro A/B evaluation (Gemini-3-pro as judge): when the human edit is Video A (*human-ai* condition), the judge selects the human edit **55.6%** of the time; when positions are swapped (*ai-human* condition), the human win rate drops to **20.2%**, a swing of **35.4 pp** driven purely by presentation order. Decomposing this swing, the video-A preference contributes $+17.7$ pp and the residual AI content advantage contributes $+12.1$ pp. As shown in Figure 8 (right), this position effect is consistent across all evaluated models. For the primary metric, we instead retain only comparisons for which the same winner is selected in both presentation orders. In the evaluation run reported in Table 3, 56.4% of comparisons are order-consistent. Among these comparisons, 72.8% prefer Gemini-3-pro in both orders and 27.2% prefer the human edit in both orders. The remaining 43.6% are order-inconsistent and are excluded from both the numerator and denominator.

Run-to-run stability. Despite the large position effect, repeated runs under the *same* presentation order are highly consistent. Pairwise agreement across three independent runs ranges from **83% to 87%** (Figure 8, left), confirming that the judge is stable and that additional trials under identical conditions do not alter the conclusions.

Correlation with human judgment. To examine how well LLM preferences align with human perception, we compare LLM-as-a-judge outcomes with human judgments on the same 540 triples in $\mathcal{D}$ for which judge outputs are available (Gemini-3-pro *vs.* human reference edits). For human preference, we use the human perceptual study results. For LLM preference, we fix the presentation order (human-first, *i.e.*, $\mathcal{J}(\mathbf{C}, \hat{\mathbf{C}})$) and query the judge three times independently, then aggregate via majority voting.[1]

We take the majority direct choice over the six evaluators as the human preference and exclude the 144 pairs with an even split, which leaves 395 pairs. The pair-wise agreement rate is **63.5%** (chance baseline = 50%), with Cohen's $\kappa = 0.27$, indicating only fair agreement above chance. Agreement rises to **86.2%** on the 29 pairs where the judge runs and all six evaluators are unanimous, suggesting the judge is most consistent when the quality difference is clear-cut. Treating judgments as continuous probabilities (fraction of votes preferring the model), Pearson's $r = 0.267$ and Spearman's $\rho = 0.263$ (both $p < 0.001$) confirm a statistically significant but modest monotonic association. Overall, these results show that the LLM judge captures a weak signal of human preference, but the low κ value underscores that it should not be treated as a reliable substitute for human evaluation.

[1]Due to the stochastic nature of LLM sampling, repeated runs under identical conditions can yield different individual judgments.

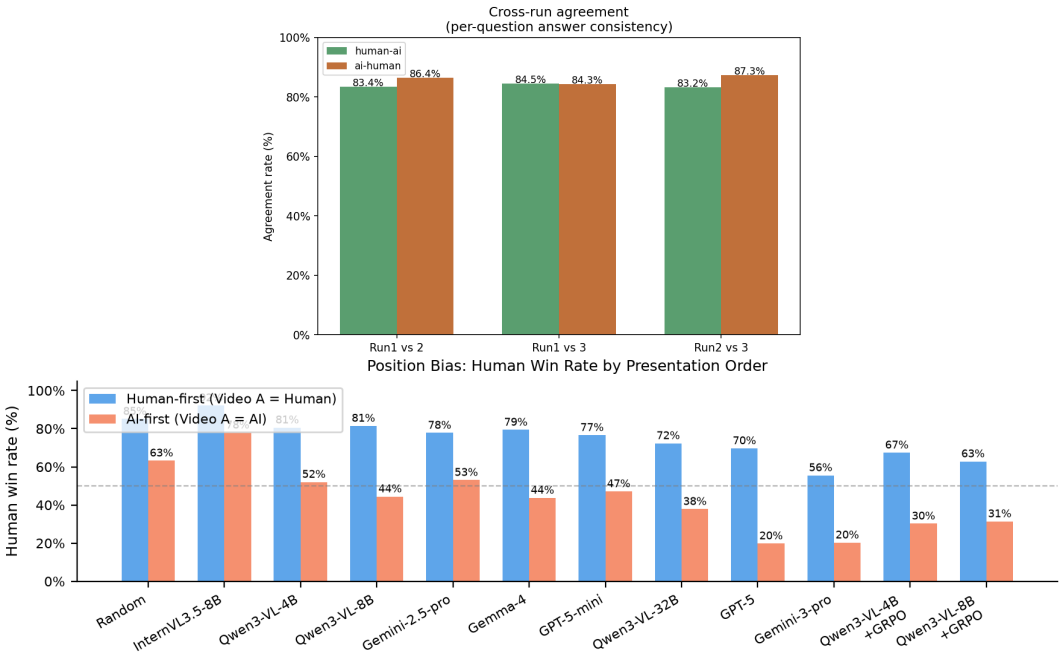

Figure 8: LLM-as-judge limitations. **Left**: cross-run pairwise agreement for Gemini-3-pro across three independent runs in both conditions; 83–87% agreement confirms the judge is stable across repeated trials. **Right**: human vs. AI win rate per model under human-ai (Video A = human) and ai-human (Video A = AI) conditions; win rates shift dramatically with presentation order, a consistent position bias across all models.

G PROMPTS

G.1 ZERO-SHOT EDITING PROMPT

All MLLM baselines (Qwen3-VL-8B, Gemma-4, InternVL3.5-8B, GPT-5, Gemini-2.5-pro, Gemini-3-pro) receive the identical prompt below, where {message} is replaced by the actual editing message.

Zero-shot Editing Prompt

```
You are a professional video editor. Review the
provided video and select the key temporal
highlights that best convey the following message to viewers:

Message: "{message}"

Respond strictly with JSON using this schema:
{
  "segments": [
    ["mm:ss.ff", "mm:ss.ff"],
    ...
  ]
}

Rules:
- Select segments that best communicate the message: "{message}"
- You may reorder segments when it helps communicate the message
- The total duration of all selected segments combined should be roughly 15% of the
video length
- Choose the most visually compelling and relevant moments
- Ensure all timestamps are within the video length
- Match the precision pattern "mm:ss.ff" (minutes, seconds, hundredths)
- Do not include descriptions, titles, or any extra commentary
- Return only valid JSON with no surrounding text

Example:
{"segments": [["00:15.46", "00:19.42"], ["01:16.05", "01:22.70"]]}
```

G.2 Pseudo-label Generation Prompt

GPT-5 generates three (message, cut sequence) pairs per source video using the prompt below, with frame-grid images as visual context.

Pseudo-label Generation Prompt

```
You are a video editor creating edited videos with clear intent and purpose.

Task:
Create 3 different edited video proposals. For each proposal:

1. First, decide on a specific MESSAGE you want to convey to viewers
   - What story do you want to tell?
   - What emotion or understanding should viewers gain?
   - What is the core intent behind this edit?

2. Then, with that MESSAGE in mind, select the clip intervals (timestamps) that will
effectively communicate it
   - Choose clips strategically to build a coherent narrative
   - Each edited video should be 1-3 minutes in total length
   - Use timestamps in [start, end] format with MM:SS:CS (e.g., ['01:23:00',
'02:45:50'])
   - You can arrange clips in any order -- chronological order is NOT required

Output exactly 3 EditedVideo objects, each with:
- A SHORT and CONCISE message (5-10 words maximum) that clearly captures the core
intent/purpose
- Timestamps selected specifically to deliver that message

IMPORTANT: Timestamp Format (MM:SS:CS):
  MM = minutes (00-59)
  SS = seconds (00-59)
  CS = centiseconds (00-99)
  e.g., "01:30:50" = 1 min 30.50 sec (NOT 1 hour 30 min)
```

G.3 MEDitor Prompt

Our GRPO-trained MEDitor uses the prompt below, kept identical at training and inference time, where {`message`} is the editing message. Unlike the JSON schema used for the zero-shot baselines, it requests a plain seconds list and states an explicit length budget of roughly 15% of the video.

MEDitor Prompt

```
You are a professional video editor. Select highlight segments conveying the message
below. The selected segments should total roughly 15% of the video length.

Message: "{message}"

List the selected highlights as time ranges in seconds, using exactly this format, with
 a single period only at the very end:
<start> to <end> seconds; <start> to <end> seconds; ...; <start> to <end> seconds.

Rules:
- Use seconds with one decimal, e.g. "12.5 to 30.0 seconds".
- Separate segments with "; "; put a period only after the final segment.
- Every time must be within the video length and start < end.
- Output only the list and stop right after the final period.

Example:
15.5 to 19.4 seconds; 76.0 to 82.7 seconds.
```

G.4 LLM-as-Judge Prompt

The pairwise preference judge (Gemini-3-pro) receives the two videos followed by the prompt below. The judge is queried twice per pair (human-first and AI-first) and the responses are combined to cancel position bias.

LLM-as-Judge Prompt

```
You are an assistant specialized in evaluating the quality of video editing,
particularly in terms of storytelling to convey the message.
Evaluate which video's editing more effectively communicates the message: "{message}"

### Criteria:
- Editing Strategy: How the choice of shots, their sequence, and the pacing create a
visual narrative for the message.
- Storytelling & Narrative Flow: How well the sequence of shots creates a logical and
compelling arc to support the message.
- Informativeness & Visual Detail: The richness of the visual information and how
effectively composition, movement, and subjects provide context.

### Output Format (JSON only):
{
  "answer": "A or B",
  "reason": "Explain the effectiveness of the editing and visual storytelling without
using labels."
}
```

G.5 Message Quality Annotation Prompt

GPT-5.5 receives batches of up to 20 messages through the following system prompt.

Message Quality Annotation Prompt

```
You are a linguistics researcher analyzing short Japanese phrases that describe video
content (e.g. scenes, moments, or subjects depicted in a video, not editing
instructions).
For each phrase, rate two dimensions on a 1-5 integer scale.

ambiguity (abstraction and subjectivity):
  How abstract or subjective are the core concepts in the phrase?
  1 = Entirely concrete and objective. The depicted content is unambiguous (e.g. "a cat
 running around")
  3 = Mix of concrete and abstract or mildly subjective elements
  5 = Highly abstract or subjective. The concept relies on personal interpretation and
many equally valid depictions exist (e.g. "a moment overflowing with love")

contextfulness (dependence on implicit assumptions):
  How much implicit background knowledge, beyond the words themselves, is required to
fully grasp what the phrase refers to?
  1 = Fully self-contained; no outside knowledge needed (e.g. "children playing on a
swing in the park")
  3 = Some domain familiarity helps but the phrase is broadly interpretable
  5 = Heavy reliance on implicit context. Specific domain knowledge, cultural
references, or situational awareness is necessary to understand what is being described
 (e.g. "that exciting moment in the final" presupposes knowing which match and which
moment)

For each phrase return: ambiguity score, ambiguity_reason (one sentence in English),
contextfulness score, contextfulness_reason (one sentence in English).
Return a JSON array with one object per phrase, in the same order as the input.
```

H Dataset Examples

Figure 9 shows representative frame strips from four MEDit-Bench source videos alongside their editing messages and corresponding human-produced cut sequences. Despite sharing the same source footage, different messages yield temporally and visually distinct edits, demonstrating the intent-driven nature of the dataset. A subset of the edited videos is available on our project page (`https://anonymous.4open.science/w/editnet/`).

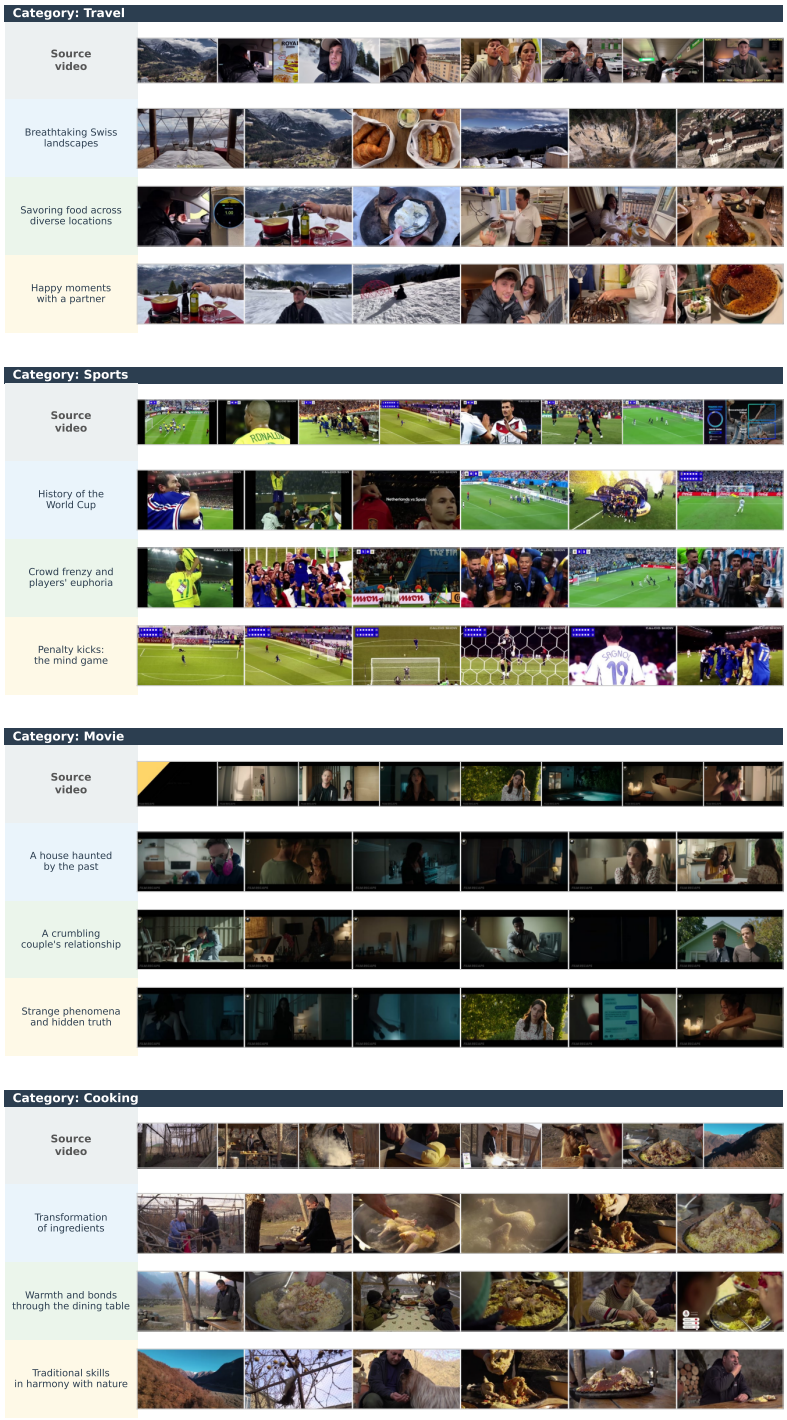

Figure 9: Representative MEDit-Bench examples across four categories (Travel, Sports, Movie, Cooking). For each source video, the top strip shows 8 uniformly sampled frames spanning the full duration (∼14–16 min). The three strips below show frames from a professional editor's cut under each of the three editing messages.