Appendix

A. Implementation Details

We mostly focus on longer videos because better frame selection plays a bigger role in longer, more complex videos, whereas shorter ones intuitively work well with uniform sampling due to their lower information content and complexity.

We ensured that the resulting sequence length of a set of visual and textual tokens did not exceed the maximum sequence length for this LLM. When evaluating models using MaxInfo, we limited the number of selected frames so that they did not exceed the maximum allowed for the context of the estimated VLLM. For the evaluation on all benchmarks, we have set the generation temperature to 0.

For the general multiple-choice question-answering evaluation, we follow the official guidelines to construct the instructions using the provided questions and options. We added a prompt to the question and options like *"Respond with only the letter (A, B, C, or D) of the correct option."* for LongVideoBench [40], Video-MME [11], MLVU [53] and MVBench [20] or *"Answer with the option's letter from the given choices directly and only give the best option."* for EgoSchema [23]. We follow the original benchmarks setup to calculate the final scores, and we also align our evaluation protocols with other evaluation toolkits, such as lmms-eval [49].

To ensure the reproducibility of our results, we have included the main hyperparameters used for all benchmarks and estimated models in the results tables, such as tolerance and rank for the MaxInfo algorithm, the number of sampled frames, and the number of initial frames (before MaxInfo).

B. Additional Experiments and Details

To further assess the impact of MaxInfo, we evaluate its performance with an additional set of models [37], [46] on the LongVideoBench and Video-MME benchmarks.

B.1. Applying MaxInfo to recent models

The results in Table 6 show that MaxInfo consistently improves model performance across both benchmarks, suggesting that precise frame selection is particularly important for long-video tasks.

B.2. Performance Analysis: MaxInfo vs. Uniform Sampling

To better understand the strengths and trade-offs of MaxInfo, we analyzed per-task accuracy across multiple benchmarks. Our results, as shown in Figure 6, indicate that MaxInfo performs superiorly in high information density tasks such as counting, summarizing and spatial reasoning, while uniform sampling has a slight advantage in tasks that rely on temporal continuity, reflecting the key trade-off between information maximization and temporal consistency.

Table 6. Adaptation of MaxInfo to current new long video understanding models

Model	**Size**	**Frame Interval**	**Avg. frames**	**LongVideoBench.**
MiniCPM [46]	9B	128	128	56.17
+ MaxInfo	9B	[8, 82]	56	59.61
△				+3.44
InternVL3.5 [37]	1B	16	16	47.7
+ MaxInfo	1B	[1, 16]	16	49.0
△				+1.3
InternVL3.5 [37]	8B	16	16	57.4
+ MaxInfo	8B	[1, 16]	16	59.0
△				+1.6
InternVL3.5 [37]	38B	16	16	60
+ MaxInfo	38B	[1, 16]	16	61.6
△				+1.6

B.3. Comparison with CLIP baseline

As shown in Table 7, we compare the experimental results of two keyframe extraction strategies based on the QwenVL2-2B model on the LongVideoBench benchmark: the **CLIP-Based** thresholding method and the **MaxInfo** module method. Both methods extract the same number of frames in the initial phase, so the encoding time is kept the same, where the similarity threshold of the CLIP-Based method is set to 0.5. The results show that the MaxInfo module outperforms the CLIP-Based method in terms of the overall performance in keyframe selection.

Table 7. Performance comparison on LVBench.

Model	Method	Accuracy
QwenVL2-2B	CLIP-Based	44.3
QwenVL2-2B	CLIP-Based + MaxInfo	44.5
QwenVL2-2B	MaxInfo + CLIP-Based	43.8
QwenVL2-2B	MaxInfo	**48.8**

In addition, we also explored combining the CLIP-Based method with MaxInfo module. The experiments show that MaxInfo is able to improve the overall information quality of the input sequences, and its information maximization strategy plays a key role in frame selection, which further enhances the performance of the model. CLIP-Based loses a lot of semantic information, which can lead to performance degradation of the model.

In order to further evaluate whether MaxInfo will lose the key frames related to the problem, we compare MaxInfo with the Uniform Sampling method under the CLIP Score metric. The experimental results shown in Table 8 that MaxInfo does not miss the frames related to the semantics of the problem, and is able to retain the semantic relevance effectively.

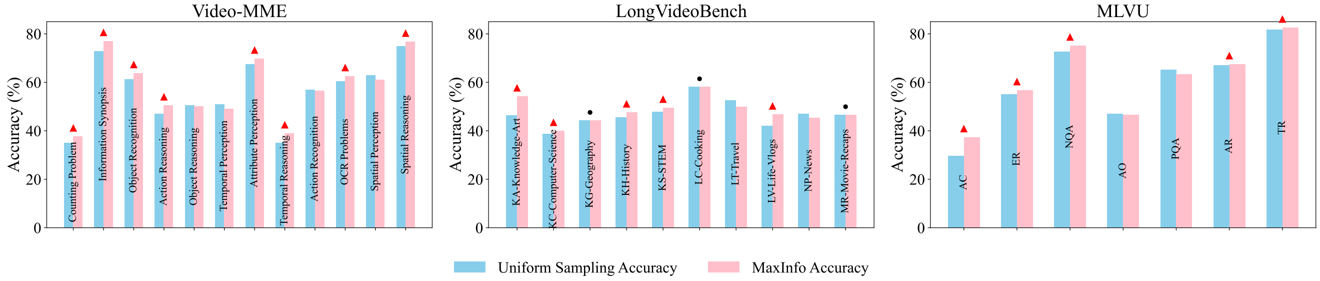

Figure 6. Accuracy comparison between Uniform Sampling and MaxInfo across three benchmarks.

Table 8. CLIP score comparison between uniform and MaxInfo sampling.

Sampling Method	CLIP-score
Uniform	0.37
MaxInfo	**0.39**

B.4. Qualitative comparison with uniform samplir

We randomly selected 50 video samples LongVideoBench and calculated the cosine similari between the frames selected by MaxInfo and the cosine similarity between the frames obtained by uniform sampling.

Figure 7 shows the distribution of cosine similarity for the same number of frames. It is clear that MaxInfo produces a more diverse distribution like a low similarity offset compared to uniform sampling, highlighting its ability to capture more diverse visual content.

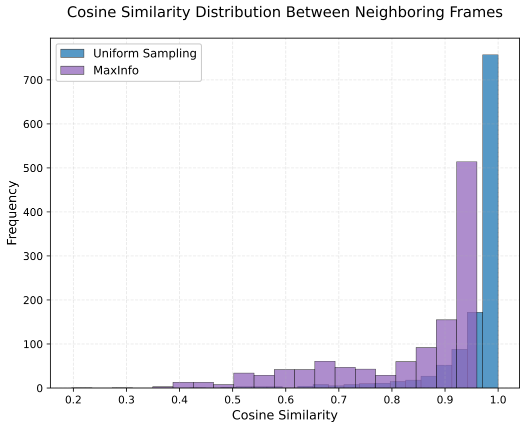

Figure 7. Similarity distribution between neighbouring frames ($frame_i$ and $frame_{i+1}$).

As shown in Figure 8, we plotted 200 sampled data points to improve visual clarity. The results show that our MaxInfo module exhibits higher diversity in frame selection compared to uniform sampling.

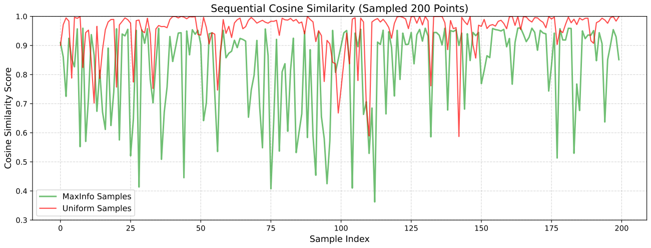

Figure 8. CLIP similarity between neighboring frames selected by MaxInfo module.

B.5. Computational Efficiency: Time and Memory Consumption

When processing long videos, the LLM is the most resource-intensive component of VLLMs due to its parameter count and the quadratic complexity of attention with respect to input length. Since most of the context is occupied by visual tokens from frames, our MaxInfo method reduces this load by selecting keyframes. Importantly, MaxInfo requires minimal and constant memory and preprocessing time, independent of LLM size, and remains significantly lighter than uniformly sampling all frames.

Time Complexity. To evaluate the latency overhead of MaxInfo in practice, we measured its runtime with the Qwen2-VL model. As shown in Table 9, the runtime of MaxInfo is almost negligible compared to the inference time of the VLLM itself, confirming that MaxInfo is a lightweight and efficient frame selection mechanism. The initial VLLM time means the inference time of the 512 frames of information directly into the Qwen2-VL model. The frame count selected by MaxInfo Block is adaptive to the information content of the input. For near-static videos (low information density), MaxInfo drastically reduces the number of processed frames. Consequently, VLLMs +

MaxInfo Block may achieve lower time compared to the initial VLLMs + MaxInfo Block configuration. The times reported in the table represent an upper bound; in practice, the reduced number of frames can lead to several-fold speedups on certain tasks. All experiments were conducted on an A100 GPU.

Table 9. Runtime of different pipeline components, based on Qwen2-VL. Frame size = 512 (UP is Upper Bound).

Model Size	CLIP (s)	MaxVol (s)	VLLMs (s)	VLLMs + MaxInfo (UP)
2B	0.296	0.0109	2.979	≤ 3.285
7B	0.296	0.0109	5.372	≤ 5.679
72B	0.296	0.0109	30.737	≤ 31.044

We also analyzed the running time of the MaxVol algorithm alone, including its chunk-based variant, under different initial numbers of frames, as shown in Table 10. The experimental results show that the running time of MaxVol remains low across settings, with minimal impact on the overall inference efficiency.

Table 10. MaxVol algorithm runtime (excluding image encoding time) for different input sizes.

Method	Input Size	MaxVol Time (s)
MaxInfo	128	0.0044
MaxInfo	256	0.0053
MaxInfo	512	0.0109
Chunks-Based MaxInfo	32×32	0.0375

Then we estimated CUDA inference time across different VLLM sizes which is shown in Figure 9. The overhead of MaxInfo remains small and nearly constant, while the overall inference time grows with model size, demonstrating that MaxInfo adds minimal cost compared to the savings from reduced visual tokens. For small models (up to 8B parameters), the relative benefit is limited since inference cost is low. However, for larger models (26B–76B), MaxInfo provides clear efficiency gains by substantially reducing the number of visual tokens, making its impact especially pronounced for long-video tasks where input length dominates computational cost.

Memory Consumption. Secondly, we precisely evaluated memory consumption of out approach. As shown in Figure 10, MaxInfo's CUDA memory usage remains constant for a fixed number of initial frames and grows much more slowly than uniform sampling as LLM size increases.

In summary, our analysis of time and memory efficiency shows that MaxInfo introduces only negligible overhead while substantially reducing the computational burden of processing long videos. Its constant preprocessing cost and slower growth in memory usage make MaxInfo particularly advantageous for large-scale VLLMs, with the benefits becoming most pronounced for models exceeding 10B parameters.

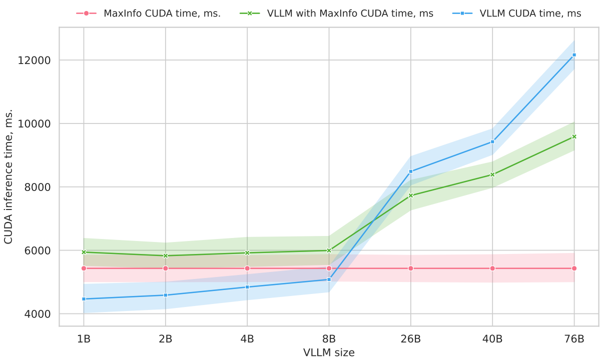

Figure 9. CUDA inference time across different VLLM sizes. The preprocessing cost of MaxInfo remains small and nearly constant, while overall inference time increases with model size.

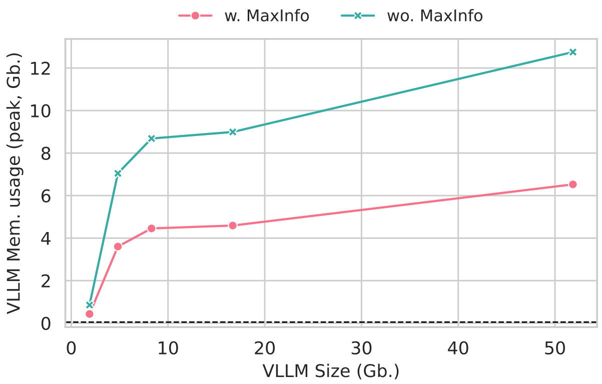

Figure 10. The comparison for memory performance on the GPU for InternVL2 models with and without MaxInfo module. The dashed line shows the CUDA memory requirements for MaxInfo.

C. Theoretical Justification

Definition 1. *Definition of the maximum volume of the video frame feature matrix.*

We consider a matrix $Q \in \mathbb{C}^{N \times r}$, where each row represents the CLIP or SigLIP etc. feature of a video frame, ordered sequentially in time, where N denotes the number of frames and r denotes the dimension of the feature.

We aim to identify a submatrix $\hat{Q} \in \mathbb{C}^{K \times r}$ of the original matrix S, such that $\hat{S}$ closely approximates S in terms of matrix volume, thereby preserving its essential structural information.

To obtain the submatrix $\hat{Q}$, we introduce a coefficient matrix C based on the minimum-norm linear combination as Equation 10

$$\tilde{C}\hat{Q} = \tilde{Q} \tag{10}$$

Here, $\tilde{Q}$ denotes a set of sample rows selected from the original matrix Q for reconstruction. By solving for $\tilde{C}$, we can approximate the reconstruction of $\tilde{Q}$ using only the representative rows in $\hat{Q}$. In addition, it is shown that the selected K rows are the most representative of the video frame information.

Solving. The submatrix $\hat{Q} \in \mathbb{C}^{K\times r}$ provides an approximation of the original matrix $Q \in \mathbb{C}^{N\times r}$ within a tolerance τ.

We start with an initial submatrix $\hat{Q} \in \mathbb{C}^{M\times r}$ and add a row $Q_i \in \mathbb{C}^{1\times r}$ to each iteration to bring the expanded submatrix up to speed in the sense of volume. The updating process can be expressed as follows Equation 11 and the volume of the updated matrix can be defined as Equation 12

$$\hat{Q} \leftarrow \begin{bmatrix} \hat{Q} \\ Q_i \end{bmatrix} \tag{11}$$

$$\mathrm{Vol}(\hat{Q})_{\text{new}} = \mathrm{Vol}(\hat{Q})_{\text{old}} \cdot \sqrt{1 + \|\tilde{C}_i\|_2^2} \tag{12}$$

where Q_i is the row selected from the original matrix $Q \in \mathbb{C}^{N\times r}$ that currently boosts the volume of the submatrix the most. Repeat this process iteratively until the conditional Equation 13 is satisfied or the target number of K rows is reached.

$$\|\tilde{C}_i\|_2 \leq \tau \tag{13}$$

Proof of maximum information entropy. To justify our approach, we use differential entropy as an information measure. Suppose our normalized frame embeddings form a matrix S. The differential entropy of a uniform distribution over the convex hull $\mathcal{C}(S)$ is given by the following Equation 14.

$$H_{\max}(S) = \ln(\mathrm{Vol}(\mathcal{C}(S))) \tag{14}$$

where $\mathrm{Vol}(\mathcal{C}(S))$ is the volume of the convex hull formed by selected embeddings. Classical results show the following Equation 15.

$$\mathrm{Vol}(\mathcal{C}(S)) = \kappa\sqrt{\det(S^\top S)} \tag{15}$$

for some constant $\kappa > 0$. Thus we get Equation 16

$$H_{\max}(S) = \ln V(S) + \text{constant} \tag{16}$$

where $V(S) = \sqrt{\det(S^\top S)}$. Since MaxVol maximizes $V(S)$, it maximizes the upper bound on differential entropy, ensuring that selected frames are more informative.

In summary, we can theoretically select the most representative frame information. The feature matrices corresponding to the selected frames have good linear independence under the constraint of the tolerance parameter τ, thus constituting an approximately optimal subset of the representation. This process achieves our goal of **information maximization**, i.e. preserving the most critical structural information while compressing redundancy.

D. Societal Impacts

This work introduces a training-free framework for improving frame sampling in Vision-Language Large Models (VLLMs), enhancing video understanding tasks. Such advancements have important implications for applications in education, accessibility, and public safety.

However, improved video analysis capabilities may also raise ethical concerns, including potential misuse in surveillance, privacy violations, or biases affecting different communities. Ensuring responsible deployment with fairness and transparency is essential to mitigate these risks.

In summary, while our approach provides significant benefits, its adoption should adhere to ethical principles to promote equitable and responsible use.