



HIPPO-VIDEO: Simulating Watch Histories with Large Language Models for History-driven Video Highlighting

Jeongeun Lee Youngjae Yu Dongha Lee*

Yonsei University

{ljeadec31, yjy, donalee}@yonsei.ac.kr

Abstract

The exponential growth of video content has made personalized video highlighting an essential task, as user preferences are highly variable and complex. Existing video datasets, however, often lack personalization, relying on isolated videos or simple text queries that fail to capture the intricacies of user behavior. In this work, we introduce HIPPO-VIDEO, a novel dataset for personalized video highlighting, created using an LLM-based user simulator to generate realistic watch histories reflecting diverse user preferences. The dataset includes 2,040 (watch history, saliency score) pairs, covering 20,400 videos across 170 semantic categories. To validate our dataset, we propose HiPHer, a method that leverages these personalized watch histories to predict preference-conditioned segment-wise saliency scores. Through extensive experiments, we demonstrate that our method outperforms existing generic and query-based approaches, showcasing its potential for highly user-centric video highlighting in real-world scenarios.



Dataset: huggingface.co/datasets/jeongeunnn/HIPPO-video



Code: github.com/jeongeunnn-e/HIPPO-Video

1 Introduction

As the scale and diversity of video content rapidly grow in the real world, it becomes increasingly important for users to digest long-form videos efficiently within limited time and resources (Huang et al., 2020; Apostolidis et al., 2021; Argaw et al., 2024a). In this context, various research tasks have emerged to generate shorter, more consumable versions of videos—such as video summarization (Park et al., 2020; Xu et al., 2024), highlight detection, and moment retrieval (Lin et al., 2023; Sun et al., 2024; Xiao et al., 2024; Xu et al., 2024).

However, these tasks often overlook the importance of personalization in the real world where *important* moments vary significantly among users. Tailoring to individual interests can better meet the demand for user-centric content delivery than a one-size-fits-all approach. While some prior works in query-focused video summarization (Vasudevan et al., 2017; Xiao et al., 2020a;b) and moment retrieval (Liu et al., 2018; Zeng et al., 2022) have explored aspects of personalization, they typically reduce user preferences to a single phrase or feature, oversimplifying the complexity of human interest. In reality, human preferences are multifaceted, evolving over time and across different types of content. To address this, we propose leveraging watch history as a richer source of user preference modeling. We contend that analyzing users’ sequential viewing behavior through their watch histories can uncover underlying preferences, leading to more effective and tailored video experiences.

In this work, we introduce **personalized video highlighting**, a novel task that leverages a user’s watch history within a single session to tailor video highlights to the user’s preferences. Inspired by how recommender systems effectively capture user interests through implicit feedback, such as interaction history (Rendle et al., 2009; Kang & McAuley, 2018), our task aims to dynamically select and present highlight segments aligned with the user’s

*Corresponding Author

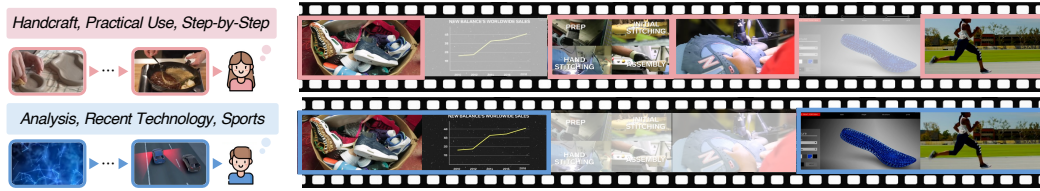


Figure 1: A video can produce varying highlights based on user interests, showing how watch history reflects implicit feedback and helps tailor highlights to individual preferences.

real-time viewing behavior and preferences during the session. For instance, as shown in Figure 1, the same video may yield distinct highlights depending on the user’s focus inferred from their watch history, emphasizing different aspects of content.

For this task, we introduce HiPPO-VIDEO: **H**ighlights Based on **P**references for **P**ersonalized **V**ideo **C**lipping, a large-scale dataset containing user watch histories and corresponding personalized saliency scores, generated by simulating real-world user behavior on video platforms. Existing video datasets (Gygli et al., 2014; Song et al., 2015; Sharghi et al., 2016) are often limited in scale due to resource-intensive nature of manual annotation, while collecting actual users’ watch histories raises privacy concerns. To address these challenges, we leverage large language models (LLMs) to simulate user interactions, enabling scalable data generation without compromising user privacy. HiPPO-VIDEO consists of 2,040 (watch history, saliency score) pairs, where each watch history comprises 10 videos, thereby totaling 20,400 videos, across 170 semantic initial preference seeds.

Through experiments, we validate our task and dataset using a simple baseline, **H**istory-Driven **P**reference-Aware Video **H**ighlighter, named **HiPHer**, which leverages user preferences derived from watch history as preference context. HiPHer outperforms existing methods by incorporating personalized preference embeddings from watch histories, while generic methods often fail to align with individual user interests, and query-focused methods struggle to capture the complexity of preferences with short queries. These results underscore the importance of incorporating detailed user histories to enhance user-specific video highlighting, demonstrating the effectiveness of history-driven preference modeling.

2 Related Work

Tasks and Datasets. Highlight detection identifies the most engaging or significant moments within a video by assigning importance scores to segments. Existing datasets (Sun et al., 2014; Song et al., 2016; Gygli et al., 2016; Sul et al., 2023) provide query-unrelated highlight clips. Moment retrieval locates specific time spans within videos that match a given natural language query, using datasets (Lei et al., 2020; Gao et al., 2017; Lei et al., 2021; Zala et al., 2023) that pair queries with annotated moments. Video summarization provides condensed versions of videos, preserving essential narrative or informational content. Traditional datasets (Gygli et al., 2014; Song et al., 2015; Sharghi et al., 2016) rely heavily on human annotations, limiting their scalability. Recent work has thus explored automated summarization using large language models (LLMs) (Argaw et al., 2024b; Hua et al., 2024). Notably, several datasets are utilized across multiple tasks (Song et al., 2015; Sul et al., 2023; Lei et al., 2021). Detailed comparisons of these datasets are provided in Table 1.

Methods. Prior work on highlight detection has mainly explored ranking-based methods that assign scores to video segments to identify highlights (Sun et al., 2014; Gygli et al., 2016; Yao et al., 2016; Rochan et al., 2020). For moment retrieval, research has centered on cross-modal alignment techniques to bridge textual queries and visual content (Lu et al., 2019; Yuan et al., 2019; Zhang et al., 2020; Lei et al., 2020). More recently, inspired by the success of DETR (Carion et al., 2020), DETR-based methods have been proposed to jointly tackle both moment retrieval and highlight detection in a unified framework (Lei et al., 2021; Moon et al., 2023; Liu et al., 2022). On the other hand, video summarization selects

Dataset	Statistics		Supported Tasks	Single Instance		Anno.
	#Videos	Avg Len(m)		Query	#Videos	
YouTubeHighlights (Sun et al., 2014)	600	2.4	MR, HD	✗	1	M
SumMe (Gygli et al., 2014)	25	2.4	VS	✗	1	M
TVSum (Song et al., 2015)	50	3.9	VS	✗	1	M
QFVS (Sharghi et al., 2016)	4	240	VS, MR	✓	1	M
Charades-STA (Gao et al., 2017)	6,700	0.5	MR	✓	1	M
TVR (Lei et al., 2020)	21,800	1.3	MR	✓	1	M
QVHighlights (Lei et al., 2021)	10,200	2.5	MR, VS	✓	1	M
Mr.HiSum (Sul et al., 2023)	31,892	3.4	VS, HD	✗	1	M
Shot2Story20K (Han et al., 2023)	20,023	0.3	VS	✗	1	M+S
Instruct-V2Xum (Hua et al., 2024)	30,000	3.1	VS	✗	1	M+S
LfVS (Argaw et al., 2024b)	1,200	12.2	VS	✗	1	M+S
 HiPPO-VIDEO	2,040(20,400)	13.9	VS, MR, HD, PV	✓	10	M+S

Table 1: Benchmark dataset comparison across tasks: **VS** (Video Summarization), **MR** (Moment Retrieval), **HD** (Highlight Detection), and **PV** (Personalized Video Highlighting). **M** denotes manually curated datasets, while **S** refers to those synthesized by models.

key moments to represent overall content (Ji et al., 2019; Argaw et al., 2024b). When a query is given (Sharghi et al., 2016; 2017; Narasimhan et al., 2021), it becomes query-focused summarization, similar to moment retrieval in aligning video content with textual input.

3 HiPPO-VIDEO

We introduce HiPPO-VIDEO, a large-scale dataset designed for personalized video highlighting. The dataset consists of (1) user watch history sequences and (2) 10-point saliency scoring annotations for target video. Each sequence consists of 10 videos and the dataset includes 2,040 sequences, resulting in a total of 20,400 videos across a variety of categories.

3.1 Simulation

Collecting real user watch histories from video platforms presents significant challenges, including privacy concerns and resource constraints. To address these limitations, we employ LLM-based user simulator¹ to generate realistic, large-scale video watch history sequences. Figure 2 provides an overview of the watch history simulation process, and detailed prompts are included in Appendix A.3.

Starting from an initial profile seed, the simulator operates iteratively, dynamically updating user preferences as it *watches* videos. Specifically, the process consists of three steps : (1) video candidate retrieval, (2) video engagement, and (3) preference update. This iterative framework enables the simulator to capture the evolving nature of real user preferences, effectively modeling the complexity and diversity of real-world video consumption.²

Initialization. To support diversity in simulating user behavior, we initialize simulators with carefully designed variables representing user interests. These variables follow the categorization from Qiu et al. (2024), comprising 170 topic and sub-topic pairs adapted from existing video datasets and popular Wikipedia topics (Zhou et al., 2018; Miech et al., 2019), capturing the breadth of content on YouTube. Additionally, we introduce a sentiment-based variable (*intent*) to model user motivations and viewing preferences. By integrating topic categorization and intent-informed preferences, we construct 2,040 profiles as initial seeds for personalized watch history simulation, contributing to adaptability across diverse users and video content. Details on initialization variables are provided in the Appendix A.1.

Video Candidate Retrieval. The simulation begins with retrieving a set of video candidates, denoted as $\mathcal{C} = \{C_1, C_2, \dots, C_l\}$, by crawling YouTube in real time. In real-world viewing

¹Hereafter, we refer to it as “the simulator” for brevity.

²In our simulation, we set the number of video candidates per turn to $l = 8$ and fix the number of watched videos in history to $m = 10$.

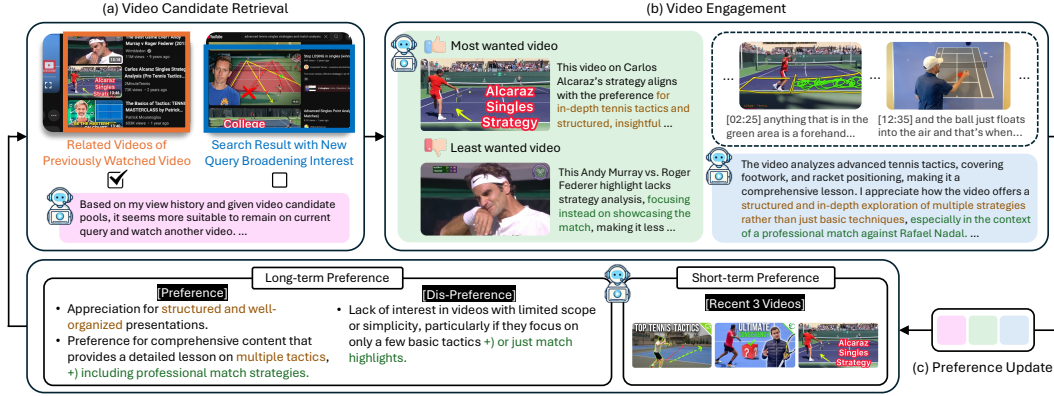


Figure 2: The overall process of our LLM-based user simulation to collect video watch histories on a video platform operates iteratively as follows: (1) retrieving video candidates either from related videos or through a new query, (2) engaging with videos, including selection and viewing, and (3) updating long-term preferences based on the simulator’s key responses obtained during video retrieval and engagement.

sessions, users typically either continue exploring within a topic or shift (or expand) to new topics. To model this behavior, the simulator is given two options: (1) exploring related videos or (2) generating a new search query. Specifically, at the i -th turn (i.e., i -th video selection), this decision is informed by previously watched videos, $H_{i-1} = \{H_1, H_2, \dots, H_{i-1}\}$, along with the user’s current preference, p_{i-1} . This approach enables the simulator to simulate natural browsing patterns, balancing topic continuity and exploration.

Video Engagement. Once the candidate pool is retrieved, the simulator selects a video to watch and engages with its content. First, the selection process accounts for two types of user preferences: *short-term* and *long-term*. Short-term preferences are based on the metadata³ of 3 most recently watched videos, while long-term preferences, denoted as p_{i-1} , are expressed as explicit likes and dislikes in natural language, providing an accumulated profile of the user’s overall interests over the video sequence H_{i-1} . To refine preference modeling, the simulator selects both the most wanted video and the least wanted one. This contrastive approach enhances user modeling by building a more fine-grained representation of preference, balancing both likes and dislikes. Additionally, the simulator provides reasoning for its selections (green box in Figure 2), reinforcing the decision-making process.

Once the most preferred video $C \in \mathcal{C}$ is determined, the simulator proceeds to *watch* it. The video is segmented into $C = \{s_1, s_2, \dots, s_n\}$ using scene change detection,⁴ ensuring each segment forms a coherent unit of content. Each segment is represented as $s_k = (v_k, t_k)$, where v_k is visual description and t_k is the corresponding transcript. To facilitate comprehensive video understanding for LLM (Wang et al., 2024), the representative frame f_k in segment s_k is converted into textual description v_k via frame captioning with Liu et al. (2024). Using this multimodal input, the simulator generates a *review*, which includes a concise summary and tailored opinion on the video, aligned with the evolving preferences p_{i-1} (blue box in Figure 2). This entire process—*selecting* and *watching* a video—replicates the human process of interacting with content, guided by both recent interactions and long-term interests.

Preference Update. After completing video engagement, the simulator updates its preference state from p_{i-1} to p_i . During the engagement, the simulator generates three key responses based on preference reasoning: (1) the rationale for selecting the most preferred video, (2) the rationale for selecting the least preferred video, and (3) a review of the watched video. These are then used to refine long-term preferences, dynamically adjusting based

³Metadata includes information identical to that displayed on YouTube, such as the title, channel, thumbnail, and view count. Detailed examples are provided in the Appendix A.2.

⁴<https://github.com/Breakthrough/PySceneDetect>

on recent interactions. In Figure 2, additional details inferred from the simulator’s key responses (highlighted in brown and green) are incorporated into the long-term preference.

3.2 Saliency Score Annotation

After simulation, the last video in each watch history is set as the target video for saliency annotation. Similar to the video engagement process, the video is segmented by detecting scene changes. The simulator then assigns relevance scores ranging from 1 to 10 to each segment. These scores are determined based on two primary sources of information: (1) final long-term preferences, consolidated over the course of watching the videos, and (2) personal reviews generated each time after watching the video. The review-driven preferences offer video-specific signals, while the long-term preferences reflect broader interests across entire session. By integrating these two preference layers, the simulator establishes session-based user inclinations, enabling the segment scoring that aligns with inferred interests.

3.3 Human Verification

Validation of Watch History Simulation Process. To evaluate the reliability of our watch history simulation, we employ Amazon Mechanical Turk (MTurk) annotators to assess two key aspects of the framework: (1) query generation and (2) video selection. Annotators are given the same preference information as the LLM-based user simulator, including previously watched videos and long-term user preferences. For query generation, annotators assess whether queries written by the simulator are plausible for next steps. Results show that 97.56% of queries are reasonable, with 85% inter-annotator agreement. For video selection, annotators are given a set of candidate videos, identical to the simulator’s pool, and asked to select the one that best aligns with the provided preferences. The simulator’s choices match human selections in 71.42% of cases, suggesting that it effectively mirrors real user behavior. More details on the evaluation process can be found in Appendix A.4.

Validation of Saliency Annotations. To validate the saliency annotations generated by the simulator, we conduct a user study adapting the methodology from Sul et al. (2023). MTurk annotators are given a video, a user preference, and the clip assigned the highest saliency score (or multiple pairs if there are tied clips). For each pair, the annotators determine whether highlighted clip aligns with the given preference by selecting one of three options: Agree (A), Unclear (U), or Disagree (D). In Table 2, the A, U, and D columns represent the number of annotators who selected each option. The results show that nearly 98% of pairs are deemed reasonable by majority agreement, confirming that the saliency scores accurately capture personalized preferences.

Agreement	A	U	D	Percentage
Agree	3	0	0	64.10%
	2	1	0	15.38%
	2	0	1	17.95%
Neutral	1	2	0	2.56%

Table 2: User agreement results

Validation of Simulated Watch Histories. To further verify the realism of our simulated watch histories, we conduct complementary evaluations using 40 real user histories (10 videos each), collected with informed consent.⁵ First, following recent LLM-as-a-judge protocols (Chiang et al., 2024; Mitchell et al., 2023; Luo et al., 2025), we task GPT-4 (Achiam et al., 2023) with binary classification to distinguish simulated from real histories initialized with the same profile seed. GPT-4 achieves only 40% accuracy, below the 50% random baseline, indicating that the simulated histories are often indistinguishable from real ones. Second, we apply Fast-DetectGPT (Bao et al., 2023) in a Hit@1 setting, where the model must identify one simulated history from a set of nine real ones. It achieves a Hit@1 score of 0.350, indicating substantial confusion and further supporting the similarity between simulated and real histories. Together, these results strongly support the validity of our simulation framework as a reliable proxy for real-world user watch histories.

⁵We refer to this dataset as HiPPO-VIDEO⁺. Human annotators recorded their watch histories and annotated the final video at the segment level, following the procedure in Section 3.2. This dataset is also used for evaluating baselines in Section 5.3.

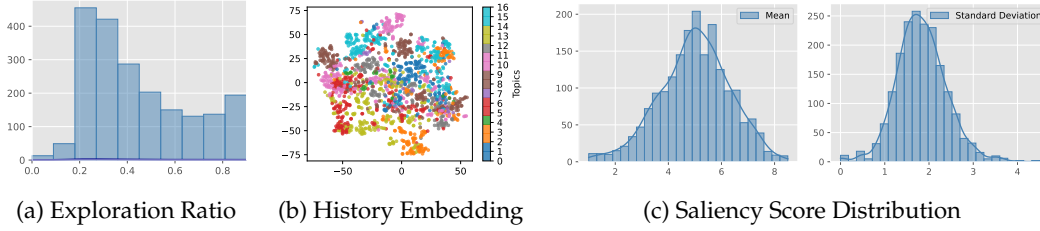


Figure 3: Dataset analysis results. (a–b) Exploration patterns and watch history embeddings visualized via t-SNE. (c) Distribution of saliency score means and standard deviations.

3.4 Dataset Analyses

We analyze key aspects of our dataset, including its overall characteristics and diversity, as shown in Figure 3, with detailed analysis settings provided in Appendix A.4.

Overall Statistics. HIPPO-VIDEO is thoroughly curated from real-time crawling, with all videos in the watch history sequences spanning from 2008 to 2024. Of these, 57.16% were published after 2023, ensuring the dataset remains up-to-date. Video durations range from 30 seconds to 119 minutes, with an average length of 13.9 minutes, reflecting typical video consumption. For annotation, target videos are divided into an average of 56.91 segments.

Intra-history Video Diversity. We analyze the *exploration ratio* within a video watch history to measure how actively the simulator broadens its interests. When the simulator chooses to watch related videos or repeats a similar query to previous ones, this is considered as non-exploration. In contrast, when the simulator generates a distinctly new query, we define this as *topic drift*, signifying an expansion of interest. As shown in Figure 3a, the exploration rate generally ranges from 0.2 to 0.6, indicating a wide spectrum of behavioral patterns among simulators—some maintaining consistent, focused interests, while others frequently shift topics and explore new content areas.

Inter-history Video Diversity. To assess the diversity of user preferences captured through simulation, we visualize the embedding space of video watch histories using t-SNE, with embeddings generated by CLIP (Radford et al., 2021). As shown in Figure 3b, the embeddings do not form tight clusters or align strictly with their initial topics. This indicates that the simulated watch histories encompass a wide range of user preferences, even when originating from predefined topics (as explained in Section 3.1 Initialization).

Saliency Score Distribution. Figure 3c shows the distribution of saliency scores with kernel density estimation (KDE). The mean saliency scores (left) represent the average segment scores per video, with most falling between 4 and 6, indicating moderate relevance. To assess fluctuations, we measure the standard deviation (right), which typically ranges from 1.5 to 2, suggesting moderate variability. Higher deviations (greater than 3) indicate substantial fluctuations, likely due to dynamic visual changes or frequent scene transitions.

4 HiPHer: History-driven Preference-aware Video Highlighter

We propose HiPHer, which generates personalized segment-wise saliency scores by modeling user preferences from their watch history. Given a video V uniformly divided into n segment, and a watch history consisting of m videos, $\mathcal{H} = \{H_1, H_2, \dots, H_m\}$, the objective is to predict saliency scores $Y = \{y_1, y_2, \dots, y_n\}$, where each y_k quantifies the relevance of the k -th segment to the user preferences. HiPHer derives a global preference embedding from the watch history to guide segment representations via cross-attention, producing relevance scores optimized with a contrastive loss to prioritize segments aligned with user interests.

Input Representations. For each of the n segments of V , we denote the representative frames as $\{f_1, f_2, \dots, f_n\}$, and the corresponding transcripts as $\{t_1, t_2, \dots, t_n\}$. The tran-

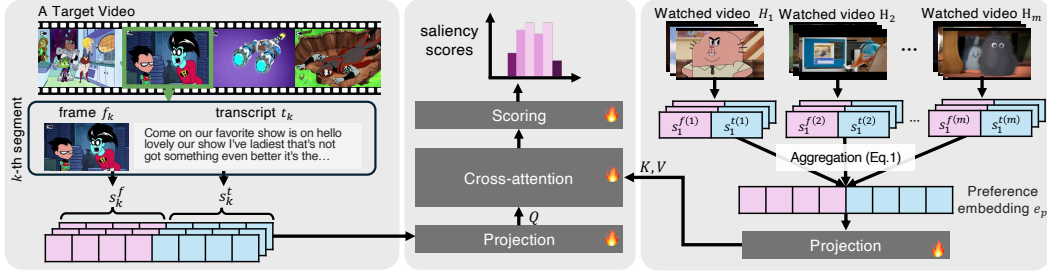


Figure 4: The architecture of HiPher consists of two modules: (1) a preference modeling module that generates a preference embedding from watched videos, and (2) a scoring module that assigns a preference score to each segment in the target video.

scripts are generated from audio using Audio Speech Recognition (ASR), as ASR has shown to enhance visual recognition tasks (Li et al., 2020). We employ the pre-trained CLIP image encoder (ViT-B/32) (Radford et al., 2021) to generate visual features $\{s_1^f, s_2^f, \dots, s_n^f\}$ for each frame. Similarly, we use the CLIP text encoder to convert the transcripts into textual features $\{s_1^t, s_2^t, \dots, s_n^t\}$. Since visual and textual features may carry distinct semantic information, we concatenate them for each segment $s_k = s_k^f \oplus s_k^t$, where s_k^f and s_k^t represent the visual and textual features, respectively, rather than directly fusing them (Kamath et al., 2021).

Preference Modeling from Watch History. HiPher first constructs a preference embedding, denoted as e_p , to encapsulate preferences inferred from a sequence of previously watched videos $\mathcal{H}$. Each video H_i in the watch history is encoded by aggregating its segment features $\{s_1^{(i)}, s_2^{(i)}, \dots, s_n^{(i)}\}$ using $\mathcal{Agg}_s$, where each segment is encoded similarly to the target video representations. This results in a single embedding $h^{(i)}$, which serves as a compact representation of the entire video. The video embeddings, $\{h^{(1)}, h^{(2)}, \dots, h^{(m)}\}$, are then aggregated into a global preference representation using $\mathcal{Agg}_h$, as follows:

$$e_p = \mathcal{Agg}_h \left(\{h^{(i)} : h^{(i)} = \mathcal{Agg}_s(s_1^{(i)}, \dots, s_n^{(i)})\}_{i=1}^m \right) \quad (1)$$

In this work, we use mean pooling as the aggregation function for $\mathcal{Agg}_h$ and $\mathcal{Agg}_s$, making e_p reflect the average characteristics of the watched videos. More advanced techniques can adjust each video’s weight based on its relevance to the target video or the user’s interests.

Segment-wise Scoring. The input representations are first processed through projection layers, each consisting of 3 sequential layers of LayerNorm and dropout. Similarly, the preference embeddings pass through the same structure of projection layers, ensuring alignment in a shared embedding space. Next, a cross-attention layer uses the input representations as queries and the preference embeddings as keys and values to condition segment representations based on preferences. Similar to (Lei et al., 2021; Narasimhan et al., 2021), the attended output is then fed into a transformer encoder, which includes a multi-head self-attention layer and a feed forward network (FFN), to compute segment-wise saliency scores that capture the relevance of each segment based on the modeled preferences.

Saliency Loss. We employ a contrastive saliency loss to ensure relevant clips receive higher saliency scores and irrelevant ones lower scores, enforcing a ranking based on user preferences. Given a target video with segments v^+ (relevant) and v^- (irrelevant), and their corresponding saliency scores y^+ and y^- , the loss is defined as:

$$\mathcal{L}_{\text{saliency}} = \sum_{(v^+, v^-)} \max(0, \gamma - (y^+ - y^-)) \quad (2)$$

If the difference between y^+ and y^- is smaller than γ , the loss function penalizes the model, encouraging it to assign a higher score to the relevant segment.

Method	RMSE ↓	mAP	Hit1@7	Hit1@9	Recall1@0.5	Recall1@0.7	Improv.
QVHighlights							
Moment-DETR (Lei et al., 2021)	0.347	0.681	0.434	0.042	<u>0.370</u>	0.205	20.7%
UMT (Liu et al., 2022)	0.527	0.547	0.409	<u>0.138</u>	0.255	0.179	20.2%
QD-DETR (Moon et al., 2023)	0.375	0.675	0.406	0.116	0.353	0.201	43.4%
UVCOM (Xiao et al., 2024)	0.330	0.710	0.489	0.149	0.413	0.183	11.4%
TR-DETR (Sun et al., 2024)	0.400	0.660	0.352	0.105	0.359	0.195	58.1%
HiPPO-VIDEO							
SL-Module (Xu et al., 2021)	0.517	0.568	0.385	0.085	–	–	96.1%
Moment-DETR (Lei et al., 2021)	<u>0.339</u>	0.705	0.432	<u>0.138</u>	0.398	0.193	38.2%
UMT (Liu et al., 2022)	0.502	<u>0.732</u>	0.429	0.132	0.320	<u>0.210</u>	6.4%
QD-DETR (Moon et al., 2023)	0.368	0.681	<u>0.456</u>	0.120	0.365	0.196	38.2%
UVCOM (Xiao et al., 2024)	0.350	0.700	0.441	0.146	0.357	0.154	13.7%
TR-DETR (Sun et al., 2024)	0.390	0.660	0.435	0.149	0.243	0.127	11.4%
HiPHer	0.301	0.766	0.507	0.166	0.452	0.245	

Table 3: Performance comparison for HD and MR. Hit1@ k and Recall@ α are computed using saliency threshold k and IoU threshold α , respectively. Gray rows indicate training datasets. The best results are shown in **bold**, and the second-best are underlined.

5 Experiments

5.1 Experimental Settings

In this section, we compare our task, personalized video highlighting (PV), with existing methods in video summarization (VS), highlight detection (HD), and moment retrieval (MR). VS and HD use only the target video to select keyframes, while MR takes a natural language query as well to retrieve a matching temporal segment. In contrast, PV uses both a user’s watch history and the target video to predict segment saliency scores.

Experimental Setup. We split HiPPO-VIDEO into training (70%) and test (30%) sets, ensuring a balanced ratio of videos across categories for content diversity. For MR and query-focused VS, we generate text-based queries by extracting key phrases that capture the essence of the user’s watch history. Additionally, we train on QVHighlights (Lei et al., 2021), a widely-used dataset for HD and MR, and evaluate on HiPPO-VIDEO. This setup assesses the generalization ability of models across different datasets, though HiPPO-VIDEO’s unique requirement for video history sequences limits direct cross-dataset generalization.

Baselines. We evaluate recent state-of-the-art methods for personalized video highlighting. For HD and MR, we include transformer-based models—SL-Module (Xu et al., 2021), UMT (Liu et al., 2022), and UVCOM (Xiao et al., 2024)—as well as DETR-based approaches: Moment-DETR (Lei et al., 2021), QD-DETR (Moon et al., 2023), and TR-DETR (Sun et al., 2024). Note that SL-Module is applied only to HD. For VS, we adapt CLIP-It (Narasimhan et al., 2021) for generic and query-focused summaries, and VSL (Chen et al., 2024) for personalized summaries. More details are provided in Appendix B.1.

Evaluation Metrics. We evaluate model performance using standard metrics used in baselines (Lei et al., 2021; Sul et al., 2023; Moon et al., 2023). For HD, we use mean average precision (mAP) and Hit@1 to assess ranking quality, with saliency score thresholds of 7 and 9 (out of 10).⁶ For MR, we compute Recall@1 with IoU thresholds of 0.5 and 0.7 to measure temporal alignment accuracy. For VS, we use F1 score to assess the balance between precision and recall in segment selection. Furthermore, since our task includes score prediction, we use Root Mean Square Error (RMSE) to evaluate segment relevance prediction accuracy. More details on evaluation metrics can be found in Appendix B.2.

⁶Liu et al. (2015); Lei et al. (2021); Moon et al. (2023) set the threshold as 4 out of 5.

Method	Query Type	F1@5	F1@7
Clip-It (Narasimhan et al., 2021)	-	0.564	0.211
Clip-It (Narasimhan et al., 2021)	phrase	0.566	0.230
Clip-It (Narasimhan et al., 2021)	sentence	0.658	0.234
VSL (Chen et al., 2024)	genre	0.466	0.187
HiPHer	history	0.726	0.486

Table 5: Performance comparison for video summarization. CLIP-It supports both generic (query-free) and query-focused summarization, while VSL provides personalized summarization based on preference queries (genres).

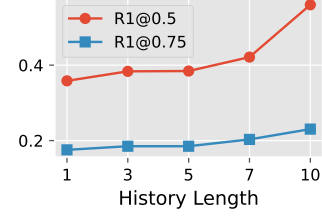


Figure 5: Performance comparison with varying numbers of history videos used for preference modeling.

5.2 Main Results

Table 3 shows that HiPHer outperforms all baselines over evaluation metrics, highlighting its effectiveness in capturing user-specific preferences. This performance gain is primarily attributed to the incorporation of personalized understanding through watch histories. As a generic approach, SL-Module effectively identifies informative segments but fails to reflect a user’s unique preferences. This limitation emphasizes the challenges of applying non-personalized methods in a personalized setting. While UMT, Moment-DETR, and QD-DETR, which leverage natural language queries, achieve better performance than generic baselines, they struggle to capture finer-grained moments. This might be because natural language queries provide a simplified representation of user intent compared to the richer contextual signals available in watch history. On the other hand, UMT tends to show competitive performance with ours, since UMT and ours use additional audio sources; this strongly indicates the importance of incorporation of multi-modal source.

5.3 Additional Results on HiPPO-VIDEO⁺

We further evaluate HiPHer and the MR/HD baselines on HiPPO-VIDEO⁺, a dataset collected from real users, to assess the practicality and generalizability of each method. As summarized in Table 4, HiPHer consistently outperforms the baselines across most metrics, showing strong robustness beyond simulated settings. These findings highlight the effectiveness of HiPHer in modeling nuanced viewing behaviors and suggest promising directions for future research in user-adaptive video understanding.

Method	RMSE	H1@7	H1@9	F1@0.5
Moment-DETR	0.419	0.472	0.389	0.417
QD-DETR	0.446	0.444	0.361	0.385
TR-DETR	0.443	0.306	0.250	0.429
HiPher	0.427	0.486	0.400	0.624

Table 4: Performance on HiPPO-VIDEO⁺.

5.4 Ablation Studies

Query Type. Table 5 reports summarization accuracy (F1 score at different thresholds) across various preference contexts: simple word-level queries, sentence-level descriptions, and user watch histories. HiPHer performs the best when leveraging history, significantly outperforming word- and sentence-based representations. These results emphasize the critical role of history-driven preference modeling for effective personalized video highlighting.

History Length. We hypothesize that user preferences, which are often highly specific, can be more accurately captured by analyzing longer watch histories, as they reveal consistent patterns across a sequence. To validate this, we conduct an ablation study by varying the number of watched videos (i.e., the length of the watch history) provided to the preference modeling module. As shown in Figure 5, performance improves as more history videos are included. These results suggest that longer histories help surface repetitive cues, leading to more effective preference modeling and improved personalization in video engagement.

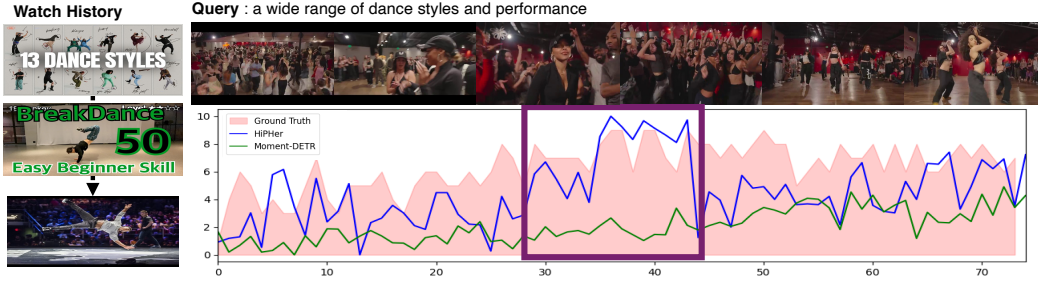


Figure 6: Case study on saliency (preference) scores between HiPHer and Moment-DETR.

Input Modalities. We conduct an ablation study to evaluate the contributions of visual and textual features. As shown in Table 6, using a single modality leads to reduced performance, with textual features (HiPHer-V) being more informative than visual ones (HiPHer-T). Combining both modalities (HiPHer) achieves the best results, demonstrating the effectiveness of our multi-modal approach in capturing fine-grained user preferences for personalized video highlighting. This is particularly important given the diversity of HIPPO-VIDEO, which reflects the range of real-world videos, including both visually rich and narrative-only content.

Method	mAP	H1@7	R1@0.5
HiPHer-V	0.67	0.12	0.32
HiPHer-T	0.74	0.15	0.39
HiPHer	0.77	0.17	0.45

Table 6: Ablation results on different input modalities.

Case Study. We present a case study that qualitatively compares the saliency (preference) scores of HiPHer and Moment-DETR. Figure 6 visualizes the segment-wise scores within a target video, contrasting a history-focused embedding approach with a query-based method. Overall, the blue line (HiPHer) closely aligns with the ground truth scores, while the green line (Moment-DETR) sometimes shows notable discrepancies (highlighted in the purple box), indicating that the history-driven embedding provides richer contextual information for personalization to a text query. Additional case studies are provided in Appendix B.4.

6 Conclusion

Motivated by the need to tailor video content to individual preferences in real-world scenarios, we introduce personalized video highlighting, a novel task that leverages user watch history to highlight relevant video segments. We also present HIPPO-VIDEO, a unique dataset generated through LLM-based user simulation, which includes user watch histories and personalized saliency scores. Through comprehensive experiments, we demonstrate that history-driven preference modeling significantly improves performance, surpassing existing methods based on generic or text-based queries. Our findings emphasize the value of integrating user-specific preferences and history for more effective video content delivery, offering promising directions for future advancements in personalized video experiences.

Acknowledgement

This work was supported by the IITP grants funded by the Korea government (MSIT) (No. RS-2020- II201361; RS-2024-00457882, AI Research Hub Project), and the NRF grant funded by the Korea government (MSIT) (No. RS-2025-00560295).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

- Evlampios Apostolidis, Eleni Adamantidou, Alexandros I Metsai, Vasileios Mezaris, and Ioannis Patras. Video summarization using deep neural networks: A survey. *Proceedings of the IEEE*, 109(11):1838–1863, 2021.
- Dawit Mureja Argaw, Mattia Soldan, Alejandro Pardo, Chen Zhao, Fabian Caba Heilbron, Joon Son Chung, and Bernard Ghanem. Towards automated movie trailer generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7445–7454, 2024a.
- Dawit Mureja Argaw, Seunghyun Yoon, Fabian Caba Heilbron, Hanieh Deilamsalehy, Trung Bui, Zhaowen Wang, Franck Deroncourt, and Joon Son Chung. Scaling up video summarization pretraining with large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8332–8341, 2024b.
- Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *arXiv preprint arXiv:2310.05130*, 2023.
- Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.
- Brian Chen, Xiangyuan Zhao, and Yingnan Zhu. Personalized video summarization by multimodal video understanding. In *CIKM*, 2024.
- Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning*, 2024.
- Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, pp. 5267–5275, 2017.
- Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part VII 13*, pp. 505–520. Springer, 2014.
- Michael Gygli, Yale Song, and Liangliang Cao. Video2gif: Automatic generation of animated gifs from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1001–1009, 2016.
- Mingfei Han, Linjie Yang, Xiaojun Chang, and Heng Wang. Shot2story20k: A new benchmark for comprehensive understanding of multi-shot videos. *arXiv preprint arXiv:2312.10300*, 2023.
- Hang Hua, Yunlong Tang, Chenliang Xu, and Jiebo Luo. V2xum-llm: Cross-modal video summarization with temporal prompt instruction tuning. *arXiv preprint arXiv:2404.12353*, 2024.
- Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pp. 709–727. Springer, 2020.
- Zhong Ji, Kailin Xiong, Yanwei Pang, and Xuelong Li. Video summarization with attention-based encoder–decoder networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(6):1709–1717, 2019.
- Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1780–1790, 2021.

- Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pp. 197–206. IEEE, 2018.
- Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pp. 447–463. Springer, 2020.
- Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34:11846–11858, 2021.
- Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. *arXiv preprint arXiv:2005.00200*, 2020.
- Kevin Qinghong Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Jinpeng Wang, Rui Yan, and Mike Zheng Shou. Univtg: Towards unified video-language temporal grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2794–2804, 2023.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.
- Meng Liu, Xiang Wang, Liqiang Nie, Xiangnan He, Baoquan Chen, and Tat-Seng Chua. Attentive moment retrieval in videos. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 15–24, 2018.
- Wu Liu, Tao Mei, Yongdong Zhang, Cherry Che, and Jiebo Luo. Multi-task deep visual-semantic embedding for video thumbnail selection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3707–3715, 2015.
- Ye Liu, Siyuan Li, Yang Wu, Chang-Wen Chen, Ying Shan, and Xiaohu Qie. Umt: Unified multi-modal transformers for joint video moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3042–3051, 2022.
- Chujie Lu, Long Chen, Chilie Tan, Xiaolin Li, and Jun Xiao. Debug: A dense bottom-up grounding approach for natural language video localization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5144–5153, 2019.
- Ziyang Luo, Haoning Wu, Dongxu Li, Jing Ma, Mohan Kankanhalli, and Junnan Li. Videoautoarena: An automated arena for evaluating large multimodal models in video analysis through user simulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 8461–8474, 2025.
- Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2630–2640, 2019.
- Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pp. 24950–24962. PMLR, 2023.
- WonJun Moon, Sangeek Hyun, SangUk Park, Dongchan Park, and Jae-Pil Heo. Query-dependent video representation for moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23023–23033, 2023.
- Medhini Narasimhan, Anna Rohrbach, and Trevor Darrell. Clip-it! language-guided video summarization. *Advances in neural information processing systems*, 34:13988–14000, 2021.

- Jungin Park, Jiyoung Lee, Ig-Jae Kim, and Kwanghoon Sohn. Sumgraph: Video summarization via recursive graph modeling. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV* 16, pp. 647–663. Springer, 2020.
- Jielin Qiu, Jiacheng Zhu, William Han, Aditesh Kumar, Karthik Mittal, Claire Jin, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Ding Zhao, et al. Mmsum: A dataset for multimodal summarization and thumbnail generation of videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 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 *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.
- Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pp. 452–461, 2009.
- Mrigank Rochan, Mahesh Kumar Krishna Reddy, Linwei Ye, and Yang Wang. Adaptive video highlight detection by learning from user history. In *European conference on computer vision*, pp. 261–278. Springer, 2020.
- Aidean Sharghi, Boqing Gong, and Mubarak Shah. Query-focused extractive video summarization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VIII* 14, pp. 3–19. Springer, 2016.
- Aidean Sharghi, Jacob S Laurel, and Boqing Gong. Query-focused video summarization: Dataset, evaluation, and a memory network based approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4788–4797, 2017.
- Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5179–5187, 2015.
- Yale Song, Miriam Redi, Jordi Vallmitjana, and Alejandro Jaimes. To click or not to click: Automatic selection of beautiful thumbnails from videos. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pp. 659–668, 2016.
- Jinhwan Sul, Jihoon Han, and Joonseok Lee. Mr. hisum: A large-scale dataset for video highlight detection and summarization. *Advances in Neural Information Processing Systems*, 36:40542–40555, 2023.
- Hao Sun, Mingyao Zhou, Wenjing Chen, and Wei Xie. Tr-detr: Task-reciprocal transformer for joint moment retrieval and highlight detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 4998–5007, 2024.
- Min Sun, Ali Farhadi, and Steve Seitz. Ranking domain-specific highlights by analyzing edited videos. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I* 13, pp. 787–802. Springer, 2014.
- Arun Balajee Vasudevan, Michael Gygli, Anna Volokitin, and Luc Van Gool. Query-adaptive video summarization via quality-aware relevance estimation. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 582–590, 2017.
- Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision*, pp. 58–76. Springer, 2024.
- Shuwen Xiao, Zhou Zhao, Zijian Zhang, Ziyu Guan, and Deng Cai. Query-biased self-attentive network for query-focused video summarization. *IEEE Transactions on Image Processing*, 29:5889–5899, 2020a.

- Shuwen Xiao, Zhou Zhao, Zijian Zhang, Xiaohui Yan, and Min Yang. Convolutional hierarchical attention network for query-focused video summarization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 12426–12433, 2020b.
- Yicheng Xiao, Zhuoyan Luo, Yong Liu, Yue Ma, Hengwei Bian, Yatai Ji, Yujiu Yang, and Xiu Li. Bridging the gap: A unified video comprehension framework for moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18709–18719, 2024.
- Minghao Xu, Hang Wang, Bingbing Ni, Riheng Zhu, Zhenbang Sun, and Changhu Wang. Cross-category video highlight detection via set-based learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7970–7979, 2021.
- Yifang Xu, Yunzhuo Sun, Benxiang Zhai, Youyao Jia, and Sidan Du. Mh-detr: Video moment and highlight detection with cross-modal transformer. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2024.
- Ting Yao, Tao Mei, and Yong Rui. Highlight detection with pairwise deep ranking for first-person video summarization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 982–990, 2016.
- Yitian Yuan, Lin Ma, Jingwen Wang, Wei Liu, and Wenwu Zhu. Semantic conditioned dynamic modulation for temporal sentence grounding in videos. *Advances in Neural Information Processing Systems*, 32, 2019.
- Abhay Zala, Jaemin Cho, Satwik Kottur, Xilun Chen, Barlas Oguz, Yashar Mehdad, and Mohit Bansal. Hierarchical video-moment retrieval and step-captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23056–23065, 2023.
- Yawen Zeng, Da Cao, Shaofei Lu, Hanling Zhang, Jiao Xu, and Zheng Qin. Moment is important: Language-based video moment retrieval via adversarial learning. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 18(2): 1–21, 2022.
- Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. *arXiv preprint arXiv:2004.13931*, 2020.
- Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

A Data Collection Process

A.1 User Initial Profile

For the initial preference seeds, we adopt 170 topic and sub-topic pairs adapted from the categorization proposed by Qiu et al. (2024), comprising 17 main categories each subdivided into 10 specific subcategories (see Table 7). Additionally, each pair is further annotated with a sentiment-based variable (*intent*), represented by one of four features: **amusing**, **emotional**, **informative**, and **recent news**.

Topics	Sub-topics
Animals	Dog, Wildlife, Cat, Fish, Birds, Insect, Snakes, Pet, Amphibians, Reptile
Education	School, Club, Teacher, Speaking, Listening, Writing, Presentation, Math, Computer, Teamwork
Health	Mental, Injury, Medication, Digestive health, Dental, Optical, Reproductive, Skin, Brain health, Cardiac
Travel	Museum, Park, Sea, Beach, Mountain, Lake, Hotel, Resort, Camping, Hiking
Movies	Action movie, Comedy, Romance, Science fiction, Horror, Drama, Cartoon, Documentary, Adventure, Crime
Cooking	Broiling, Grilling, Roasting, Baking, Sauteing, Boiling, Steaming, Poaching, Simmering, Stewing
Job	Manager, Researcher, Chef, Police, Lawyer, Salesman, Mechanic, Banker, Doctor, Waiter
Electronics	Laptop, TV, Phone, Software, Internet, Camera, Audio, Headphone, Hardware, Monitor
Art	Crafts, Photography, Painting, Collection, Drawing, Digital art, Sculpting, Pottery, Glass craft, Calligraphy
Personal Style	Grooming, Fashion, Personal Hygiene, Tattoos, Scarf, Hair Style, Makeup, Dressing, Tie, Formal
Clothes	Sweater, Jeans, Shirt, Socks, Coat, Pants, Hat, Gloves, Dress, Shoes
Sports	Outdoor recreation, Team sports, Tennis, Football, Basketball, Climbing, Skiing, Swimming, Fishing, Yoga
House	Building, Garden, Pool, Bathroom, Bedroom, Kitchen, Repairment, Moving, Decoration, Furniture
Food	Fruit, Vegetable, Drink, Meat, Seafood, Snacks, Dessert, Breakfast, Lunch, Dinner
Holiday	Halloween, Christmas, Labor day, Thanksgiving, Valentine’s day, Mother’s day, Birthday, National day, New year, Father’s day
Transportation	Car, Train, Bus, Boat, Bike, Airplane, Motorcycle, Truck, Trailer, Scooter
Hobbies	Dancing, Singing, Playing cards, Reading, Chess, Board games, Team games, Volunteer work, Instrument, Exercise

Table 7: Topic and sub-topic pairs.

A.2 YouTube Crawling

We aim to replicate the environment where real users interact with YouTube to ensure realism in the LLM-based user simulator. To achieve this, as shown in Figure 7, we provide the simulator with the exact metadata as displayed on the YouTube website, including the video title, channel name, description, view count, publication date, thumbnail URL, video link, and duration.

A.3 Prompts

A.3.1 Video Candidate Retrieval

Table 8 shows the prompt used by the LLM-based user simulator to determine whether to explore related videos or generate a new search query.

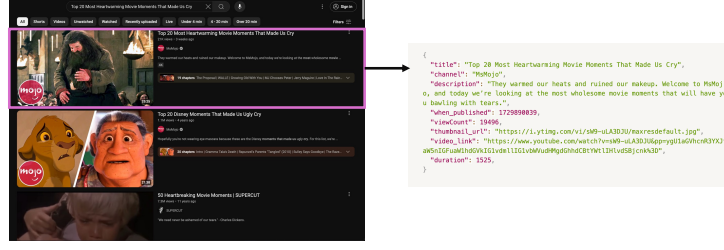


Figure 7: The process of converting YouTube video metadata into a structured JSON format.

A.3.2 Video Engagement

Table 9 presents the prompt for selecting the most and least preferred videos from the candidate pool, while Table 10 shows the prompt for engaging with the most preferred video.

A.3.3 Preference Update

Table 11 shows the prompt used to update long-term preferences after interaction with the newly watched video.

A.3.4 Saliency Scoring Annotation

Table 12 shows the prompt used to assign saliency scores to each segment in the video based on the provided preference information.

Prompt for Video Candidate Retrieval

You are finding {intent} videos about {search query}.

You have watched the following videos:
{watch history}

Your preferences have previously been defined as:
{preference}

For reference, current related videos are:
{related videos}

Now, decide whether to:
Explore the current query further by watching related videos.
Search for a new query to broaden your interest.

If you search for a new query, suggest one based on your interests, preferences, and history.

Answer Format:
Decision: ["Explore" or "Search for a new query"]
New query: [new query suggestion if "Search for a new query"]

Table 8: Instructions for an LLM-based user simulator to decide between exploring related videos or searching for new queries based on historical preferences.

A.4 Details of Human Verification

To assess the reliability and plausibility of the watch history generated by our LLM-based user simulator, we conducted human evaluations on two key components of the simulation framework: (1) query generation and (2) video selection. These evaluations were performed using Amazon Mechanical Turk (MTurk), with three independent annotators assigned to each task.

Prompt for Video Selection

You are a video quality rater, responsible for selecting the most relevant and least relevant videos based on your preferences.

Previously, you have watched the following videos:

{history}

So far, you have defined your preferences as:

{preference}

You now want to watch a video related to {query}.

From the list of candidate videos, choose the most wanted video (the one that best matches your preferences and the query) and the least wanted video (the one that least matches your preferences and the query).

Explain why each video is the most or least relevant to your preferences and the query.

Index starts from 1. (If you want to select the first video, you should write 1, not 0.)

If there is no appropriate candidate for the most or least wanted video, you should write [None].

Candidate Videos:

{candidate}

Answer Format: Fill [] with your response. Do not return anything else.

Most Wanted: [video number]

Explanation: [Why this video best matches your preferences and the query]

Least Wanted: [video number]

Explanation: [Why this video least matches your preferences and the query]

Table 9: Instructions for an LLM-based user simulator to select the most and least wanted videos based on preferences and a query.

Prompt for Watching a Video

You are a YouTube viewer with your preferences, and you should create a video summary based on how well it aligns with your personal preferences.

Context: Your latest updated preferences are as follows:

{preference}

Now, you are watching a new video, presented as a series of (frame description, transcript) pairs.

Video:

{video}

Write your summary of the video, followed by your personal opinion of the video.

The summary and personal opinion should be 2 sentences each.

For personal opinion, you may refer to 1 or 2 preferences that are related to reviewing the video.

But make sure your opinion should be mainly based on the video content, not just your preferences.

You may like or dislike the video.

Return only one paragraph for the answer.

Table 10: Instructions for an LLM-based user simulator to summarize and review a video based on its content and user preferences.

Annotators were provided with the exact same input as the LLM-based user simulator—namely, the set of previously watched videos and the user’s long-term preference at the time of decision. For video selection, annotators were also given the same video candidate pool, including metadata such as titles, thumbnails, and descriptions (as detailed in Appendix A.2).

Query Generation. For each query generated by the simulator, annotators were asked if this is a reasonable next search query given the user’s watch, search history, and preference. Each response was labeled as either reasonable or not reasonable. The agreement rate refers

Prompt for Preference Update

You are a preference analyzer, responsible for refining user preferences based on user-written reviews and reasons for the most and least wanted videos.

Your preferences have previously been defined as:
{preference}

Next, your previous reviews on videos you watched are as follows:
{reviews}

Next, you selected the following video as the most wanted video based on your preferences:
{selected video}
Reason:
{selected reason}

Next, you selected the following video as the least wanted video based on your preferences:
{least wanted video}
Reason:
{least wanted reason}

Based on the previous reviews and reasoning for the most and least wanted videos, re-define your preferences and dis-preferences in bullet points.

Let’s break it down step by step:

1. **ADDITION:** Add new details that were introduced in the video and are not in your previous preferences.
 2. **REFINEMENT:** Refine or adjust your preferences with specific terms where necessary to encompass both your original preferences and new insights from the video. Unless the content itself needs to change, reuse the exact words from the existing preferences.
 3. **REMOVAL:** Remove any details from your preferences where the newly watched video and your original preferences do not align.
-

Table 11: Instructions for an LLM-based user simulator to update user preferences based on previous video reviews and reasoning.

Prompt for Scoring Video Clips Based on Viewer Preferences

You are a viewer with specific content preferences. Evaluate multiple video clips and provide a score from 1 to 10 based on their appeal to you.

Preference Profile:
{preference.profile}

For each clip below, determine how appealing it would be to you. Consider engagement, pacing, and overall impact. Provide a score and a short justification for each clip.

Clips to Evaluate:
{clip info}

Output Format:

A list of responses for each clip, using this format:

- Clip ID: clip_0, Score: 8, Justification: "Interesting content and engaging pacing."
 - Clip ID: clip_1, Score: 5, Justification: "Too slow and not aligned with my interests."
-

Table 12: Instructions for an LLM-based user simulator to rate the appeal of video clips based on personal preferences and provide justifications.

to the proportion of queries that were rated reasonable by the majority (i.e., at least two out of three annotators), which amounted to 97.56%. We also report inter-annotator agreement, computed as Fleiss’ κ , which was 0.85, indicating substantial agreement among annotators and reinforcing the consistency of the task design and the reliability of the simulator’s outputs.

Video Selection. Annotators were asked to select the most preferred video from a set of candidates based on the provided preference information. The simulator’s choice was then compared to the majority choice of the annotators. If at least two annotators selected the same video as the simulator, the decision was considered a match. The agreement rate between the simulator and human annotators was 68.42%, showing that the simulator aligned with human choices in the majority of cases. This demonstrates its ability to mimic realistic user behavior when making preference-based decisions.

These results validate the plausibility of the generated watch histories and support the reliability of the simulation framework used in constructing HiPPO-VIDEO. While simulated behavior cannot perfectly replicate real user actions, our evaluations suggest that the LLM-based user simulator closely approximates human decision-making and offers a scalable foundation for studying personalized video summarization.

A.5 Details on Analyses

Inter-history video diversity. To generate an embedding for each watch history, we first extract visual features from each video using CLIP-ViT/B-32. These features are then averaged per video to capture its overall content, resulting in a representative feature for each video. Finally, we average the features across all videos in watch history, yielding a single embedding for the entire history. This embedding serves as a compact representation of the user’s viewing patterns and preferences, capturing both individual video content and the broader interests reflected in the sequence of watched videos.

Saliency Score Distribution. Figure 3c shows the distribution of saliency scores using kernel density estimation (KDE). The mean saliency score (left) is computed as the average of the saliency scores for all segments in a video, given by the formula: $\text{mean} = \frac{1}{n} \sum_{k=1}^n y_k$ where y_k is the saliency score for segment k , and n is the total number of segments in the video. The mean provides a measure of the average relevance of the segments within a video. In our dataset, most videos have a mean score between 4 and 6, indicating that the majority of videos are considered moderately important overall.

To evaluate how much the saliency scores fluctuate across segments, we compute the standard deviation, which measures the spread of the scores around the mean. The standard deviation is calculated as: $\text{std} = \sqrt{\frac{1}{n} \sum_{k=1}^n (y_k - \text{mean})^2}$ where y_k represents the individual saliency score for segment k , and mean is the mean saliency score for the video. The standard deviation quantifies the variability in the importance of the segments. Standard deviations typically range from 1.5 to 2 in our dataset, suggesting moderate variability in how segments are scored. Higher standard deviations (greater than 3) indicate greater fluctuations in segment importance, which may result from dynamic visual changes, such as scene transitions or shifts in the video’s content, where some segments may be much more relevant than others.

A.6 Long-term Preference Modeling

Table 13 illustrates how the LLM-based user simulator refines its long-term preferences as it iteratively interacts with more videos. Initially, the preferences are broad and general, but as the simulator processes more content, they become increasingly specific and nuanced. Through repeated engagement with the content, the simulator develops a detailed preference model that captures both high-level interests and subtle distinctions, emphasizing the importance of iterative interactions in accurately modeling complex user preferences.

B More Details on Experiments

B.1 Baseline Details

SL-module replaces conventional pair-based learning in video highlight detection with a set-based approach. Instead of comparing segment pairs, it evaluates a set of video

Watch History (0 Videos)	
• I like recent news on Job, especially about Lawyer.	
Watch History (3 Videos)	
[Likes] • Informative insights with dynamic and visually appealing presentations. • Interest in latest trends and technology in the legal profession, especially related to law firm management, including generative AI.	[Dis-likes] • Lack of engaging visuals and narrative structure. • Does not directly address current trends affecting the legal profession.
Watch History (10 Videos)	
[Likes] • Combines informative insights with dynamic, visually appealing presentations. • Interest in technology and trends in the legal profession, including generative AI, LawTech, and RegTech. • Prefers a cohesive flow and engaging narrative exploring risks and ethical challenges of AI. • Values insights on how law firms use AI to improve services and move beyond billable hour models. • Enjoys practical applications of AI and their role in advancing legal workflows and access to justice. • Likes content highlighting how technology supports underserved populations and enables judicial reform. • Interested in the ethical implications of AI and its effect on trust in the legal system.	[Dis-likes] • Lacks engaging visuals and narrative structure. • Omits discussion of technological trends, including those influenced by the pandemic. • Too focused on patent examination without broader technological context. • Emphasizes marketing/scaling law firms without connecting to tech impact.

Table 13: Long-term user preferences for video content related to legal professions and technological advancements, organized by watch history length.

segments to predict highlight scores by modeling inter-dependencies among segments within the same video. A fixed visual feature extractor processes each segment, followed by a transformer encoder (without positional encoding) to capture contextual relationships. The transformer output is passed to a scoring model that outputs highlight scores. The model is trained by minimizing the KL divergence between predicted and ground-truth highlight score distributions over the set.

Moment-DETR projects video and query features into a shared embedding space, concatenates them, and processes the result with a transformer encoder using positional encodings. A linear layer predicts saliency scores from the encoder output. A transformer decoder, initialized with moment queries, predicts temporal moments; its output feeds into a three-layer FFN for normalized coordinates and a softmax classifier for moment-level scores.

UMT starts by processing visual and audio features with separate transformer encoders. Then, a bottleneck transformer fuses these features into multi-modal representations. If a text query is provided, it is used to generate temporally-aligned, clip-specific moment queries via attention between the text and the multi-modal features. The generated queries are then decoded to obtain joint representations for both tasks. Finally, the model produces clip-level saliency scores for highlight detection and moment boundaries (center, window, and offset) for moment retrieval.

VSL summarizes videos based on user-preferred genres using a similarity-based approach. It generates scene-level textual summaries from visual captions and transcripts, then computes similarity between these summaries and genre embeddings derived from text prompts.

B.2 Evaluation Metrics

To comprehensively assess our model’s performance across tasks, we utilize several widely accepted evaluation metrics. These metrics are chosen based on the nature of each task—whether it involves ranking video segments, retrieving specific moments, or predicting saliency scores. Below, we explain each metric and its purpose in detail.

For highlight detection, the goal is to rank video segments based on how salient or important they are. To evaluate how well our model performs this ranking, two key metrics are usually used:

- Mean Average Precision (mAP): mAP measures the quality of ranked results. It checks whether the most relevant (i.e., salient) segments appear near the top of the list. For each relevant segment, it calculates the precision (i.e., how many of the top-ranked results are correct), and then averages this across all relevant segments. Finally, the average is taken over all test samples. Higher mAP values indicate better performance, meaning the model ranks relevant segments closer to the top more consistently.
- Hit@1: This metric simply checks whether the top-ranked video segment is actually one of the ground truth salient segments. It’s a straightforward way to see if the model gets the very best answer right.

To decide which segments are “*salient*,” we use saliency score thresholds. These scores are given on a scale from 1 to 10. Following the methodology of Liu et al. (2015), we treat segments with scores ≥ 7 and ≥ 9 as salient (i.e., ground truth highlights). In (Liu et al., 2015), the threshold was 4 out of 5, which corresponds to a similar percentile cut-off.

In Moment Retrieval, the task is to retrieve a specific time segment in the video that corresponds to a given query (e.g., “when the player scores a goal”). Here, we want to know whether the model correctly identifies the right moment in time.

- Recall@1: This metric evaluates whether the top segment predicted by the model has sufficient overlap with the ground truth moment. It focuses on the single top-ranked result. If this result matches the correct segment well enough, it’s counted as a success.

To define what counts as a good match, we use Intersection over Union (IoU), a standard metric in temporal and spatial localization tasks. IoU compares how much the predicted time span overlaps with the ground truth. It is calculated as the length of the overlap divided by the length of the union of both time spans. If $\text{IoU} \geq 0.5$: The prediction is considered correct if at least 50% of the predicted and ground truth segments overlap. If $\text{IoU} \geq 0.7$, the match must be even more precise, with at least 70% overlap. Higher Recall@1 values mean that the model is retrieving relevant moments more accurately.

B.3 Ablation Studies

While we initially set $\gamma = 1$ following prior works (Lei et al., 2021; Moon et al., 2023), we conducted an ablation study to assess its impact on HiPHER’s performance. We observed that smaller margins ($\gamma = 0.1$ – 0.2) consistently yield better results, as they enable finer-grained preference modeling. In contrast, larger margins tend to encourage overconfident separation between segments, which reduces generalization. We will include these ablation results in the final version to specify the impact of γ .

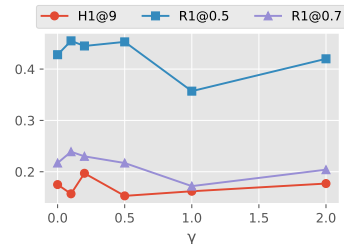


Figure 8: Performance of varying γ .

B.4 Case Study

Figure 9 presents qualitative case studies across various video topics (e.g., movies, sports, holidays), comparing the predicted scores of HiPHer and the baseline (Moment-DETR).

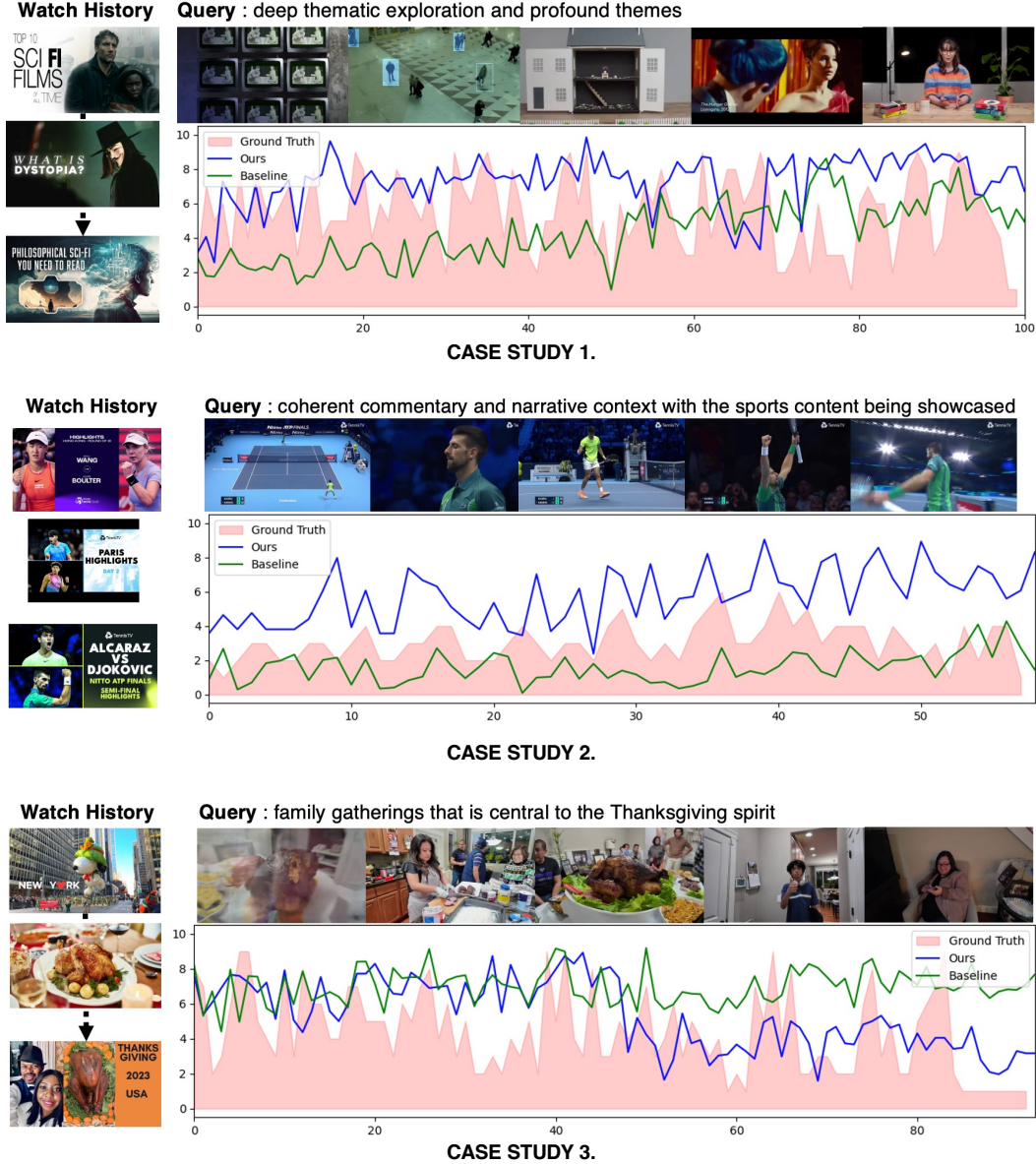


Figure 9: Qualitative Case Studies: “Ours” refers to HiPHer, and “Baselines” refers to Moment-DETR.