FOCUS: EFFICIENT KEYFRAME SELECTION FOR LONG VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models (MLLMs) represent images and video frames as visual tokens. Scaling from single images to hour-long videos, however, inflates the token budget far beyond practical limits. Popular pipelines therefore either uniformly subsample or apply keyframe selection with retrieval-style scoring using smaller vision-language models. However, these keyframe selection methods still rely on pre-filtering before selection to reduce the inference cost and can miss the most informative moments.

We propose FOCUS, *Frame-Optimistic Confidence Upper-bound Selection*, a training-free, model-agnostic keyframe selection module that selects query-relevant frames under a strict token budget. FOCUS formulates keyframe selection as a combinatorial pure-exploration (CPE) problem in multi-armed bandits: it treats short temporal clips as arms, and uses empirical means and Bernstein confidence radius to identify informative regions while preserving exploration of uncertain areas. The resulting two-stage exploration-exploitation procedure reduces from a sequential policy with theoretical guarantees, first identifying high-value temporal regions, then selecting top-scoring frames within each region On two long-video question-answering benchmarks, FOCUS delivers substantial accuracy improvements while processing less than 2% of video frames. For videos longer than 20 minutes, it achieves an 11.9% gain in accuracy on LongVideoBench, demonstrating its effectiveness as a keyframe selection method and providing a simple and general solution for scalable long-video understanding with MLLMs.

1 INTRODUCTION

"The art of being wise is the art of knowing what to overlook." — William James

Recent advances in large language models (LLMs) and multimodal LLMs (MLLMs) have significantly improved visual understanding and reasoning. In current frameworks, images are encoded into visual tokens aligned with text and jointly processed by the LLM. Extending this paradigm to videos—especially long, untrimmed ones—introduces a key challenge: the sheer number of frames leads to an overwhelming number of visual tokens, making inference computationally prohibitive.

A common solution is aggressive downsampling (Wang et al., 2022; Lin et al., 2023; Maaz et al., 2023a; Wang et al., 2024b; Zhang et al., 2025b), but uniformly sampling a handful of frames (e.g., 64 from a one-hour video) often misses critical content. Increasing the frame rate, on the other hand, causes token explosion. This trade-off motivates the need for keyframe selection: choosing a small set of informative frames that preserve semantics while staying within token limits.

Recent methods address this by scoring frame relevance with pre-trained vision-language encoders (e.g., CLIP (Radford et al., 2021) or BLIP (Li et al., 2022)) and then pick the highest-relevance frames (Tang et al., 2025; Zhang et al., 2025a). These text–image matching approaches are typically training-free and plug in easily before the visual encoder in MLLM stacks, retrieving frames with higher relevance other than uniform sampling. Despite their success, current keyframe selection methods still face scalability and efficiency limitations. For a one-hour video at 30 fps (over 10^5 frames), exhaustively scoring all frames entails on the order of 10^{11}-10^{12} FLOPs with a vision-language encoder like BLIP (Li et al., 2022). This scaling pressure forces existing methods (Zhang et al., 2025a; Tang et al., 2025) to uniformly sample the video to lower frame rate before the scoring

process. This pre-filtering process before keyframe selection undermines the goal of identifying most informative keyframes from all frames.

In this work, we propose FOCUS, *Frame-Optimal Confidence Upper-Bound Selection*, a training-free, plug-and-play keyframe selection method designed to process extremely long videos with minimal computational overhead. FOCUS is easy to implement in practice while offering an elegant theoretical foundation.

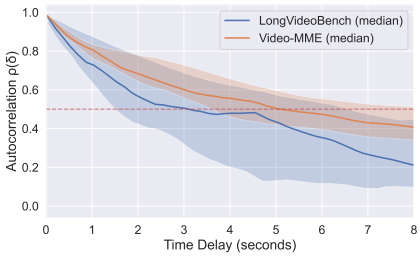

Figure 1: Temporal autocorrelation (ACF) of the frame-query relevance r_t on LongVideoBench and Video-MME. Solid lines show the median ACF across videos and shaded bands indicate the interquartile range.

The key insight behind FOCUS is grounded in the observation that natural videos exhibit strong temporal locality: adjacent frames are highly correlated in appearance and motion (Wiegand et al., 2003; Wang et al., 2016; 2022). This local smoothness naturally extends to frame–query relevance scores. As illustrated in Figure 1, we compute the autocorrelation function (ACF) of relevance scores r_t on LongVideoBench and VideoMME. The results show a strong local correlation structure, with a half-life of approximately 5 seconds.

This observation implies that exhaustive scoring of all frames is unnecessary. Instead, we can adopt an exploration–exploitation strategy to adaptively allocate computation: quickly filtering out irrelevant temporal regions and focusing scoring efforts on promising segments, ultimately prioritizing the most informative keyframes.

FOCUS first partitions the video into short temporal clips, each treated as an arm in a multi-armed bandit. The clip selection is then framed as a Combinatorial Pure-Exploration (CPE) problem: the goal is to identify a subset of arms that maximizes expected cumulative relevance under a strict token budget.

Each arm maintains an empirical mean relevance and a Bernstein-style confidence radius. Computation is adaptively allocated to clips that are either promising (high mean) or uncertain (large confidence radius), following an optimism-in-the-face-of-uncertainty principle. This iterative process enjoys theoretical convergence guarantees. To leverage parallel computation without sacrificing optimism, we reduce the iterative strategy to a coarse-to-fine schedule: optimistic means guide exploration, while unbiased empirical means inform final arm selection. Within each selected arm, we extract the top-relevance frames to construct the final keyframe set.

We validate the effectiveness of our approach on two video understanding benchmarks, including LongVideoBench (Wu et al., 2024) and Video-MME (Fu et al., 2025). The proposed FOCUS is tested as an off-the-shelf module on with four popular MLLMs. FOCUS improves answer accuracy over state-of-the-art keyframe selection baselines across benchmarks while maintaining lower inference cost. The gains are especially pronounced on long-form videos: for videos longer than 20 minutes on LongVideoBench, FOCUS delivers a 1.9% accuracy improvement while still cutting inference cost.

In summary, our main contributions are three-fold: (1) We formulate query-aware keyframe selection as a budgeted *combinatorial pure-exploration* (CPE) problem in a multi-armed bandit setting; (2) We introduce FOCUS, a training-free, model-agnostic keyframe selection module that selects query-relevant frames under a strict token budget; (3) We validate the effectiveness of FOCUS on two long-video understanding benchmarks, achieving consistent gains across four popular MLLMs.

2 METHOD

2.1 PROBLEM FORMULATION

Keyframe Selection Setup. Let a video be $V = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$ and denote the corresponding text query as q. Let the frame index set be $\mathbb{T} = \{1, \ldots, T\}$. A downstream multimodal LLM Φ consumes a subset of frames indexed by $\mathbb{K} \subseteq \mathbb{T}$ with $|\mathbb{K}| = k$ and produces an answer $\hat{a} = \Phi\big(q, \{\boldsymbol{x}_t\}_{t\in\mathbb{K}}\big)$. Let

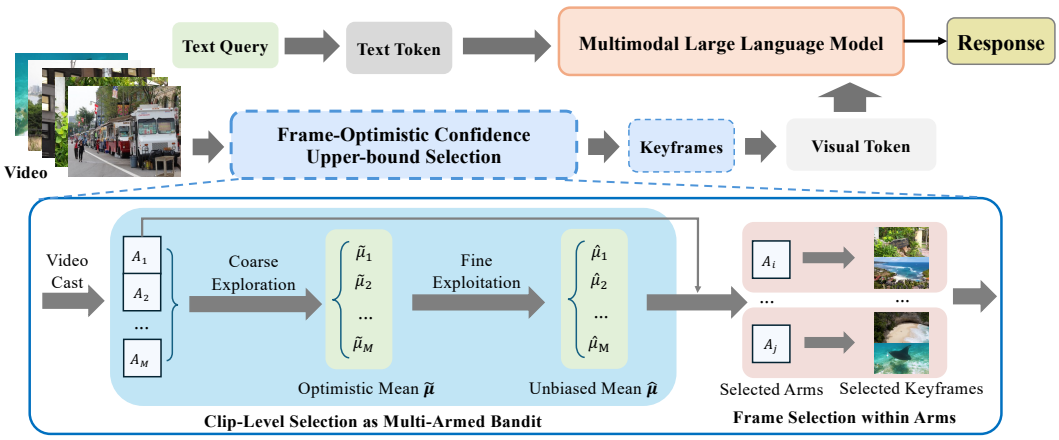

Figure 2: Overview of FOCUS. FOCUS partitions videos into fixed-length clips as bandit arms, applies optimistic confidence upper-bound arm selection and selects final keyframes within each promising arms.

$R_\Phi(\mathbb{K} \mid V, q)$ denote the task-level utility of the selected frames (e.g., quality of generated answer, relevance to query, or other performance metrics).

Oracle and Surrogate Objective. The oracle objective chooses $\mathbb{K}$ to maximize expected utility:

$$\mathbb{K}^{\text{oracle}}(V, q) = \underset{\mathbb{K}\subseteq\mathbb{T},\,|\mathbb{K}|=k}{\arg\max}\ \mathbb{E}\big[R_\Phi(\mathbb{K} \mid V, q)\big], \tag{1}$$

Direct optimization to equation 1 is infeasible due to the combinatorial search space and the high cost of black-box evaluations of Φ. We further expand the task-level utility $R_\Phi(\mathbb{K} \mid V, q)$ to a summation of frame-level utility $y_t \in [0, 1]$:

$$\mathbb{K}^\star = \underset{\mathbb{K}\subseteq\mathbb{T},\,|\mathbb{K}|=k}{\arg\max}\ \mathbb{E}\big[\sum_{t\in\mathbb{K}} y_t\big]. \tag{2}$$

However, estimating the contribution of each frame t to the task-level utility is also intractable. We therefore posit that y_t is indirectly observable via a vision-language encoder ψ that outputs a relevance score $r_t = \psi(\boldsymbol{x}_t, q; \boldsymbol{\theta}) = y_t + \epsilon_\psi$, where ϵ_ψ denotes encoder-induced noise. We assume ϵ_ψ follows some distribution that are supported on $[0, 1]$ and with zero mean and σ_ψ^2 variance. Under this assumption, the relevance score r_t is a unbiased estimator of y_t which is also commonly used in many works (Tang et al., 2025; Yu et al., 2024) implicitly.

Exhaustively scoring all T frames to get $\{r_t\}$ is computationally prohibitive, especially for hourly long videos which contains over 10^5 frames. This computational constraint motivates us to model keyframe selection under budget constraints, where we strategically allocate a limited sampling budget to identify the most promising temporal segments before producing the final set of k keyframes. Instead of directly optimizing equation 2 at the frame level, we will approximate it through a combinatorial pure-exploration multi-armed bandit formulation at the clip level, which significantly reduces exploration cost.

2.2 CLIP-LEVEL SELECTION AS MULTI-ARMED BANDIT

For a video $V = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$, we partition the timeline into M non-overlapping fixed-length clips $\mathcal{A} = \{A_a\}_{a=1}^M$, where each clip $A_a \subseteq \mathbb{T}$ spans frames $[s_a, e_a]$ and is treated as a bandit arm. We assume that frame-level utility within the same arm share the same distribution: $y_t \sim \nu_a$ for all $t \in [s_a, e_a]$, where ν_a has mean μ_a and variance σ_a^2. We define pulling the arm a as randomly sampling one frame from that clip and observing its query relevance score r_t as a reward.

Intuitively, our goal is to focus on the most promising clips which means we have to identify the optmal subset $S^\star \subseteq \mathcal{A}$. Formally, we define the a *decision class* $\mathbb{S} \in 2^{\mathcal{A}}$ as a subset of the power set

Algorithm 1 Iterative Optimistic Confidence Upper-bound Arm Selection

Require: Maximization oracle $\mathrm{TopM}(\{\mu_a\}, m) \to \mathbb{A} \subseteq \mathcal{A}$

Initialize: Empirical means $\hat{\mu}_0(a) \leftarrow 0$ and $N_0(a) \leftarrow 0$ for all a.
Pull each arm $a \in \mathcal{A}$ for q times and observe the rewards.
Update empirical means $\hat{\mu}_a$ for all a.
$N_{mq}(a) \leftarrow q$ for all a.
for $n \leftarrow mq, mq+1, \ldots$ **do**
  $\mathbb{A}_n \leftarrow \mathrm{TopM}(\hat{\boldsymbol{\mu}}, m)$
  Compute confidence radius $\beta_a(n)$ for all $a \in \mathcal{A}$ ▷ $\beta_a(n)$ defined in equation 5
  for $a \leftarrow 1$ **to** M **do**
    if $a \in \mathbb{A}_n$ **then**
      $\tilde{\mu}_n(a) \leftarrow \hat{\mu}_n(a) - \beta_a(n)$
    else
      $\tilde{\mu}_n(a) \leftarrow \hat{\mu}_n(a) + \beta_a(n)$
    end if
  end for
  $\tilde{\mathbb{A}}_n \leftarrow \mathrm{TopM}(\tilde{\boldsymbol{\mu}}, m)$
  if $\tilde{\mathbb{A}}_n = \mathbb{A}_n$ **then**
    return $\mathbb{A}_n$
  end if
  $p_n \leftarrow \underset{a \in (\tilde{\mathbb{A}}_n \setminus \mathbb{A}_n) \cup (\mathbb{A}_n \setminus \tilde{\mathbb{A}}_n)}{\arg\max} \beta_a(n)$ ▷ break ties arbitrarily
  Pull arm p_n and observe the reward
  Update empirical means $\hat{\mu}(p_n)$ with the observed reward
  $N_{n+1}(p_n) \leftarrow N_n(p_n) + 1$
end for

of $\mathcal{A}$. The optimal member $S^\star$ of decision class $\mathbb{S}$ is defined as

$$S^\star = \underset{S \in \mathbb{S}}{\arg\max} \sum_{a \in S} \mu_a. \tag{3}$$

Under the classic CPE framework, the learner's objective is to identify $S^\star$ after interacting with the arms over a sequence of rounds. In the keyframe selection setting, our final goal is further to select k keyframes from the selected arms. Denote $\{k_a\}_{a=1}^{|S^\star|}$ as the number of keyframes allocated to the a-th selected arm. We further define the frame-level optimal keyframe subset $\mathbb{K}_a^\star$ as

$$\mathbb{K}_a^\star = \underset{\mathbb{K}_a \subseteq A_a,\, |\mathbb{K}_a| = k_a}{\arg\max} \sum_{t \in \mathbb{K}_a} y_t. \tag{4}$$

The final keyframe subset $\mathbb{K}^\star$ is then defined as $\mathbb{K}^\star = \bigcup_{a \in S^\star} \mathbb{K}_a^\star$. Empirically, we assume the decision class $\mathbb{S}$ is all size-m subsets of $\mathcal{A}$ and keyframes are equally distributed across the promising arms. This setting gives us a elegant theoretical guarantee of regret bound as shown in section C and is also proved to be effective in our experiments.

2.3 Optimistic Confidence Upper-bound Arm Selection

2.3.1 Optimal Arm Selection.

Generally, we play a exploration game by pulling an arm a and observing the reward r_t at each round n. We maintain two core empirical statistics for each arm a during this process: an empirical mean $\hat{\mu}_a$ and an empirical Bernstein confidence radius (variance-adaptive) β_a, following the UCV-V style bound (Audibert et al., 2009):

$$\beta_a(n) \;=\; \sqrt{\frac{2\,\hat{\sigma}_a^2\,\ln n}{\max(1, N_a(n))}} \;+\; \frac{3\,\ln n}{\max(1, N_a(n))}. \tag{5}$$

Here $N_a(n)$ is the number of pulls for arm a at round n and $n = \sum_{a \in \mathcal{A}} N_a(n)$ is the total number of pulls. The confidence radius ensures that the empirical mean is within the confidence radius of the

Algorithm 2 Optimistic Confidence Upper-bound Arm Selection

Require: Maximization oracle $\mathrm{TopM}(\{\mu_a\}, m) \to \mathbb{A} \subseteq \mathcal{A}$
1: **Initialize:** Empirical means $\hat{\mu}_0(a) \leftarrow 0$ and $N_0(a) \leftarrow 0$ for all a.
// Stage I: Coarse exploration
2: Pull each arm $a \in \mathcal{A}$ for q times and observe the rewards.
3: Update empirical means $\hat{\mu}_a$ for all a.
4: $N_{mq}(a) \leftarrow q$ for all a.
5: Compute confidence radius $\beta_a(n)$ for all $a \in \mathcal{A}$
6: $\tilde{\mu}_n(a) \leftarrow \hat{\mu}_n(a) + \beta_a(n)$ for all $a \in \mathcal{A}$
7: $\mathbb{A}_{\text{coarse}} \leftarrow \mathrm{TopM}(\tilde{\boldsymbol{\mu}}, m)$ ▷ Optimistic Means UCB
// Stage II: Fine-grained exploitation
8: Pull each arm $a \in \mathbb{A}_{\text{coarse}}$ for z times and observe the rewards.
9: Update empirical means $\hat{\mu}_a$ for $a \in \mathbb{A}_{\text{coarse}}$
10: $\mathbb{A}_{\text{fine}} \leftarrow \mathrm{TopM}(\hat{\boldsymbol{\mu}}, m)$ ▷ Unbiased Smpirical Means
11: **return** $\mathbb{A}_{\text{fine}}$

true mean with high probability, *i.e.,*

$$\mathcal{P}\left[|\hat{\mu}_a - \mu_a| \leq \beta_a(n)\right] \geq 1 - \frac{6}{n}. \tag{6}$$

Please refer to Appendix B for the detailed proof.

As shown in Algorithm 1, the optimistic confidence upper-bound arm selection starts with an initialization phase where we pull each arm for q times and observe the relevance scores as rewards. We then update the empirical means $\hat{\mu}_a$ and compute the confidence radius $\beta_a(n)$ for each arm a. Note the relevance score r_t is an unbiased estimator of y_t so we have $\mathbb{E}[\hat{\mu}_a] = \mu_a$. Then we choose the best m arms using the empirical means $\hat{\mu}_a$, *i.e.,* $\mathbb{A}_n = \mathrm{TopM}(\hat{\boldsymbol{\mu}}, m)$, where $\hat{\boldsymbol{\mu}}$ is the vector of all arms' empirical means and $\mathrm{TopM}(\cdot, m)$ returns a set of the m arms with the largest empirical means.

We further refine the arm selection by evaluating the "potential" of each arm. To be specific, for arm $a \in \mathbb{A}_n$, we compute the lower confidence bound of the empirical mean, *i.e.,* $\mathrm{LCB}_a(n) = \hat{\mu}_a - \beta_a(n)$; for arm $a \notin \mathbb{A}_n$, we compute the upper confidence bound of the empirical mean, *i.e.,* $\mathrm{UCB}_a(n) = \hat{\mu}_a + \beta_a(n)$. If

$$\max_{a \notin \mathbb{A}_n} \mathrm{UCB}_a(n) \geq \min_{a \in \mathbb{A}_n} \mathrm{LCB}_a(n), \tag{7}$$

this indicates that some arms outside the current top-m set are still potential to be included in the top-m set. Thus, we choose the arm a that we are most uncertain about, *i.e.,*

$$a = \underset{a \in (\tilde{\mathbb{A}}_n \setminus \mathbb{A}_n) \cup (\mathbb{A}_n \setminus \tilde{\mathbb{A}}_n)}{\arg\max} \beta_a(n). \tag{8}$$

We then pull this arm a for q times and repeat the process until the top-m set is unchanged, *i.e.,* $\mathbb{A}_{n+1} = \mathbb{A}_n$. We then return the top-m set $\mathbb{A}_n$.

It is easy to see Algorithm 1 is guaranteed to return the optimal top-m set $\mathbb{A}_n$ with high probability (see Section C for the detailed proof). However, the iterative process is empirically inefficient (or intractable) as the sequential arm-pulls and updating can not be parallelizable. We have to pull the arms one-by-one which means forward the vision-language model with batch size 1 sequentially. This costs significant waste of GPU utilization.

2.3.2 Two-stage Arm Selection.

To make the procedure practical and easy to parallelize, we specialize Algorithm 1 into the two-stage, batch variant in Algorithm 2.

Stage I: Coarse initialization. We pull each arm q times in parallel and update the empirical means $\hat{\mu}_a$ and confidence radii $\beta_a(n)$ for all $a \in \mathcal{A}$. This stage coincides with the initialization phase of Algorithm 1 and serves as a coarse exploration pass that produces reliable per-arm statistics at low coordination cost.

Stage II: Fine-grained exploration (batched). Using the optimistic scores $\tilde{\mu}a = \hat{\mu}a + \beta_a(n)$, we select the top αm arms, $\mathcal{A}\text{coarse} = \mathrm{TopM}(\tilde{\boldsymbol{\mu}}, , \alpha m)$, and allocate an additional z pulls to each $a \in \mathcal{A}_{\text{coarse}}$ (performed in a single batch). Here, α is a hyperparameter that controls the ratio of the coarse exploration budget to the fine-grained exploration budget. This stage is a batched counterpart of the iterative loop in Algorithm 1: it implements the "optimism in the face of uncertainty" principle by concentrating samples on arms with the largest UCB values, while avoiding per-step scheduling overhead.

Final Arm Selection. After the fine exploitation, we form the final set by selecting the best m arms according to the unbiased empirical means, $\mathbb{A}_{\text{fine}} = \mathrm{TopM}(\hat{\boldsymbol{\mu}}, m)$. This choice mirrors δ-PAC identification routines, where optimistic scores guide exploration but the recommendation itself is based on $\hat{\mu}_a$ rather than $\tilde{\mu}_a$.

2.4 Frame Selection within Selected Arms

Given the selected arm set $\mathbb{A}_{\text{fine}}$ and a total budget of K frames, we sample k_a frames per arm $a \in \mathbb{A}_{\text{fine}}$ with equal allocation (i.e., $k_a = \mathrm{round}(k/|\mathbb{A}_{\text{fine}}|)$, adjusted to sum to K). For each arm a with index set $\mathbb{T}_a$ and observed rewards $\{r_{a,s}\}_{s \in S_a}$ at sampled indices $T_a \subseteq \mathbb{T}_a$, we simply interpolate all rewards $\hat{r}_{a,t}$ within the arm using the nearest-neighbor assignment. We then form a per-arm sampling distribution according to the interpolated rewards and draw k_a frames *without replacement* from p_a. The final keyframe set is $\mathcal{K} = \bigcup_{a \in \mathcal{A}_{\text{fine}}} \mathcal{K}_a$.

3 Experiments

3.1 Experimental Setup

Benchmarks We follow the LMMs-Eval framework Zhang et al. (2024) and adopt the open-source evaluation protocol from AKS for benchmarks, prompts, and scoring. Our experiments focus on two long-video multiple-choice QA benchmarks: LongVideoBench Wu et al. (2024) and VideoMME Fu et al. (2025). These datasets feature videos lasting up to an hour, where effective keyframe selection becomes crucial for performance. To ensure fair comparison (Tang et al., 2025), we disable subtitles, perform zero-shot evaluation, and keep model parameters frozen—varying only the frame selection strategy (our method versus uniform sampling).

Implementation Details We test both open-source video MLLMs (Qwen2VL (Wang et al., 2024a), LLaVA-OV (Li et al., 2025), LLaVA-Video (Zhang et al., 2025b) and Qwen2-7B (Yang et al., 2024) language model) and the commercial GPT-4o (0513). For frame relevance scoring, we use BLIP ITM (Li et al., 2022) to compute $r_t = \psi(\boldsymbol{x}_t, q; \boldsymbol{\theta})$, where r_t estimates the latent frame-level utility as described in Section 2.1, which is justified as a promising choice by Tang et al. (2025).

3.2 Performance Analysis

We evaluate FOCUS by using it to select keyframes as the visual input for the four aforementioned MLLMs, and compare it against the commonly used uniform sampling strategy. The results on LongVideoBench and Video-MME are summarized in Table 1.

Improved Performance via Frame Selection. As shown in Table 1, FOCUS consistently outperforms uniform sampling across both open-source and closed-source MLLMs on both LongVideoBench and Video-MME.

Specifically, on LongVideoBench, FOCUS improves accuracy by 3.2% on GPT-4o, 6.7% on Qwen2-VL-7B, 5.9% on LLaVA-OV-7B, and 4.6% on LLaVA-Video-7B. On Video-MME, the gains are 0.7%, 2.1%, 1.8%, and 1.0% on the same models, respectively.

We observe a clear trend that larger MLLMs with more frame inputs tend to achieve better performance. However, FOCUS significantly narrows this gap by identifying the most informative frames, thereby boosting the performance of smaller MLLMs. For instance, Qwen2-VL-7B with FOCUS outperforms Gemini-1.5-Flash on LongVideoBench, despite using 8× fewer input frames. This

Model	#Frame	LLM	LongVideoBench	Video-MME
GPT-4V	256	–	61.3	59.9
Gemini-1.5-Flash	256	–	61.6	70.3
Gemini-1.5-Pro	256	–	64.0	75.0
VideoLLaVA	8	7B	39.1	39.9
MiniCPM-V 2.6	64	8B	54.9	60.9
InternVL2-40B	16	40B	59.7	61.2
LLaVA-Video-72B	64	72B	63.9	70.6
GPT-4o	32	–	51.6	61.8
GPT-4o ***w/*** Ours	32	–	**54.8**	**62.5**
Qwen2-VL-7B	32	7B	55.6	57.6
Qwen2-VL-7B ***w/*** Ours	32	7B	**62.3**	**59.7**
LLaVA-OV-7B	32	7B	54.8	56.5
LLaVA-OV-7B ***w/*** Ours	32	7B	**60.7**	**58.3**
LLaVA-Video-7B	64	7B	58.9	64.4
LLaVA-Video-7B ***w/*** Ours	64	7B	**63.5**	**65.4**

Table 1: Video-question answering accuracy (%) of various MLLMs on LongVideoBench and Video-MME. FOCUS is integrated into GPT-4o, Qwen2-VL, LLaVA-OV, and LLaVA-Video. The suffix "***w/*** Ours" denotes models using keyframes selected by our method; otherwise, frames are uniformly sampled. **#Frame** indicates the number of frames provided to the MLLM, and **LLM** denotes the language model size. We also include performance of additional popular MLLMs for reference.

highlights the effectiveness of FOCUS as a plug-and-play keyframe selection module for a wide range of MLLMs.

Interpretability through Visualizations. We visualize the frames selected by FOCUS alongside uniformly sampled frames for two examples from LongVideoBench and Video-MME in Figure 3.

Note that LongVideoBench and Video-MME differ substantially in how their video-question pairs are constructed. In general, LongVideoBench features more detailed and specific questions, while Video-MME focuses on concise, high-level queries. Moreover, LongVideoBench tends to ask about specific scenes or events, whereas Video-MME emphasizes global understanding of the video content.

To highlight this distinction, we manually mark the most informative frames relative to the query using yellow stars. These frames are more temporally concentrated in LongVideoBench (around specific events) and more uniformly distributed across the timeline in Video-MME.

This difference helps explain why FOCUS achieves greater performance gains on LongVideoBench: our method assumes that frame-level relevance scores are i.i.d., a common setting in multi-armed bandit formulations. This assumption neglects temporal dependencies between video segments. Consequently, retrieval-based methods for keyframe selection typically require regularization (Tang et al., 2025; Yu et al., 2024) to promote diversity and ensure coverage.

If temporal dependencies between segments (arms) are taken into account, the problem setting shifts toward Lipschitz or Metric Bandits (Kleinberg et al., 2008; Bubeck et al., 2011), or Contextual Bandits (Chu et al., 2011; Agarwal et al., 2014). We leave such extensions to future work.

3.3 COMPARISON WITH STATE-OF-THE-ART

To further validate the effectiveness of FOCUS, we compare it against state-of-the-art training-free keyframe selection methods on both LongVideoBench and Video-MME. Specifically, we consider recent approaches based on vision-language similarity:

- **Top-K**: Computes relevance scores between each frame and the query, then selects the top-K scoring frames. Due to computational constraints, we apply a pre-filtering step by downsampling videos to 1 frame per second.

Figure 3: Comparison between uniformly sampled frames and those selected by FOCUS. The left column shows two examples from LongVideoBench; the right column shows two from Video-MME. Yellow stars indicate manually annotated frames that are most informative to the query, many of which are successfully captured by FOCUS.

Method	LongVideoBench				Video-MME			
	Short	Medium	Long	Overall	Short	Medium	Long	Overall
Uniform	67.5	57.4	51.8	58.9	76.4	62.6	54.3	64.4
Top-K	**72.3**	58.0	60.5	62.3	75.4	60.4	53.0	62.9
AKS	**72.3**	**59.2**	56.1	62.1	76.3	62.8	54.7	64.6
FOCUS (ours)	**72.3**	59.0	**63.7**	**63.5**	**76.5**	**63.5**	**56.1**	**65.4**

Table 2: Comparison between our method and state-of-the-art keyframe selection baselines under matched keyframe count. Results are reported by video length buckets: Short, Medium, and Long. For Video-MME, we adopt its original categorization: *Short* (<2 min), *Medium* (4-15 min), and *Long* (30-60 min). For LongVideoBench, we define *Short* as videos shorter than 3 minutes, *Medium* as 3-20 minutes, and *Long* as over 20 minutes to ensure a balanced distribution.

- **AKS** (Tang et al., 2025): A recent method that adaptively balances frame relevance and temporal coverage. It is considered the current state-of-the-art and also incorporates pre-filtering via downsampling to 1 frame per second (Tang et al., 2025).

Fair comparison protocol. We ensure a fair comparison by: (i) evaluating all methods using LLaVA-Video-7B, the best-performing MLLM in our setup; (ii) fixing the number of selected keyframes to $k = 64$; (iii) using the same vision-language encoder (e.g., BLIP) for frame scoring whenever possible. Results are summarized in Table 2.

Consistency across different lengths. FOCUS achieves consistent performance gains across all video length categories, with particularly strong improvements on long videos. On LongVideoBench, FOCUS outperforms uniform sampling by 11.9% and Top-K by 7.6% on videos longer than 20 minutes. On Video-MME, the respective improvements are 1.8% and 1.4%.

We also observe that on short videos, all keyframe selection methods perform similarly and consistently outperform uniform sampling. We attribute this to a possible saturation in the reasoning capabilities of the underlying MLLM (LLaVA-Video-7B), where input selection plays a less critical role.

Method	Filtering-free	Frames Seen (%)	GPU hours
AKS ***w/o*** pre-filtering	✗	100	255
AKS ***w/*** pre-filtering	✗	3.7	9.3
FOCUS (Ours)	✔	1.6	5.5

Table 3: Efficiency comparison of keyframe selection methods on LongVideoBench. "Pre-filtering" refers to downsampling videos to 1 fps prior to selection. Note that the official AKS pipeline includes this pre-filtering step by default. "Frames Seen (%)" counts the proportion of frame-level BLIP forward passes relative to scoring all frames; GPU hours are measured on a single H100 (80GB).

Efficiency comparison. We report the efficiency of each method in Table 3, measuring both the number of frames "seen" (i.e., scored by a vision-language model) and the total GPU hours required to perform keyframe selection. All GPU hours are measured using a single NVIDIA H100 (80GB) GPU on the LongVideoBench dataset.

As shown, AKS without pre-filtering is computationally infeasible in practice, as it requires scoring all video frames—amounting to over 255 GPU hours by the optimistic estimation. With pre-filtering, the cost drops significantly to 9.3 GPU hours. In contrast, FOCUS is the most efficient: it requires only 1.6% of the BLIP forward passes and just 5.5 GPU hours, while simultaneously achieving the best overall performance.

3.4 EFFICIENCY-ACCURACY TRADE-OFF

FOCUS exposes a natural trade-off between accuracy and computational cost through a single hyperparameter α, which controls the fraction of arms selected for fine-grained exploration. We report accuracy and efficiency under different α settings in Table 4.

	Accuracy (%)	Frames Seen (%)	GPU hours
$\alpha = 0.1$	62.9	1.1	3.5
$\alpha = 0.25$	63.5	1.6	5.5
$\alpha = 0.5$	63.6	2.5	9.2

Table 4: Effect of α on the performance and efficiency of FOCUS. "Frames Seen (%)" counts the proportion of frame-level BLIP forward passes relative to scoring all frames; GPU hours are measured on a single H100 (80GB).

We observe that choice of α has a significant impact on the efficiency while remain stable on the performance. When $\alpha = 0.1$, FOCUS requires around 1.1% of the frames BLIP forward passes while only 3.5 GPU hours. When $\alpha = 0.5$, FOCUS requires around 2.5% of the frames BLIP forward passes while only 9.2 GPU hours.

Exhaustively exploiting all arms would require 9.3 GPU hours, while the performance gain compared to $\alpha = 0.25$ is negligible.

4 CONCLUSION

We addressed the core bottleneck of long-video understanding in MLLMs—the explosion of visual tokens—by introducing FOCUS, a training-free, plug-and-play keyframe selection method that allocates computation under a strict budget. FOCUS first partitions the video into temporal clips, treats each as an arm in a bandit problem, and then identifies query-relevant regions via a combinatorial pure-exploration strategy using empirical means and Bernstein confidence bounds. To improve efficiency, we reduce the iterative bandit process to a coarse-to-fine two-stage procedure that preserves optimism while enabling parallel inference.

Experiments on two challenging long-video QA benchmarks demonstrate that FOCUS consistently improves accuracy across four MLLMs while processing fewer than 2% of video frames. Our results show that lightweight, training-free keyframe selection—when guided by statistical principles—can significantly enhance the scalability and practicality of MLLMs for long-video understanding.

5 Reproducibility Statement

We provide a theoretical analysis of the method in the Appendix B and Appendix C.

References

Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1638–1646, Bejing, China, 22–24 Jun 2014. PMLR. URL https://proceedings.mlr.press/v32/agarwalb14.html.

Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. *Journal of the ACM (JACM)*, 65(3):1–55, 2018.

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. x-armed bandits. *Journal of Machine Learning Research*, 12(46):1655–1695, 2011. URL http://jmlr.org/papers/v12/bubeck11a.html.

Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 208–214. JMLR Workshop and Conference Proceedings, 2011.

Thomas Desautels, Andreas Krause, and Joel W Burdick. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *J. Mach. Learn. Res.*, 15(1):3873–3923, 2014.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24108–24118, 2025.

Aurélien Garivier and Olivier Cappé. The kl-ucb algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th annual conference on learning theory*, pp. 359–376. JMLR Workshop and Conference Proceedings, 2011.

Weiyu Guo, Ziyang Chen, Shaoguang Wang, Jianxiang He, Yijie Xu, Jinhui Ye, Ying Sun, and Hui Xiong. Logic-in-frames: Dynamic keyframe search via visual semantic-logical verification for long video understanding. *arXiv preprint arXiv:2503.13139*, 2025.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pp. 681–690, 2008.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=zKv8qULV6n.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023b.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023a.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023b.

Vianney Perchet, Philippe Rigollet, Sylvain Chassang, and Erik Snowberg. Batched bandit problems. In Peter Grünwald, Elad Hazan, and Satyen Kale (eds.), *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pp. 1456–1456, Paris, France, 03–06 Jul 2015. PMLR. URL https://proceedings.mlr.press/v40/Perchet15.html.

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.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 29118–29128, 2025.

Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pp. 20–36. Springer, 2016.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024a.

Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022.

Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. Internvideo2: Scaling foundation models for multimodal video understanding. In *European Conference on Computer Vision*, pp. 396–416. Springer, 2024b.

Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7): 560–576, 2003.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report, 2024. *URL https://arxiv.org/abs/2407.10671*, 7:8, 2024.

Sicheng Yu, Chengkai Jin, Huanyu Wang, Zhenghao Chen, Sheng Jin, Zhongrong Zuo, Xiaolei Xu, Zhenbang Sun, Bingni Zhang, Jiawei Wu, et al. Frame-voyager: Learning to query frames for video large language models. *arXiv preprint arXiv:2410.03226*, 2024.

Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. Lmms-eval: Reality check on the evaluation of large multimodal models. *arXiv preprint arXiv:2407.12772*, 2024.

Shaojie Zhang, Jiahui Yang, Jianqin Yin, Zhenbo Luo, and Jian Luan. Q-frame: Query-aware frame selection and multi-resolution adaptation for video-llms. *arXiv preprint arXiv:2506.22139*, 2025a.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *Transactions on Machine Learning Research*, 2025b.

A Appendix

A.1 Related Work

A.1.1 Multimodal Large Language Models (MLLMs)

Recent MLLMs extend language models with visual encoders and cross-modal alignment to support open-ended video understanding and VQA (Liu et al., 2023b;a; 2024; Zhang et al., 2023; Maaz et al., 2023b; Zhang et al., 2025b). A common paradigm encodes images or frames into visual tokens that are fused with text inside an LLM. While such models have shown impressive progress, scaling them to hour-long videos exacerbates token and latency bottlenecks: naively increasing sampling rates or sequence lengths rapidly renders inference impractical. These constraints motivate *pre-LLM* selection mechanisms—such as keyframe selection—that reduce visual tokens while preserving task-relevant content for long-video VQA.

A.1.2 Vision-Language Pretrained Models

Contrastive and generative vision-language pretraining provides strong building blocks for frame scoring. CLIP learns image-text alignment via contrastive objectives and is widely adopted for retrieval-style relevance scoring (Radford et al., 2021). The BLIP family introduces unified vision-language pretraining and task heads (e.g., ITM/ITC) that can produce frame-query similarity signals without task-specific training (Li et al., 2022). These models enable training-free or lightly-supervised keyframe selection, but dense per-frame scoring on long videos remains expensive; thus, efficient sampling and exploration strategies are crucial to realize their full potential under tight inference budgets.

A.1.3 Keyframe Selection

Classical approaches include fixed-rate *uniform sampling*, shot/change detection, or saliency heuristics. Uniform sampling is simple and budget-friendly but query-agnostic, often missing short, critical events. Earlier RL-based summarization methods can be adaptive but typically require non-trivial training and may show unstable generalization.

Recent methods leverage vision-language similarity to *query-condition* the selection. AKS (Tang et al., 2025) uses BLIP-based scoring to balance relevance and temporal coverage via an optimization objective, providing a plug-and-play selector for VQA pipelines. Q-Frame (Zhang et al., 2025a) proposes a query-aware frame selection module with multi-resolution adaptation, directly predicting informative frames for a given question. Frame-Voyager (Yu et al., 2024) learns to query frames for video-LLMs under limited budgets, and Logic-in-Frames/VSLS (Guo et al., 2025) incorporates semantic-logical verification to search keyframes that support reasoning. While these methods improve over naive baselines, many either (i) require task-specific training, (ii) incur high inference costs by densely scoring frames (even after downsampling), or (iii) depend on complex hand-crafted logic for coverage/diversity. Our work differs by casting clip-level selection as a *bandit exploration* problem with a variance-aware index and explicit batching, yielding a budget-aware, GPU-friendly pipeline that avoids full scanning.

A.1.4 Multi-Armed Bandits and Batched Exploration

Bandits formalize explore–exploit trade-offs under limited budget. UCB (Auer et al., 2002) and its variance-aware variant UCB-V (Audibert et al., 2009) provide optimism-in-the-face-of-uncertainty indices with finite-time guarantees, while KL-UCB refines confidence bounds for bounded rewards (Garivier & Cappé, 2011). For parallel settings, GP-BUCB introduces hallucinated ("fantasy") updates to select batches without sacrificing theoretical performance up to constant factors (Desautels et al., 2014). Batched bandits study regret under a small number of interaction rounds, showing strong performance with limited adaptivity (Perchet et al., 2015). Budgeted extensions such as Bandits with Knapsacks (BwK) incorporate resource constraints directly into decision-making (Badanidiyuru et al., 2018). Our approach adapts these principles to long-video VQA: we treat each temporal clip as an arm, use a variance-aware UCB-V score to drive selection, and adopt batched (GPU-parallel) exploration with simple budget guards. This yields an efficient keyframe extractor that integrates seamlessly with CLIP/BLIP scoring and standard VQA pipelines.

B Bernstein Confidence Radius

Theorem B.1. *Let $N_a(n)$ be the number of pulls for arm a at round n and $n = \sum_{a\in\mathcal{A}} N_a(n)$ is the total number of pulls. Let $\hat{\mu}_a(n)$ be the empirical mean of arm a at round n and $\hat{\sigma}_a^2(n)$ be the empirical variance of arm a at round n. We define the empirical Bernstein Confidence Radius $\beta_a(n)$ as*

$$\beta_a(n) \;=\; \sqrt{\frac{2\,\hat{\sigma}_a^2\,\ln n}{\max(1, N_a(n))}} \;+\; \frac{3\,\ln n}{\max(1, N_a(n))}.$$

Then we have the following bound holds with probability at least $1-\delta$:

$$\mathcal{P}\left[|\hat{\mu}_a - \beta_a(n)| \le \mu_a\right] \ge 1 - \frac{6}{n}.$$

Proof. Under the setting of frame-query relevance setting, the reward r_t and latent frame reward y_t is naturally bounded in $[0,1]$. Therefore, according to Bernstein inequality, for any $\delta \in (0,1)$, we have

$$\mathcal{P}\left[\mu_a \le \hat{\mu}_a(n) + \sqrt{\frac{2\hat{\sigma}_a^2 \ln\frac{3}{\delta}}{N_a(n)}} + \frac{3\ln\frac{3}{\delta}}{N_a(n)}\right] \ge 1-\delta.$$

And symmetrically, we have

$$\mathcal{P}\left[\mu_a \ge \hat{\mu}_a(n) - \sqrt{\frac{2\hat{\sigma}_a^2 \ln\frac{3}{\delta}}{N_a(n)}} - \frac{3\ln\frac{3}{\delta}}{N_a(n)}\right] \ge 1-\delta.$$

Therefore, we have

$$\mathcal{P}\left[|\hat{\mu}_a - \mu_a| \le \sqrt{\frac{2\hat{\sigma}_a^2 \ln\frac{3}{\delta}}{N_a(n)}} + \frac{3\ln\frac{3}{\delta}}{N_a(n)}\right] \ge 1-2\delta.$$

Choose $\delta = \frac{3}{n}$, then we have

$$|\mu_a - \hat{\mu}_a(n)| \le \sqrt{\frac{2\hat{\sigma}_a^2 \ln\frac{3}{\delta}}{N_a(n)}} + \frac{3\ln\frac{3}{\delta}}{N_a(n)}.$$

holds with probability at least $1-\frac{6}{n}$.

When $N_a(n) = 0$, the statement is trivially true. Thus, we have the following bound holds with probability at least $1-\frac{6}{n}$:

$$|\mu_a - \hat{\mu}_a(n)| \le \beta_a(n).$$

□

C Regret Bound

Arm-level Regret Bound

Theorem C.1. *Algorithm 2 returns the oracle top-s set $S^\star$ with probability at least $1 - \frac{6M}{n}$ when terminated.*

Proof. When Algorithm 2 terminates, the following condition holds:

$$\max_{a \notin \hat{S}} \hat{\mu}_n(a) + \beta_a(n) \leq \min_{a \in \hat{S}} \hat{\mu}_n(a) - \beta_a(n).$$

According to Theorem B.1, with probability at least $1 - \frac{6}{n}$, we have $|\mu_a - \hat{\mu}_a(n)| \leq \beta_a(n)$ for all arms a. Therefore, for any $a \notin \hat{S}$,

$$\mathcal{P}\left[a \in S^\star\right] \leq 1 - \frac{6}{n}.$$

Thus, the probability that there does not exist such an arm a is at least $1 - \frac{6(M-m)}{n}$, where m is size of the $\hat{S}$ set. And this completes the proof. □

Frame-level Regret Bound We define the frame-level regret as the difference between the optimal frame-level reward and the reward of the selected frames.

$$r_N^{\text{frame}} = \sum_{t \in \mathbb{K}^\star} y_t - \sum_{t \in \widehat{\mathbb{K}}_n} y_t.$$

As long as we obtain the oracle top-s set $S^\star$, the frame-level regret is also guaranteed to be small. As Frame-level sampling is actually finite so we can always find the top-k frames with the highest rewards.

$$\mathbb{E} r_N^{\text{frame}} = \mathbb{E} \sum_{t \in \mathbb{K}^\star} y_t - \sum_{t \in \widehat{\mathbb{K}}_n} y_t = \mathbb{E} \sum_{a \in S^\star} \sum_{t \in \mathbb{K}_a^\star} 2\epsilon_\psi = 0.$$

For tighter bound, we leave this to future work.

D Limitations

In this work, we assume the frame-query relevance scores are i.i.d. and the temporal dependencies between frames are not considered. However, in practice, the frame-query relevance scores are dependent on the temporal dependencies between frames. As different parts may have strong correlations, this assumption may not hold. In this setting, we can use the Lipschitz/Metric Bandit problem (Kleinberg et al., 2008; Bubeck et al., 2011) or Contextual Bandit problem (Chu et al., 2011; Agarwal et al., 2014) to model the problem. We leave this as future work.

E The Use of Large Language Models (LLMs)

We used GPT-5 and Claude 4 solely for proofreading and light copy-editing (typos, grammar, and minor phrasing). All technical content, scientific claims, mathematical proofs, algorithms, experiment design and execution, dataset handling, figures, and evaluations were authored and verified by the human authors. LLMs were not used to generate ideas, code, data, results, or reviews; they did not contribute content at the level of a co-author. All suggested edits were manually inspected and accepted or rejected by the authors.