# SEGMENTS DIVIDE, FRAMES DECIDE: COMPOSABLE RELEVANCE FOR LONG VIDEOS FROM QA ALONE

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

A vision-language model (VLM) sees a long video only through a handful of frames, so the choice of frames determines what it can answer. Queries differ in the temporal scale of the evidence they need, yet existing relevance signals operate at a single, fixed scale. Selectors that require no temporal grounding annotations score each frame in isolation, missing motion and event structure. Supervised grounding methods capture this structure but require such annotations and expensive model passes. This paper introduces SPAN: *Segment relevance via Propagated, Annotation-free, Nested composition*, which scores relevance at multiple temporal scales without grounding labels, using a query-conditioned composition operator that propagates relevance from base clips to arbitrarily coarse segments. Its weighting and composition functions read only the defined base-clip representations and the query to construct segments and obtain their relevance, so the same operator applies unchanged across scales. Self-distillation learns this operator from video-question-answer tuples alone: composed clips must reconstruct the video's own representation. Finally, a nested, tree-structured knapsack allocates SPAN's frame budget across the segment hierarchy. Across four long-video benchmarks and two VLM families, SPAN improves on uniform sampling by an average of $5.47$ points and beats the strongest annotation-free baseline in 18/20 settings. It recovers over $95.0\%$ of the performance achieved by a selector that runs a VLM during selection, without running one itself. For long-video question answering, this makes query-aware frame selection a lightweight plug-in step, thereby letting any frozen VLM answer from relevant evidence without temporal grounding annotations.

## 1 INTRODUCTION

Long-video question answering (Video QA) imposes a strict budget: a vision-language model (VLM) can be shown only a small number of frames, so the choice of frames determines what it can answer (Li et al., 2024; Maaz et al., 2024; Cheng et al., 2024; Zhang et al., 2025a; Wang et al., 2025a; Bai et al., 2025). Their selection is not simply a frame-ranking problem because different queries are answered by evidence at different temporal scales Zhou et al. (2025). One query (*"What was the person trying to accomplish today?"*) may be answered by evidence spread thinly throughout the video, while another (*"What did the person do right after picking up the cup?"*) is answered by a short, contiguous segment. Even within a single scale, a frame scored in isolation misses the motion and event structure that a clip carries. A frame selector operating under a strict budget must therefore be able to operate at multiple temporal scales.

We introduce *Segment relevance via Propagated, Annotation-free, Nested composition (SPAN)* which learns (a) how to represent short clips from the video in a latent space, and (b) a query-conditioned operator that composes their representations into longer segments, weighting each clip by its relevance to the query. This operator's weights depend only on a clip's representation and the query, never on a segment's length or position, so the same operator applies unchanged across scales and can score the relevance of any segment at inference time, even when no answer is available for comparison. This lets a search procedure cheaply explore the video during inference, composing segments and checking their relevance on demand rather than embedding and scoring every candidate from scratch.

A crucial design requirement is that SPAN learn these representations and the composition operator *without temporal grounding annotations* indicating which frames are necessary to answer a query.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

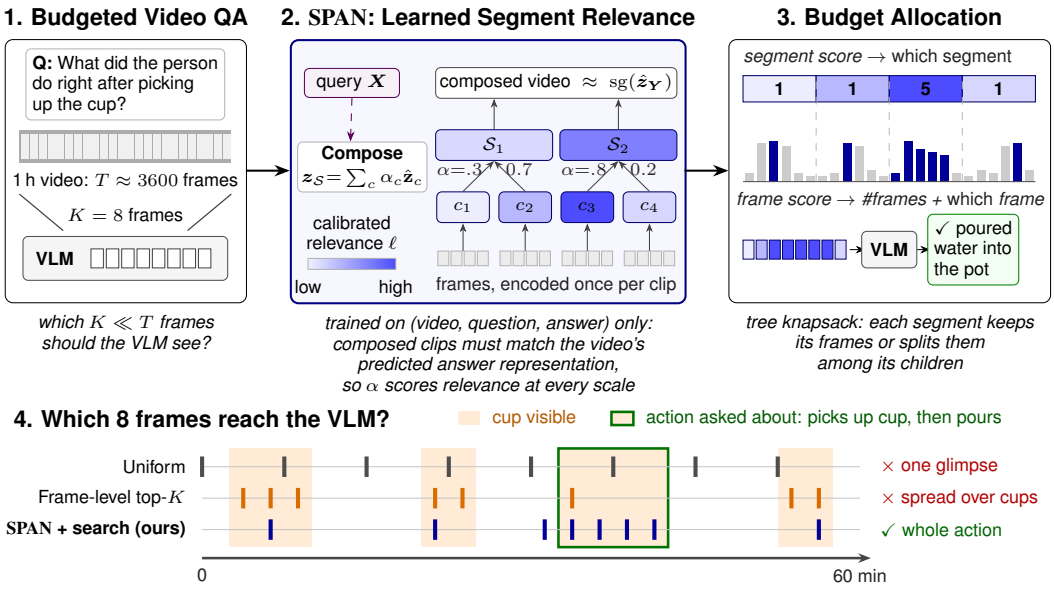



Figure 1: Segments divide, frames decide. (1) A VLM sees only $K \ll T$ frames of a long video. (2) SPAN learns query-conditioned segment relevance without temporal annotations. (3) Segment value decides which segment; an off-the-shelf frame scorer decides how many frames each region receives and which. (4) Illustrative: frame scores alone spread the budget over every appearance of the cup; segment relevance concentrates it on the action.

Such annotations are rarely available, expensive to create, and often ambiguous (e.g., how does one pick a specific set of clips to answer the first query above?). Instead, SPAN learns from video question-answer pairs alone, which are far more readily available (Zhang et al., 2024). With self-distillation, SPAN learns to match a video's representation to its answer; it simultaneously learns clip-level representations and the composition operator by requiring that composed clips reconstruct the video's own predicted answer representation. Temporal grounding for frame selection is thus learned without direct supervision via timestamps, and without a VLM anywhere in the loop.

These properties position SPAN against three lines of prior work. Training-free selectors such as BOLT (Liu et al., 2025) and AKS (Tang et al., 2025) score frames in isolation, so they are blind to motion and event structure. Supervised selectors such as VideoITG (Wang et al., 2026) model temporal structure, but require a temporally annotated training corpus and a full VLM forward pass over candidate frames at selection time. Selectors can also be learned without temporal labels: reinforcement-learning approaches optimise a sampling policy against a downstream VLM's answers, requiring a VLM during training even when not at inference (Yu et al., 2025; Tang et al., 2026). SPAN differs from all three: it learns relevance at multiple temporal scales via composition, from QA pairs alone, with no VLM anywhere in training or selection, and scales to hours-long videos.

We evaluate on MLVU (Zhou et al., 2025), EgoSchema (Mangalam et al., 2023), LongVideoBench (Wu et al., 2024), and CG-Bench (Chen et al., 2025) that include videos from three minutes to two hours using Qwen3-VL (Bai et al., 2025) and VideoLLaMA3 (Zhang et al., 2025a) at 2B–8B parameters, under budgets of 8 to 32 frames. SPAN improves on uniform sampling by 5.47 points on average and beats the strongest annotation-free baseline in 18/20 settings. Against VideoITG, which scores candidate frames with a full VLM, SPAN recovers over 95.0% of its performance with neither temporal labels nor a VLM in the selection loop. Further, unlike VideoITG, SPAN can index a 3,600-frame candidate pool in 11.7 seconds/query (Table 3).

## 2 Related Work

**Query-aware frame selection for Video QA under a budget.** Most budgeted selectors score frames with an off-the-shelf image-text model such as CLIP (Radford et al., 2021). BOLT (Liu et al., 2025)

samples frames based on these scores; AKS (Tang et al., 2025) recursively splits the budget across the video; FOCUS (Zhu et al., 2026) allocates it coarse-to-fine over segments. A concurrent line instead classifies each query as global or localised and switches between uniform and query-aware sampling accordingly (Li et al., 2026), offering a discrete, two-mode alternative to continuously scoring relevance across scale. In all of these, a segment's score is a statistic of its frames' scores (or a binary query-type decision), insufficient to capture the motion or event structure that no single frame carries. Supervised selectors such as VideoITG (Wang et al., 2026) learn temporal structure instead, but run a VLM forward pass over candidate frames. SPAN differs from both families: it learns segment scores across scales without temporal labels, and its selection runs no VLM.

**From relevance scores to a frame budget.** Many selectors retain the top-$k$ frames by score (Yu et al., 2025; Buch et al., 2025). MDP$^3$ (Sun et al., 2025) instead selects a diverse, non-redundant set via a determinantal point process, but still over a flat frame pool. Both rules ignore two properties of long videos. Within a chosen region, a budget can buy either breadth (one frame from each of many clips) or local density (every frame of a few clips); the better trade depends on the query, which this paper evaluates in Section 4.2. Across regions, a coarse segment and the finer segments inside it compete for the same frames (a nested structure that a flat pool, irrespective of whether it is diversified or not, does not represent), so a segment's budget must go either to the segment itself or to the segments nested within it, never both. SPAN casts selection as a tree-structured, nested multiple-choice knapsack (Sinha & Zoltners, 1979; Armstrong et al., 1982; Johnson & Niemi, 1983), solved by dynamic programming over the segment hierarchy. Each node's value combines calibrated segment relevance with a content-aware curve built from an off-the-shelf frame scorer.

**Weakly supervised temporal grounding.** Reading relevance, based on query-conditioned attention trained with only a video-level objective, dates back to weakly supervised moment retrieval (Mithun et al., 2019). The same recipe serves as the post hoc baseline for grounded Video QA (Xiao et al., 2024). Such attention tends to collapse onto a single spuriously correlated clip, the standard failure mode of weakly supervised localisation. SPAN constrains its weights with a harder target to avoid collapse: composed clips must reconstruct the video's own context-rich predicted representation rather than the ground-truth answer embedding. The predictive losses make the video representation answer-bearing, while self-distillation ensures that its decomposition is faithful.

**Attention-weighted and compositional video representations.** Attention-weighted pooling of frame embeddings is routine in text-video retrieval (Gorti et al., 2022; Bain et al., 2022). There, the pooled embedding serves retrieval, and its weights are neither read out as relevance nor reused across temporal scales. SPAN's operator reads only the constituent base-clip representations and the query, never the length, position, or identity of the segment being formed. The same trained operator, therefore, scores segments at multiple scales.

# 3 SPAN: Segment Relevance via Propagated, Annotation-free, Nested composition

This section describes SPAN. We first formalise the task of video question-answering with constrained temporal grounding in Section 3.1. Section 3.2 describes how segment-level temporal grounding is learned with SPAN in the absence of temporal labels. Finally, Section 3.3 presents the knapsack-based algorithm for allocating the frame budget to segments across temporal scales.

## 3.1 Formulation

**Task Formulation.** Formally, Video QA can be represented with the tuple $(\mathbb{V}, \boldsymbol{X}, \boldsymbol{Y})$, where the video $\mathbb{V}$ can be further represented as a discrete sequence of $T$ frames: $\mathbb{V} = \{v_1, \ldots, v_T\}$. $v_t$ represents the visual frame extracted at the $t$-th timestamp, $\boldsymbol{X}$ is the textual query and $\boldsymbol{Y}$ is the expected semantic response grounded in the video.

Let the embedding for a given single frame $v_t$ be $\boldsymbol{Z}_t = (\boldsymbol{z}_{t,1}, \ldots, \boldsymbol{z}_{t,P})$, where $\boldsymbol{Z}_t = f_{\theta_{ViT}}(v_t)$ with $\theta_{ViT}$ parameterising the ViT-based image-encoding function of a VLM, $\theta_{VLM}$ and $P$ representing the number of visual patch tokens generated per frame. The resulting visual context provided to the VLM is the concatenated sequence of all frame embeddings: $\boldsymbol{Z}_{\mathbb{V}} = \text{concat}(\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_T)$. Encoding $T$ frames translates to a visual prompt of length $T \times P$. Because the self-attention mechanism in foundation models scales quadratically with sequence length (Vaswani et al., 2017), processing visual

context becomes increasingly expensive as the video gets longer. Therefore, identifying relevant frames under a strict computational budget is required.

**Constrained Temporal Grounding.** Let the downstream VLM $\theta_{VLM}$ be frozen and constrained to process at most $K \ll T$ frames. Correctly answering a query $\boldsymbol{X}$ under this budget requires the query-dependent subset

$$\mathbb{V}_{\boldsymbol{X}} = \arg \max_{\mathbb{V}' \subseteq \mathbb{V},\ |\mathbb{V}'|=K} \log p_{\theta_{VLM}}(\boldsymbol{Y} \mid \boldsymbol{X}, \mathbb{V}'). \tag{1}$$

Exactly solving Eq. 1 requires evaluating $\theta_{VLM}$ on all $\binom{T}{K}$ subsets, which is combinatorially intractable. We therefore learn a frame-proposal function $\hat{\mathbb{V}}_{\boldsymbol{X}} = \mathcal{R}_{\theta_{prop}}(\mathbb{V}, \boldsymbol{X}, K)$, parameterised by $\theta_{prop}$, to approximate $\mathbb{V}_{\boldsymbol{X}}$.

Learning $\theta_{prop}$ is subject to two additional constraints: (1) no temporally annotated subset $\mathbb{V}^*_{\boldsymbol{X}}$ is available to supervise $\theta_{prop}$ directly, and (2) $\theta_{VLM}$ is a frozen black box, through which we cannot backpropagate to obtain a learning signal for $\theta_{prop}$. Eq. 1 is therefore inaccessible both as a target ($\mathbb{V}^*_{\boldsymbol{X}}$ unavailable) and as an objective (no gradient through $\theta_{VLM}$). We instead train $\theta_{prop}$ against a surrogate defined in a continuous latent space rather than $\theta_{VLM}$'s discrete output space using a Joint-Embedding Predictive Architecture (JEPA) (LeCun et al., 2022; Chen et al., 2026), which embeds $\mathbb{V}$ such that its query-conditioned representation predicts response, $\boldsymbol{Y}$ embedding directly, without ever querying $\theta_{VLM}$.

**Joint Embedding Predictive Architecture.** The JEPA architecture employs an input-image encoder $\theta_{vis}$ to map vision inputs $\mathbb{V}$ into the embedding $\boldsymbol{Z}_{\mathbb{V}}$, a target-text encoder $\theta_{txt}$ to map the target response $\boldsymbol{Y}$ into an embedding $\boldsymbol{z}_{\boldsymbol{Y}}$, and a predictor network $\theta_{pred}$ to learn the semantic mapping $f_{\theta_{pred}} : (\boldsymbol{Z}_{\mathbb{V}}, \boldsymbol{X}) \mapsto \boldsymbol{z}_{\boldsymbol{Y}}$. This work employs an instruction-guided Query-Transformer (Q-Former) (Li et al., 2023) to realise the predictor network. JEPA-based models typically optimise a joint objective that balances prediction error against representation-collapse regularisation. This is formally represented as:

$$\mathcal{L}_{JEPA} = \mathcal{L}_{L1} + \mathcal{L}_{InfoNCE} \tag{2}$$

where $\mathcal{L}_{InfoNCE}$ calculates the contrastive InfoNCE loss (Oord et al., 2018) between the encoded $\boldsymbol{z}_{\boldsymbol{Y}}$ and the predicted $\hat{\boldsymbol{z}}_{\boldsymbol{Y}}$ in an abstract concept space, after each is projected via separate MLP networks $\boldsymbol{W}^Y_{reg}$ and $\boldsymbol{W}^V_{reg}$, respectively. Concurrently, $\mathcal{L}_{L1}$ calculates the L1-norm in the text-encoder's embedding space to enforce strict semantic alignment. For this, $\hat{\boldsymbol{z}}_{\boldsymbol{Y}}$ is projected via an MLP network $\boldsymbol{W}^Y_{error}$ before calculating the absolute error against $\boldsymbol{z}_{\boldsymbol{Y}}$. Therefore,

$$\mathcal{L}_{L1} = \left\| \boldsymbol{W}^Y_{error} \hat{\boldsymbol{z}}_{\boldsymbol{Y}} - \boldsymbol{z}_{\boldsymbol{Y}} \right\|_1, \quad \mathcal{L}_{InfoNCE} = -\log \frac{\exp\left(\langle \boldsymbol{W}^Y_{reg} \boldsymbol{z}_{\boldsymbol{Y}},\ \boldsymbol{W}^V_{reg} \hat{\boldsymbol{z}}_{\boldsymbol{Y}} \rangle / \tau\right)}{\sum \exp\left(\langle \boldsymbol{W}^Y_{reg} \boldsymbol{z}_{\boldsymbol{Y}'},\ \boldsymbol{W}^V_{reg} \hat{\boldsymbol{z}}_{\boldsymbol{Y}'} \rangle / \tau\right)} \tag{3}$$

This JEPA instantiation gives us a frozen-VLM-free, temporal grounding annotation-free training signal, but a trained embedding space alone does not yet yield $\hat{\mathbb{V}}_{\boldsymbol{X}}$. We still require a mechanism to obtain a discrete, size-$K$ subset of frames, which we introduce next.

### 3.2 Composable Relevance Scores from Question–Answer Pairs

With SPAN, we realise $\mathcal{R}_{\theta_{prop}}$ in two stages - segment relevance scoring and a knapsack search that takes these scores and the budget $K$ as input and solves for the discrete subset $\hat{\mathbb{V}}_{\boldsymbol{X}}$, with no learned parameters of its own. $\mathcal{R}_{\theta_{prop}}(\mathbb{V}, \boldsymbol{X}, K)$ is thus the composition of a learned scorer, $\text{Score}_{\theta_{prop}}(\mathbb{V}, \boldsymbol{X})$, and a fixed search, $\text{Search}_K(\cdot)$, applied to its output.

First, to efficiently extract representations at varying temporal granularities (e.g., specific 3-second vs. 3-minute segments) without a separate forward pass through the Q-Former for every possible temporal window, and to quantify their relevance to the query, this work posits that *a long video's semantic representation should be derivable from its shorter constituent segments*. This is what SPAN does: it makes composition a learned operation.

Consider the video is partitioned into a set of $N$ base clips, i.e. $\mathbb{C} = \{c_1, \ldots, c_N\}$. Each clip is of length $M$ frames i.e. $c = \{v_i, \ldots, v_{i+M}\}$ encoded as - $\boldsymbol{Z}_c = \text{concat}(\boldsymbol{Z}_i, \ldots, \boldsymbol{Z}_{i+M})$, with corresponding predicted representations $\hat{\boldsymbol{z}}_c = f_{\theta_{pred}}(\boldsymbol{Z}_c, \boldsymbol{X})$. For a given query $\boldsymbol{X}$ and any coarser segment, the corresponding set of temporally contiguous base clips $\mathbb{S} = \{c_j, \ldots, c_k\}$ defines a shared

relevance scorer $\ell(\boldsymbol{z_X}, \hat{\boldsymbol{z}}_c)$ and a composition weight normalised over $\mathbb{S}$:

$$\alpha(\boldsymbol{z_X}, \hat{\boldsymbol{z}}_c; \mathbb{S}) = \frac{\exp\left(\ell(\boldsymbol{z_X}, \hat{\boldsymbol{z}}_c)\right)}{\sum_{c' \in \mathbb{S}} \exp\left(\ell(\boldsymbol{z_X}, \hat{\boldsymbol{z}}_{c'})\right)} \tag{4}$$

where $\boldsymbol{z_X}$ is the query embedding obtained using $\theta_{txt}$. The composition operator aggregates $\mathbb{S}$ into a single representation directly from its constituents' already-computed embeddings, without re-running $\theta_{vis} \circ \theta_{pred}$ over its constituent frames -

$$\boldsymbol{z}_{\mathbb{S}} = \text{Compose}(\mathbb{S}; \boldsymbol{X}) = \sum_{c \in \mathbb{S}} \alpha(\boldsymbol{z_X}, \hat{\boldsymbol{z}}_c; \mathbb{S})\, \hat{\boldsymbol{z}}_c \tag{5}$$

Crucially, $\ell$ and $\alpha$ depend only on the constituent base clips' representations and the query — never on the identity or size of $\mathbb{S}$ beyond the renormalisation above — so Compose is defined identically whether $\mathbb{S}$ is the full set of base clips or an arbitrary contiguous subset. During training, we instantiate Compose over $\mathbb{S} = \mathbb{C}$, projected into each of the two loss spaces of JEPA from Section 3.1:

$$\boldsymbol{z}^{L1}_{composed} = \sum_{i=1}^{N} \alpha^{L1}\big(\boldsymbol{z_X}, \boldsymbol{W}^{Y}_{error}\hat{\boldsymbol{z}}_{c_i}; \mathbb{C}\big)\, \boldsymbol{W}^{Y}_{error}\hat{\boldsymbol{z}}_{c_i} \tag{6}$$

$$\boldsymbol{z}^{InfoNCE}_{composed} = \sum_{i=1}^{N} \alpha^{InfoNCE}\big(\boldsymbol{W}^{Y}_{reg}\boldsymbol{z_X}, \boldsymbol{W}^{V}_{reg}\hat{\boldsymbol{z}}_{c_i}; \mathbb{C}\big)\, \boldsymbol{W}^{V}_{reg}\hat{\boldsymbol{z}}_{c_i} \tag{7}$$

and enforce that this composition reconstructs the full video's own representation via a self-distillation objective, with the target detached:

$$\mathcal{L}_{self\text{-}distill} = \mathcal{L}_{L2}\big(\boldsymbol{z}^{L1}_{composed},\ \text{sg}\big(\boldsymbol{W}^{Y}_{error}\hat{\boldsymbol{z}}_{\boldsymbol{Y}}\big)\big) + \mathcal{L}_{L2}\Big(\boldsymbol{z}^{InfoNCE}_{composed},\ \text{sg}\big(\boldsymbol{W}^{V}_{reg}\hat{\boldsymbol{z}}_{\boldsymbol{Y}}\big)\Big) \tag{8}$$

where sg denotes stop-gradient. The overall objective is

$$\mathcal{L}_{\text{SPAN}} = \mathcal{L}_{L1} + \mathcal{L}_{InfoNCE} + \lambda_{self\text{-}distill}\, \mathcal{L}_{self\text{-}distill} \tag{9}$$

A segment's own relevance score, used by the search during inference in Section 3.3, is the weight $\ell^{L1}(\boldsymbol{z_X}, \boldsymbol{z}_{\mathbb{S}})$ assigned during composition: a segment that must be upweighted to accurately reconstruct the full video's representation is, by construction, one that the model deems important for predicting the answer. For notational convenience, we use $\ell(\mathbb{S})$ to refer to the segment's relevance.

**Design choices.** (a) We route the self-distillation target through the video's own stop-gradient predicted representation, $\text{sg}(\hat{\boldsymbol{z}}_{\boldsymbol{Y}})$, rather than the ground-truth answer embedding $\boldsymbol{z_Y}$ directly: this decouples the two objectives — $\mathcal{L}_{L1}$ and $\mathcal{L}_{InfoNCE}$ alone make the full-video representation answer-predictive, while $\mathcal{L}_{self\text{-}distill}$ enforces that composition is faithful to it. The stop-gradient itself prevents the composed and target branches from co-adapting to trivially satisfy the matching loss, following standard self-distillation practice (Grill et al., 2020; Chen & He, 2021). (b) The relevance scorer is realised as $\ell(\boldsymbol{z_X}, \boldsymbol{z}_s) = (\boldsymbol{W}^{q}\boldsymbol{z_X})^{\top}(\boldsymbol{W}^{s}\boldsymbol{z}_s)$, where $\boldsymbol{z_X}$ is the query embedding, produced by $\theta_{txt}$, and $\boldsymbol{z}_s$ is a base clip or composed segment's embedding. Applying $\ell$ to segments from $\mathbb{V}$ is precisely $\text{Score}_{\theta_{prop}}(\mathbb{V}, \boldsymbol{X})$: the set of scores that the search stage consumes to generate $\hat{\mathbb{V}}_{\boldsymbol{X}}$.

### 3.3 Inference-time Search: From Segment Relevance to Frame Selection

Allocating a fixed budget of frames to segments with different relevance values can be posed as a knapsack problem, specifically within the tree-structured multiple-choice knapsack family Sinha & Zoltners (1979); Armstrong et al. (1982); Johnson & Niemi (1983) since segments are of varying granularity and they can contribute anywhere from zero frames up to their full length. Solving it lets us decide whether to allocate the budget directly to the coarser segment or split it across the finer segments it contains. Refer to Section E for further discussion.

**Step-1: Tree-Structured Relevance Index.** We instantiate the search space as a tree-structured relevance index, $\mathcal{T}$ built by recursively applying Compose from Section 3.2 to the base clips $\mathbb{C}$ and obtaining $\ell(\mathbb{S})$ for segments across temporal scales in the following order - (i) A base clip of length $M$ is defined as the shortest unit, and the resulting $N$ base clips representing the leaves are

encoded and scored to obtain their values $\ell(c), \forall c \in \mathbb{C}$. (ii) For a tree of depth $D$, each depth-$d(\geq 1)$ internal node represents a segment obtained by applying Compose to $2^d$ contiguous base clips and then scoring to obtain its value $\ell(\mathbb{S})$. (iii) With a stride, representing the offset in terms of base-clip units, consecutive segments at a given depth are built and scored. Note that we keep the stride equal to the depth's segment width to avoid overlapping. (iv) Add an edge between the coarser segment at depth $d$ and the finer segment at $d-1$ if there are overlapping base clips between them. Overall, this results in a Binary Tree as shown in the Figure 3.

**Step-2: Score Calibration.** SPAN's raw relevance scores are not directly comparable across depths: a score at depth $d$ is computed relative to a different set of sibling segments than a score at depth $d' \neq d$. For meaningful cross-depth comparison, scores within each depth are calibrated independently, for instance, by softmax or z-scoring. This results for every segment $\mathbb{S}$, Calibrate : $\ell(\mathbb{S}) \mapsto \ell_{cal}(\mathbb{S})$.

**Step-3: Node Value Assignment.** While each node or segment $\in \mathcal{T}$ spanning frames $\mathbb{V}_\mathbb{S}$ receives the relevance exacting to $\ell(\mathbb{S})$ from SPAN, we argue that the value of the node should be a configurable function of frames sampled, or else sampling anywhere from $1 \dots |\mathbb{V}_\mathbb{S}|$ frames from it would be identical once the node is selected. Further, which frames within it should be extracted? To break the tie and assist with frame extraction, we assume access to an off-the-shelf model to score a query against frames. Thus, for $k \in \{1, \dots, |\mathbb{V}_\mathbb{S}|\}$ we now have the following node value function -

$$\mathcal{V}(\mathbb{S}, k) = \Psi\big(C_\mathbb{S}(k),\ \ell_{cal}(\mathbb{S})\big) \tag{10}$$

where $C_\mathbb{S}(k)$ is a content-aware value curve over $\mathbb{V}_\mathbb{S}$, built using the query-frame scorer, quantifying how much of $\mathbb{S}$'s content is captured by its $k$ most relevant frames. We instantiate $\Psi$ as a weighted combination, $\Psi(a, b) = \lambda_{frame}\, a + \lambda_{segment}\, b$ and examine other realisations in Section D.

**Step-4: Knapsack Solver ($\mathbf{Search}_K(\cdot)$).** Given $\mathcal{V}$, budget $K$, and the tree $\mathcal{T}$, the search reduces to finding the value-maximising allocation of $K$ across $\mathcal{T}$'s nodes, subject to each internal node's chosen budget being met by some feasible split across its children — tractable via dynamic programming, rather than requiring a combinatorial search. Refer Algorithm 1 for the summary.

# 4 Experiments

## 4.1 Implementation Details

**Training Configuration.** SPAN is trained on open-ended video question-answering data extracted from the LLaVA-Video-178K dataset Zhang et al. (2024), filtered to videos between 1 and 3 minutes in length, resulting in 202576 samples. Visual context is extracted by uniformly sampling each video at 1 frame per second (FPS), capped at 128 frames per sequence. Architecturally, SPAN uses off-the-shelf, lightweight, frozen encoders: the visual encoder $\theta_{vis}$ is a 300M-parameter `dinov2-large` backbone Oquab et al. (2023), and the text encoder $\theta_{txt}$, used for the target answer, is `Qwen3-Embedding-0.6B` Zhang et al. (2025b). The predictor $\theta_{pred}$ (Section 3.1) is instantiated as a Q-Former Li et al. (2023) with 8 learnable, instruction-guided query tokens which cross-attend the frame embeddings of a clip and are pooled to obtain the base clip-embedding $\boldsymbol{z}_c$. $M$ is fixed to 8, the finest granularity seen during training. The model is trained for a single epoch with the AdamW optimiser and a linear warmup over the first 10% of steps to a constant learning rate of $1 \times 10^{-5}$. The self-distillation term is weighted by $\lambda_{self\text{-}distill} = 0.5$ in Eq. 9. Training is distributed across $8\times$ NVIDIA V100 32GB GPUs with a per-device batch size of 4.

**Evaluation.** Frame-selection methods are evaluated by the downstream VLM's Video QA performance using the frames they select. We evaluate on a wide range of benchmarks spanning three minutes to two hours in video length — MLVU (Zhou et al., 2025), EgoSchema (Mangalam et al., 2023), LongVideoBench (Wu et al., 2024), and CG-Bench (Chen et al., 2025). Videos are sampled at 1 FPS, with candidate frame pools capped at 3600 frames. We evaluate across two model families, Qwen3-VL (Bai et al., 2025) and VideoLLaMA3 (Zhang et al., 2025a), at parameter scales 2B-8B.

**SPAN-specific setup.** The relevance tree $\mathcal{T}$ is built with base clips of 8 frames (Section 3.2), depth $D = 4$. The frame-level relevance scorer is instantiated using `openai/clip-vit-large-patch14`. The node-value weights $(\lambda_{frame}, \lambda_{segment})$ are tuned per benchmark via grid search on $[0.0, 1.0]$ on each benchmark such that $\lambda_{frame} + \lambda_{segment} = 1.0$.

Table 1: **Comparison of annotation-free frame-sampling strategies** (Uniform Sampling, BOLT, AKS, SPAN) across foundation model backbones on VideoQA benchmarks, at a frame budget of $K = 8$. $\Delta_{Avg.}$ reports each method's average accuracy improvement (in points), across all benchmarks, over a *Language Bias* baseline. Best result per backbone in **bold**.

| Backbone | Method | MLVU (3m – 2h) | EgoSchema (3m) | LongVideoBench (8s – 1h) | CGBench (10m – 80m) | $\Delta_{Avg.}$ |
|---|---|---|---|---|---|---|
| Qwen3-VL-2B | Lang Bias | 35.97 | 26.2 | 34.51 | 23.07 | – |
| | Uniform | 49.45 | 53.0 | 44.53 | 27.77 | 13.75 |
| | Bolt | 57.64 | 50.4 | 45.71 | 28.37 | 15.59 |
| | AKS | 54.46 | 53.6 | 47.05 | 28.33 | 15.92 |
| | *SPAN(Ours)* | **60.12** | **57.2** | **49.66** | **32.03** | **19.82** |
| Qwen3-VL-4B | Lang Bias | 43.84 | 29.2 | 41.41 | 24.23 | |
| | Uniform | 54.28 | 57.8 | 49.92 | 28.27 | 12.90 |
| | Bolt | **63.89** | 58.4 | 52.86 | 29.67 | 16.54 |
| | AKS | 59.20 | 58.6 | 51.77 | 30.20 | 15.27 |
| | *SPAN(Ours)* | 63.80 | **62.0** | **54.46** | **33.70** | **18.82** |
| Qwen3-VL-8B | Lang Bias | 45.22 | 34.4 | 35.35 | 27.90 | – |
| | Uniform | 55.70 | 58.2 | 48.32 | 33.90 | 13.31 |
| | Bolt | 61.68 | 58.4 | 50.34 | 34.77 | 15.58 |
| | AKS | 60.17 | 59.2 | 52.36 | 34.23 | 15.77 |
| | *SPAN(Ours)* | **65.41** | **61.4** | **53.45** | **37.27** | **18.66** |
| VideoLLaMA3-2B | Lang Bias | 44.30 | 24.4 | 39.90 | 24.70 | – |
| | Uniform | 55.84 | 46.6 | 48.73 | 30.53 | 12.10 |
| | Bolt | 63.02 | 46.8 | 48.91 | 32.67 | 14.53 |
| | AKS | 61.04 | 48.6 | 52.19 | 34.40 | 15.73 |
| | *SPAN(Ours)* | **64.08** | **50.0** | **53.28** | **36.27** | **17.58** |
| VideoLLaMA3-7B | Lang Bias | 50.23 | 29.4 | 39.31 | 25.27 | – |
| | Uniform | 60.30 | 58.2 | 49.41 | 28.97 | 13.17 |
| | Bolt | 66.70 | 57.0 | 50.93 | 30.53 | 15.24 |
| | AKS | 65.04 | **58.2** | 52.78 | 31.57 | 15.85 |
| | *SPAN(Ours)* | **68.12** | 57.0 | **55.38** | **34.53** | **17.71** |

### 4.2 Results and Discussion

**Comparison with temporal grounding annotation-free baselines.** We compare SPAN against Uniform Sampling, BOLT Liu et al. (2025), and AKS Tang et al. (2025) in Table 1. Because benchmarks differ in how much of their accuracy can be achieved from language priors alone, we report the average improvement over a language-only (no visual input) baseline. SPAN is the best-performing method in 18 of 20 (VLM, benchmark) settings for $K = 8$ considered here, improving over the strongest baseline by 1.85–3.9 pts on average. See Table 6 for comparison when $K = 32$.

As Figure 2 shows, frame-selection methods converge as the frame budget grows; for Qwen3-VL-8B, the gap between Uniform and SPAN narrows on both MLVU and LongVideoBench. This is expected because benchmark videos have bounded length; any strategy approaches full-video coverage as $K$ nears the total number of available frames. Therefore, the value of informed frame selection is largest under the tight budgets this work targets. This can also be observed empirically: the improvement range over the strongest baseline shrinks from 1.85–3.9 pts at $K$=8 to 1.0–2.78 pts at $K$=32 (Table 6). SPAN still outperforms the strongest baseline at this higher budget, showing that its segment-relevance signal remains informative even as the selection problem itself grows easier.

Crucially, SPAN sampling only 8 frames matches, and even exceeds uniform sampling's performance at $4\times$–$8\times$ more frames: 65.41 (SPAN) vs. 65.92 (Uniform at 64 frames) on MLVU, and 53.45 (SPAN) vs. 53.20 (Uniform at 32 frames) on LongVideoBench (Figure 2). This shows that SPAN uses the fixed budget more efficiently. And, since the computational cost of a VLM's self-attention scales

Table 2: **Video QA performance of SPAN relative to the temporally-supervised VideoITG baseline**, at frame budgets $K \in \{8, 32\}$. $\mu_{rec}(\%)$ shows the share of VideoITG's accuracy that SPAN recovers averaged over benchmarks without any temporal annotation.

| Backbone | Budget | Method | MLVU (3m – 2h) | LongVideoBench (8s – 1h) | CGBench (10m – 80m) | $\mu_{rec}(\%)$ |
|---|---|---|---|---|---|---|
| Qwen3-VL-8B | 8 Frames | VideoITG | 66.42 | 55.56 | 39.40 | 96.43 |
| | | *SPAN* | 65.41 | 53.45 | 37.27 | |
| | 32 Frames | VideoITG | 74.66 | 59.93 | 42.53 | 96.80 |
| | | *SPAN* | 70.61 | 59.26 | 41.23 | |
| VideoLLaMA3-7B | 8 Frames | VideoITG | 70.10 | 58.25 | 36.77 | 95.39 |
| | | *SPAN* | 68.12 | 55.38 | 34.53 | |
| | 32 Frames | VideoITG | 73.51 | 56.73 | 34.83 | 95.34 |
| | | *SPAN* | 70.01 | 54.71 | 32.86 | |

Table 3: Relevance-indexing efficiency across scoring mechanisms, for two candidate frame-pool sizes $T \in \{512, 3600\}$. **Encoding** is the one-time cost of encoding a video's frames independent of query; **Scoring** is the per-query relevance-scoring cost. Peak memory reflects the full indexing pipeline. ↓: lower is better. *VideoITG uses KV-caching and FlashAttention for efficiency; **–** indicates the metric could not be computed at $T$=3600, since it exceeds VideoITG's 16K-token context limit.

| Scoring Mechanism | Parameters | Encoding (s)↓ | | Scoring (s)↓ | | Peak Mem. (GB)↓ | |
|---|---|---|---|---|---|---|---|
| | | $T$=512 | $T$=3600 | $T$=512 | $T$=3600 | $T$=512 | $T$=3600 |
| Per-Frame (CLIP-based) | 0.4B | 3.30 | 25.29 | 0.01 | 0.07 | 6.2 | 6.2 |
| SPAN (Ours) | 1.1B | 1.44 | 9.82 | 0.36 | 1.88 | 6.1 | 14.5 |
| VideoITG* | 8B | 2.92 | – | 0.61 | – | 22.0 | – |

quadratically with the number of visual tokens (Vaswani et al., 2017), matching Uniform's accuracy at a fraction of its frame count translates directly into a proportional reduction in the downstream inference cost — precisely the constraint that motivates frame selection in Section 3.1.

**Comparison with supervised selector.** We study how much of the performance of a temporally-supervised frame selector VideoITG (Wang et al., 2026) is recovered by SPAN. Table 2 shows that SPAN is able to recover over 95.0% overall. Refer to Table 7 for recovery on other VLMs.

We next examine the relevance-indexing efficiency in Table 3 for a scenario in which multiple queries are posed against a single video, as in agentic frameworks (Xiao et al., 2021; Zhong et al., 2022; Goletto et al., 2024). This setting naturally separates a *one-time* indexing cost, paid once per video, from a *per-query* cost, paid for every new question. VideoITG cannot score a pool of 3,600 frames at all: at 16 visual tokens per frame, this requires $3{,}600 \times 16 = 57{,}600$ tokens, which exceeds its 16K context limit. Quadratic constraint of VLMs that motivates the frame selection in Section 3.1 now acts on the selector itself: a VLM-based selector becomes intractable at candidate-pool scales where selection matters the most. SPAN's one-time encoding cost is also the lowest of all three mechanisms at both scales (1.44s and 9.82s, vs. 3.30s and 25.29s for per-frame scoring, and 2.92s at $T$=512 for VideoITG). Its per-query scoring cost is higher than flat per-frame scoring (0.36s vs. 0.01s at $T$=512), since SPAN's score is computed by composing a full multi-scale segment hierarchy (including encoding base clips per query) rather than a single per-frame dot product. Because encoding is amortised across all queries against a video, the total time for selecting frames with SPAN including per-frame scoring with CLIP is 0.37s, which is 1.7× faster than VideoITG's (0.61s) at $T$=512.

SPAN's peak memory also grows with pool size (6.1GB at $T$=512 to 14.5GB at $T$=3600), since scoring requires holding clip/segment representations at every level (up to $D = 4$). This growth is sublinear relative to the 7× increase in candidate frames, and remains well below VideoITG's 22GB even at the larger scale. Overall, the SPAN's residual accuracy gap relative to VideoITG is the direct cost of requiring no temporal grounding labels; given SPAN's scalability to far larger frame pools and lower memory footprint, a favourable trade-off for the long-video setting this paper targets.

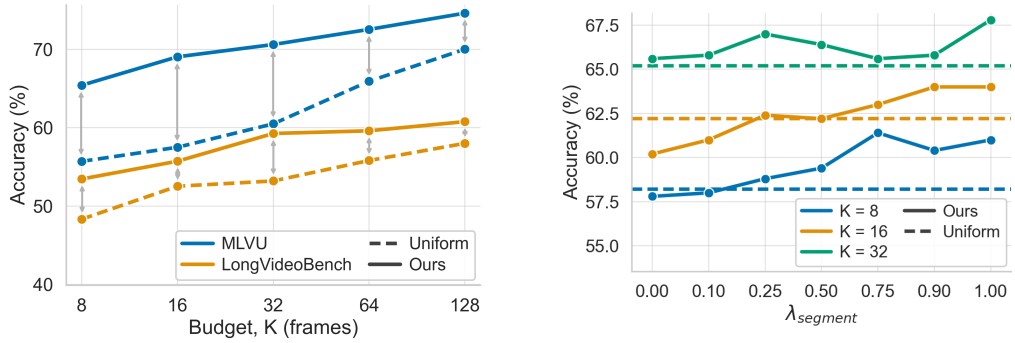



Figure 2: Qwen3-VL-8B Video QA performance with SPAN when varying. (Left) Frame Budget $K$. (Right) $\lambda_{segment}$ on EgoSchema for $K \in \{8, 16, 32\}$. Comparison shown against uniform sampling.

**Knapsack vs. flat allocation.** Table 4 isolates the value of adaptive allocation by fixing SPAN's scores and varying only how the $K$=8 budget is split: uniformly across $N \in \{1, 4, 8\}$ top-scored base clips, versus the knapsack's search over the full hierarchy. The knapsack outperforms every fixed $N \times k$ split on both benchmarks, confirming that the right breadth-density trade-off requires query-dependent search, not fixed, and a flat search to a single granularity (base clips) is subpar.

Table 4: **Knapsack vs. flat frame allocation**, at budget $K$=8 on Qwen3-VL-8B. $N \times k$ denotes $N$ top-scored base clips with $k$ frames sampled from each ($N{\cdot}k$=8); Best in **bold**.

| Frame Allocation | MLVU | LongVideoBench |
|---|---|---|
| $1 \times 8$ (max. density) | 55.84 | 51.01 |
| $4 \times 2$ | 58.10 | 49.07 |
| $8 \times 1$ (max. breadth) | 59.43 | 50.93 |
| SPAN (Knapsack) | **65.41** | **53.45** |

**Segment vs. frame relevance in the node value.** Figure 2 sweeps $(\lambda_{frame}, \lambda_{segment})$ in the node-value function (Eq. 10) for EgoSchema with Qwen3-VL-8B. Weighing $\lambda_{segment}$ higher consistently yields the best accuracy, confirming that SPAN's learned segment-level signal drives performance, not frame-level relevance alone. At $\lambda_{segment}$=0, the node value collapses to the frame-content curve alone, and the knapsack degenerates to selecting the top-$K$ frames by CLIP score directly – the weakest setting tested, at times even below Uniform Sampling.

For additional ablations on the choice of encoders and node value function, refer Appendix B, D.

## 5 Conclusion

SPAN learns a multi-scale segment relevance index from video-question-answer pairs alone, without temporal labels, using a composition function that builds coarser segment representations – and their relevance – from base clips. We pose budgeted frame selection as a tree-structured knapsack problem over this index: segment relevance determines which regions receive budget, and an off-the-shelf frame scorer determines which frames within them to sample. Across four benchmarks and two VLM families, SPAN consistently outperforms uniform and frame-level baselines, and recovers most of a temporally-supervised selector's accuracy at a fraction of its cost. Together, these make SPAN a lightweight, annotation-free plug-in that lets any frozen VLM answer from relevant evidence.

**Limitations and Future Work.** VideoITG scoring candidates with a full VLM remains more accurate; closing this gap likely requires selectively unfreezing $\theta_{vis}/\theta_{txt}$ or scaling embedding size. We validate segment relevance only via downstream VLM accuracy; future work would focus on CG-Bench's own clue-level grounding annotations to validate relevance directly. Finally, SPAN's composition operator is trained only up to the segment scale seen during training; coverage of the hour-long videos in our benchmarks is achieved by tiling multiple root-level segments, whose relevance the knapsack compares without any learned signal spanning across tiles – extending training to longer-range composition, or learning an explicit cross-tile relevance signal, are natural next steps.

## AI USE STATEMENT

In this work, we used generative AI tools to improve the readability of this paper and to assist with creating figures and implementing methods. and finding related work on Video QA. We have not used generative AI tools to conduct original research, and the remaining required disclosure tasks are not applicable to this work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REFERENCES

Ronald D Armstrong, Prabhakant Sinha, and Andris A Zoltners. The multiple-choice nested knapsack model. *Management Science*, 28(1):34–43, 1982.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. A clip-hitchhiker's guide to long video retrieval. *arXiv preprint arXiv:2205.08508*, 2022.

Leonard Bärmann, Chad DeChant, Joana Plewnia, Fabian Peller-Konrad, Daniel Bauer, Tamim Asfour, and Alex Waibel. Episodic memory verbalization using hierarchical representations of life-long robot experience. In *2025 IEEE-RAS 24th International Conference on Humanoid Robots (Humanoids)*, pp. 783–790. IEEE, 2025.

Shyamal Buch, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. Flexible frame selection for efficient video reasoning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29071–29082. IEEE, 2025.

Delong Chen, Mustafa Shukor, Theo Moutakanni, Willy Chung, Lei Yu, Tejaswi Kasarla, Allen Bolourchi, Yann LeCun, and Pascale Fung. Vl-jepa: Joint embedding predictive architecture for vision-language. In *International Conference on Learning Representations*, volume 2026, pp. 89791–89807, 2026.

Guo Chen, Yicheng Liu, Yifei Huang, Baoqi Pei, Jilan Xu, Yuping He, Tong Lu, Yali Wang, and Limin Wang. Cg-bench: Clue-grounded question answering benchmark for long video understanding. In *International Conference on Learning Representations*, volume 2025, pp. 45647–45682, 2025.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 15745–15753. IEEE, 2021.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024.

Anxhelo Diko, Tinghuai Wang, Wassim Swaileh, Shiyan Sun, and Ioannis Patras. Rewind: Understanding long videos with instructed learnable memory. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13734–13743, 2025.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. Videoagent: A memory-augmented multimodal agent for video understanding. In *European Conference on Computer Vision*, pp. 75–92. Springer, 2024.

Gabriele Goletto, Tushar Nagarajan, Giuseppe Averta, and Dima Damen. Amego: Active memory from long egocentric videos. In *European Conference on Computer Vision*, pp. 92–110. Springer, 2024.

Satya Krishna Gorti, Noël Vouitsis, Junwei Ma, Keyvan Golestan, Maksims Volkovs, Animesh Garg, and Guangwei Yu. X-pool: Cross-modal language-video attention for text-video retrieval. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4996–5005. IEEE, 2022.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13504–13514, 2024.

David S Johnson and KA Niemi. On knapsacks, partitions, and a new dynamic programming technique for trees. *Mathematics of Operations Research*, 8(1):1–14, 1983.

Yann LeCun et al. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

Jialuo Li, Bin Li, Jiahao Li, and Yan Lu. Divide, then ground: Adapting frame selection to query types for long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11369–11380, 2026.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Shuming Liu, Chen Zhao, Tianqi Xu, and Bernard Ghanem. Bolt: Boost large vision-language model without training for long-form video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3318–3327. IEEE, 2025.

Yongdong Luo, Xiawu Zheng, Guilin Li, Shukang Yin, Haojia Lin, Chaoyou Fu, Jinfa Huang, Jiayi Ji, Fei Chao, Jiebo Luo, et al. Video-rag: Visually-aligned retrieval-augmented long video comprehension. *Advances in Neural Information Processing Systems*, 38:168008–168033, 2026.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12585–12602, 2024.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023.

Niluthpol Chowdhury Mithun, Sujoy Paul, and Amit K Roy-Chowdhury. Weakly supervised video moment retrieval from text queries. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11584–11593. IEEE, 2019.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

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.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Prabhakant Sinha and Andris A Zoltners. The multiple-choice knapsack problem. *Operations Research*, 27(3):503–515, 1979.

Hui Sun, Shiyin Lu, Huanyu Wang, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Ming Li. Mdp 3: a training-free approach for list-wise frame selection in video-llms. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24090–24101. IEEE, 2025.

Quan Sun, Jinsheng Wang, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, and Xinlong Wang. Eva-clip-18b: Scaling clip to 18 billion parameters. *arXiv preprint arXiv:2402.04252*, 2024.

Canhui Tang, Zifan Han, Hongbo Sun, Sanping Zhou, Xuchong Zhang, Xin Wei, Ye Yuan, Huayu Zhang, Jinglin Xu, and Hao Sun. Tspo: Temporal sampling policy optimization for long-form video language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 9368–9376, 2026.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29118–29128. IEEE, 2025.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, et al. Embeddinggemma: Powerful and lightweight text representations. *arXiv preprint arXiv:2509.20354*, 2025.

Shihao Wang, Guo Chen, De-an Huang, Zhiqi Li, Minghan Li, Guilin Liu, Jan Kautz, Jose M Alvarez, Lei Zhang, and Zhiding Yu. Videoitg: Multimodal video understanding with instructed temporal grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24640–24650, 2026.

Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, et al. Internvideo2. 5: Empowering video mllms with long and rich context modeling. *arXiv preprint arXiv:2501.12386*, 2025a.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 3272–3283, 2025b.

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.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9777–9786, 2021.

Junbin Xiao, Angela Yao, Yicong Li, and Tat-Seng Chua. Can i trust your answer? visually grounded video question answering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13204–13214. IEEE, 2024.

Hanrong Ye, Haotian Zhang, Erik Daxberger, Lin Chen, Zongyu Lin, Yanghao Li, Bowen Zhang, Haoxuan You, Dan Xu, Zhe Gan, et al. Mmego: Towards building egocentric multimodal llms for video qa. In *International Conference on Learning Representations*, volume 2025, pp. 71705–71723, 2025.

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. In *International Conference on Learning Representations*, volume 2025, pp. 84154–84179, 2025.

Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025a.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025b.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

Yaoyao Zhong, Wei Ji, Junbin Xiao, Yicong Li, Weihong Deng, and Tat-Seng Chua. Video question answering: Datasets, algorithms and challenges. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pp. 6439–6455, 2022.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, et al. Mlvu: Benchmarking multi-task long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13691–13701, 2025.

Zirui Zhu, Hailun Xu, Yang Luo, LIU YONG, Kanchan Sarkar, Zhenheng Yang, and Yang You. Focus: Efficient keyframe selection for long video understanding. In *International Conference on Learning Representations*, volume 2026, pp. 98463–98486, 2026.

## A Illustration of Binary Tree built using SPAN

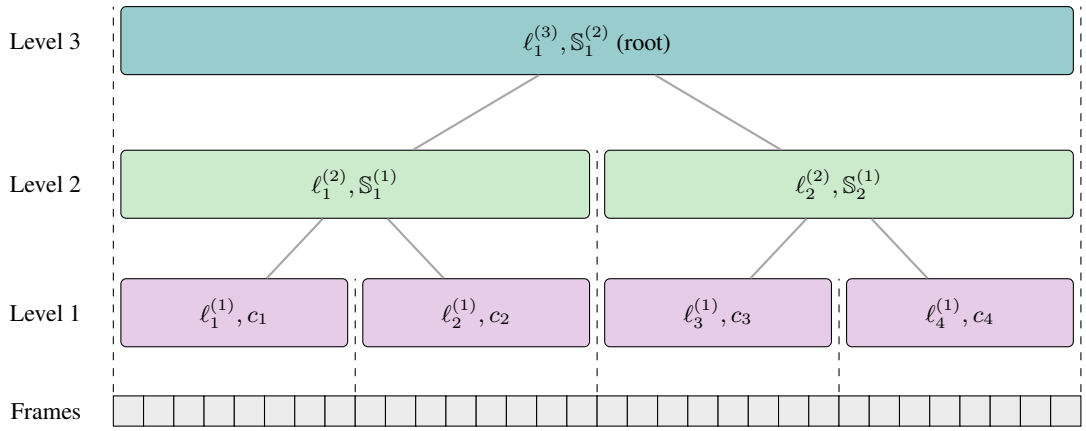



Figure 3: (Bottom-up) Multiscale binary tree-structured relevance index constructed by SPAN. Base clip representations (Level 1) are initially computed from raw frames and scored against the query. Consecutive clips are iteratively merged into coarser segments (Level 1, Level 2, ...) via a query-conditioned linear combination. This hierarchical aggregation simultaneously synthesises a merged semantic representation and assigns a temporal relevance score at each granularity. Here, $\ell_i^{(d)}$ denotes the computed relevance of the $i$-th base clip or segment at depth $d$ to the specific query.

## B Ablation Study: Choice of Encoders

Table 5: **Effect of encoder choice ($\theta_{vis}$, $\theta_{txt}$) on SPAN's learned relevance**, on MLVU across Qwen3-VL-2B/4B/8B at budgets $K \in \{8, 32\}$. To isolate the quality of the relevance signal itself from the benefit of hierarchical search, frames are selected via flat top-$K$ base clips (one frame each) – the $K\times1$ setting from Table 4 – rather than the full knapsack, so accuracy differences reflect encoder choice alone.

| **Frame Encoder**, $\theta_{vis}$ | **Text Encoder**, $\theta_{txt}$ | **Qwen3-VL-2B** | | **Qwen3-VL-4B** | | **Qwen3-VL-8B** | |
|---|---|---|---|---|---|---|---|
| | | $K=8$ | $K=32$ | $K=8$ | $K=32$ | $K=8$ | $K=32$ |
| siglip2_img | siglip2_text | 48.30 | 53.59 | 54.42 | 59.34 | 54.97 | 60.99 |
| dinov2-large | embedding_gemma | 52.35 | 57.36 | 58.42 | 62.60 | 58.56 | 64.49 |
| dinov3-l | embedding_gemma | 51.93 | 56.44 | 57.45 | 62.60 | 58.05 | 63.94 |
| dinov2-large | qwen3_0.6B | **54.00** | **57.77** | **58.83** | **63.16** | **59.43** | **64.63** |

In Table 5, we study SPAN's choice of encoder. Moving from the SigLIP family (1.0B `siglip2-so400m-patch14-384`) (Tschannen et al., 2025) to the DINO family (Oquab et al., 2023; Siméoni et al., 2025) for frame encoding generally improves performance. With a fixed text encoder, EmbeddingGemma (Vera et al., 2025), and image encoders matched in size – `dinov2-large` and `dinov3-l` are both 300M-parameter ViT-L backbones – `dinov2-large` outperforms `dinov3-l`, isolating a genuine architectural or training-data effect rather than a capacity difference. Increasing text-encoder capacity, from EmbeddingGemma (300M) to Qwen3-Embedding-0.6B (Zhang et al., 2025b) (600M, $2\times$ larger), yields a clear downstream Video QA improvement overall, suggesting relevance scoring benefits more from text-encoder capacity than image-encoder capacity at this scale. To further isolate this trend, we leave pairing `dinov2-large` with the larger 4B and 8B Qwen3-Embedding variants to future work.

## C Additional Benchmarking Results

Table 6: **Comparison of annotation-free frame-sampling strategies** (Uniform Sampling, BOLT, AKS, SPAN) across foundation model backbones on VideoQA benchmarks, at a frame budget of $K = 32$. $\Delta_{Avg.}$ reports each method's average accuracy improvement (in points), across all benchmarks, over a *Language Bias* baseline. Best result per backbone in **bold**.

| Backbone | Method | MLVU (3m – 2h) | EgoSchema (3m) | LongVideoBench (8s – 1h) | CGBench (10m – 80m) | $\Delta_{Avg.}$ |
|---|---|---|---|---|---|---|
| Qwen3-VL-2B | Lang Bias | 35.97 | 26.2 | 34.51 | 23.07 | |
| | Uniform | 54.32 | 59.0 | 48.82 | 28.60 | 17.75 |
| | Bolt | 61.27 | 60.8 | 50.84 | 32.07 | 21.31 |
| | AKS | 58.00 | 61.0 | 50.17 | 30.63 | 20.01 |
| | *SPAN(Ours)* | **63.06** | **61.4** | **51.26** | **34.97** | **22.73** |
| Qwen3-VL-4B | Lang Bias | 43.84 | 29.2 | 41.41 | 24.23 | |
| | Uniform | 60.44 | 63.4 | 53.45 | 30.87 | 17.37 |
| | Bolt | 68.12 | 64.2 | 57.66 | 33.17 | 21.12 |
| | AKS | 63.62 | 65.4 | 55.22 | 32.37 | 19.48 |
| | *SPAN(Ours)* | **69.83** | **65.8** | **58.16** | **36.67** | **22.94** |
| Qwen3-VL-8B | Lang Bias | 45.22 | 34.4 | 35.35 | 27.90 | |
| | Uniform | 60.49 | 65.2 | 53.20 | 36.00 | 18.01 |
| | Bolt | 69.00 | 64.6 | 56.14 | 38.07 | 21.23 |
| | AKS | 65.22 | 67.2 | 55.30 | 38.23 | 20.77 |
| | *SPAN(Ours)* | **70.61** | **67.8** | **59.26** | **41.23** | **24.01** |
| VideoLLaMA3-2B | Lang Bias | 44.30 | 24.4 | 39.90 | 24.70 | |
| | Uniform | 61.04 | 44.8 | 51.01 | 29.30 | 13.21 |
| | Bolt | 63.02 | 45.4 | 50.42 | 31.93 | 14.37 |
| | AKS | 63.02 | **47.4** | 50.84 | 31.63 | 14.90 |
| | *SPAN(Ours)* | **64.12** | 47.2 | **52.94** | **33.47** | **16.11** |
| VideoLLaMA3-7B | Lang Bias | 50.23 | 29.4 | 39.31 | 25.27 | |
| | Uniform | 64.35 | 56.0 | 50.67 | 29.77 | 14.15 |
| | Bolt | 68.95 | 55.6 | 52.95 | 32.23 | 16.38 |
| | AKS | 66.47 | **56.6** | 53.11 | 31.47 | 15.86 |
| | *SPAN(Ours)* | **70.01** | 56.0 | **54.71** | **32.86** | **17.34** |

Table 7: **Video QA performance of SPAN relative to the temporally-supervised VideoITG baseline**, at frame budgets $K \in \{8, 32\}$. $\mu_{rec}(\%)$ shows the share of VideoITG's accuracy gain that SPAN recovers averaged over benchmarks without any temporal annotation.

| **Backbone** | **Budget** | **Method** | **MLVU** (3m – 2h) | **LongVideoBench** (8s – 1h) | **CGBench** (10m – 80m) | $\mu_{rec}(\%)$ |
|---|---|---|---|---|---|---|
| Qwen3-VL-2B | 8 Frames | VideoITG | 62.70 | 50.25 | 32.37 | 97.89 |
| | | *SPAN* | 60.12 | 49.66 | 32.03 | |
| | 32 Frames | VideoITG | 65.23 | 53.54 | 35.37 | 97.09 |
| | | *SPAN* | 63.06 | 51.26 | 34.97 | |
| Qwen3-VL-4B | 8 Frames | VideoITG | 66.38 | 56.14 | 35.33 | 96.17 |
| | | *SPAN* | 63.80 | 54.46 | 33.70 | |
| | 32 Frames | VideoITG | 72.77 | 60.52 | 36.63 | 97.39 |
| | | *SPAN* | 69.83 | 58.16 | 36.67 | |
| VideoLLaMA3-2B | 8 Frames | VideoITG | 67.11 | 56.40 | 35.70 | 97.18 |
| | | *SPAN* | 64.08 | 53.28 | 36.27 | |
| | 32 Frames | VideoITG | 66.97 | 54.21 | 35.17 | 96.19 |
| | | *SPAN* | 64.12 | 52.94 | 33.47 | |

## D Ablation Study: Node Value Function

Table 8: **Comparison of different realisations of node value function $\Psi$ in knapsack search.** We compare the performance of Qwen3-VL-8B on MLVU, EgoSchema and LongVideoBench Benchmarks with the product $C_{\mathbb{S}}(k) \times \ell_{cal}(\mathbb{S})$ and addition-based realisations $\lambda_{frame} C_{\mathbb{S}}(k) + \lambda_{segment} \ell_{cal}(\mathbb{S})$ of the node value functions in knapsack search. Frame Budget, $K \in \{8, 32\}$. **Bold** shows the best performance. The results show that the addition performs best on average.

| **Node Value Function**, $\Psi$ | **MLVU** | | **EgoSchema** | | **LongVideobench** | |
|---|---|---|---|---|---|---|
| | $K = 8$ | $K = 32$ | $K = 8$ | $K = 32$ | $K = 8$ | $K = 32$ |
| $C_{\mathbb{S}}(k) \times \ell_{cal}(\mathbb{S})$ | **65.82** | 70.09 | 57.0 | 64.2 | 53.20 | 56.56 |
| $\lambda_{frame} C_{\mathbb{S}}(k) + \lambda_{segment} \ell_{cal}(\mathbb{S})$ | 65.41 | **70.61** | **61.4** | **67.8** | **53.45** | **59.26** |

## E WHY KNAPSACK FOR BUDGET ALLOCATION?

Section 3.2 establishes a way to obtain relevance scores for video segments. But using them to sample frames for the downstream VLM requires allocating the frame budget $K$ across many candidate segments with varying granularity and value. This is naturally posed as a knapsack problem: how to allocate the available frame budget across candidate segments, each with its own value, to maximise the total value while staying within budget. However, unlike the classic 0/1 knapsack problem, our formulation differs in two ways. First, a segment is not simply "in or out"; it can contribute anywhere from zero frames up to its full length. Second, coarser segments can have overlapping or identical underlying frames with finer segments, so they cannot be treated as independent candidates. A feasible allocation for every segment must either spend its budget directly on itself or split across the finer segments contained in it, but never both. Together, these properties place our problem in the tree-structured multiple-choice knapsack family Sinha & Zoltners (1979); Armstrong et al. (1982); Johnson & Niemi (1983).

## F ALGORITHMIC SKETCH FOR INFERENCE-TIME SEARCH WITH KNAPSACK

**Algorithm 1** Inference-Time Search for Budgeted Frame Sampling

**Require:** Frame budget $K$; query $\boldsymbol{X}$; base clips $\mathbb{C}$ and scorer $\ell(\cdot)$; depth $D$; value function $\Psi(\cdot)$; per-frame query-relevance scores

**Ensure:** Sampled frame set $\hat{\mathbb{V}}_{\boldsymbol{X}}$

1: $\mathcal{T} \leftarrow$ segment tree built from levels $1, \ldots, D$, linking each segment to its finer-level children

2: Calibrate segment-level relevance scores independently within each level

3: For each segment $\mathbb{S}$, compute a *content-aware value curve* $C_{\mathbb{S}}(k)$ = cumulative relevance mass captured by the top-$k$ ranked frames in $\mathbb{S}$

4: $\mathcal{V}^{best}(\mathbb{S}, k) \leftarrow$ best value from segment $\mathbb{S}$ under budget $k$:

$$\mathcal{V}^{best}(\mathbb{S},k) = \max\Bigg( \underbrace{\Psi\big(C_{\mathbb{S}}(k),\ \ell_{cal}(\mathbb{S})\big)}_{\text{sample k most relevant frames from } \mathbb{S}} \ , \ \underbrace{\max_{\sum_i k_i = k} \sum_{\mathbb{S}^{child} \in \text{children}(\mathbb{S})} \mathcal{V}^{best}(\mathbb{S}^{child}, k_i)}_{\text{allocate budget to children}} \Bigg)$$

5: Solve $\mathcal{V}^{best}$ bottom-up over $\mathcal{T}$, allocating total budget $K$ across root segments

6: $\hat{\mathbb{V}}_{\boldsymbol{X}} \leftarrow$ top-$k$ most query-relevant frames extracted from each selected segment $\mathbb{S}^*$

7: **return** $\hat{\mathbb{V}}_{\boldsymbol{X}}$

## G Extended Discussion of Related Work

The closest prior work is Text-Guided Attention (TGA) Mithun et al. (2019), which similarly learns attention weights from video-level weak supervision and later repurposes them as localisation scores. TGA differs from SPAN in two respects: it learns weights over individual frames rather than a coarse-to-fine segment hierarchy, and it is trained on dense video captions — natural-language descriptions of video content that constitute a comparatively heavier supervision signal than the video-question-answer pairs SPAN relies on. SPAN's relevance weights, by contrast, are defined for segments of arbitrary length up to the temporal scale seen during training, and extend to arbitrarily long videos at inference by tiling multiple such segments across the video's duration — designed specifically to supply a frozen VLM with query-relevant frames. Because TGA's frame-level attention plays a functionally similar role to the off-the-shelf frame relevance signal used in our hierarchical search, it could, in principle, serve as a drop-in substitute for that component.

**Compression-based Methods.** An alternative paradigm focuses on intermediate compression layers to reduce the number of visual tokens extracted by Vision Transformers (ViTs) Dosovitskiy et al. (2020) and to feed VLMs Li et al. (2024); Maaz et al. (2024); Cheng et al. (2024); Zhang et al. (2025a); Wang et al. (2025a); Bai et al. (2025). They can be categorised into three distinct strategies.

a. Spatial and Token-Level Compression. *Video-LLaMA* Zhang et al. (2024) utilises a learnable Q-Former Li et al. (2023) to project frames into features for a fixed number of visual tokens regardless of input resolution. Interestingly, simpler spatial compression techniques often outperformed this; *LLaVA-OneVision* Li et al. (2024) utilises simple bilinear interpolation to drastically reduce visual tokens, and subsequent iterations of Video-LLaMA Cheng et al. (2024); Zhang et al. (2025a) abandoned the Q-Former for straightforward 3D convolutions and down-sampling.

*Drawbacks:* While effective at reducing spatial redundancy, these methods fail to resolve the core temporal-length bottleneck in multi-hour videos.

b. Compression via Temporal Memory Representation. These online processing approaches ingest frames sequentially, storing features in a compressed temporal memory bank. *MA-LLM* He et al. (2024) maintains a visual memory bank of raw frame features alongside a Q-Former that generates and stores high-level semantic query embeddings. As frames arrive, the Q-Former cross-attends to the current frame and the memory bank, passing only the final semantic embeddings to the foundation model. Similarly, *ReWind* Diko et al. (2025) utilises a complex read-perceive-write cycle involving multiple Q-Formers to iteratively read historical context, perceive the current frame, and write updated representations to a memory bank.

*Drawbacks:* In both systems, the cross-attention computational cost grows linearly with the memory bank size. More critically, memory compression is irreversible. For instance, in MA-LLM, compression is largely based on feature similarity; subtle yet critical anomalous events may be permanently discarded or merged with neighbouring frames that visually resemble them, thereby corrupting episodic memory. Furthermore, ReWind requires updating the foundation model during training.

**Agentic Methods.** One class of methods indirectly performs visual search by first converting video into an intermediate textual representation via off-the-shelf captioning models, then applying agentic, LLM-based reasoning over this description to identify query-relevant content (Fan et al., 2024; Goletto et al., 2024; Bärmann et al., 2025; Wang et al., 2025b; Luo et al., 2026). *VideoTree* Wang et al. (2025b) is a training-free method that constructs a query-dependent hierarchical tree. It utilises Adaptive Breadth Expansion (via K-Means clustering on EVA-ViT features Sun et al. (2024)) to discard query-irrelevant clusters, followed by Relevance-Guided Depth Expansion to recursively sub-cluster highly relevant segments before passing them to the downstream foundation model.

*Drawbacks:* While this decouples perception from reasoning, it introduces an information bottleneck: downstream reasoning is fundamentally limited by what the captioning model chooses to describe. Visual detail that is omitted, abstracted, or incorrectly captioned is permanently unavailable to any subsequent reasoning step.

**Other Query-Conditioned Frame Selection Methods.** Concurrent works also include fully training-based selection models, such as *MM-EGO* Ye et al. (2025). MM-EGO concatenates anchor embeddings with frame and query embeddings, then computes dot products with representations extracted from the *final* layer of the foundation model. Finally, hybrid approaches like *LongVU* attempt to

remove redundant frames, identify the top query-relevant frames, and subsequently remove redundant spatial tokens within those surviving frames.

*Drawbacks:* The flaw in these frame selection architectures is their computational requirement. MM-EGO requires executing a forward pass through a massive, full-scale foundation model for *every* frame and query just to determine relevance. For multi-hour videos and dynamic querying, running a large model as a frame filter quickly becomes operationally intractable as the videos grow longer.