mDP^3: A Training-free Approach for List-wise Frame Selection in Video-LLMs

Supplementary Material

Appendix Contents

A. Proofs

A.1. Preliminaries

Definition 1 (Submodular Maximization [36]). *Let Ω denote a finite set, and let $f : 2^\Omega \to \mathbb{R}_{\geq 0}$ be a set function, where 2^Ω is the power set of Ω. The function f is called submodular if it satisfies one of the following three equivalent conditions:*

- *For every $X, Y \subseteq \Omega$ with $X \subseteq Y$, and for all $x \in \Omega \setminus Y$, we have*

$$f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y)\,, \tag{20}$$

- *For every $X, Y \subseteq \Omega$, we have*

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y)\,, \tag{21}$$

- *For every $X \subseteq \Omega$ and $x_1, x_2 \in \Omega \setminus X$ such that $x_1 \neq x_2$, we have*

$$f(X \cup \{x_1\}) + f(X \cup \{x_2\}) \geq f(X \cup \{x_1, x_2\}) + f(X)\,. \tag{22}$$

These three conditions are equivalent, and the first condition is the most commonly used, as it directly reflects the law of diminishing marginal utility as the number of items increases.

Definition 2 (Sub-additive). *A set function $f : 2^\Omega \to \mathbb{R}_{\geq 0}$ is sub-additive if for every two sets $X, Y \in \Omega$, we have*

$$f(X \cup Y) \leq f(X) + f(Y)\,. \tag{23}$$

Lemma 1. *A non-negative submodular set function $f : 2^\Omega \to \mathbb{R}_{\geq 0}$ is sub-additive.*

Proof. As the second condition Eq. (21) in Definition 1. for $X, Y \subseteq \Omega$, we have $f(X)+f(Y) \geq f(X \cup Y)+f(X \cap Y)$. So, $f(X) + f(Y) \geq f(X \cup Y)$ as $f(X \cap Y) \geq 0$. □

Lemma 2. *Let $f : 2^\Omega \to \mathbb{R}$ be submodular. Let $S \subseteq \Omega$, and $f_S(X) = f(S \cup X) - f(S)$ for every $X \subseteq \Omega$. (f_S is the marginal value function for set S.) Then f_S is also submodular.*

Proof. Let $X, Y \subseteq \Omega \setminus S$; it suffices to consider ground set $\Omega \setminus S$.

$$\begin{aligned}
&(f_S(X \cup Y) + f_S(X \cap Y)) - (f_S(X) - f_S(Y)) \\
&\quad = f(S \cup X \cup Y) - f(S) + f(S \cup (X \cap Y)) - f(S) \\
&\qquad - (f(S \cup X) - f(S) + f(S \cup Y) - f(S)) \\
&\quad = f(S \cup X \cup Y) + f(S \cup (X \cap Y)) \\
&\qquad - f(S \cup X) - f(S \cup Y) \\
&\quad \leq 0\,.
\end{aligned} \tag{24}$$

The last inequality is by $S \cup X \cup Y = (S \cup X) \cup (S \cup Y)$, $S \cup (X \cap Y) = (S \cup X) \cap (S \cup Y)$ and submodularity of f. Therefore, f_S is also submodular is proved. □

A.2. Proof of $(1 - 1/e)$-approximation

Submodular maximization is NP-hard in general. Therefore, most research in this field focuses on approximation algorithms with polynomial-time complexity. While the submodular function is monotone, *i.e.*, for every $X, Y \subseteq \Omega$, we have $f(X) \leq f(Y)$. The problem of maximizing a monotone submodular function subject to a cardinality constraint admits a $(1 - 1/e)$-approximation greedy algorithm (as introduced in Sec. 3.2) [42].

In this section, we provide a concise proof of the $1 - 1/e$ approximation ratio for the greedy algorithm.

Theorem 1 ($(1 - \frac{1}{e})$-approximation of Greedy Algo.)**.** *There exists a greedy algorithm for the submodular maximization problem, which starts with an empty set $S = \emptyset$ and iteratively selects the item that maximizes the marginal gain:*

$$j = \underset{i \in \Omega \setminus S}{\operatorname{argmax}} \quad f(S \cup \{i\}) - f(S)\,. \tag{25}$$

The algorithm continues until the selected set S reaches the cardinality limit k.

This greedy algorithm provides a solution $\hat{S}$, which guarantees a $(1 - 1/e)$ approximation, where the optimal solution is denoted as S^*:

$$f(\hat{S}) \leq f(S^*) \tag{26}$$

Proof. According to Lemma 2, f_S is submodular, and by Lemma 1, it is also sub-additive. Therefore, we have:

$$f_S(S^*) \leq \sum_{x \in S^*} f_S(x)\,, \tag{27}$$

which implies that:

$$\exists x \in S^*, \quad f_S(x) \geq \frac{1}{k} f_S(S^*)\,. \tag{28}$$

For this x, we have the following margin lower bound:

$$f(S \cup \{x\}) - f(S) \geq \frac{f(S^*) - f(S)}{k}\,. \tag{29}$$

Let S_t denote the selected subset after the t-th step of the greedy algorithm. According to Eq. (25), in the greedy algorithm, we have:

$$f(S_{t+1}) - f(S_t) \geq f(S_t \cup \{x\}) - f(S_t), \quad \forall x \in \Omega \setminus S_t\,. \tag{30}$$

Therefore, in the greedy algorithm, the marginal gain is lower-bounded by:

$$f(S_{t+1}) - f(S_t) \geq \frac{f(S^*) - f(S_t)}{k}\,. \tag{31}$$

This implies:

$$f(S^*) - f(S_{t+1}) \leq \left(1 - \frac{1}{k}\right)(f(S^*) - f(S_t))\,. \tag{32}$$

Hence, when the greedy algorithm selects a subset $\hat{S} = S_k$ after k steps, we have:

$$\begin{aligned} f(S^*) - f(S_k) &\leq \left(1 - \frac{1}{k}\right)(f(S^*) - f(S_{k-1})) \\ &\leq \left(1 - \frac{1}{k}\right)^2 (f(S^*) - f(S_{k-2})) \\ &\vdots \\ &\leq \left(1 - \frac{1}{k}\right)^k (f(S^*) - f(S_0))\,, \end{aligned} \tag{33}$$

where S_0 is the initial set at $t = 0$, with $S_0 = \emptyset$, such that $f(S_0) = 0$. Therefore, we obtain:

$$f(S^*) - f(\hat{S}) \leq \left(1 - \frac{1}{k}\right)^k f(S^*)\,. \tag{34}$$

Hence, we have:

$$\begin{aligned} f(\hat{S}) &\geq \left(1 - \left(1 - \frac{1}{k}\right)^k\right) f(S^*) \\ &\geq \lim_{k \to +\infty} \left(1 - \left(1 - \frac{1}{k}\right)^k\right) f(S^*) \\ &= \left(1 - \frac{1}{e}\right) f(S^*)\,. \end{aligned} \tag{35}$$

The proof is complete. □

Previous works [14, 19] have established that the determinantal point process (DPP) is a monotone submodular function. Therefore, when selecting 8 frames using the standard DPP, the approximation ratio is at least $1 - \left(1 - \frac{1}{8}\right)^8 = 65.6\%$.

A.3. Proof of Time Complexity

The time complexity of Algorithm 1 is closely $\mathcal{O}(nk^3)$, where n represents the number of candidate frames and k denotes the selection size.

Proof. As indicated by the pseudocode in Algorithm 1, there are three loops, with the iteration sizes specified as follows:

- Line 3: $\lceil \frac{n}{m} \rceil$;
- Line 6: $k + 1$;
- Line 7: $k_{t-1} \leq \min(m, k, k - C_{t-1}) \leq \min(m, k)$.

Here, m denotes the segment size. The DPP update in each iteration (Line 9) has a time complexity of $\mathcal{O}(mk_{t-1})$, utilizing Cholesky decomposition for incremental computation.

Therefore, the overall time complexity can thus be expressed as:

$$\mathcal{O}\left(\frac{n}{m} \cdot (k \cdot mk_{t-1}^2)\right) = \mathcal{O}\left(n \cdot \min(m^2k, k^3)\right) \tag{36}$$

Additionally, the time complexity of computing the determinant in line 5 is $\mathcal{O}(m^3)$, resulting in the following time complexity:

$$\mathcal{O}\left(\frac{n}{m} \cdot m^3\right) = \mathcal{O}\left(nm^2\right) \tag{37}$$

Next, we analyze the relationship between m and k, leading to the following total time complexity:

$$\begin{cases} \mathcal{O}(nm^2), & \text{if } k < m^{2/3} \\ \mathcal{O}(nk^3), & \text{if } m^{2/3} \leq k < m \\ \mathcal{O}(nm^2k) < \mathcal{O}(nk^3), & \text{if } k \geq m \end{cases} \tag{38}$$

In practice, the segment size m is typically small, so in most practical cases, the time complexity of Algorithm 1 is upper-bounded by $\mathcal{O}(nk^3)$. □

This algorithm exhibits pseudo-polynomial time complexity, analogous to the knapsack problem. In the dynamic programming approach proposed in this paper, the runtime of the pseudo-polynomial complexity is practically comparable to that of a polynomial-time algorithm.

In practical applications, the segment size m and selection size k are both much smaller than the total number of video frames n, *i.e.*, $m \ll n$ and $k \ll n$. Thus, regardless of the relationship between m and k, the total time complexity is much smaller than feeding all frames into transformer-based LLMs, which have a time complexity of $O((n \cdot \#\text{tokens\_per\_image})^2)$ per layer and attention head.

Additionally, the iteration in line 6 of Algorithm 1 is independent, enabling parallel updates. This results in a time complexity of $\mathcal{O}(nk^2)$, making its efficiency comparable to that of the standard DPP. Moreover, without lazy strategies and parallel updates, the time complexity is closely $\mathcal{O}(nk^4)$ in most practical cases, and the proof is similar.

B. Experiments Details and More Discussions

B.1. Experiments Compute Resoureces

We integrate MDP^3 as a plug-and-play process during inference with 7B SOTA Video-LLMs. We conduct experiments using LMMs-Eval [21] and VLMEvalKit [11] on three long video benchmarks. All experiments are run on NVIDIA A100-PCIE-40GB or NVIDIA A100-PCIE-80GB GPUs, using 256 AMD EPYC 7H12 64-Core @ 2.600GHz CPUs, with Ubuntu 20.04.6 as the operating system, adhering to a rigorous experimental protocol to ensure fair comparisons among compared methods.

B.2. Dataset Details

Video-MME [13]: Video Multi-Modal Evaluation (Video-MME) is a dataset designed to enhance video understanding for Multimodal Large Language Models (MLLMs). It consists of 900 videos spanning 6 visual domains, with durations ranging from 11 seconds to 1 hour, capturing a variety of contextual dynamics. All videos are manually annotated by experts, generating 2700 question-answer pairs, ensuring high-quality and reliable data for model evaluation. Experiments on Video-MME will be conducted both with and without subtitles to assess the impact of multi-modal inputs.

MLVU [75]: Multi-task Long Video Understanding Benchmark (MLVU) is a new dataset designed to evaluate Long Video Understanding (LVU) performance. It addresses the limitations of existing benchmarks by offering longer video durations, diverse video genres (such as movies, surveillance footage, and cartoons), and a range of evaluation tasks. The benchmark includes 2593 tasks across 9 categories, with an average video duration of 12 minutes, providing a comprehensive assessment of MLLMs' capabilities in understanding long videos. This allows for a more comprehensive assessment of MLLMs' capabilities in understanding long videos.

LongVideoBench [61]: It is a recent benchmark designed to evaluate long-term video-language understanding for MLLMs. It consists of 3763 web-collected videos of varying lengths, up to one hour, with subtitles, covering a wide range of themes. The dataset is tailored to assess models' ability to process and reason over detailed multimodal information from long video inputs. It includes 6678 human-annotated multiple-choice questions across 17 fine-grained categories, making it one of the most comprehensive benchmarks for long-form video understanding. In this paper, we focus on the validation set without subtitles, denoted as LVB$_{val}$, which contains 1337 question-answer pairs and has an average video length of 12 minutes.

B.3. Implementation Details

Visual Encoder: The primary baselines, LLaVA-OneVision-7B [22] and MiniCPM-V2.6-7B [66], both use SigLIP [71] as the visual encoder, which we also adopt as the VLM in Eq. (2) to avoid including excessive additional parameters. Additionally, since the context length of the text encoder in SigLIP is 64, potentially insufficient for the entire question, we split the text sequence into multiple sequences of equal length, each no longer than 64 tokens. We then extract multiple text embeddings and aggregate them into a final text embedding using pooling.

Multi-kernel: Eq. (3) presents a principled approach for selecting the optimal kernel. Factors exist to combine positive semi-definite (PSD) kernels. For implementation, we use the Gaussian kernel as the base PSD kernel, defined by $k(x, y) = \exp\left(-\frac{\|(x-y)/h\|^2}{2\sigma^2}\right)$. Consequently, the base kernel with a combination factor can be expressed as:

$$\beta_u \cdot k_u(x, y) = \beta_u \cdot \exp\left(-\frac{\|x - y\|^2}{2(h_u\sigma_u)^2}\right). \quad (39)$$

We denote $\alpha_u = (h_u\sigma_u)^2$ as a single hyperparameter. Following the multi-kernel maximum mean discrepancy (MK-MMD) framework [35], the optimal β_u can be optimized using a quadratic program (QP). However, optimizing β_u is orthogonal of this work. In the recent official implementation of Long et al. [35], average weights $\beta_u = 1/U$ were employed, yielding good performance. Hence, we adopt average weights for β_u and concentrate on configuring α_u. Consistent with Long et al. [35], Sun and Li [49], for both $g, k \in \mathcal{K}$, we set α_u to 2^i where $i \in \{-3, -2, 0, 1, 2\}$ and use an averaged ensemble of multiple Gaussian kernels. Besides, the difference between g and k is modulated by λ.

Hyper-parameters: We set the trade-off hyperparameter $\lambda = 0.2$ and the segment size $m = 32$ for all tasks and benchmarks in MDP^3, determined through cross-validation on a subset of the LLaVA-OneVision-mid [22] training set.

B.4. Hyperparameter Sensitivity Analysis

To evaluate the robustness of MDP^3, we perform a sensitivity analysis of the score-trade-off parameter λ and the segment size m using the VideoMME benchmark (without subtitles). The results are shown in Figs. 4 and 5.

Score-trade-off (λ). Figure 4 varies λ logarithmically from 0.0125 to 12.8. Accuracy stays within a narrow range of 51–53 %, peaking at 53.3 % when $\lambda = 0.2$; every value surpasses the baseline of 49.2 %.

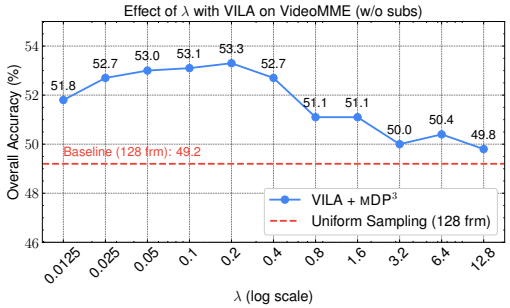

Figure 4. Hyper-parameter sensitivity with trade-off λ.

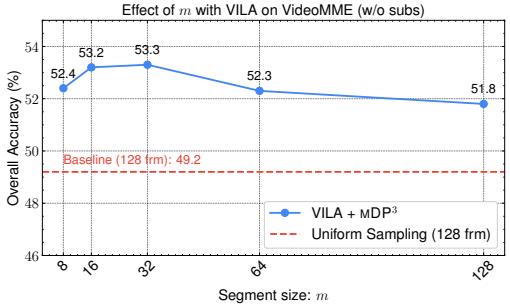

Figure 5. Hyper-parameter sensitivity with segment size m.

Segment size (m). Figure 5 evaluates segment sizes ranging from 8 to 128 frames. All configurations exceed the baseline. The best accuracy, 53.3 %, is achieved at $m = 32$; accuracy stays above 52 % for $m \leq 64$ and remains competitive (51.8 %) even at $m = 128$. These findings indicate that MDP^3 is insensitive to the choice of m.

Across all examined values of λ and m, MDP^3 consistently outperforms uniform sampling, confirming its robustness and practical ease of tuning.

B.5. Additional Parameters Analysis

MDP^3 is a training-free, model-agnostic method that leverages pretrained VLMs. The primary baselines, VILA-V1.5-8B [29], MiniCPM-V2.6-7B [66], and LLaVA-OneVision-7B [22], all integrate the vision encoder from SigLIP [71], so MDP^3 only needs to introduce the additional parameter from the text encoder in SigLIP. The parameter scales of them are reported in Tab. 4. The results indicate that the additional parameters from the text encoder amount to no more than 6% of the original MLLM scale, which is negligible. More importantly, these parameters are pretrained in VLMs and do not require tuning with the specific MLLMs.

B.6. Latency Comparison

Our method, MDP^3, is training-free, which eliminates any additional latency during the training phase. Accordingly, we report the average latency of various processes during inference with MiniCPM-V2.6 on Video-MME (with-

MLLM	Used LLM	Used VLM	MLLM Params	Additional Params	Increase
VILA-V1.5-8B	LLama3-8B	SigLIP-400M	8.494B	0.450B	5.298%
MiniCPM-V2.6-7B	Qwen2-7B	SigLIP-400M	8.099B	0.450B	5.556%
LLaVA-OneVision-7B	Qwen2-7B	SigLIP-400M	8.027B	0.450B	5.606%

Table 4. Parameter scales for VILA-V1.5-8B, MiniCPM-V2.6-7B, and LLaVA-OneVision-7B, along with the increase due to the additional parameters introduced by MDP^3. Here, "MLLM Params" refers to the parameter scale of the baseline, including the LLM, the visual encoder in the VLMs, and the projector between them. "Additional Params" comes from the text encoder of the pretrained SigLIP, introduced by MDP^3.

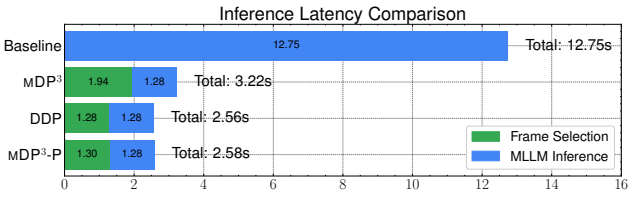

Figure 6. Latency during inference with MiniCPM-V2.6 on Video-MME (without subtitles). Where MDP^3-P refers o the MDP^3 with parallel computation in dynamic programming (*i.e.* line 9 in Algorithm 1). For direct comparison and better visualiztion, we omit the latency of identical processes across all compared models, including image loading and processing in the visual processor and encoder.

out subtitles), as illustrated in Fig. 6. The baseline refers to MiniCPM-V2.6 processing all 128 candidate frames without applying frame selection. Compared to the baseline, MDP^3 selects only 8 essential frames as input to the MLLMs, achieving a significant speedup during inference. It reduces MLLM inference time by 11.47s (90% of the baseline) while requiring only an additional 1.94s (15% of the baseline) for frame selection. MDP^3-P accelerates frame selection using parallel computation in dynamic programming (*i.e.* line 9 in Algorithm 1), requiring only an additional 1.30s for frame selection while reducing MLLM inference time by 11.47s (90% of the baseline). Furthermore, MDP^3-P has a latency comparable to the basic DPP method, which accounts only for list-wise diversity. This accords to our theoretical analysis, as both MDP^3-P and DPP share the same time complexity of $\mathcal{O}(nk^2)$.

B.7. Additional Experiments about Ablation Study

We conduct detailed ablation experiments on three Video-LLMs (VILA-V1.5, MiniCPM-V2.6, LLaVA-OneVision) and three benchmarks (Video-MME, MLVU, LVB$_{val}$), demonstrating the consistent superiority of MDP^3 over alternative strategies.

MDP^3 attains peak scores across all configurations: 53.3/58.6/50.8 for VILA-V1.5, 58.0/66.6/57.1 for MiniCPM-V2.6, and 59.6/69.8/59.0 for LLaVA-OneVision. The most significant improvements are observed on MLVU,

		Video-MME	MLVU	LVB$_{val}$
VILA-V1.5	+ uniform	47.5	46.3	47.1
	+ SigLIP	50.6	53.9	46.4
	+ MDP^3 w. MGK	48.9	48.6	53.5
	+ DPP w. CMGK	51.8	56.8	47.1
	+ MDP^3 w. cos. sim.	50.2	54.4	48.8
	MDP^3	**53.3**	**58.6**	**50.8**
MiniCPM-V2.6	+ uniform	52.6	55.4	51.2
	+ SigLIP	56.3	60.3	51.4
	+ MDP^3 w. MGK	52.6	62.0	52.6
	+ MDP^3 w. cos. sim.	53.1	63.0	54.1
	+ DPP w. CMGK	55.2	64.7	52.1
	MDP^3	**58.0**	**66.6**	**57.1**
LLaVA-OV	+ uniform	53.6	59.3	54.2
	+ SigLIP	57.0	62.7	51.6
	+ MDP^3 w. MGK	54.9	63.3	52.7
	+ DPP w. CMGK	56.4	68.6	52.8
	+ MDP^3 w. cos. sim.	55.9	65.2	54.6
	MDP^3	**59.6**	**69.8**	**59.0**

Table 5. Ablation study of MDP^3 across three Video-LLMs (VILA-8B, MiniCPM-V2.6, LLaVA-OneVision) and benchmarks (Video-MME, MLVU, LVB$_{val}$).

with gains of +4.8 to +6.2 over the baselines, underscoring its effectiveness in complex video understanding tasks.

Baseline uniform sampling underperforms by 3.0 to 9.5 points across all models, highlighting the importance of frame selection. Furthermore, intermediate variants exhibit limited gains due to partial adherence to our proposed principles; neglecting query relevance, list-wise diversity, or sequentiality fundamentally restricts selection quality. Moreover, the ablation results further validate the design choices of MDP^3. For example, although the cosine similarity variant outperforms uniform sampling by 2.1 to 4.8 points, it still lags behind MDP^3 by the same margin. This difference underscores the advantage of our multi-kernel similarity in the reproducing kernel Hilbert space (RKHS), which better captures high-dimensional feature relationships.

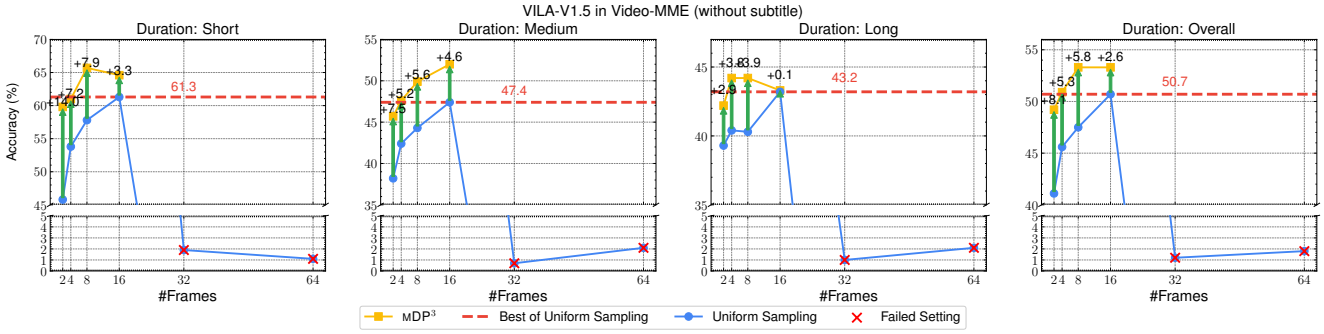

Figure 7. Detailed accuracy (%) on Video-MME (without subtitles) for VILA-V1.5 grouped by various durations, comparing uniform sampling with MDP^3 selection from 128 candidate frames.

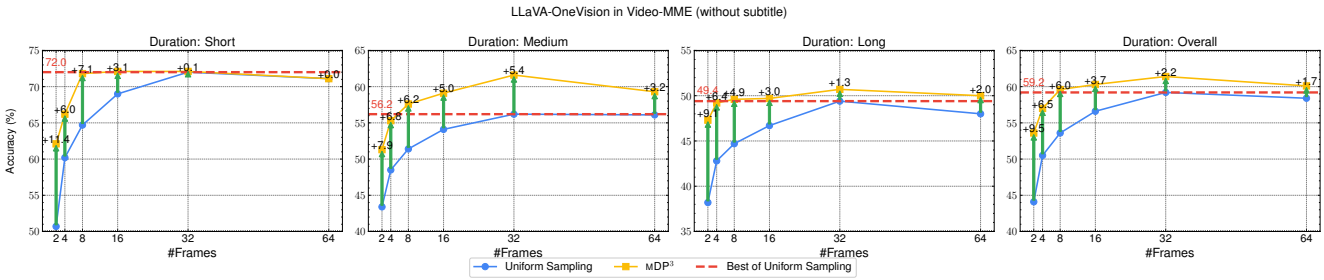

Figure 8. Detailed accuracy (%) on Video-MME (without subtitles) for LLaVA-OneVision grouped by various durations, comparing uniform sampling with MDP^3 selection from 128 candidate frames.

B.8. More Experiments with Various Selection Size

We have reported the results for various selection sizes k in Sec. 4.2.2 and Fig. 1. In this section, we provide additional results and more detailed discussions. In Figs. 7 and 8, we present the performance of VILA-V1.5 and LLaVA-OneVision on Video-MME (without subtitles) across various durations by varying the selection size k. The results show that, regardless of duration, VILA-V1.5's performance improves with more input frames when $k \leq 16$, while LLaVA-OneVision's performance improves with additional frames when $k \leq 32$. They employ the distinct context lengths and numbers of frames used during training. When the number of input frames exceeds the training stage, the input becomes out-of-distribution (OOD), leading to a performance drop. Specifically, when VILA-V1.5 receives more than 16 input frames, it experiences a catastrophic decline, whereas LLaVA-OneVision maintains some generalization ability when presented with more than 32 frames. Although some Video-LLMs are trained with more frames to enhance long-video understanding, this approach is costly and the number of input frames cannot be increased indefinitely. Therefore, designing an effective frame selection algorithm is crucial.

There should be a meaningful discussion about the question: **"Are more input frames better for video understanding?"** In Tab. 1, we select 8 frames following the state-of-the-art frame selection approach, Frame-Voyage, settings to ensure a fair comparison. However, 8 frames may not be optimal for all Video-LLMs, benchmarks, and durations. Our answer to the question **"Are more input frames better for video understanding?"** is **No!** We provide the following reasons:

1. **Training Constraints:** Video-LLMs are typically trained on a limited number of frames. When the number of input frames exceeds the training configuration, the model encounters OOD data, leading to degraded performance. Although training with more frames can mitigate this issue, it is computationally expensive and unsustainable in the long run (*i.e.*, the number of input frames cannot be increased indefinitely.).
2. **Inference Constraints:** Edge-deployed LLMs have limited resources, and proprietary models charge based on token usage. Consequently, processing excessive frames during inference stage is both resource-intensive and expensive.
3. **Diminishing Returns:** The marginal gain from adding frames diminishes exponentially (as indicated by the theoretical bound $\mathcal{O}((1/k)^k)$ in Eq. (35) and supported by experiments in Figs. 1, 7 and 8), while the Transformer's computational cost increases quadratically ($\mathcal{O}(n^2)$). Consequently, adding more frames is not

cost-efficient.

4. **Redundancy and Noise:** Incorporating additional frames may introduce redundant or irrelevant information, which can dilute salient features and add noise.
5. **Latency and Real-Time Constraints:** In applications requiring real-time processing, increasing the number of frames can lead to higher latency, which may be unacceptable in time-sensitive scenarios (as shown in Fig. 6).

These considerations underscore the importance of developing an effective frame selection algorithm that balances performance improvements with computational efficiency and resource limitations. Instead of merely increasing the number of input frames, a well-designed selection strategy can extract the most informative frames, ensuring that the model focuses on quality over quantity and optimizes both accuracy and cost.

Additionally, the above discussion is supported across various durations, encompassing short, medium, and long video understanding scenarios.

B.9. Qualitative Analysis

We sample six representative cases from Video-MME [13], illustrated in Figs. 9 to 14, to compare different frame selection methods, including uniform sampling (marked by ▲ in the top right corner of frame) and top-k query-frame matching with SigLIP [71] (marked by ◆). The selection by MDP^3 is marked by ●. Besides, the ground-truth of option is colored by green.

We categorize the issues observed in the baseline frame selection process. Especially, the point-wise top-k query-frame matching with SigLIP, which is the best-performing baseline frame selection method in Tab. 3, but still falls significantly short compared to MDP^3. The observed issues are as follows:

1. **Over-matching Specific Text** (Fig. 9): As shown in Fig. 9, when asked, "How many people are wearing ties in the video?", the query-frame matching over-focuses on the keyword "tie", resulting in the selection of numerous duplicate frames featuring an individual with a prominent, visible tie. This leads to the omission of frames where multiple people are wearing ties that are smaller or less noticeable. In contrast, the selection by MDP^3 demonstrates better balance between query relevance and frames diversity, effectively addressing this over-matching issue.
2. **Failure of Counting Across Frames** (Figs. 10 and 11): As shown in Figs. 10 and 11, when posed with questions such as, "What is the total number of bird species visible in the video?" or "How many different kinds of animal faces appear in this video?", these counting questions differ from the example in Fig. 9. Unlike the case in Fig. 9, the counted items are distributed across different frames and do not appear in a single frame. To answer accurately, it is necessary to select frames that include various relevant counted items. However, the baseline method of uniform sampling fails to identify specific counted items, while the query-frame matching approach struggles to avoid duplication, often focusing repeatedly on frames with the same item. In contrast, the frame selection by MDP^3 demonstrates greater diversity, allowing for the inclusion of different frames with various items, thereby aiding MLLMs in accurately counting across frames.
3. **Failure of Summarization** (Fig. 12): In this case, when the question is a summarization-type query such as "What is the genre of this video," the frame selection requires a comprehensive representation of the entire video rather than focusing on specific event clips. Since there is no specific key text to match frames, the query-matching fails, performing even worse than uniform sampling. In contrast, the frame selection by MDP^3 demonstrates a global understanding of the entire video, exhibiting good diversity and assisting MLLMs in summarizing the content effectively.
4. **Failure of Reverse Question Answering** (Fig. 13): This case is particularly interesting as it focuses on identifying events or items that do *NOT* appear in the video, such as "Which of the following elements does not appear in the video?" This type of reverse QA poses a significant challenge for query-frame matching since there is no key text for matching. However, MDP^3 demonstrates strong performance, ensuring diversity in the selected frames and providing a more comprehensive representation of the video.
5. **Failure of Transition Awareness** (Fig. 14): As shown in Fig. 14, when asked, "How many times does the interviewed girl appear in the video?", this question represents a special type of counting task: it not only requires counting across frames but also identifying the distinct *number of times* the interviewed girl appears. While the concept of the "interviewed girl" is singular, the actual item to be counted is the "number of appearances", making temporality and sequentiality crucial. As shown in Fig. 14, the query-frame matching fails to recognize the transitions between appearances of the interviewed girl. This oversight leads to missing frames between appearances, resulting in multiple occurrences being merged into a single one. Consequently, MLLMs cannot accurately count the number of appearances with such a selection. In contrast, MDP^3 effectively captures the sequential nature of the video and recognizes transitions between appearances, enabling accurate counting.

Additionally, this case study not only provides a qualitative analysis of various frame selection methods, but also reveals the limitations of baseline frame selection and highlights the strengths of MDP^3. Besides, it raises several chal-

lenges in VidQA, and serves as a guide for constructing a more comprehensive benchmark to evaluate the video understanding capabilities of MLLMs. This study is highly valuable to the technology community focused on MLLMs.

C. Limitations of MDP^3 and Future Directions

MDP^3 is a training-free, model-agnostic method that, for the first time, fully addresses query relevance, list-wise diversity, and sequentiality. Theoretically, MDP^3 offers a $(1-1/e)$-approximate solution to the NP-hard list-wise frame selection problem, achieving pseudo-polynomial time complexity and demonstrating its efficiency. Empirically, MDP^3 outperforms existing methods significantly, confirming its effectiveness and robustness.

However, MDP^3 still has certain limitations that merit further exploration in future research.

1. **Limitation:** The use of pretrained VLMs to develop a training-free, model-agnostic method is a double-edged sword. While MDP^3 can be seamlessly integrated into existing Video-LLMs, pretrained VLMs often have limitations in understanding complex instructions. **Future Directions:** Fortunately, MDP^3 is highly adaptable for future extensions. Fine-tuning the VLMs within MDP^3 with more complex instructions could significantly improve frame selection. Specifically, although the selection process is discrete and not directly optimizable, paired selection data can be gathered, and contrastive learning methods (such as DPO) can be applied for fine-tuning. The selection order could be supervised using existing LLMs, with the list-wise score finetuned to align with this supervision.
2. **Limitation:** The selection size k is fixed in MDP^3, and as shown in the case study, MDP^3 may occasionally select some useless frames. This issue can be mitigated by adjusting the trade-off between relevance and diversity, but such strategies are not feasible to apply on each sample. **Future Directions:** Therefore, exploring how to set an adaptive selection size k is a promising area for future research. In MDP^3, during dynamic programming, the optimal selection for any size $i < k$ has been captured in the trace matrix $\mathcal{T}_{T,i}$. This provides a convenient framework for determining the optimal k, but the challenge of identifying the best $i < k$ still remains and warrants further investigation.

Question: How many people are wearing ties in the video?

A. 4.

B. 5.

C. 3.

D. 2.

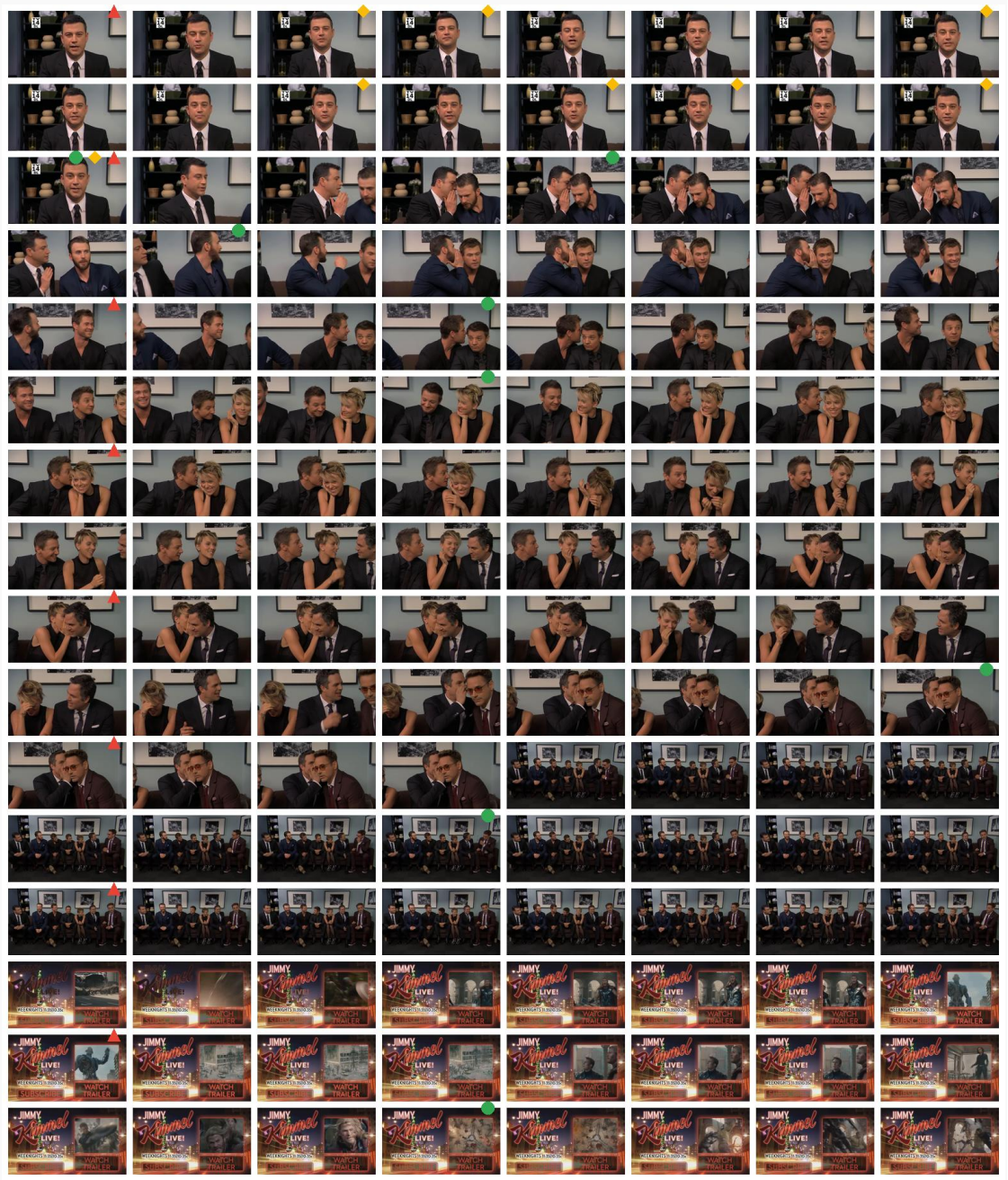

Figure 9. ▲: uniform sampling; ◆: top-k query-frame matching with SigLIP; ●: MDP^3. Over-matching of the keyword "tie" leads to duplicate frames being selected, omitting frames where multiple people wear ties. MDP^3 addresses this issue by balancing query relevance with frame diversity.

Question: What is the total number of bird species that are visible in the video?

A. 2.

B. 3.

C. 1.

D. 0.

Figure 10. ▲: uniform sampling; ◆: top-k query-frame matching with SigLIP; ●: MDP^3. In counting tasks across frames, such as "How many bird species or animal faces are in the video?", uniform sampling and query-frame matching struggle with item duplication. MDP^3 improves diversity, aiding in accurate counting across frames.

Question: How many different kinds of animal faces are made in this video?

A. 4.

B. 3.

C. 5.

D. 2.

Figure 11. ▲: uniform sampling; ◆: top-k query-frame matching with SigLIP; ●: MDP^3. In counting tasks across frames, such as "How many different kinds of animal faces are made in this video?", uniform sampling and query-frame matching struggle with item duplication. MDP^3 improves diversity, aiding in accurate counting across frames.

Question: What is the genre of this video?

A. It is a news report that introduces the history behind Christmas decorations.

B. It is a documentary on the evolution of Christmas holiday recipes.

C. It is a travel vlog exploring Christmas markets around the world.

D. It is a tutorial on DIY Christmas ornament crafting.

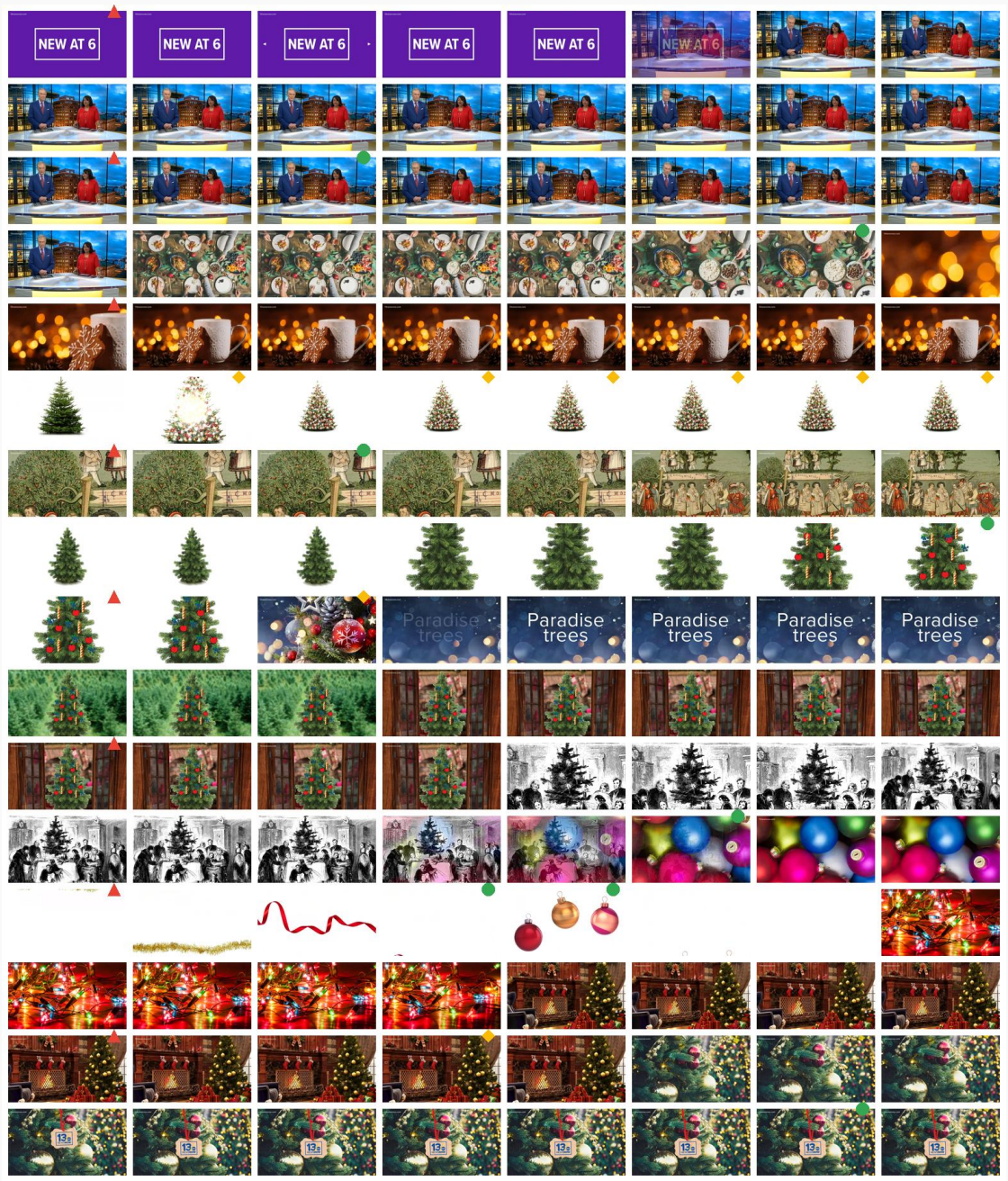

Figure 12. ▲: uniform sampling; ◆: top-k query-frame matching with SigLIP; ●: MDP^3. For summarization queries like "What is the genre of this video?", query-frame matching fails to represent the entire video. MDP^3 shows a global understanding of the video, enhancing diversity and assisting in summarization.

Question: Which of the following elements does not appear in the video?

A. Iceberg.

B. Moon.

C. Earth.

D. River.

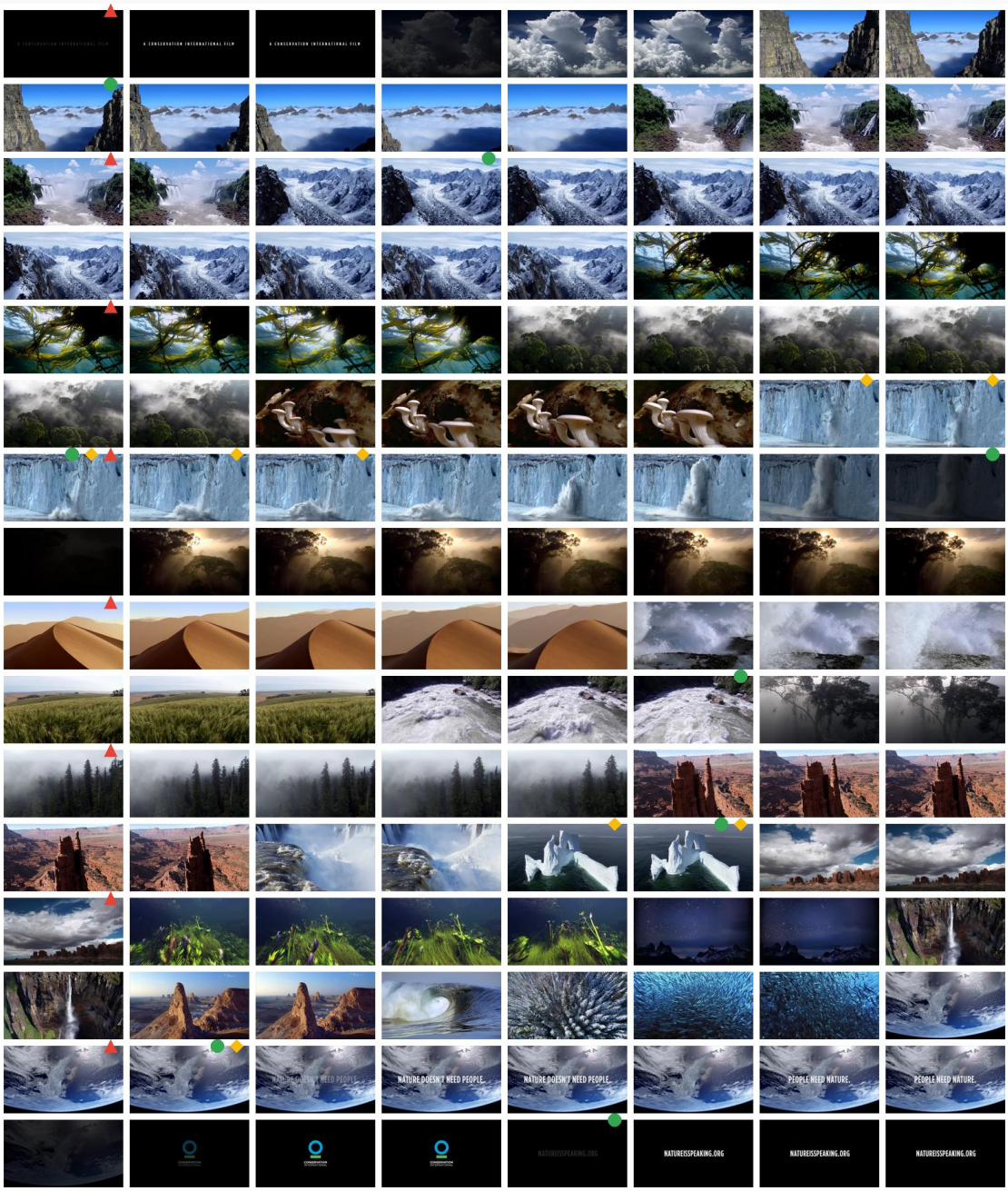

Figure 13. ▲: uniform sampling; ◆: top-k query-frame matching with SigLIP; ●: MDP^3. Reverse QA, such as "Which elements *DO NOT* appear in the video?", presents a challenge due to the lack of a specific key text for matching. MDP^3 excels by ensuring diversity in selected frames and providing a comprehensive video representation.

Question: How many times does the interviewed girl appear in the video?

A. 4.

B. 1.

C. 2.

D. 3.

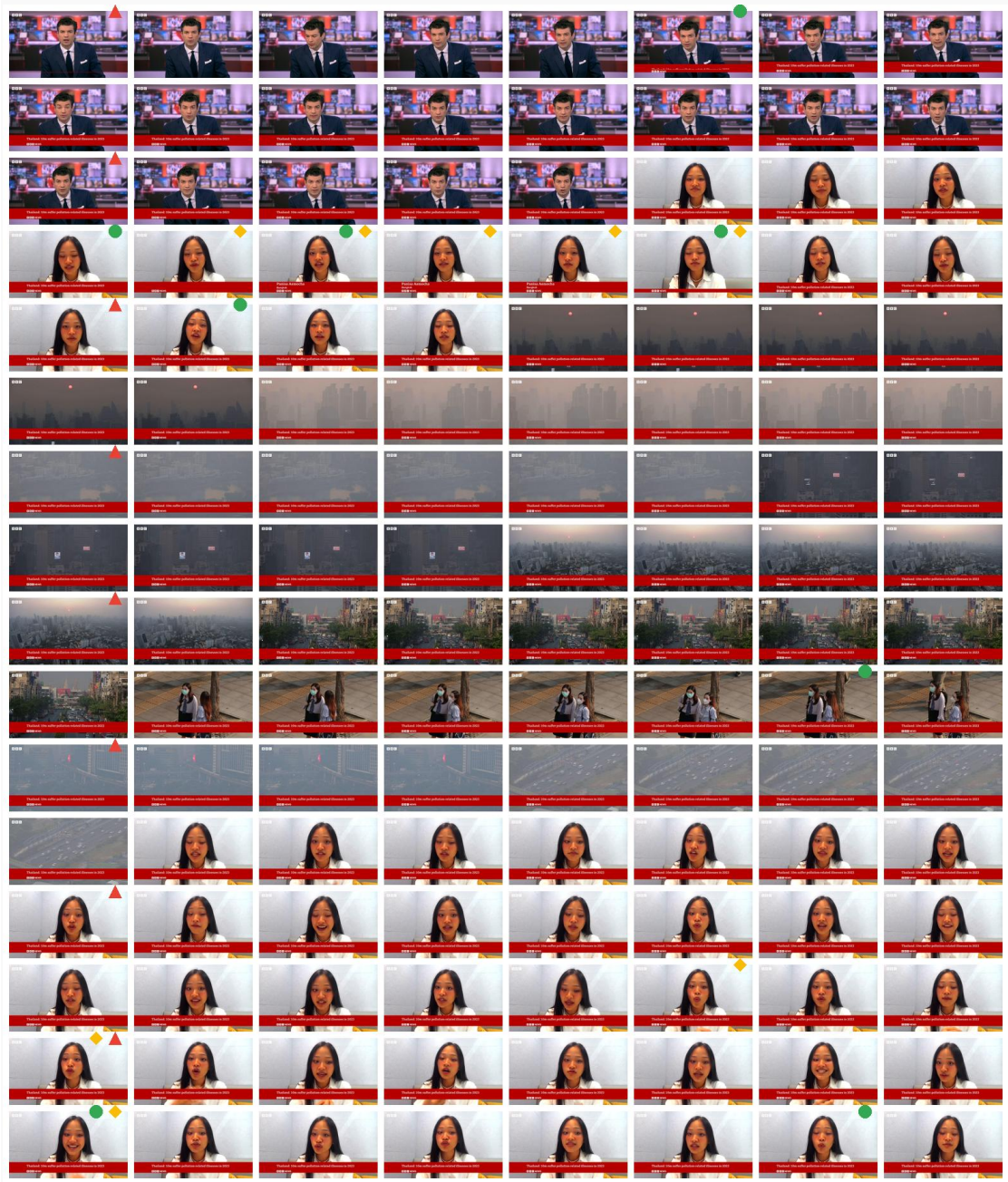

Figure 14. ▲: uniform sampling; ◆: top-k query-frame matching with SigLIP; ●: MDP^3. For questions like "How many times does the interviewed girl appear?", query-frame matching fails to capture the transitions between appearances. MDP^3 accurately counts the number of appearances by considering sequentiality.