VIDEO ENCODERS BUILT ON IMAGE REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

The design of a video encoder determines when frames begin to interact and which frame-specific visual evidence remains accessible to the language model. Native video pathways couple neighboring frames during visual encoding, whereas image pathways preserve independently computed frame representations but incur a much larger visual-token cost when all image tokens are forwarded. We ask a basic question: *whether a compact video encoder can instead be built on image representations*. To answer this question, we separate three operations that are often coupled: per-frame representation, cross-frame token allocation, and temporal interaction. A frozen image encoder first produces frame-specific candidates. A question-aware selector then allocates a fixed token budget across frames using relevance, diversity, and cross-frame correspondence, after which a lightweight learned refiner reads neighboring-frame context and writes residual updates only to the retained anchors. This preserves source positions and keeps the visual output at the fixed budget. Across 13 benchmarks and three vision–language backbones, the resulting pathway matches full-image aggregate performance while using only about 28%–35% of its visual tokens. Specifically, on Qwen3-VL-8B, it achieves a 13-benchmark macro-average of 62.75 with $1{,}535$ visual tokens, compared with 62.58 for the full Image pathway at $4{,}424$ tokens and 59.49 for native Conv3D at $2{,}212$ tokens. On Qwen3-VL-32B, it reaches a 13-benchmark macro-average of 66.28, compared with 66.09 for Image, while providing a $2.16\times$ end-to-end speedup. These results show that compact video encoding does not require early temporal mixing: frame-specific evidence can be preserved first, allocated jointly, and temporally contextualized after selection.

1 INTRODUCTION

A video may show the same scene repeatedly, while a question depends on a brief change in a small region. For instance, in Fig. 2, repeated background views cannot substitute for the instant of take-off or contact at landing. A useful representation must retain the observations that distinguish these states without spending its budget on repeated evidence. Therefore, *making a representation compact* and *keeping its decisive evidence accessible* are not the same operation.

Native video pathways introduce temporal interaction inside the visual encoder through spatiotemporal convolutions or attention (Tran et al., 2015; Carreira & Zisserman, 2017; Bertasius et al., 2021). In the Conv3D considered here, neighboring frames contribute to shared features before question-aware selection. The resulting sequence is compact, but the selector acts on temporally coupled tokens rather than separately addressable image tokens from each frame. A brief contact change may therefore be harder to isolate from neighboring states. The concern is not that Conv3D inevitably erases motion, but that frame-specific evidence is no longer independently selectable.

In contrast, independent image encoding exposes those frame-specific candidates, but forwarding them all leaves relevance and redundancy unresolved. Repeated backgrounds, persistent objects, and overlapping views occupy token slots alongside observations needed for the question. The language model must then process a much longer sequence and identify the useful evidence within it. Full-image inference uses roughly twice the native video token count in our setting. More input does not itself identify what matters: retaining the dense candidate pool is not the same as allocating its contents according to the question.

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.

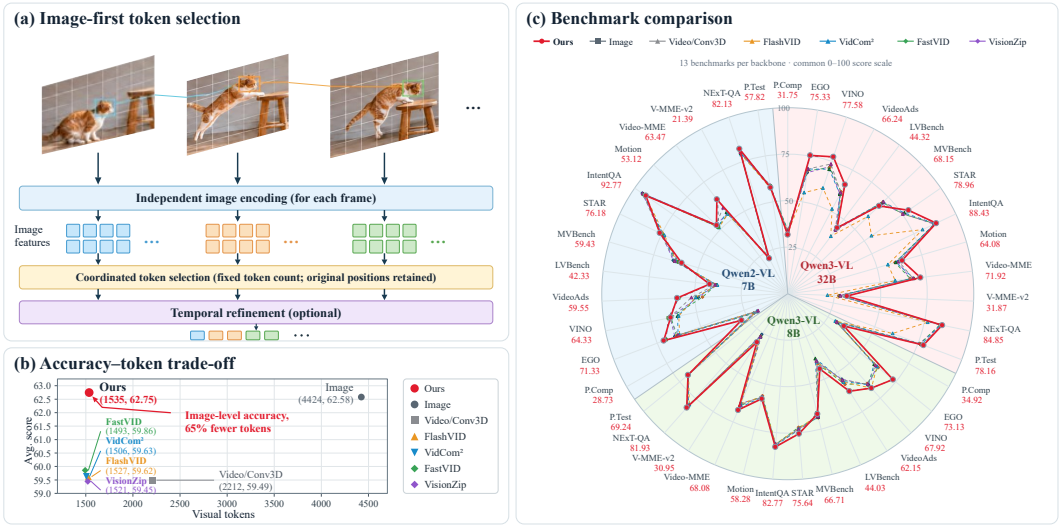


Figure 1: **Image-first video encoding.** (a) Independent encoding preserves frame-specific candidates; coordinated selection allocates the token budget; and optional refinement adds temporal context without extra output tokens. (b–c) Results compare complete pathways.

The representation–budget question. Can we retain frame-specific image evidence under a compact budget, without either mixing frames before selection or forwarding a redundant image sequence?

We address this question with an *image-first* pathway summarized in Fig. 1(a). A frozen image encoder first represents frames independently. A question-aware selector then allocates a fixed token budget across the video using relevance, diversity, and adjacent-frame correspondence. Here, *key evidence* means evidence that contributes to answering the question together with the other retained observations, not simply the largest feature change or highest individual score. Two similar-looking observations may both be needed to establish a transition, so the selector coordinates across frames without averaging their embeddings. Only afterward does an optional lightweight refiner read neighboring-frame context and write one residual update per retained anchor. Source positions and intermediate features remain aligned, and the output remains at the same fixed token budget.

This ordering separates representation, allocation, and interaction. It also makes the limits of selective compression explicit. A conditional KL decomposition distinguishes evidence made unavailable by selection from error in using the evidence that remains: a larger refiner cannot recover information it cannot observe. A complementary fixed-operator analysis identifies the temporal interactions omitted when refinement cannot access unselected context. Together, these results motivate preserving candidates before compression and allowing selected anchors to read additional temporal context without returning the full image sequence to the language model. They do not imply that later interaction is universally better.

Empirically, this decomposition produces a strong accuracy–budget trade-off. Across 13 benchmarks and three vision–language backbones, the image-first pathway matches full-image aggregate performance while using only about 28%–35% of its visual tokens. On Qwen3-VL-8B, it reaches a 13-benchmark macro-average of 62.75 with 1,535 visual tokens, compared with 62.58 for the full Image pathway at 4,424 tokens. On Qwen3-VL-32B, it reaches 66.28 versus 66.09 for Image while providing a 2.16× end-to-end speedup. Fig. 1(b–c) summarizes the accuracy–token and cross-benchmark comparisons.

Image and video pathways also differ in positions and intermediate streams (Wang et al., 2024; Bai et al., 2025); we therefore preserve the image source interface rather than attributing complete-pipeline differences to token count alone. The resulting design principle is to *preserve candidates before selection, allocate the compact budget jointly, and add temporal context where needed*, while keeping the output budget fixed. Our contributions are:

- We formulate compact video encoding by separating per-frame representation, cross-frame token allocation, and temporal interaction, and derive analyses that distinguish missing evidence from prediction error and isolate interaction lost across the selection boundary.

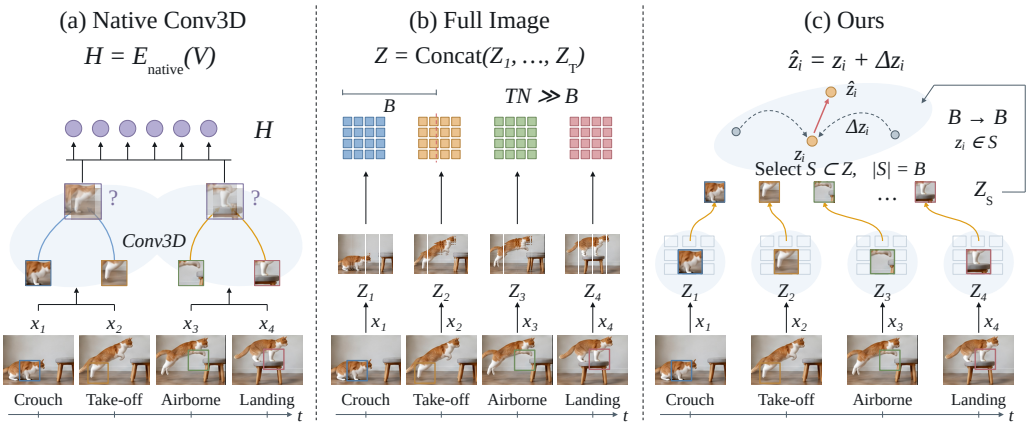


Figure 2: **Three video-input pathways.** (a) Conv3D mixes neighboring frames before selection. (b) Full Image preserves frame-specific tokens but forwards the dense sequence. (c) Ours selects B source anchors before temporal refinement, preserving $B \to B$. Highlighted patches indicate source support only.

- We introduce an image-first pathway that combines frozen framewise encoding with question-aware coordinated selection and a lightweight learned temporal refiner, preserving source alignment while keeping the visual output at a fixed token budget.
- We evaluate the resulting design across 13 video-understanding benchmarks and three vision–language backbones, showing that it matches full-image aggregate performance with about 28%–35% of the visual tokens while providing substantial inference savings.

2 Problem Formulation

Given a sampled video $\mathcal{V} = (x_1, \ldots, x_T)$, a question q, and a budget B, we seek a compact input that retains useful frame-specific evidence. Native Conv3D couples neighboring observations before selection; Full Image retains every candidate without a question-specific output allocation. We instead allocate B slots to relevant, complementary source evidence, keeping additional temporal context separately accessible (Fig. 2).

2.1 Source-Aligned Candidates and a Budgeted Interface

A frozen image encoder produces $\mathbf{Z}_t = E_{\text{img}}(x_t)$ with N_t tokens of dimension D. Concatenation gives $\mathbf{Z} = \text{Concat}(\mathbf{Z}_1, \ldots, \mathbf{Z}_T)$ with $M = \sum_t N_t$ rows. Computing $\mathbf{Z}_t$ uses no other sampled frame; this is computational separation, not statistical independence of frames. The native pathway instead produces $\mathbf{H} = E_{\text{native}}(\mathcal{V})$ with L_{native} rows. We focus on $B \le L_{\text{native}} < M$: the dense image sequence is a pool of selectable observations, not the intended language-model input. Each candidate carries an embedding $\mathbf{z}_i$, a source position $p_i = (\tau_i, u_i, v_i)$, and intermediate features d_i; $\mathbf{P}$ and $\mathbf{D}$ collect the associated streams.

Source-aligned selection. A selector returns B distinct indices S and applies the same row-selection matrix P_S to every stream:

$$\mathbf{Z}_S = P_S\mathbf{Z}, \qquad \mathbf{P}_S = P_S\mathbf{P}, \qquad \mathbf{D}_S = P_S\mathbf{D}, \qquad |S| = B. \tag{1}$$

The mask $S = S(\mathcal{V}, q)$ may depend on the whole video, but selection changes neither the retained embeddings nor their source metadata. Optional refinement subsequently writes $\widehat{\mathbf{Z}}_S = \mathbf{Z}_S + \Delta_\theta \in \mathbb{R}^{B\times D}$ while leaving $\mathbf{P}_S$ and $\mathbf{D}_S$ unchanged. *Source alignment identifies the output anchor; it does not restrict refined content to one frame.*

2.2 Separating Missing Evidence from Prediction Error

Discarding a repeated background and discarding the only visible landing contact both save tokens, but need not remove the same predictive evidence. To distinguish missing evidence from difficulty using what remains, fix a distribution over $(\mathcal{V}, q)$ and let $p_F = p_{\text{img}}(\cdot \mid \mathcal{V}, q)$ be the frozen

full-image pathway's first-answer-token distribution. Without extra context, the available record is $\mathcal{O} = (q, S, \mathbf{Z}_S, \mathbf{P}_S, \mathbf{D}_S)$. The mask S is included because adaptive selection can itself convey information. For any predictor $p_\theta(\cdot \mid \mathcal{O})$ measurable with respect to this record, define $\bar{p}_\mathcal{O} = \mathbb{E}[p_F \mid \mathcal{O}]$.

Proposition 1 (A conditional prediction-error decomposition)**.** *For a finite output vocabulary and finite expected KL divergence,*

$$\mathbb{E}\, D_{\mathrm{KL}}(p_F \| p_\theta) = \underbrace{\mathbb{E}\, D_{\mathrm{KL}}(p_F \| \bar{p}_\mathcal{O})}_{\mathcal{I}(\mathcal{O})\text{: unavailable evidence}} + \underbrace{\mathbb{E}\, D_{\mathrm{KL}}(\bar{p}_\mathcal{O} \| p_\theta)}_{\mathcal{A}_\theta(\mathcal{O})\text{: prediction error}}. \tag{2}$$

Thus $\mathcal{I}(\mathcal{O})$ is the minimum expected divergence over all predictors observing $\mathcal{O}$.

The first term concerns which evidence is accessible: two videos with identical selected records but different unobserved contact states may induce different teacher predictions. No predictor of that record can distinguish them. The second concerns how available evidence is used: a frozen language model with a small refiner may not realize the best predictor even when the clue is retained. More capacity cannot remove the first limitation. Teacher agreement is not answer correctness.

Reading context $\mathcal{C}$ expands the record to $\mathcal{O}^+ = (\mathcal{O}, \mathcal{C})$, including all retrieved indices, features, and other accessed information. These nested records satisfy

$$\mathcal{I}(\mathcal{O}) = \mathcal{I}(\mathcal{O}^+) + \mathbb{E}\, D_{\mathrm{KL}}(\bar{p}_{\mathcal{O}^+} \| \bar{p}_\mathcal{O}), \tag{3}$$

and therefore $\mathcal{I}(\mathcal{O}^+) \le \mathcal{I}(\mathcal{O})$. Additional context cannot increase the information floor, although a learned predictor can still perform worse. Crucially, this context need not consume output token slots. Appendix F gives the identity and proofs.

2.3 When Can Temporal Interaction Be Postponed?

A selected contact patch may need neighboring-frame evidence that received no output slot. To expose this interaction gap, consider $F_A(\mathbf{Z}) = (I + A)\mathbf{Z}$ with fixed $A \in \mathbb{R}^{M \times M}$. For index sets $R, Q \subseteq \{1, \dots, M\}$, write $A_{RQ} = P_R A P_Q^\top$ and let $\bar{S}$ denote the unselected indices. This diagnostic model does not assert that the native backbone is linear or equals $F_A \circ E_{\mathrm{img}}$.

Proposition 2 (The interaction omitted by selection)**.** *Fix A and S. The two operation orders differ by*

$$\underbrace{P_S(I + A)\mathbf{Z}}_{\text{interact, then select}} - \underbrace{(I_B + A_{SS})\mathbf{Z}_S}_{\text{select, then interact}} = A_{S\bar{S}}\mathbf{Z}_{\bar{S}}. \tag{4}$$

They agree for every $\mathbf{Z}$ if and only if $A_{S\bar{S}} = 0$.

Both orders produce B rows; their difference is interaction across the selection boundary, not output length. Reading context $C \subseteq \bar{S}$ restores $A_{SC}\mathbf{Z}_C$. For the remaining unobserved indices $U = \bar{S} \setminus C$, the exact worst-case omitted-interaction residual is

$$\sup_{\|\mathbf{Z}_U\|_F \le R} \|A_{SU}\mathbf{Z}_U\|_F = R\|A_{SU}\|_2. \tag{5}$$

This statement fixes the operator, selection set, retrieved context, and observed rows; it is not an accuracy bound for an adaptive neural encoder. Appendix G gives the proof.

The design must therefore avoid two different mistakes: spending output slots on repeated observations, and making a selected observation unusable by cutting off its relevant context. Selection determines which source evidence occupies the compact sequence; refinement controls which additional temporal evidence can inform it. Neither task requires mixing frames when the image candidates are first formed.

3 Method

The formulation above suggests two design requirements: spend a fixed output budget on useful source-aligned evidence, and allow retained anchors to access relevant temporal context beyond the selected set without expanding that output. The first controls *which evidence survives the budget*, as captured by the information floor in Proposition 1; the second controls *which interactions remain accessible after selection*, as exposed by Proposition 2. Cross-frame information may therefore guide allocation without modifying the frame-specific candidates themselves, while additional temporal context can be introduced after the source anchors have been chosen.

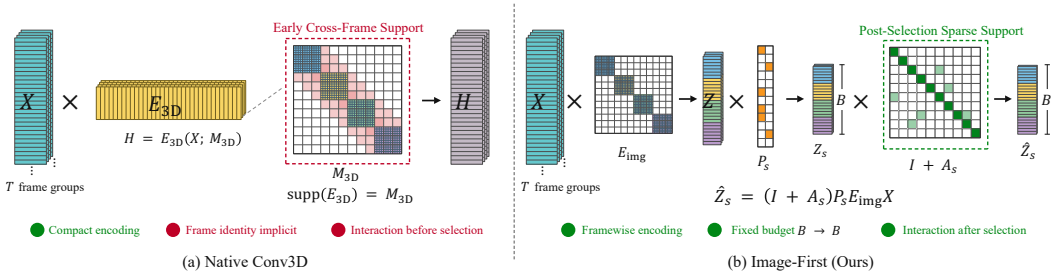


Figure 3: **Selection before temporal refinement.** Left: native encoding mixes neighboring frames before selection. Right: framewise candidates remain separate through budgeted selection, followed by sparse residual interaction. The displayed $(I_B + A_{SS})\mathbf{Z}_S$ uses selected-token context; the refiner may also read unselected neighbors. Matrix blocks indicate dependencies, not linearity.

We instantiate these requirements with an *image-first* pathway. Independent encoding keeps frame-specific candidates separately addressable; coordinated selection allocates the budget to relevant, non-repeated, and complementary evidence; and optional refinement writes local temporal context into the retained anchors without restoring the dense output. The image encoder, language model, and selector remain frozen; only the refiner is learned. Fig. 3 shows this ordering, and Algorithm 1 (Appendix) summarizes inference.

3.1 Preserve Frame-Specific Candidates Before Selection

Independent encoding keeps observations from different frames separately addressable until selection. In the running example, take-off and landing patches remain distinct candidates rather than entering selection only through a joint neighboring-frame feature. This preserves the choice of which image features to retain; it is not a claim of lossless pixel encoding. In our Qwen3-VL instantiation, each candidate carries its final embedding, source MRoPE position, and associated DeepStack features (Bai et al., 2025). The image and video pathways share the pretrained visual model, but differ in how temporal content enters their visual representations. Selection applies the same retained indices to every image stream, without pooling embeddings or assigning new coordinates. Unselected features can remain accessible until refinement finishes: the candidate pool need not be the language-model input. The point is not to remove temporal information, but to postpone where it enters: selection acts on source-aligned image candidates, while cross-frame content is written only after the budgeted anchors are fixed.

3.2 Select Question-Relevant and Complementary Evidence

A useful subset is not merely a collection of individually salient tokens. Question relevance identifies what to look for, diversity limits repeated evidence, and correspondence preserves observations whose value comes from being seen together. Because the conditional distributions in Proposition 1 are unavailable at inference, we use frozen surrogates. Let $\bar{\mathbf{z}}_i$ denote the mean representation over nearby-frame correspondences and $m_i(q) \in [-1, 1]$ the question-relevance score. Our guiding preference is

$$\Phi(S; q) = \sum_{i \in S} \|\mathbf{z}_i - \bar{\mathbf{z}}_i\|_2 m_i(q) - \lambda \sum_{i,j \in S, i \neq j} \text{Sim}(\mathbf{z}_i, \mathbf{z}_j), \qquad |S| = B. \tag{6}$$

The first term favors question-relevant changes rather than motion alone, while the second discounts repeated evidence. Eq. 6 is a selection principle rather than an exact optimization of $\mathcal{I}(\mathcal{O})$.

Allocate across frames. We inherit integer frame quotas b_t from a frozen allocator, with $0 \leq b_t \leq N_t$ and $\sum_t b_t = B$, requiring $|S \cap I_t| = b_t$ for each frame index set I_t. Quotas determine where across time to spend the budget; selection within each frame determines which regions occupy slots.

Select relative to what is already retained. Within each frame, a feature-norm- and question-weighted determinantal point process (DPP) favors candidates that add new directions to the subset (Kulesza & Taskar, 2012). Across adjacent frames, a sparse correspondence graph uses feature similarity, spatial consistency, matching confidence, and local state change to reward jointly retaining linked observations. This distinction matters because visually similar endpoints may be redundant within a frame yet complementary across time when their comparison identifies a transition.

The resulting greedy procedure considers feasible singleton and adjacent-frame pair moves under the remaining quotas. It terminates with exactly B distinct source anchors but makes no claim of global subset optimality. Selection changes the retained mask, not the retained embeddings: it does not average, merge, or temporally update source features. Appendix E gives the DPP marginal, pair objective, complementarity analysis, and feasibility proof.

3.3 Zero-Expansion Temporal Refinement

Selection determines which source evidence occupies the compact sequence; refinement determines which additional temporal context can inform that evidence. A retained contact patch, for example, may still require a nearby frame to clarify whether contact has just occurred or has already been broken. Motivated by Proposition 2, each selected anchor $i \in S$ therefore retrieves a fixed number of feature-similar neighboring-frame tokens, with context indices $C(i)$ not restricted to S.

Bottleneck attention uses relative temporal and spatial information to compute normalized weights α_{ij}. With value projection W_V and a gated, bounded residual branch G_θ, the refined anchor is

$$\widehat{\mathbf{z}}_i = \mathbf{z}_i + G_\theta\left(\mathbf{z}_i, \sum\nolimits_{j \in C(i)} \alpha_{ij} W_V \mathbf{z}_j\right), \qquad i \in S. \tag{7}$$

The gate and bounded residual output limit the modification to each source embedding. Context may come from preceding or following frames in the offline setting, but every selected anchor still produces exactly one output embedding. Hence $\widehat{\mathbf{Z}}_S \in \mathbb{R}^{B \times D}$, while source positions p_i and intermediate features d_i remain unchanged.

In the language of Proposition 1, context retrieval changes what evidence can be read, whereas the residual branch changes how that evidence is encoded. Only the refiner parameters θ are learned; the image encoder, language model, and selector remain frozen. Training combines answer supervision with KL distillation from the full-image teacher at temperature τ. The teacher is used only during training, and zero-initialized residuals make optimization begin from the unrefined selected-token pathway. Disabling refinement therefore recovers the training-free selection pathway.

Zero expansion refers specifically to the visual output interface: the refiner maps B retained anchors to B refined anchors while being allowed to read additional context. It does not remove the cost of frame encoding, context retrieval, or refinement, and it does not by itself imply an end-to-end speedup. Appendix H gives additional architectural and learning-objective details.

4 Experiments

We evaluate the image-first pathway on 13 video-understanding benchmarks across three vision–language model backbones. The experiments ask two questions: whether coordinated selection improves performance at a fixed visual-token budget, and whether a compact image-derived pathway can retain the aggregate performance of full-input references while reducing inference cost.

Experimental setup. Our evaluation spans general video understanding, temporal reasoning, long-video understanding, intent recognition, and fine-grained perception. Table 1 reports six temporal, long-video, and perception benchmarks, while Fig. 4 presents the remaining seven. We follow the evaluation protocol of each benchmark and report accuracy (%). We additionally report unweighted benchmark-level macro-averages: $\mathrm{Avg.}(6)$ over the six benchmarks in Table 1 and $\mathrm{Avg.}(13)$ over all 13 benchmarks. Both are computed directly from the individual benchmark scores.

We evaluate Qwen2-VL-7B, Qwen3-VL-8B, and Qwen3-VL-32B (Wang et al., 2024; Bai et al., 2025). For each backbone, *Video/Conv3D* uses the native video-encoding pathway, whereas *Image* supplies the sampled frames through the image pathway. Both uniformly sample 32 frames without an additional visual-token cap and therefore serve as full-input references rather than budget-matched compression baselines. We compare against Uniform, FlashVID, VidCom^2, FastVID, and VisionZip (Fan et al., 2026; Liu et al., 2025; Shen et al., 2025; Yang et al., 2025), and additionally evaluate FastV, SparseVLM, and DyCoke on Qwen3-VL-8B (Chen et al., 2024; Zhang et al., 2025a; Tao et al., 2025). All compression methods target approximately 1.5K visual tokens. We use public implementations and recommended configurations, adapted to this target budget; actual mean token counts vary slightly with frame organization, merging, and padding and are reported in Table 1.

Table 1: **Six benchmarks.** Avg.(6) is the unweighted mean. Bold and underlined values mark best and second-best budget-matched results; Image and Video/Conv3D are full-input references.

Method	Visual tokens ↓	Motion Bench	Video-MME	Video-MME-v2	NExT-QA	Perception Test	Perception Comp	Avg. (6) ↑
Qwen2-VL-7B								
Video/Conv3D	2,780	53.61	57.78	22.50	80.33	59.17	18.39	48.63
Image	5,561	51.62	62.78	22.03	81.55	58.65	28.25	50.81
Uniform	1,536	50.11	51.65	20.35	76.88	55.42	17.65	45.34
FlashVID	1,536	51.27	53.64	21.67	78.89	57.39	19.13	47.00
VidCom^2	1,529	51.84	<u>54.78</u>	22.37	79.77	<u>58.02</u>	19.24	47.67
FastVID	1,522	52.02	53.49	**22.84**	<u>80.24</u>	57.79	<u>21.13</u>	47.92
VisionZip	1,532	**53.26**	54.03	<u>22.70</u>	79.53	**58.34**	21.06	<u>48.15</u>
Ours	1,536	<u>53.12</u>	**63.47**	21.39	**82.13**	57.82	**28.73**	**51.11**
Qwen3-VL-8B								
Video/Conv3D	2,212	57.34	65.00	27.50	80.91	68.99	30.49	55.04
Image	4,424	58.08	67.96	30.78	81.84	69.28	34.53	57.08
Uniform	1,536	55.48	61.22	24.35	77.12	66.85	27.65	52.11
FastV	1,536	56.12	63.45	25.88	78.45	67.24	29.15	53.38
SparseVLM	1,536	56.45	63.85	26.15	78.95	67.65	29.55	53.77
DyCoke	1,536	56.85	64.25	26.65	79.35	68.12	29.85	54.18
FlashVID	1,527	57.12	64.40	<u>27.69</u>	79.75	68.85	29.58	54.57
VidCom^2	1,506	57.31	<u>66.88</u>	27.22	80.51	68.94	<u>30.98</u>	<u>55.31</u>
FastVID	1,493	56.63	65.69	27.23	81.08	68.72	30.07	54.90
VisionZip	1,521	<u>57.43</u>	64.12	26.05	<u>81.18</u>	<u>69.11</u>	30.08	54.66
Ours	1,535	**58.28**	**68.08**	**30.95**	**81.93**	**69.24**	**34.92**	**57.23**
Qwen3-VL-32B								
Video/Conv3D	2,212	62.19	67.96	28.44	84.53	77.53	33.63	59.05
Image	4,424	63.93	71.85	31.72	84.76	78.00	31.39	60.28
Uniform	1,536	58.25	62.15	24.85	80.12	72.55	30.15	54.68
FlashVID	1,527	56.14	58.29	21.46	76.89	60.42	32.71	50.99
VidCom^2	1,504	60.52	<u>68.38</u>	<u>28.23</u>	84.01	<u>76.92</u>	<u>33.68</u>	<u>58.62</u>
FastVID	1,485	60.12	66.42	27.55	83.59	76.12	**34.48**	58.05
VisionZip	1,521	<u>61.64</u>	66.54	27.61	<u>84.12</u>	75.89	33.22	58.17
Ours	1,535	**64.08**	**71.92**	**31.87**	**84.85**	**78.16**	31.75	**60.44**

Our method uses the image-first pathway described in Section 3: the image encoder, language model, and selector remain frozen, while the lightweight temporal refiner is learned. Within each backbone, all methods use the same evaluation samples, questions, prompt templates, generation settings, and answer-parsing procedures. We report mean accuracy over three runs with seeds 42, 3407, and 114514. Visual-token counts denote the mean number of visual tokens supplied to the language model, excluding text prompts and generated answers. Rankings among compression methods use only budget-matched methods.

4.1 Main Results

Table 1 reports MotionBench, Video-MME, Video-MME-v2, NExT-QA, Perception Test, and PerceptionComp (Hong et al., 2025; Fu et al., 2025; 2026; Xiao et al., 2021; Pătrăucean et al., 2023; Li et al., 2026). Unless stated otherwise, result triplets follow Qwen2-VL-7B, Qwen3-VL-8B, and Qwen3-VL-32B.

Compact image representations outperform budget-matched compression. At comparable visual-token budgets, our method achieves Avg.(6) scores of **51.11**, **57.23**, and **60.44**, exceeding the strongest compression baseline by **2.96**, **1.92**, and **1.82** percentage points, respectively. The corresponding strongest baselines by six-benchmark macro-average are VisionZip, VidCom^2, and VidCom^2. Our method obtains the highest score among compression methods on 3/6, 6/6, and 5/6 benchmarks, respectively. The gain is not uniform: on Qwen3-VL-32B, PerceptionComp remains below FastVID (31.75 vs. 34.48). **Most of the full-image token sequence is unnecessary for aggregate performance.** Our method uses 1,536, 1,535, and 1,535 visual tokens, compared with 5,561, 4,424, and 4,424 for Image, corresponding to reductions of **72.4%**, **65.3%**, and **65.3%**.

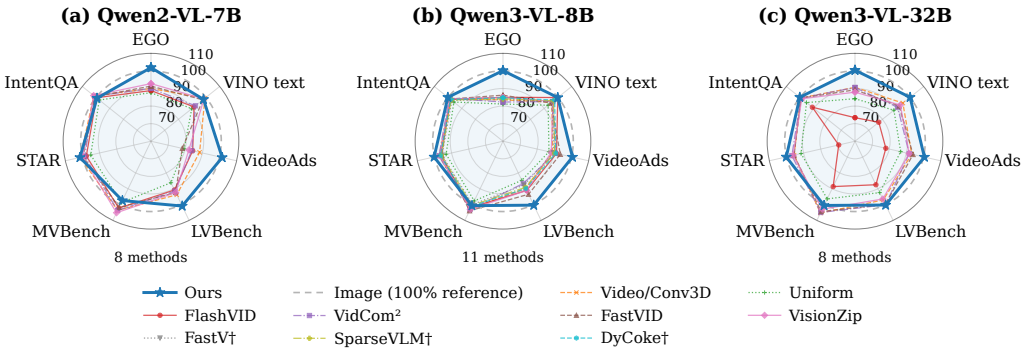


Figure 4: **General video understanding across three backbones.** Scores are normalized to the corresponding Image pathway (100%). The shared radial range is 60%–110% and does not start at zero. FastV, SparseVLM, and DyCoke are evaluated only on Qwen3-VL-8B. Exact scores are reported in Appendix B.

Despite these reductions, its six-benchmark macro-averages are comparable to, and slightly above, Image's 50.81, 57.08, and 60.28. The main result is therefore the retention of aggregate full-image performance under a substantially smaller visual-token budget, rather than the small numerical gains over the full-input reference.

4.2 Cross-Benchmark Generalization

We further evaluate EgoSchema (EGO), VINO text, VideoAds, LVBench, MVBench, STAR, and IntentQA (Mangalam et al., 2023; Zhang et al., 2024; 2025b; Wang et al., 2025; Li et al., 2024; Wu et al., 2021; Li et al., 2023). Fig. 4 normalizes each score to the Image reference for the same backbone and benchmark. Exact scores and 13-benchmark macro-averages are reported in Appendix B.

The trade-off persists across task types and model scales. Relative to the strongest compression baseline on each benchmark, our method achieves higher scores on **4/7**, **5/7**, and **6/7** benchmarks, with an additional tie on VINO text for Qwen3-VL-8B. Improvements on EGO, VideoAds, LVBench, and STAR hold across all three backbones. MVBench is a consistent exception, where our scores trail the strongest compression baseline by 4.69, 2.13, and 3.01 percentage points. Thus the advantage extends across several task families without implying uniform improvement on every benchmark. Across all 13 benchmarks, our method reaches $\mathrm{Avg.}(13)$ scores of **59.43**, **62.75**, and **66.28**. Compared with the strongest compression baseline by macro-average (VisionZip, FastVID, and VidCom^2, respectively), our method improves by **2.55**, **2.89**, and **2.42** percentage points. They are also close to and slightly above the corresponding Image results of 59.26, 62.58, and 66.09. Overall, the image-first pathway retains full-input aggregate performance while using only approximately **28%–35%** of Image's visual tokens.

4.3 Efficiency Analysis

We profile inference on Qwen3-VL-32B-Instruct using a single GPU with batch size 1. Time to first token (TTFT) includes visual encoding, token selection, temporal refinement, and language-model prefill; end-to-end latency additionally includes autoregressive decoding. The reported $\mathrm{Avg.}(13)$ is the three-seed macro-average from the main evaluation. As Table 2 shows:

Compression reduces the pre-generation cost. Relative to Image, Ours reduces visual tokens from 4,424 to 1,535 (**65.3%**), TTFT from 849.1 ms to 365.8 ms (**56.9%**), and end-to-end latency from 911.2 ms to 421.8 ms (**53.7%**). Throughput increases from 1.10 to 2.37 queries/s, corresponding to a 2.16× speedup, while peak GPU memory decreases by 1.39 GB. These savings retain aggregate performance: 66.28 for Ours versus 66.09 for Image.

The main savings occur before autoregressive decoding. Fig. 5 in the Appendix shows that TTFT accounts for most of the reduction, falling from 849.1 ms for Image to 365.8 ms for Ours, whereas post-first-token generation changes only from 62.1 ms to 56.0 ms. The 14.52 ms selection cost is already included in TTFT and is offset by the shorter visual prefill. At the same ∼1.5K-token budget, Ours adds only 8.0 ms over FastVID for a **2.70**-point macro-average gain and 18.5 ms over

Table 2: **Inference efficiency on Qwen3-VL-32B-Instruct.** Speedup is relative to Image using end-to-end latency.

Method	Visual tokens ↓	Selection (ms) ↓	TTFT (ms) ↓	E2E (ms) ↓	Throughput (q/s) ↑	Memory (GB) ↓	Avg.(13) ↑	Δ Image (pp) ↑	Speedup ↑
Video/Conv3D	2212.0	–	454.0	508.5	1.97	67.97	64.20	-1.89	1.79×
Image	4424.0	–	849.1	911.2	1.10	69.04	66.09	0.00	1.00×
FlashVID	**1507.4**	43.59	**297.7**	**330.5**	**3.03**	**67.52**	54.32	-11.77	**2.76**×
VidCom^2	1515.3	**3.28**	348.4	403.3	2.48	67.88	63.86	-2.23	2.26×
FastVID	1515.0	15.38	356.7	413.8	2.42	67.81	63.58	-2.51	2.20×
VisionZip	1532.0	57.16	409.2	459.4	2.18	67.82	63.29	-2.80	1.98×
Ours	1535.0	14.52	365.8	421.8	2.37	67.65	**66.28**	**+0.19**	2.16×

VidCom^2 for a **2.42**-point gain. It also exceeds VisionZip by **2.99** points while being 37.6 ms faster. FlashVID achieves the lowest latency, but with an 11.77-point reduction relative to Image. The resulting trade-off is therefore not minimum latency alone, but substantial inference savings while retaining full-input aggregate performance.

4.4 Ablation Study

We isolate dynamic token allocation, cross-frame coordination, and temporal refinement, with component and system-level ablations reported in Appendix C. On Qwen3-VL-8B, dynamic allocation raises the 13-benchmark macro-average from 60.12 to 61.74, cross-frame coordination further increases it to 62.64, and temporal refinement reaches 62.75. Additional Qwen2-VL-7B ablations show sensitivity to correspondence quality, candidate-frame coverage, and visual-token budget.

5 Related Work

Video encoders differ in when temporal information is introduced. Frame-based methods aggregate independently encoded images over time, while C3D, I3D, and video Transformers introduce spatiotemporal interaction within the encoder (Karpathy et al., 2014; Simonyan & Zisserman, 2014; Donahue et al., 2017; Tran et al., 2015; Carreira & Zisserman, 2017; Bertasius et al., 2021; Arnab et al., 2021). Video large language models inherit this choice under a sequence-length constraint: some aggregate image-derived features before language modeling, while unified models such as Qwen2-VL and Qwen3-VL share much of the image–video stack but differ in how temporal content enters the visual representation (Maaz et al., 2024; Zhang et al., 2023; Weng et al., 2024; Wang et al., 2024; Bai et al., 2025). Recent methods also reduce visual sequences through token selection, pruning, or merging (Yang et al., 2025; Shen et al., 2025; Liu et al., 2025; Fan et al., 2026). Our study instead separates representation source, token allocation, and temporal interaction. Related distinctions among temporal-fusion stages have been studied in MotionBench (Hong et al., 2025); here we hold image representations and output budgets fixed while varying coordinated selection and post-selection refinement. We provide a more complete discussion of related work in Appendix A.

6 Conclusion

We study whether compact video encoding can be built on image representations without either mixing frames before selection or forwarding the full dense image sequence. Our image-first design separates *representation*, *allocation*, and *interaction*: frame-specific candidates are preserved first, a fixed output budget is allocated jointly across the video, and temporal context is introduced afterward through lightweight refinement. The accompanying analysis distinguishes evidence lost at selection from error in using retained evidence, and characterizes the interaction omitted across the selection boundary. Across 13 benchmarks and three vision–language backbones, this design retains full-image aggregate performance while using only about 28%–35% of the visual tokens, with substantial inference savings. These results suggest a simple design principle for video representation: preserve source evidence before compression, allocate the limited budget to complementary observations, and add temporal context only where it is needed.

AI USE STATEMENT

Generative AI tools are employed in a supporting capacity during the preparation of this work, primarily for tasks such as assisting with code development and troubleshooting, improving linguistic clarity, and organizing manuscript content. All substantive scientific contributions, including the formulation of the methodological framework, the design and execution of experiments, the interpretation of results, and the determination of the claims presented in the paper, are made by the authors. Any large language models discussed in the manuscript are treated as objects of experimental investigation rather than as contributors to the research.

Where AI-assisted programming is used, the resulting code is independently inspected, tested, and validated by the authors to confirm that the implemented procedures match the intended computations. Reported numerical values are cross-checked against the corresponding experimental records, and all AI-assisted textual material is critically reviewed, edited, and, where necessary, rewritten. Responsibility for the accuracy, integrity, and final form of the manuscript, mathematical formulations, experimental results, software, and supplementary materials rests entirely with the authors.

REFERENCES

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6836–6846, 2021.

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.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 813–824. PMLR, 2021.

João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4733, 2017. doi: 10.1109/CVPR.2017.502.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *Computer Vision – ECCV 2024*, pp. 19–35, 2024. doi: 10.1007/978-3-031-73004-7_2.

Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):677–691, 2017. doi: 10.1109/TPAMI.2016.2599174.

Junhao Du, Jialong Xue, Anqi Li, Jincheng Dai, and Guo Lu. Unified spatiotemporal token compression for video-llms at ultra-low retention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17661–17671, 2026.

Ziyang Fan, Keyu Chen, Ruilong Xing, Yulin Li, Li Jiang, and Zhuotao Tian. FlashVID: Efficient video large language models via training-free tree-based spatiotemporal token merging. In *International Conference on Learning Representations (ICLR)*, 2026.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118, 2025. doi: 10.1109/CVPR52734.2025.02245.

Chaoyou Fu, Haozhi Yuan, Yuhao Dong, Yi-Fan Zhang, Yunhang Shen, Xiaoxing Hu, Xueying Li, Jinsen Su, Chengwu Long, Xiaoyao Xie, Yongkang Xie, Xiawu Zheng, Xue Yang, Haoyu Cao, Yunsheng Wu, Ziwei Liu, Xing Sun, Caifeng Shan, and Ran He. Video-mme-v2: Towards the next stage in benchmarks for comprehensive video understanding. *arXiv preprint arXiv:2604.05015*, 2026.

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 (CVPR)*, pp. 13504–13514, 2024. doi: 10.1109/CVPR52733.2024.01282.

Wenyi Hong, Yean Cheng, Zhuoyi Yang, Weihan Wang, Lefan Wang, Xiaotao Gu, Shiyu Huang, Yuxiao Dong, and Jie Tang. MotionBench: Benchmarking and improving fine-grained video motion understanding for vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8450–8460, 2025.

Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, 2014. doi: 10.1109/CVPR.2014.223.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2-3):123–286, 2012.

Jiapeng Li, Ping Wei, Wenjuan Han, and Lifeng Fan. Intentqa: Context-aware video intent reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11963–11974, 2023.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22195–22206, 2024.

Shaoxuan Li, Zhixuan Zhao, Hanze Deng, Zirun Ma, Shulin Tian, Zuyan Liu, Yushi Hu, Haoning Wu, Yuhao Dong, Benlin Liu, Ziwei Liu, and Ranjay Krishna. Perceptioncomp: A video benchmark for complex perception-centric reasoning. In *Computer Vision – ECCV 2026*, pp. 241–257, 2026.

Xuyang Liu, Yiyu Wang, Junpeng Ma, and Linfeng Zhang. Video compression commander: Plug-and-play inference acceleration for video large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 1910–1924. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.98.

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. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.679.

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

Viorica Pătrăucean, Lucas Smaira, Ankush Gupta, Adrià Recasens, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Joseph Heyward, Mateusz Malinowski, Yi Yang, Carl Doersch, Tatiana Matejovicova, Yury Sulsky, Antoine Miech, Alexandre Fréchette, Hanna Klimczak, Raphael Koster, Junlin Zhang, Stephanie Winkler, Yusuf Aytar, Simon Osindero, Dima Damen, Andrew Zisserman, and João Carreira. Perception test: A diagnostic benchmark for multimodal video models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Kele Shao, Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Holitom: Holistic token merging for fast video large language models. In *Advances in Neural Information Processing Systems*, 2025.

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, and Guiguang Ding. FastVID: Dynamic density pruning for fast video large language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4118.

Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, pp. 568–576, Cambridge, MA, USA, 2014. MIT Press.

Reuben Tan, Ximeng Sun, Ping Hu, Jui-hsien Wang, Hanieh Deilamsalehy, Bryan A. Plummer, Bryan Russell, and Kate Saenko. Koala: Key frame-conditioned long video-llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13581–13591, 2024.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Dycoke: Dynamic compression of tokens for fast video large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18992–19001, 2025.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4489–4497, 2015. doi: 10.1109/ICCV.2015.510.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, and Jie Tang. Lvbench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22958–22967, 2025.

Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. LongVLM: Efficient long video understanding via large language models. In *Computer Vision – ECCV 2024*, pp. 453–470, 2024. doi: 10.1007/978-3-031-73414-4_26.

Bo Wu, Shoubin Yu, Zhenfang Chen, Joshua B. Tenenbaum, and Chuang Gan. STAR: A benchmark for situated reasoning in real-world videos. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

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 (CVPR)*, pp. 9777–9786, 2021.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. VisionZip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19792–19802, 2025. doi: 10.1109/CVPR52734.2025.01843.

Hang Zhang, Xin Li, and Lidong Bing. Video-LLaMA: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 543–553. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-demo.49.

Jianrui Zhang, Mu Cai, and Yong Jae Lee. Vinoground: Scrutinizing lmms over dense temporal reasoning with short videos. *arXiv preprint arXiv:2410.02763*, 2024.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A. Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. Sparsevlm: Visual token sparsification for efficient vision-language model inference. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 74840–74857. PMLR, 2025a.

Zheyuan Zhang, Wanying Dou, Linkai Peng, Hongyi Pan, Ulas Bagci, and Boqing Gong. Videoads for fast-paced video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21812–21821, 2025b.

Zijia Zhao, Yuqi Huo, Tongtian Yue, Longteng Guo, Haoyu Lu, Bingning Wang, Weipeng Chen, and Jing Liu. Efficient motion-aware video mllm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24159–24168, 2025.

A Extended Related Work

Where temporal interaction enters. A recurring design choice in video representation is when information is exchanged across frames. Early frame-based systems process images separately and aggregate them afterward, whereas C3D and I3D introduce temporal coupling inside the visual encoder through three-dimensional convolutions (Karpathy et al., 2014; Simonyan & Zisserman, 2014; Donahue et al., 2017; Tran et al., 2015; Carreira & Zisserman, 2017). Video Transformers express related choices through spatial–temporal attention factorizations (Bertasius et al., 2021; Arnab et al., 2021). MotionBench explicitly studies pre-, post-, and through-encoder temporal fusion (Hong et al., 2025), while Efficient Motion-Aware Video MLLM uses compressed-video structure to combine RGB appearance and motion information within a compact video representation (Zhao et al., 2025). Our focus is not to argue for one universal fusion stage, but to ask what remains selectable when temporal interaction is introduced before or after a fixed-budget selection step.

Image-derived representations for long video understanding. Many VideoLLMs build video representations from image-derived features while differing in how temporal information is aggregated. Video-LLaMA uses a Video Q-Former over visual features, and LongVLM introduces hierarchical local–global aggregation for long videos (Zhang et al., 2023; Weng et al., 2024). Koala conditions learnable spatiotemporal queries on sparse key-frame tokens to adapt pretrained VideoLLMs to longer temporal horizons (Tan et al., 2024), while MA-LMM processes frames online and maintains compressed visual memory for long-term reference (He et al., 2024). Unified multimodal models such as Qwen2-VL and Qwen3-VL further share substantial image–video infrastructure while differing in how temporal content enters visual tokens (Wang et al., 2024; Bai et al., 2025). These works establish that video understanding can be built on image-derived representations; our setting instead asks how to preserve independently addressable image candidates under a fixed output-token budget before adding temporal context.

Visual-token pruning, merging, and compression. A complementary line of work reduces the visual sequence supplied to or processed inside the language model. FastV uses early-layer attention to prune visual tokens in later language-model layers (Chen et al., 2024), while SparseVLM performs text-guided, training-free visual-token sparsification with adaptive pruning and token recycling (Zhang et al., 2025a). For video, VisionZip, FastVID, VidCom^2, and FlashVID exploit token importance, temporal density, frame uniqueness, or spatiotemporal redundancy (Yang et al., 2025; Shen et al., 2025; Liu et al., 2025; Fan et al., 2026). DyCoke combines temporal token compression with dynamic KV-cache reduction during decoding (Tao et al., 2025), and HoliTom combines outer-LLM spatiotemporal merging with inner-LLM token merging (Shao et al., 2025). More recently, unified spatiotemporal token compression has treated frame- and token-level reduction jointly under very low retention ratios (Du et al., 2026). These methods primarily optimize how an existing visual stream is pruned, merged, or compressed, whereas our controlled comparisons additionally vary the representation available *before* selection.

Relation to our formulation. Our study therefore separates three factors that are often entangled in prior systems: *representation source*, *cross-frame token allocation*, and *temporal interaction*. We first preserve frame-specific image candidates, then coordinate a fixed output budget across the full video, and only afterward allow retained anchors to read temporal context. This decomposition is complementary to prior work on temporal fusion and token compression: rather than treating compression as a single post-encoding operation, it distinguishes which evidence remains individually selectable, which observations receive output slots, and which additional context can inform those selected anchors.

Table 3: **Additional seven-benchmark results.** Avg.(13) is the unweighted mean over all 13 benchmarks. Bold and underlined values denote the best and second-best results within each backbone.

Method	EGO	VINO text	VideoAds	LVBench	MVBench	STAR	IntentQA	Avg.(13) ↑
Qwen2-VL-7B								
Video/Conv3D	63.00	65.00	51.82	39.35	63.19	76.06	94.85	57.31
Image	70.00	**65.50**	58.64	41.94	60.97	75.07	93.44	59.26
Uniform	61.35	58.74	46.52	36.15	60.18	71.44	91.22	53.67
FlashVID	62.18	59.83	49.46	38.08	62.33	73.37	93.14	55.41
VidCom^2	63.05	60.48	49.12	38.36	63.22	74.11	94.42	56.06
FastVID	63.96	64.05	45.88	38.74	62.05	74.82	94.11	56.24
VisionZip	64.95	64.54	48.15	38.75	**64.12**	74.96	**95.05**	56.88
Ours	**71.33**	64.33	**59.55**	**42.33**	59.43	**76.18**	92.77	**59.43**
Qwen3-VL-8B								
Video/Conv3D	61.00	65.00	55.00	39.35	68.19	73.38	81.26	59.49
Image	73.00	**68.00**	61.82	43.87	66.25	75.49	82.67	62.58
Uniform	59.25	63.14	53.68	37.15	64.72	70.36	79.15	56.93
FastV	61.15	65.25	55.34	38.45	66.18	72.15	80.35	58.40
SparseVLM	61.45	65.88	55.72	39.12	66.55	72.48	80.75	58.81
DyCoke	61.85	66.35	56.15	39.45	66.92	72.85	81.12	59.21
FlashVID	62.05	67.92	56.40	39.95	67.42	72.84	81.08	59.62
VidCom^2	60.08	65.46	55.95	38.02	68.38	73.47	82.02	59.63
FastVID	62.94	64.55	57.78	41.02	**68.84**	72.21	81.46	59.86
VisionZip	61.95	65.56	54.51	39.96	68.25	73.04	81.56	59.45
Ours	**73.13**	67.92	**62.15**	**44.03**	66.71	**75.64**	**82.77**	**62.75**
Qwen3-VL-32B								
Video/Conv3D	68.00	73.50	61.36	42.90	**71.39**	75.35	87.82	64.20
Image	75.00	77.50	65.91	44.19	67.78	78.87	88.29	66.09
Uniform	63.15	68.45	57.12	40.85	65.34	72.15	84.12	59.94
FlashVID	55.06	59.92	51.42	38.65	60.08	54.81	80.37	54.32
VidCom^2	67.92	70.56	60.41	44.24	71.16	76.36	87.77	63.86
FastVID	66.95	71.44	61.88	44.14	70.75	75.36	87.64	63.57
VisionZip	65.94	72.08	60.39	42.63	69.78	75.80	87.18	63.29
Ours	**75.33**	**77.58**	**66.24**	**44.32**	68.15	**78.96**	**88.43**	**66.28**

B Additional Experimental Results

Table 3 reports the exact scores for the seven benchmarks summarized in Fig. 4, together with the macro-average over all 13 benchmarks. Each benchmark receives equal weight: Avg.(13) is computed directly from the 13 individual benchmark scores rather than by averaging the six- and seven-benchmark group means.

C Extended Ablations and Training Details

This section provides additional experiments that isolate the source of our gains, document the training of the temporal refiner, and examine robustness to correspondence structure, candidate-frame coverage, and visual-token budget.

C.1 Training-Fair Comparison

Most compression baselines in the main comparison are training-free, whereas our complete pathway additionally contains a learned temporal refiner. To isolate this factor, we disable temporal refinement while keeping the coordinated selector and visual-token budget unchanged. The resulting *selection-only* variant is fully training-free. Table 4 reports the matched comparison.

Training-free selection accounts for the primary gain. The selection-only variant reaches Avg.(13) scores of 59.37, 62.64, and 66.22 on Qwen2-VL-7B, Qwen3-VL-8B, and Qwen3-VL-

Table 4: **Training-fair comparison.** TF denotes training-free compression; *Selection* disables the learned refiner. Δ is measured against the strongest prior TF baseline.

Method	Train	Visual tokens ↓	Avg.(6) ↑	Avg.(13) ↑	Δ prior TF ↑
Qwen2-VL-7B					
Full Image	–	5,561	50.81	59.26	–
VisionZip	TF	1,532	48.15	56.88	0.00
Ours: selection	**TF**	1,536	50.97	59.37	**+2.49**
Ours: + refiner	Learned	1,536	**51.11**	**59.43**	**+2.55**
Qwen3-VL-8B					
Full Image	–	4,424	57.08	62.58	–
FastVID	TF	1,493	54.90	59.86	0.00
Ours: selection	**TF**	1,535	57.01	62.64	**+2.78**
Ours: + refiner	Learned	1,535	**57.23**	**62.75**	**+2.89**
Qwen3-VL-32B					
Full Image	–	4,424	60.28	66.09	–
VidCom^2	TF	1,504	58.62	63.86	0.00
Ours: selection	**TF**	1,535	60.33	66.22	**+2.36**
Ours: + refiner	Learned	1,535	**60.44**	**66.28**	**+2.42**

Table 5: **Component and refiner-objective ablations on Qwen3-VL-8B.** All variants use 1,535 visual tokens; dashes indicate that no learned refiner is used.

Dynamic Allocation	Cross-Frame Coordination	Answer Loss	KD Loss	VideoAds	Video-MME	LVBench	P.Test	Avg.(13)
✗	✗	–	–	55.45	63.85	39.80	67.85	60.12
✓	✗	–	–	60.30	65.40	42.15	68.30	61.74
✓	✓	–	–	**62.55**	66.25	42.26	68.65	62.64
✓	✓	✓	✗	61.85	66.80	42.85	68.80	62.67
✓	✓	✗	✓	61.95	67.55	43.60	69.10	62.71
✓	✓	✓	✓	62.10	**68.05**	**44.05**	**69.25**	**62.75**

32B, respectively. These exceed the strongest prior training-free compression baselines by **2.49**, **2.78**, and **2.36** percentage points. They also match or slightly exceed the corresponding Full Image references while using only approximately 1.5K visual tokens. Thus the main improvement does not depend on additional training.

Temporal refinement provides a targeted complement. Enabling the learned refiner increases Avg.(13) by **+0.06**, **+0.11**, and **+0.06** points across the three backbones. Larger gains appear on temporally demanding benchmarks, while several other benchmarks change only slightly or decrease. This pattern is consistent with the intended division of labor: coordinated selection determines which evidence occupies the compact sequence, while refinement adds temporal context after the output anchors are fixed.

C.2 Component and Refiner-Objective Ablations

We jointly examine dynamic allocation, cross-frame coordination, and the two objectives used to train the temporal refiner. All variants use Qwen3-VL-8B with the same 1,535-token budget. Table 5 summarizes these comparisons.

Allocation and coordination provide the primary gain. Dynamic allocation raises the 13-benchmark macro-average from 60.12 to 61.74, and cross-frame coordination further increases it to 62.64. The latter is already the fully training-free selection-only pathway.

Refiner training provides a smaller complement. Starting from 62.64 without a learned refiner, answer supervision alone reaches 62.67 and KD alone reaches 62.71. Combining both objectives gives 62.75. The aggregate improvement is modest, while Video-MME, LVBench, and Perception Test benefit more strongly; VideoAds decreases slightly.

Table 6: **System-level ablations on Qwen2-VL-7B.** We vary the selection strategy, number of candidate frames, and visual-token budget. Δ is measured relative to the default configuration.

Dimension	Setting	Candidate frames	Token budget	Actual tokens ↓	Avg.(13) ↑	Δ
Selection	Independent	32	1,536	1,534.9	58.50	−0.93
	Shuffled correspondence	32	1,536	1,534.9	57.53	−1.90
	Coupled (Ours)	**32**	**1,536**	**1,534.9**	**59.43**	**0.00**
Candidate frames	16 frames	16	1,536	1,517.7	57.97	−1.46
	24 frames	24	1,536	1,530.1	58.85	−0.58
	32 frames (Ours)	**32**	**1,536**	**1,534.9**	**59.43**	**0.00**
	768 tokens	32	768	767.8	56.63	−2.80
	1,536 tokens (Ours)	**32**	**1,536**	**1,534.9**	**59.43**	**0.00**
	3,072 tokens	32	3,072	3,035.3	61.11	+1.68

C.3 Refiner Training Details

The temporal refiner is intentionally lightweight. It consists of a bottleneck cross-frame attention block and a gated feed-forward network, introducing approximately 18M trainable parameters. The vision encoder, language model, and coordinated selector remain frozen throughout training. We train only the refiner for two epochs on 8 A100 GPUs for 12 hours (approximately 96 GPU-hours) using AdamW, a learning rate of 2×10^{-5}, and a batch size of 64. For knowledge distillation, we use temperature $\tau = 2.0$ and weight $\beta = 1.0$.

Training data and decontamination. We construct a 65K-example training subset from the Video-ChatGPT training split. To reduce train–test overlap, we filter this subset against the evaluation prompts and available metadata of all 13 downstream benchmarks using both exact n-gram matching and a semantic-similarity filter, removing every sample flagged by either criterion.

C.4 System-Level Ablations

We next examine three system-level choices on Qwen2-VL-7B: cross-frame correspondence, candidate-frame coverage, and the visual-token budget. All comparisons use the same 13-benchmark macro-average. Table 6 summarizes these system-level ablations.

Correct cross-frame correspondence matters. Ignoring cross-frame correspondence reduces $\mathrm{Avg.}(13)$ from 59.43 to 58.50. Randomly shuffling the correspondences decreases it further to 57.53, suggesting that incorrect pair structure can be actively harmful rather than merely uninformative.

Broader candidate coverage improves performance. Increasing the candidate pool from 16 to 24 frames raises $\mathrm{Avg.}(13)$ from 57.97 to 58.85, and increasing it from 24 to 32 frames gives a further 0.58-point improvement. The monotonic increase over the evaluated range is consistent with broader temporal coverage.

The output budget controls the accuracy–compression trade-off. Reducing the budget from 1,536 to 768 tokens lowers $\mathrm{Avg.}(13)$ by 2.80 points, whereas increasing it to 3,072 tokens raises the score by 1.68 points. We use approximately 1.5K tokens in the main experiments because it matches the operating regime of prior compression baselines while retaining a substantial reduction relative to the Full Image pathway.

C.5 Additional Efficiency Breakdown

The latency reduction primarily occurs before autoregressive decoding: TTFT falls from 849.1 ms for Image to 365.8 ms for Ours, while post-first-token generation changes from 62.1 ms to 56.0 ms. The selection overhead is included in TTFT and is offset by the shorter visual prefill.

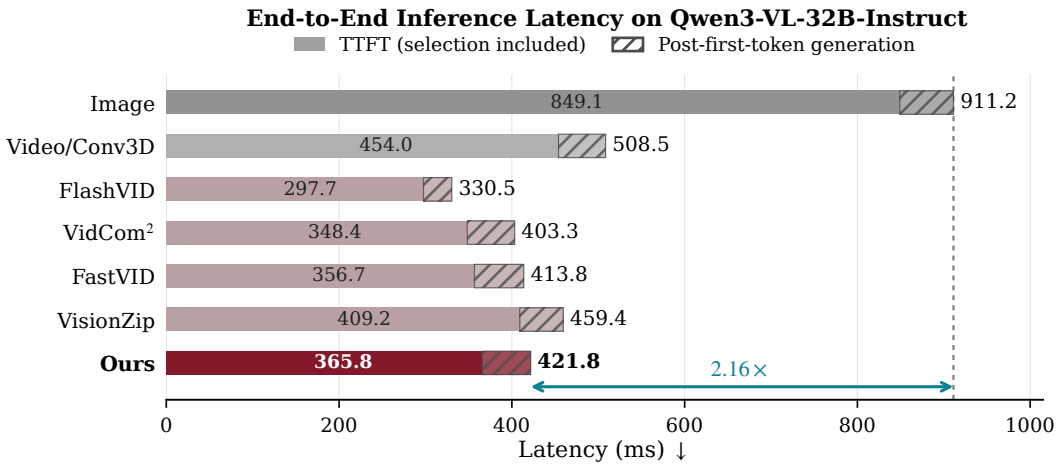
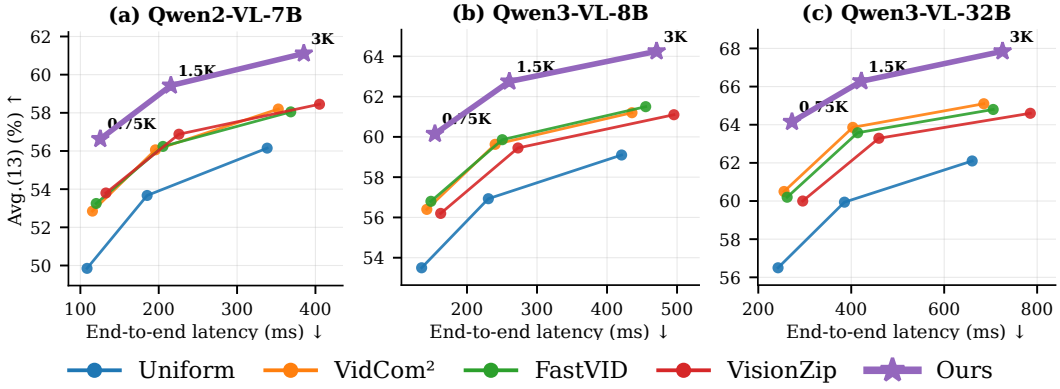


Figure 5: **Inference latency on Qwen3-VL-32B-Instruct.** Selection is included in TTFT. Ours reduces end-to-end latency from 911.2 ms to 421.8 ms relative to Image.



Points on each curve correspond to 0.75K, 1.5K, and 3K token budgets; labels are shown on Ours.

Figure 6: **Multi-budget accuracy–latency trade-off.** We sweep 0.75K, 1.5K, and 3K visual-token budgets on three backbones. Each curve connects the three operating points of one method.

C.6 Multi-Budget Accuracy–Latency Trade-off

Fig. 6 summarizes the accuracy–latency trade-off across the three evaluated budgets and backbones. The main experiments use an approximately 1.5K visual-token budget to match prior compression methods. We further sweep the output budget over $\{768, 1536, 3072\}$ tokens on Qwen2-VL-7B, Qwen3-VL-8B, and Qwen3-VL-32B. For each budget, we measure the actual visual-token count, end-to-end inference latency, and the macro-average over all 13 benchmarks.

The advantage persists across budgets. At 768 tokens, our method obtains Avg.(13) scores of 56.63, 60.15, and 64.15 on Qwen2-VL-7B, Qwen3-VL-8B, and Qwen3-VL-32B, respectively. These exceed the strongest prior compression baseline at the same budget by 2.83, 3.35, and 3.65 points. At the 1.5K budget, the corresponding gains are 2.55, 2.89, and 2.42 points, and at 3K they remain 2.66, 2.75, and 2.75 points. The relative advantage therefore persists over a $4\times$ range of output budgets.

A smaller coordinated representation can outperform larger baselines. The 1.5K operating point of our method also outperforms every prior method evaluated at the 3K budget. Ours reaches 59.43, 62.75, and 66.28, compared with the strongest 3K baselines at 58.45, 61.50, and 65.10, respectively. The corresponding end-to-end latencies are 215.5 vs. 405.2 ms, 260.5 vs. 455.2 ms, and 421.8 vs. 685.2 ms.

Returns diminish beyond 1.5K tokens. Increasing our budget from 768 to approximately 1.5K tokens improves Avg.(13) by 2.80, 2.60, and 2.13 points across the three backbones. Doubling

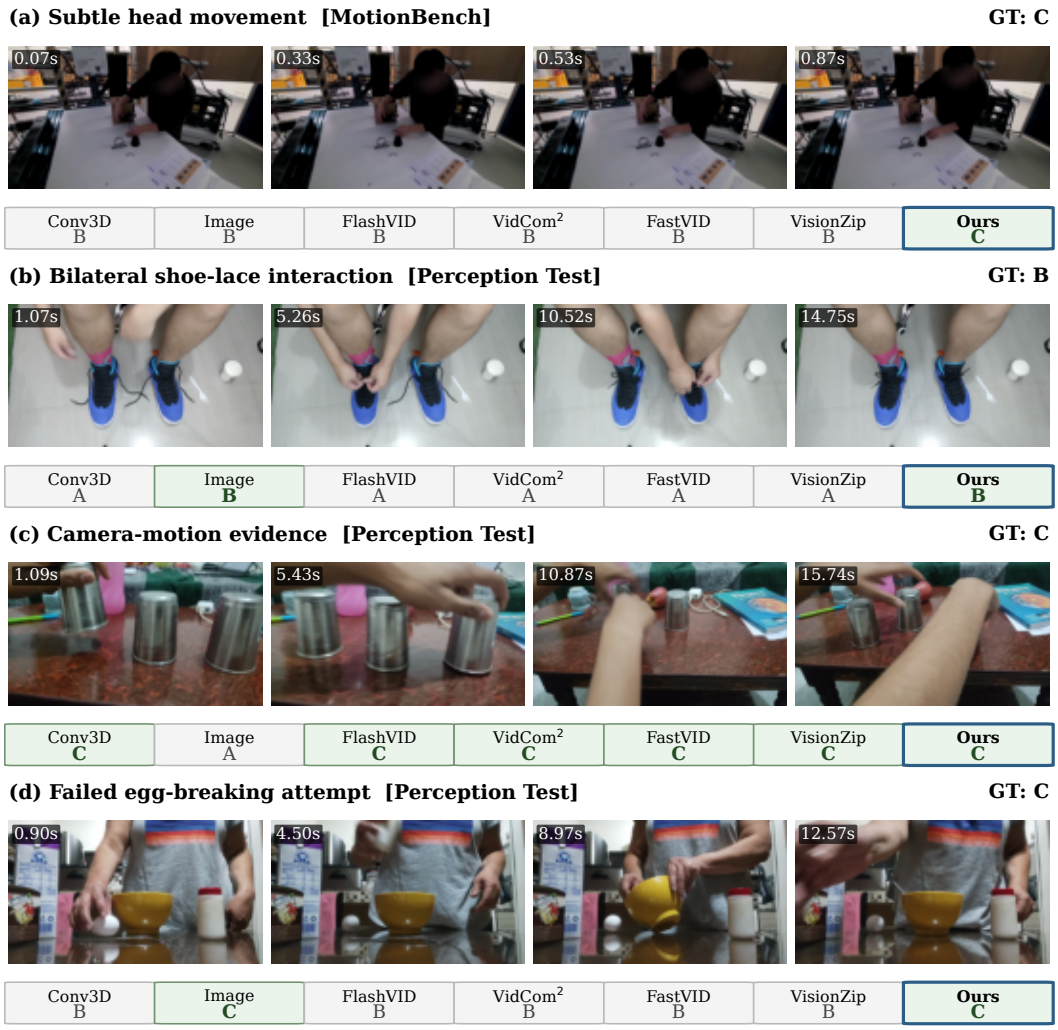


Figure 7: **Representative qualitative successes (I).** Four examples illustrate different forms of evidence preservation. In (a), the answer depends on a subtle downward head movement. In (b) and (d), the decision requires preserving interactions among multiple objects across time. In (c), camera motion is visible from the relative displacement of the cup group.

the budget again to approximately 3K yields smaller gains of 1.68, 1.50, and 1.57 points while increasing latency. This supports the 1.5K setting as a practical accuracy–efficiency operating point.

D Qualitative Analysis

We complement the aggregate benchmark results with qualitative examples from Qwen3-VL-8B. The examples include eight representative successes and one representative failure case. They are intended to illustrate the kinds of visual evidence that matter under a compact budget rather than to estimate how frequently a particular behavior occurs.

Visualization protocol. For each example, we show four representative frames from the 32-frame input together with their timestamps and the parsed multiple-choice prediction of every evaluated method. Green cells indicate correct predictions, and our method is additionally outlined for identification. The displayed frames are selected only for visualization; they should not be interpreted as a reconstruction of any method's internal token mask or retained-token support. We include cases where other compression methods also succeed, rather than restricting the visualization to exclusive wins.

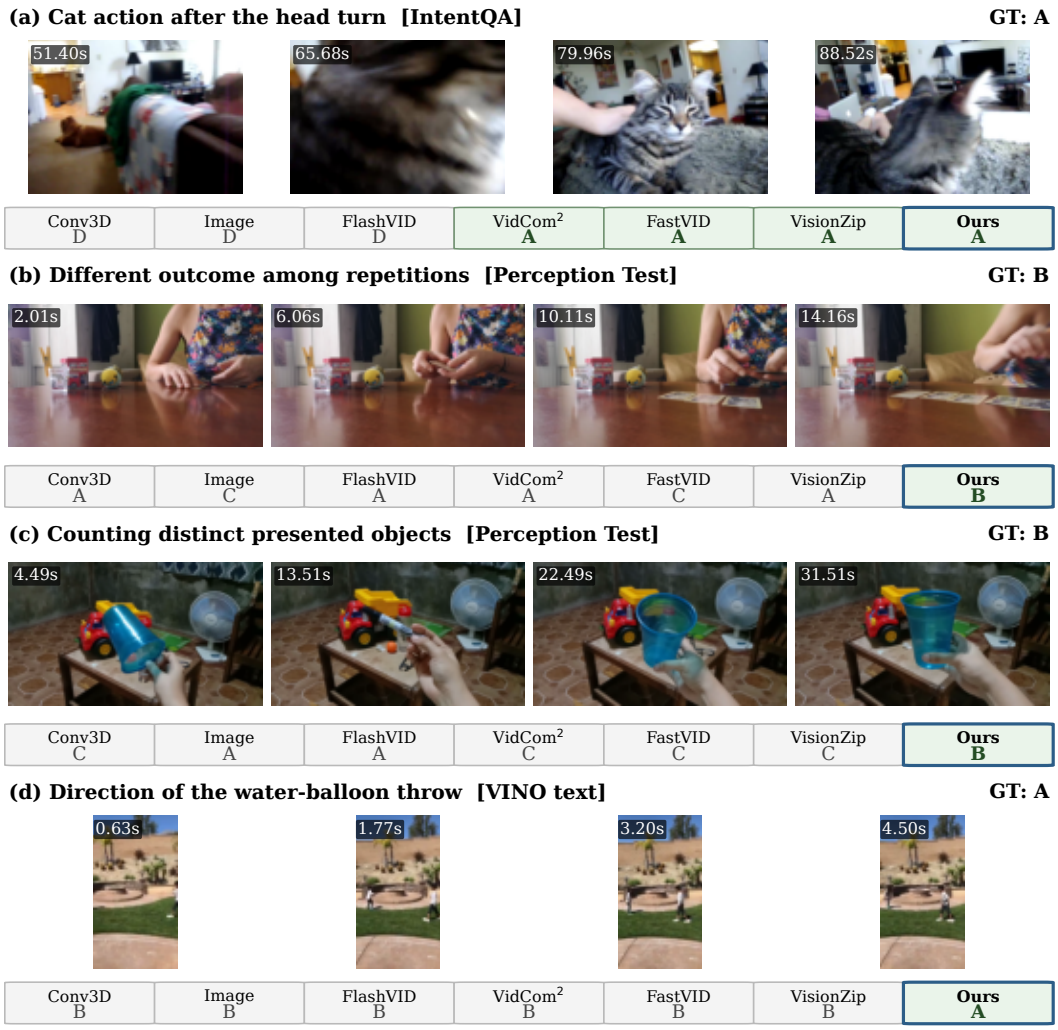


Figure 8: **Representative qualitative successes (II).** The examples cover late evidence, repeated outcomes, object counting, and directional temporal reasoning. In (a), the decisive event occurs near the end of the clip when the cat approaches and licks the hand. In (b), the answer requires comparing repeated trials and identifying the one with a different outcome. In (c), the model must avoid double-counting repeated views of objects. In (d), temporal order and spatial roles determine the throw direction. Ours predicts the ground-truth option in all four cases.

What the successful cases illustrate. The examples suggest several recurring evidence patterns. Fine-grained local change matters when the answer depends on a small motion rather than a dominant object or pose (Fig. 7a). Relational evidence matters when two observations must be interpreted jointly, as in bilateral shoe-lace interaction, the failed egg-breaking attempt, and the water-balloon direction (Fig. 7b,d; Fig. 8d). Long-range coverage is useful when the decisive evidence occurs late or when outcomes must be compared across repeated actions (Fig. 8a,b). Finally, repeated views should not be confused with distinct evidence, as illustrated by the object-counting example in Fig. 8c. These examples are consistent with the intended role of coordinated selection: allocate the limited output budget to complementary source evidence rather than to individually salient but redundant observations.

Failure analysis. This case stresses long-range subject binding rather than local appearance recognition. The model must associate a late action with the correct parrot among multiple visually similar birds, foliage, and cage structure. Here, only the full Image pathway retains enough evidence to recover the correct option, while all compressed pathways shown in Fig. 9 converge to the same distractor. This result indicates a limitation of compact visual selection when the answer depends on



Figure 9: **Representative failure case: long-range subject binding.** The question asks what the farther parrot does after spreading its wing near the end of the clip. The full Image pathway predicts the correct answer (**E**: continuing to clean itself), whereas Ours, Video/Conv3D, and all compact baselines shown here predict **C**. The example suggests that aggressive visual compression can make it difficult to preserve the identity of the queried subject over a long temporal span when several visually similar instances are present.

maintaining instance identity across a long temporal span. It also motivates identity-aware retrieval and longer-range temporal state tracking as complementary directions for future refinement.

E Coordinated Selection: Objective and Analysis

This appendix gives the details behind the determinantal diversity term, the adjacent-frame pair reward, and the feasibility of the greedy selection procedure in Section 3.2. The analysis concerns the selection operators themselves and does not assume that the resulting subset is globally optimal.

Determinantal marginal gain. Consider one frame and suppress the frame index. Let $L \succeq 0$ be its DPP kernel, let U be the currently selected candidates from that frame, and let $i \notin U$. Whenever L_{UU} is positive definite, the determinant of the enlarged principal block satisfies

$$\det L_{U\cup\{i\},U\cup\{i\}} = \det L_{UU}\left(L_{ii} - L_{iU}L_{UU}^{-1}L_{Ui}\right). \tag{8}$$

Consequently,

$$\begin{aligned}\Delta_i^{\mathrm{DPP}}(U) &= \log\det L_{U\cup\{i\},U\cup\{i\}} - \log\det L_{UU} \\ &= \log\left(L_{ii} - L_{iU}L_{UU}^{-1}L_{Ui}\right). \end{aligned} \tag{9}$$

Proof. After ordering the indices with those in U first and i last, the enlarged principal block is

$$L_{U\cup\{i\},U\cup\{i\}} = \begin{bmatrix} L_{UU} & L_{Ui} \\ L_{iU} & L_{ii} \end{bmatrix}.$$

Since L_{UU} is invertible, the block-determinant identity gives

$$\det\begin{bmatrix} L_{UU} & L_{Ui} \\ L_{iU} & L_{ii} \end{bmatrix} = \det L_{UU}\det\left(L_{ii} - L_{iU}L_{UU}^{-1}L_{Ui}\right).$$

The Schur complement is scalar, yielding Eq. 8. Taking logarithms and subtracting $\log\det L_{UU}$ proves Eq. 9. □

Because $L \succeq 0$, the Schur complement is nonnegative whenever the inverse exists. If the enlarged principal block is positive definite, it is strictly positive. The expression therefore measures the additional volume contributed by candidate i after accounting for the directions already represented by U. A candidate nearly contained in the span represented by U receives a small marginal gain even when its individual quality is high.

Empty and degenerate subsets. For $U = \varnothing$, we use the standard convention $\det L_{\varnothing,\varnothing} = 1$. The first marginal is therefore

$$\Delta_i^{\mathrm{DPP}}(\varnothing) = \log L_{ii}, \tag{10}$$

whenever $L_{ii} > 0$.

Eq. 9 is written in inverse form only when L_{UU} is nonsingular. If a kernel contains degenerate directions, the same analysis can be applied to the strictly positive definite kernel

$$L^{(\varepsilon)} = L + \varepsilon I, \qquad \varepsilon > 0, \tag{11}$$

for which every principal block is invertible. The determinant and Schur-complement identities then hold exactly with $L^{(\varepsilon)}$ in place of L. Alternatively, when the current principal determinant is positive but adding i makes the enlarged determinant zero, its unregularized log-determinant marginal is $-\infty$, reflecting that the candidate adds no new determinantal volume. None of the arguments below depends on a particular choice of numerical regularization.

Adjacent-frame complementarity. The DPP term discourages redundant directions within a frame, but visual similarity across neighboring frames does not necessarily imply redundancy. To formalize the complementary pair term, let E be the sparse correspondence graph and let $w_{ij} \geq 0$ be the reward associated with edge $\{i, j\} \in E$. Define

$$G(S) = \sum_{\substack{\{i,j\}\in E \\ i,j\in S}} w_{ij}. \tag{12}$$

For an unselected candidate $i \notin S$, its marginal contribution is

$$\begin{aligned} G(S \cup \{i\}) - G(S) &= \sum_{j\in S:\{i,j\}\in E} w_{ij} \\ &= \Delta_i^{\mathrm{pair}}(S), \end{aligned} \tag{13}$$

which recovers the pair reward used in the main text.

Because all w_{ij} are nonnegative, this marginal exhibits complementarity. If $S \subseteq S'$ and $i \notin S'$, then

$$\Delta_i^{\mathrm{pair}}(S) \leq \Delta_i^{\mathrm{pair}}(S'). \tag{14}$$

Thus selecting one endpoint can increase the value of selecting its corresponding endpoint later.

Proof. The set of neighbors of i contained in S is a subset of those contained in S'. Therefore

$$\sum_{j\in S:\{i,j\}\in E} w_{ij} \leq \sum_{j\in S':\{i,j\}\in E} w_{ij},$$

because every added term is nonnegative. □

Eq. 14 is the opposite direction from the diminishing-returns condition of a submodular set function. In particular, G is supermodular. This also shows why adding the pair reward to a diversity criterion does not, in general, preserve submodularity.

A two-candidate example makes the point explicit. Let i and j be joined by a single edge with $w_{ij} > 0$, and take a DPP kernel $L = I$, for which the log-determinant contribution is zero for every selected subset. Then

$$G(\varnothing) = G(\{i\}) = G(\{j\}) = 0, \qquad G(\{i, j\}) = w_{ij}.$$

The marginal value of adding i is therefore zero when starting from $\varnothing$ but w_{ij} after j has been selected:

$$G(\{i\}) - G(\varnothing) < G(\{i, j\}) - G(\{j\}).$$

Hence the combined selection score need not satisfy a standard submodular diminishing-returns property, and we make no corresponding approximation guarantee for the greedy procedure.

Quota-constrained feasible moves. Let I_t denote the candidates belonging to frame t, and let the prescribed integer quotas satisfy

$$0 \le b_t \le N_t, \qquad \sum_{t=1}^{T} b_t = B. \tag{15}$$

At a current subset S, a singleton or pair move U is feasible only if

$$U \cap S = \varnothing, \qquad |S \cup U| \le B, \qquad |(S \cup U) \cap I_t| \le b_t \quad \text{for every } t. \tag{16}$$

The pair moves allow the selector to retain correspondence endpoints jointly, while singleton moves ensure that the quota constraints can always be completed.

Termination at exactly B anchors. Assume Eq. 15 and that $\mathrm{FeasibleMoves}$ includes every unselected singleton satisfying Eq. 16. Starting from $S = \varnothing$, the loop in Algorithm 1 cannot terminate or become stuck with $|S| < B$ and reaches exactly B distinct selected indices.

Proof. Suppose $|S| < B$. Since

$$\sum_{t=1}^{T} b_t = B,$$

it cannot be the case that $|S \cap I_t| = b_t$ for every frame; otherwise summing over frames would give $|S| = B$. Hence there exists at least one frame t such that

$$|S \cap I_t| < b_t.$$

Because $b_t \le N_t = |I_t|$, this frame contains at least one unselected candidate $i \in I_t \setminus S$. The singleton move $\{i\}$ respects the frame quota. It also respects the global budget because $|S| < B$. Therefore at least one feasible move exists whenever the budget is not yet filled.

Each accepted move contains only previously unselected candidates, so $|S|$ strictly increases. Eq. 16 prevents a move from exceeding the global budget or any frame quota. In particular, when only one global slot remains, no two-element pair is feasible and a singleton remains available by the preceding argument. Since the budget is finite, the procedure therefore reaches $|S| = B$ after finitely many moves. All selected indices are distinct by construction. □

This result establishes feasibility only. The move chosen by $\mathrm{GreedyMove}$ depends on the configured relevance, diversity, and correspondence scores, and the complementarity term above prevents a general claim of global optimality. The role of the procedure is instead to construct exactly B source-aligned anchors while allowing the value of a candidate to depend on what has already been retained.

F Proof of the Conditional Prediction-Error Decomposition

We prove Proposition 1 and the monotonicity statement for additional context. Let $\mathcal{Y}$ denote the finite output vocabulary. For $y \in \mathcal{Y}$, write

$$P_y = p_F(y), \qquad Q_y = p_\theta(y \mid \mathcal{O}), \qquad \bar{P}_y = \bar{p}_{\mathcal{O}}(y) = \mathbb{E}[P_y \mid \mathcal{O}].$$

Because conditional expectation is taken coordinatewise, $\bar{p}_{\mathcal{O}}$ is itself a probability distribution: $\bar{P}_y \ge 0$ and $\sum_{y \in \mathcal{Y}} \bar{P}_y = 1$ almost surely.

Proof of Proposition 1. Using the standard convention for zero-probability terms, condition on $\mathcal{O}$. For every coordinate with $\bar{P}_y = 0$, nonnegativity of P_y and $\mathbb{E}[P_y \mid \mathcal{O}] = 0$ imply that $P_y = 0$ almost

surely conditional on $\mathcal{O}$, so that coordinate contributes zero. On the remaining coordinates,

$$\begin{aligned}
&\mathbb{E}[D_{\mathrm{KL}}(p_F\|p_\theta) - D_{\mathrm{KL}}(p_F\|\bar{p}_\mathcal{O}) \,|\, \mathcal{O}] \\
&\quad= \mathbb{E}\left[\sum_{y\in\mathcal{Y}} P_y \log\frac{\bar{P}_y}{Q_y}\,\middle|\,\mathcal{O}\right] \\
&\quad= \sum_{y\in\mathcal{Y}} \mathbb{E}[P_y \mid \mathcal{O}] \log\frac{\bar{P}_y}{Q_y} \\
&\quad= \sum_{y\in\mathcal{Y}} \bar{P}_y \log\frac{\bar{P}_y}{Q_y} = D_{\mathrm{KL}}(\bar{p}_\mathcal{O}\|p_\theta). \qquad (17)
\end{aligned}$$

Taking expectation over $\mathcal{O}$ gives

$$\mathbb{E}\, D_{\mathrm{KL}}(p_F\|p_\theta) = \mathbb{E}\, D_{\mathrm{KL}}(p_F\|\bar{p}_\mathcal{O}) + \mathbb{E}\, D_{\mathrm{KL}}(\bar{p}_\mathcal{O}\|p_\theta),$$

which is Eq. 2.

Since KL divergence is nonnegative,

$$\mathbb{E}\, D_{\mathrm{KL}}(p_F\|p_\theta) \geq \mathbb{E}\, D_{\mathrm{KL}}(p_F\|\bar{p}_\mathcal{O}) = \mathcal{I}(\mathcal{O}).$$

Equality is attained by the predictor $p_\theta(\cdot \mid \mathcal{O}) = \bar{p}_\mathcal{O}$. Hence

$$\mathcal{I}(\mathcal{O}) = \inf_{p_\theta\ \mathcal{O}\text{-measurable}} \mathbb{E}\, D_{\mathrm{KL}}(p_F\|p_\theta).$$

□

Additional context can only reduce the information floor. Let $\mathcal{O}^+ = (\mathcal{O}, \mathcal{C})$ be an enriched record and define

$$\bar{p}_{\mathcal{O}^+} = \mathbb{E}[p_F \mid \mathcal{O}^+].$$

Because $\mathcal{O}$ is contained in $\mathcal{O}^+$, $\bar{p}_\mathcal{O}$ is also measurable with respect to $\mathcal{O}^+$. Applying Proposition 1 with $\mathcal{O}^+$ as the observed record and $\bar{p}_\mathcal{O}$ as the predictor gives

$$\begin{aligned}
\mathcal{I}(\mathcal{O}) &= \mathbb{E}\, D_{\mathrm{KL}}(p_F\|\bar{p}_\mathcal{O}) \\
&= \mathbb{E}\, D_{\mathrm{KL}}(p_F\|\bar{p}_{\mathcal{O}^+}) + \mathbb{E}\, D_{\mathrm{KL}}(\bar{p}_{\mathcal{O}^+}\|\bar{p}_\mathcal{O}) \\
&= \mathcal{I}(\mathcal{O}^+) + \mathbb{E}\, D_{\mathrm{KL}}(\bar{p}_{\mathcal{O}^+}\|\bar{p}_\mathcal{O}). \qquad (18)
\end{aligned}$$

Therefore

$$\mathcal{I}(\mathcal{O}^+) \leq \mathcal{I}(\mathcal{O}),$$

with equality if and only if $\bar{p}_{\mathcal{O}^+} = \bar{p}_\mathcal{O}$ almost surely on the relevant support. Thus additional context lowers the irreducible teacher-prediction gap exactly to the extent that it changes the conditional teacher distribution.

G Proof of the Selection–Interaction Identity

We prove Proposition 2 and the worst-case residual in Eq. 5. For any index set R, let P_R be its row-selection matrix, and write

$$\mathbf{Z}_R = P_R\mathbf{Z}, \qquad A_{RQ} = P_R A P_Q^\top.$$

Proof of Proposition 2. Because S and $\bar{S}$ partition the M candidate indices,

$$P_S^\top P_S + P_{\bar{S}}^\top P_{\bar{S}} = I_M,$$

and therefore

$$\mathbf{Z} = P_S^\top \mathbf{Z}_S + P_{\bar{S}}^\top \mathbf{Z}_{\bar{S}}.$$

Expanding the interact-then-select operator gives

$$\begin{aligned} P_S(I+A)\mathbf{Z} &= \mathbf{Z}_S + P_S A\mathbf{Z} \\ &= \mathbf{Z}_S + P_S A P_S^\top \mathbf{Z}_S + P_S A P_{\bar{S}}^\top \mathbf{Z}_{\bar{S}} \\ &= (I_B + A_{SS})\mathbf{Z}_S + A_{S\bar{S}}\mathbf{Z}_{\bar{S}}. \end{aligned} \tag{19}$$

Subtracting the select-then-interact term yields

$$P_S(I+A)\mathbf{Z} - (I_B + A_{SS})\mathbf{Z}_S = A_{S\bar{S}}\mathbf{Z}_{\bar{S}},$$

which proves Eq. 4.

If $A_{S\bar{S}} = 0$, the two orders clearly agree for every $\mathbf{Z}$. Conversely, suppose they agree for every $\mathbf{Z}$. Then

$$A_{S\bar{S}}\mathbf{Z}_{\bar{S}} = 0$$

for every possible $\mathbf{Z}_{\bar{S}}$. Choosing a matrix whose one feature column is an arbitrary vector $v \in \mathbb{R}^{|\bar{S}|}$ and whose remaining columns are zero gives $A_{S\bar{S}}v = 0$ for every v. Hence $A_{S\bar{S}} = 0$. □

Partial context retrieval. Let $C \subseteq \bar{S}$ denote the retrieved context indices and let

$$U = \bar{S} \setminus C$$

be the remaining unobserved indices. The full selected output after applying the fixed interaction operator can be written as

$$P_S(I+A)\mathbf{Z} = (I_B + A_{SS})\mathbf{Z}_S + A_{SC}\mathbf{Z}_C + A_{SU}\mathbf{Z}_U. \tag{20}$$

If the selected representation has access to S and C but not U, then the term attributable to the unavailable indices is exactly

$$E_U = A_{SU}\mathbf{Z}_U.$$

We now compute its worst-case magnitude under the constraint $\|\mathbf{Z}_U\|_F \le R$. By submultiplicativity of the Frobenius norm,

$$\|A_{SU}\mathbf{Z}_U\|_F \le \|A_{SU}\|_2 \|\mathbf{Z}_U\|_F \le R\|A_{SU}\|_2.$$

Thus

$$\sup_{\|\mathbf{Z}_U\|_F \le R} \|A_{SU}\mathbf{Z}_U\|_F \le R\|A_{SU}\|_2.$$

To show that the bound is attained, let $v \in \mathbb{R}^{|U|}$ be a unit right singular vector of A_{SU} associated with its largest singular value $\sigma_{\max}(A_{SU}) = \|A_{SU}\|_2$, and let $e \in \mathbb{R}^D$ be any unit vector. Set

$$\mathbf{Z}_U = Rve^\top.$$

Then

$$\|\mathbf{Z}_U\|_F = R\|v\|_2\|e\|_2 = R,$$

while

$$\begin{aligned} \|A_{SU}\mathbf{Z}_U\|_F &= R\|A_{SU}ve^\top\|_F && (21) \\ &= R\|A_{SU}v\|_2\|e\|_2 && (22) \\ &= R\|A_{SU}\|_2. && (23) \end{aligned}$$

Hence

$$\sup_{\|\mathbf{Z}_U\|_F \le R} \|A_{SU}\mathbf{Z}_U\|_F = R\|A_{SU}\|_2. \tag{24}$$

This result is conditional on the fixed operator A, the fixed selection set S, and the retrieved context set C. It characterizes the largest interaction contribution hidden behind the selection boundary under a norm-bounded unobserved feature matrix; it is not a bound on downstream answer accuracy or on an adaptive nonlinear video encoder.

H TEMPORAL REFINEMENT AND LEARNING OBJECTIVE

This appendix gives additional details for the temporal refinement module in Section 3.3. The purpose of refinement is to let each selected source anchor read temporally relevant evidence without increasing the number of visual tokens forwarded to the language model. Throughout refinement, the selected index set S and its source metadata remain fixed.

Context retrieval. For each selected anchor $i \in S$, let $C(i)$ denote a set of retrieved tokens from neighboring frames. The context set is not restricted to the selected anchors:

$$C(i) \subseteq \{1, \dots, M\} \setminus \{i\}, \qquad C(i) \not\subseteq S \text{ in general.} \tag{25}$$

Thus selection determines which tokens occupy the output sequence, whereas context retrieval determines which additional tokens may be read while constructing those outputs. Tokens used only as context do not become new language-model input tokens.

For a selected anchor i and context token $j \in C(i)$, the refinement module uses their visual features together with relative temporal and spatial information to produce normalized attention weights α_{ij}. We require

$$\alpha_{ij} \geq 0, \qquad \sum_{j \in C(i)} \alpha_{ij} = 1. \tag{26}$$

The aggregated context for anchor i is

$$\mathbf{c}_i = \sum_{j \in C(i)} \alpha_{ij} W_V \mathbf{z}_j, \tag{27}$$

where W_V is the value projection used by the bottleneck attention module.

Residual refinement. The selected anchor is updated through the gated residual branch G_θ:

$$\widehat{\mathbf{z}}_i = \mathbf{z}_i + G_\theta(\mathbf{z}_i, \mathbf{c}_i), \qquad i \in S. \tag{28}$$

This is the same update as Eq. 7, with Eq. 27 making the retrieved context explicit.

The residual branch is gated and its output is bounded by construction. Writing its bound abstractly as ρ, the update satisfies

$$\left\|\widehat{\mathbf{z}}_i - \mathbf{z}_i\right\| = \left\|G_\theta(\mathbf{z}_i, \mathbf{c}_i)\right\| \leq \rho. \tag{29}$$

The particular parameterization of the gate does not affect the structural property used here: refinement modifies the content associated with an existing anchor rather than creating additional output anchors.

Zero-expansion interface. Let

$$\widehat{\mathbf{Z}}_S = \mathrm{Concat}\left(\widehat{\mathbf{z}}_i : i \in S\right).$$

Since each of the B selected anchors produces exactly one refined output,

$$\widehat{\mathbf{Z}}_S \in \mathbb{R}^{B \times D}. \tag{30}$$

The source positions and intermediate feature streams remain

$$\widehat{\mathbf{P}}_S = \mathbf{P}_S, \qquad \widehat{\mathbf{D}}_S = \mathbf{D}_S. \tag{31}$$

Hence refinement is a $B \to B$ operation at the visual output interface, even though its computation may read tokens outside S.

This distinction is important. The dense image pool can serve as temporary context memory without becoming the sequence supplied to the language model. Zero expansion therefore refers to output-token count, not to the amount of computation or memory used while producing the refined anchors.

Relation to the interaction boundary. Proposition 2 shows that selecting before interaction omits cross-boundary terms of the form $A_{S\bar{S}}\mathbf{Z}_{\bar{S}}$ in the fixed linear diagnostic model. Retrieving context $C \subseteq \bar{S}$ restores access to the corresponding component

$$A_{SC}\mathbf{Z}_C,$$

while the remaining unavailable component is associated with $U = \bar{S} \setminus C$.

The learned refiner is not assumed to implement this linear operator exactly. The proposition instead motivates its access pattern: selected anchors should be allowed to read relevant unselected context when useful. In this sense, selection fixes the output support, while refinement relaxes the interaction boundary induced by that support.

Relation to the information decomposition. The same distinction can be expressed through Proposition 1. Before refinement, the selected record is

$$\mathcal{O} = (q, S, \mathbf{Z}_S, \mathbf{P}_S, \mathbf{D}_S).$$

After retrieving context $\mathcal{C}$, the available record becomes

$$\mathcal{O}^+ = (\mathcal{O}, \mathcal{C}).$$

As shown in Eq. 3,

$$\mathcal{I}(\mathcal{O}) = \mathcal{I}(\mathcal{O}^+) + \mathbb{E}D_{\mathrm{KL}}\left(\bar{p}_{\mathcal{O}^+} \| \bar{p}_{\mathcal{O}}\right). \tag{32}$$

Thus context retrieval can reduce the information floor by making additional evidence observable. The residual branch then determines how effectively the available record is converted into a prediction. More expressive refinement can improve the latter operation, but it cannot recover evidence that neither selection nor context retrieval exposes.

Learning objective. Only the temporal refiner parameters θ are optimized. The image encoder, language model, and selector remain frozen. Let $p_\theta(\cdot \mid \mathcal{V}, q)$ denote the selected-and-refined pathway's first-answer-token distribution, and let $p_F(\cdot \mid \mathcal{V}, q)$ denote the frozen Full Image first-answer-token distribution.

Training combines answer supervision and teacher distillation:

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{ans}}(\theta) + \beta\, \mathcal{L}_{\mathrm{KD}}(\theta), \tag{33}$$

where β controls the relative contribution of the distillation term. At temperature τ, write $p_F^{(\tau)}$ and $p_\theta^{(\tau)}$ for the corresponding softened distributions. The distillation term is

$$\mathcal{L}_{\mathrm{KD}}(\theta) = D_{\mathrm{KL}}\left(p_F^{(\tau)} \| p_\theta^{(\tau)}\right). \tag{34}$$

The Full Image pathway is used as a teacher only during training; it is not required at inference.

Initialization and the training-free limit. The residual branch is initialized so that its initial output is zero:

$$G_{\theta_0}(\mathbf{z}_i, \mathbf{c}_i) = 0. \tag{35}$$

Consequently,

$$\widehat{\mathbf{z}}_i = \mathbf{z}_i \qquad \text{at initialization.} \tag{36}$$

Optimization therefore begins from the unrefined selected-token pathway rather than from an arbitrary modification of the retained image features.

Disabling the residual branch gives the exact training-free limit

$$\widehat{\mathbf{Z}}_S = \mathbf{Z}_S. \tag{37}$$

The coordinated selector and the fixed-budget source interface remain unchanged. The learned component is therefore confined to temporal refinement: representation and selection do not require additional training.

Algorithm 1: **Image-first encoding with a fixed output budget.**

Input: video $\mathcal{V}$, question q, budget B, frozen image encoder, frozen allocator, optional refiner G_θ.

Encode every frame independently; collect $(\mathbf{Z}, \mathbf{P}, \mathbf{D})$.
Inherit quotas b_t from the frozen allocator, with $0 \le b_t \le N_t$ and $\sum_t b_t = B$.
Build question-weighted DPP kernels $\{L_t\}$ and pair graph (E, w).
$S \leftarrow \emptyset$.
while $|S| < B$ **do**
  $\mathcal{U} \leftarrow \mathrm{FeasibleMoves}(S, \{b_t\}, E)$.
  $U \leftarrow \mathrm{GreedyMove}(\mathcal{U}; S, \{L_t\}, w)$.
  $S \leftarrow S \cup U$.
end while
Sort S by source time and spatial position.
$(\widehat{\mathbf{Z}}_S, \mathbf{P}_S, \mathbf{D}_S) \leftarrow (\mathbf{Z}[S], \mathbf{P}[S], \mathbf{D}[S])$.
if temporal refinement is enabled **then**
  for each selected anchor $i \in S$ **do**
    Retrieve context $C(i)$ from the unchanged dense pool $\mathbf{Z}$.
    Write one residual update using Eq. 7.
  end for
end if
return $(\widehat{\mathbf{Z}}_S, \mathbf{P}_S, \mathbf{D}_S)$: exactly B source-aligned tokens.

Scope of the refinement claim. The refinement module does not claim to reconstruct the discarded dense image sequence or reproduce the native video encoder. Its role is narrower: it allows a fixed set of source-aligned anchors to incorporate selected temporal context while preserving the output-token budget and source interface. Accordingly, the $B \to B$ property is a statement about the representation passed to the language model, not a claim that context retrieval and refinement incur zero computational cost.

I Complete Selector Specification and Hyperparameters

The main text presents the selector at the level of design principles. This section gives the implementation-consistent specification used in all experiments. In particular, it clarifies the question-relevance score, the per-frame DPP kernel, the inherited frame quotas, the adjacent-frame correspondence graph, the exact greedy move scores, and the context size used by the temporal refiner. Unless otherwise noted, all quantities below are fixed at inference time and are shared across backbones.

I.1 Question-Conditioned Relevance

For each question, we encode the complete question together with up to eight question/option fragments. Let Q_r denote the input-token indices of text fragment r. We form one vector per fragment by averaging the corresponding language-model input embeddings,

$$q_r = \frac{1}{|Q_r|} \sum_{k \in Q_r} e_k. \tag{38}$$

For a visual candidate z_{ti} from frame t, both the visual feature and text vector are ℓ_2-normalized,

$$\bar{z}_{ti} = \frac{z_{ti}}{\|z_{ti}\|_2}, \qquad \bar{q}_r = \frac{q_r}{\|q_r\|_2}. \tag{39}$$

The question relevance used by the implementation is the maximum cosine similarity across the encoded text fragments,

$$m_{ti}(q) = \max_r \bar{z}_{ti}^\top \bar{q}_r. \tag{40}$$

Because no additional clipping is applied, $m_{ti}(q) \in [-1, 1]$. This is the range used by the implementation.

Table 7: **Frozen SITE allocator configuration.**

Question weight	Motion weight	Saliency weight	Diversity weight	Frame floor
0.8	0.6	0.4	0.4	12
Preview frames	**Preview boost**	**Uncertainty weight**	**Temporal NMS gap**	**DPP ratio**
5	0.75	0.45	2	0.5

I.2 Token Quality and Per-Frame DPP Kernel

The unnormalized quality of candidate i in frame t is

$$\widetilde{a}_{ti} = \max\{10^{-6}, \|z_{ti}\|_2 \, [1 + \eta m_{ti}(q)]\}, \qquad \eta = 0.8. \tag{41}$$

We normalize the qualities by their mean within each frame,

$$a_{ti} = \frac{\widetilde{a}_{ti}}{\frac{1}{N_t}\sum_j \widetilde{a}_{tj}}. \tag{42}$$

The DPP kernel for frame t is then

$$L_t(i,j) = a_{ti} a_{tj} \, \bar{z}_{ti}^{\top} \bar{z}_{tj} + \epsilon_t \mathbf{1}[i=j], \tag{43}$$

with

$$\epsilon_t = 10^{-3} \frac{1}{N_t} \sum_i a_{ti}^2. \tag{44}$$

The diagonal term is a numerical stabilizer. Selection within each frame is therefore driven jointly by candidate quality and diversity in the normalized visual-feature space.

I.3 Frame Quotas Inherited from the Frozen Allocator

The implementation does not compute a new frame allocation from a separate softmax or learned quota predictor. Instead, it inherits the non-uniform frame quotas from the frozen upstream SITE allocator and optimizes only the source-token identities under those quotas.

The SITE allocator first returns B compressed slots together with one representative source-token index r_ℓ for each slot ℓ. In the implementation, each sampled frame contributes the same number N of source candidates. Using one-based frame indices for exposition, the source frame of representative r_ℓ is

$$\mathrm{frame}(r_\ell) = 1 + \left\lfloor \frac{r_\ell}{N} \right\rfloor, \tag{45}$$

where r_ℓ is the zero-based global source index stored by the implementation. The representative count for frame t is

$$c_t = \sum_{\ell=1}^{B} \mathbf{1}[\mathrm{frame}(r_\ell) = t]. \tag{46}$$

The initial quota is

$$b_t = \min\{c_t, N\}. \tag{47}$$

If clipping causes $\sum_t b_t < B$, the remaining slots are assigned iteratively to frames with the largest residual capacity $N - b_t$ until

$$\sum_t b_t = B. \tag{48}$$

Thus, our coordinated selector preserves the frame-level budget profile produced by the frozen allocator while changing which source tokens occupy those slots. Table 7 lists the frozen allocator configuration used in all experiments.

I.4 Adjacent-Frame Correspondence Graph

Correspondence edges are constructed only between adjacent frames. For candidates i in frame t and j in frame $t+1$, define the cosine similarity

$$s_{ij} = \bar{z}_{ti}^{\top} \bar{z}_{t+1,j}. \tag{49}$$

A candidate pair is considered only when j is the top-1 match of i and i is also the top-1 reverse match of j. Let $\delta_{i\to j}$ and $\delta_{j\to i}$ denote the corresponding top-1 minus top-2 similarity margins in the forward and reverse directions. We use

$$\Delta_{ij} = \min\{\delta_{i\to j}, \delta_{j\to i}\}. \tag{50}$$

Let $p_i, p_j \in [0,1]^2$ be normalized patch coordinates. Their normalized spatial distance is

$$d_{ij} = \frac{\|p_i - p_j\|_2}{\sqrt{2}}. \tag{51}$$

Let s_t^{scene} denote the adjacent-frame scene-similarity statistic used by the implementation. A mutual match is retained as an edge only if

$$s_{ij} \geq 0.65, \qquad \Delta_{ij} \geq 0.005, \qquad d_{ij} \leq 0.5, \qquad s_t^{\text{scene}} \geq 0.65. \tag{52}$$

For retained edges, the matching confidence is

$$c_{ij} = \text{clip}\left(\frac{s_{ij} - 0.65}{0.35}, 0, 1\right) \text{clip}\left(\frac{\Delta_{ij}}{0.05}, 0, 1\right), \tag{53}$$

and the state-change score is

$$h_{ij} = \max\left\{\text{clip}\left(\frac{d_{ij}}{0.25}, 0, 1\right), \text{clip}(2[1 - s_{ij}], 0, 1)\right\}. \tag{54}$$

The final correspondence reward is

$$w_{ij} = c_{ij} h_{ij}. \tag{55}$$

This weighting favors mutually reliable correspondences whose endpoints also exhibit a measurable spatial or feature change.

I.5 Implemented Selection Objective and Greedy Solver

Let S_t denote the selected candidates from frame t, and let E be the correspondence graph defined above. The implementation-level objective is

$$J(S) = \sum_{t=1}^{T} \log\det L_t[S_t, S_t] + \gamma \sum_{\{i,j\}\in E} w_{ij} \mathbf{1}[i, j \in S], \qquad \gamma = 2.0, \tag{56}$$

subject to

$$|S_t| = b_t, \qquad \sum_t b_t = B. \tag{57}$$

The simpler selection expression in the main text should therefore be interpreted as design intuition; Eq. 56 is the objective implemented in our experiments.

For greedy evaluation, let U_t be the candidates already selected in frame t. The DPP Schur-complement residual for adding candidate i is

$$\rho_i^{\text{DPP}} = \begin{cases} L_t(i,i), & U_t = \varnothing, \\ L_t(i,i) - L_t(i,U_t) L_t[U_t, U_t]^{-1} L_t(U_t, i), & \text{otherwise.} \end{cases} \tag{58}$$

For numerical stability, the residual is floored at 10^{-14} before taking the logarithm. The singleton move score is

$$g_i = \log\max\{\rho_i^{\text{DPP}}, 10^{-14}\} + \gamma \sum_{j\in S:\{i,j\}\in E} w_{ij}. \tag{59}$$

For a feasible correspondence edge $\{i,j\} \in E$, the pair score is

$$g_{ij} = \frac{g_i + g_j + \gamma w_{ij}}{2}. \tag{60}$$

Table 8: **Selection and refinement implementation constants.**

Relevance weight η	Minimum raw quality	Diagonal-jitter coefficient	Residual floor	Cosine threshold	Margin threshold	Maximum spatial distance
0.8	10^{-6}	10^{-3}	10^{-14}	0.65	0.005	0.5
Scene-similarity threshold	**Confidence margin scale**	**Spatial-change scale**	**Feature-change multiplier**	**Pair-reward weight γ**	**Context size K**	**Neighbors per context frame**
0.65	0.05	0.25	2.0	2.0	8	4

Implemented `GreedyMove`. Given the current selected set S, one greedy iteration performs the following steps:

1. enumerate all quota-feasible unselected singletons and compute g_i using Eq. 59;
2. let $i^\star$ be the feasible singleton with the largest score;
3. enumerate all quota-feasible correspondence edges $\{i, j\} \in E$ and compute g_{ij} using Eq. 60;
4. let $\{u^\star, v^\star\}$ be the feasible pair with the largest pair score; if its score is strictly larger than $g_{i^\star}$, add both endpoints, otherwise add $i^\star$;
5. repeat until all frame quotas are satisfied and $|S| = B$.

The availability of feasible singleton moves guarantees completion whenever slots remain. The solver is greedy and makes no claim of globally maximizing Eq. 56.

I.6 Temporal Context Retrieval

The temporal refiner uses a fixed context size of

$$K = 8. \tag{61}$$

For a selected anchor i in an interior frame t, we retrieve the four most similar tokens by cosine similarity from the previous frame and the four most similar tokens from the next frame,

$$C(i) = \mathrm{Top4}_{t-1}(i) \cup \mathrm{Top4}_{t+1}(i). \tag{62}$$

For the first frame, the two context frames are frames 2 and 3; for the last frame, they are frames $T - 1$ and $T - 2$. Context is retrieved from the unchanged dense image pool and may therefore include tokens that were not selected into the B output anchors.

For each anchor-context pair, the refiner receives the relative-position feature

$$[\mathrm{sign}(\Delta t) \log(1 + |\Delta t|), \Delta x, \Delta y], \tag{63}$$

where Δx and Δy are normalized spatial offsets. The retrieved context is used only to update the retained anchors; it does not increase the number of visual tokens passed to the language model. Table 8 summarizes the remaining selection and context-retrieval constants.