ADADPC: ADAPTIVE DUAL-PATH COMPUTATION FOR EFFICIENT LONG-VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-video understanding requires models to process large visual token sequences, yet useful visual evidence does not necessarily require full updates at every layer. The key challenge is therefore to reduce redundant visual state refinement without losing access to information that may still support the answer. We propose AdaDPC, an adaptive dual-path computation framework that explicitly separates visual state refinement from visual evidence access. The Selective Refinement Path performs full attention and FFN updates on dynamically selected visual blocks, while the Query-Guided Access Path allows text queries to retrieve evidence from the remaining visual states at low cost. Both paths operate on a shared evolving visual state sequence, allowing visual blocks to change computational roles across layers. Cross-Layer Priority Feedback further uses question-relevant evidence discovered during access to guide subsequent refinement within a fixed budget. Across eight benchmarks, AdaDPC outperforms the dense model, improving MLVU Dev from 65.3 to 69.7 and TimeScope from 76.3 to 81.9, gains of 4.4 and 5.6 percentage points, respectively. Meanwhile, AdaDPC reduces FLOPs by 55.5%, demonstrating that broad visual evidence access can be preserved without uniformly applying expensive refinement.

1 INTRODUCTION

Long-video understanding requires multimodal large language models to identify events, track changes, and integrate observations across time (Wu et al., 2024a; Zhou et al., 2025; Shangguan et al., 2025). Longer videos and denser sampling produce large visual token sequences that repeatedly undergo self-attention and FFN computation (Vaswani et al., 2017). Yet useful visual content does not necessarily require full updates at every layer. The central question is how much further computation each visual state needs, rather than simply whether its content is useful.

Figure 1(b) distinguishes *visual-state refinement* from *visual-evidence access*: refinement develops a representation, whereas access uses its encoded information for answering. A repeated action may already be adequately represented, yet each occurrence remains counting evidence. Earlier scenes provide references for state changes, and distributed observations establish temporal order. Refinement needs vary across layers: a state may need full updates at one depth and lightweight access at another. Thus, states needing little refinement can remain useful evidence.

Dense processing maintains access while repeatedly updating all states, whereas reducing computation can also change when and how their information is available. Figure 1(a) provides a diagnostic: on 320 video-QA examples randomly sampled from the eight benchmarks used in our evaluation, 68.6% of high-access tokens in Qwen2.5-VL-3B lie in the lower-update half. The token-level analysis is described in Appendix A. The measured interventions in Figure 1(c) show 83.8% accuracy with updates disabled but access retained, compared with 85.3% with both enabled and 56.3% with both disabled. These results motivate selective refinement with continued evidence access.

We propose **AdaDPC**, an Adaptive Dual-Path Computation framework for *selective refinement with broad evidence access* in offline long-video understanding. Adaptive Block Routing assigns learned refinement priorities to visual blocks at each layer, with MoD-inspired token reweighting (Raposo et al., 2024) supporting task-supervised Router learning. Tokens within each block share a routing decision without pooling or merging their representations. The Selective Refinement Path, abbreviated as SRP, jointly updates selected visual states and text through full attention and FFN com-

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.

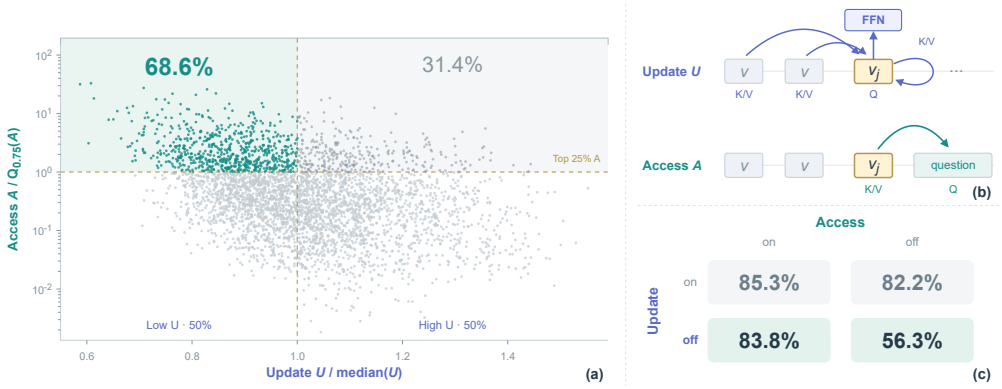

Figure 1: **Refine selectively, preserve visual evidence.** (a) On 320 video-QA examples randomly sampled from the eight evaluation benchmarks, 68.6% of Qwen2.5-VL-3B visual tokens in the top access quartile fall within the lower-update half. Scores are normalized within each example by the median update magnitude and the 75th-percentile access score, respectively; the vertical axis is logarithmic, and both dashed thresholds are at one. (b) Update refines a visual state through attention and FFN; Access allows question queries to read its keys and values. (c) Measured accuracies under Update–Access interventions are 85.3% with both enabled, 82.2% with only Update enabled, 83.8% with only Access enabled, and 56.3% with both disabled. Retaining access substantially mitigates the accuracy loss from suppressing updates.

putation. The Query-Guided Access Path, abbreviated as QGA, uses few text queries to read the remaining visual keys and values, directly enriching current text representations without full self-attention or FFN updates of those visual states. Both paths provide evidence access; they differ in whether visual states receive intensive refinement.

The paths share one evolving visual state sequence rather than separately constructed representations with fixed responsibilities. States retain their original positions and can switch roles across layers. Previously refined states can later be accessed cheaply, while states read without full updates can regain refinement. Evidence access also informs future refinement: Cross-Layer Priority Feedback transfers indices of question-relevant blocks identified during evidence access to prioritize next-layer updates within the existing budget. New Router scores fill the remaining slots. This connects refinement allocation, evidence access, relevance discovery, and subsequent reallocation over the same accumulated visual states.

Experiments across eight video-language benchmarks show that AdaDPC improves over Dense and the matched Dense + SFT baseline while reducing LLM inference FLOPs by 55.5% relative to Dense in the long-context efficiency setting. These results support selective visual-state refinement with continued evidence access.

Our contributions are threefold:

- We introduce AdaDPC, combining selective refinement with broad evidence access through adaptive routing over unified visual states.
- We develop Cross-Layer Priority Feedback to guide subsequent refinement using relevance discovered during evidence access within fixed layer budgets.
- We outperform Dense and Dense + SFT on eight benchmarks with 55.5% fewer LLM inference FLOPs than Dense; QGA gains cost 0.36% extra TFLOPs over SRP.

2 Related Work

Video-Language Modeling. Visual instruction tuning (Liu et al., 2023), query-based visual interfaces (Li et al., 2023), and gated cross-attention (Alayrac et al., 2022) provide complementary mechanisms for connecting visual representations to language models. For long videos, LLaMA-VID compresses each frame into context and content tokens (Li et al., 2025), while LongVU applies

spatiotemporal adaptive compression (Shen et al., 2025b). MovieChat (Song et al., 2024) and MA-LMM (He et al., 2024) use memory mechanisms to retain information across long video sequences.

Efficient Video and Multimodal Understanding. Existing work reduces the cost of video MLLMs mainly through visual token pruning, merging, sparse attention, or compact multi-scale representations. DynamicViT learns token sparsification in vision transformers (Rao et al., 2021); PuMer (Cao et al., 2023) and SparseVLM (Zhang et al., 2025b) use text information to guide visual token reduction. VisionZip (Yang et al., 2025b), DyCoke (Tao et al., 2025), and FastVID (Shen et al., 2025a) further explore efficient visual or video token compression. FastV (Chen et al., 2025a) prunes low-importance visual tokens in deeper layers, while ToMe (Bolya et al., 2023) and FrameFusion (Fu et al., 2025b) reduce redundancy by merging similar tokens, with FrameFusion further combining merging and pruning for videos. Other methods preserve broader visual coverage through sparse attention or KV compression, such as MoBA (Lu et al., 2025), AdaSpark (Li et al., 2026), and StreamMem. Sparsity Forcing optimizes token sparsity through reinforcement learning (Chen et al., 2026). SlowFast-style approaches (Feichtenhofer et al., 2019; Xu et al., 2024) instead construct visual streams with different temporal or spatial resolutions, assigning different representational roles before LLM processing. These methods demonstrate substantial redundancy in long-video inputs, but typically reduce computation by removing, merging, sparsifying, or pre-separating visual representations. AdaDPC instead operates on a single evolving visual state sequence and separates expensive visual-state refinement from lightweight question-guided evidence access.

Conditional Computation for Multimodal Models. Mixture-of-Depths dynamically routes only selected tokens through attention and FFN under a fixed computation budget (Raposo et al., 2024), and p-MoD extends this idea to multimodal models with depth-dependent visual computation (Zhang et al., 2025a). Related adaptations include γ-MoD (Luo et al., 2025) and VideoLLM-MoD (Wu et al., 2024b). AdaDPC adopts this conditional-computation principle for visual refinement, while keeping unselected states accessible through QGA. Cross-Layer Priority Feedback further uses question-conditioned relevance revealed during access to guide subsequent refinement, allowing visual blocks to switch computational roles across layers without constructing separate fixed visual streams.

3 METHOD

AdaDPC assigns visual blocks to the Selective Refinement Path or Query-Guided Access Path, abbreviated as SRP and QGA, over a unified visual state sequence. Roles can change across layers while reusing accumulated representations. The Router and transformation operators absorb the backbone's standard input preprocessing; outer residual additions are explicit. Figure 2 illustrates the overall architecture.

3.1 ADAPTIVE BLOCK ROUTING

We adapt MoD-style conditional computation (Raposo et al., 2024) to spatially grouped visual states in long videos while preserving access to non-refined content.

Block Priority Scoring. Let $\mathbf{H}^\ell$ be the layer input, $\mathcal{V}$ its visual positions, and $\mathbf{v}_i^\ell = \mathbf{H}_i^\ell$ for $i \in \mathcal{V}$. We partition each temporal position's 2D visual token grid into blocks of width b_w and height b_h, with nominal size $B = b_w b_h$. The resulting N_b blocks $\{\mathcal{B}_j\}_{j=1}^{N_b}$ contain original sequence indices and retain fixed spatial membership without pooling or merging representations. At each routed layer, a linear Router with bias recomputes priorities:

$$\rho_i^\ell = \mathrm{Router}^\ell(\mathbf{v}_i^\ell), \qquad s_i^\ell = \tanh(\rho_i^\ell), \qquad R_j^\ell = \max_{i \in \mathcal{B}_j} \rho_i^\ell. \tag{1}$$

Raw-score maxima determine block priorities without diluting high-priority tokens; bounded scores support token reweighting.

Depth-wise Budget Scheduling. Because QGA reads existing states, shallow layers receive broader refinement to build contextualized representations; deeper layers increasingly access accumulated information without repeated full updates. All L decoder layers are routed. With

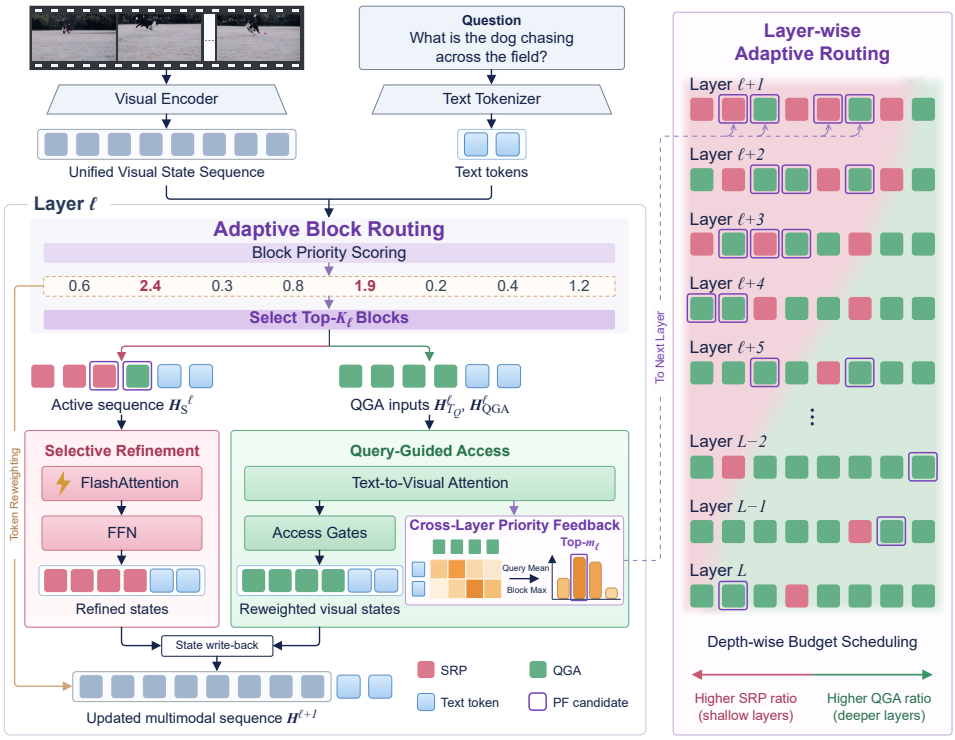

Figure 2: **Overview of AdaDPC.** Adaptive Block Routing assigns visual blocks to SRP or QGA over a unified visual state sequence, with Cross-Layer Priority Feedback informing subsequent refinement allocation.

$\ell \in \{1, \ldots, L\}$ denoting the global decoder-layer index, we use a shifted cosine schedule with a minimum refinement ratio (Zhang et al., 2025a):

$$r_\ell = \frac{1}{2}\cos\left(\frac{\pi \ell}{L}\right) + \beta, \qquad \hat{r}_\ell = \max\{r_{\min}, r_\ell\}, \qquad K_\ell = \lceil \hat{r}_\ell N_b \rceil . \tag{2}$$

With shift $\beta = 0.5$ and $r_{\min} = 0.2$, refinement decays to a 20% floor. The budget fixes how many blocks are refined; current Router scores and feedback determine which, allowing roles to change across layers. With subscripts S and Q denoting SRP and QGA, respectively, the selected and remaining block-index sets are $\mathcal{G}_{\mathrm{S}}^\ell$ and $\mathcal{G}_{\mathrm{Q}}^\ell$; their visual-position sets are $\mathcal{V}_{\mathrm{S}}^\ell = \bigcup_{j \in \mathcal{G}_{\mathrm{S}}^\ell} \mathcal{B}_j$ and $\mathcal{V}_{\mathrm{Q}}^\ell = \mathcal{V} \setminus \mathcal{V}_{\mathrm{S}}^\ell$.

Cross-Layer Priority Feedback. Cross-Layer Priority Feedback, abbreviated as PF, uses QGA attention to identify question-relevant content for subsequent refinement. Feedback scores derive from its attention weights rather than output features or Router logits. Let $\boldsymbol{\iota}^\ell = \mathrm{Sort}(\mathcal{V}_{\mathrm{Q}}^\ell)$ map local visual-key indices to original positions. Feedback queries $\mathcal{Q}_{\mathrm{fb}}$ include only valid input-question positions after the video, excluding padding and answers. These queries can access all valid visual keys in this path. For query q, local key k, and head h of dimension d_h,

$$p_{hqk}^\ell = \frac{\exp\left(\langle \mathbf{Q}_{hq}^\ell, \mathbf{K}_{hk}^\ell \rangle / \sqrt{d_h}\right)}{\sum_{u=1}^{|\mathcal{V}_{\mathrm{Q}}^\ell|} \exp\left(\langle \mathbf{Q}_{hq}^\ell, \mathbf{K}_{hu}^\ell \rangle / \sqrt{d_h}\right)}. \tag{3}$$

We average over H heads and $N_{\mathrm{fb}} = |\mathcal{Q}_{\mathrm{fb}}|$ queries, then apply the same within-block maximum as in routing to these attention-derived token scores:

$$a_{\iota_k^\ell}^\ell = \frac{1}{H N_{\mathrm{fb}}} \sum_{h=1}^{H} \sum_{q \in \mathcal{Q}_{\mathrm{fb}}} p_{hqk}^\ell, \qquad \eta_j^\ell = \max_{i \in \mathcal{B}_j} a_i^\ell, \quad j \in \mathcal{G}_{\mathrm{Q}}^\ell. \tag{4}$$

Let $(b_1, \ldots, b_{M_\ell})$ list the $M_\ell = |\mathcal{G}_{\mathrm{Q}}^{\ell}|$ QGA blocks. For $\ell < L$ and feedback ratio λ, we select $m_\ell = \min\{\lceil \lambda K_{\ell+1} \rceil, M_\ell\}$ candidates and map local Top-K positions to block IDs:

$$\mathcal{J}^{\ell} = \mathrm{TopK}\Big((\eta_{b_t}^{\ell})_{t=1}^{M_\ell}, m_\ell\Big), \quad \mathcal{P}^{\ell} = \{b_t : t \in \mathcal{J}^{\ell}\}. \tag{5}$$

These candidates occupy existing slots in $K_{\ell+1}$; other blocks are ranked by newly computed next-layer Router scores to fill the remaining slots. All valid text and template positions $\mathcal{T}$ are included, and original order is restored:

$$\mathcal{I}_{\mathrm{S}}^{\ell+1} = \mathrm{Sort}\left(\mathcal{T} \cup \bigcup_{j \in \mathcal{G}_{\mathrm{S}}^{\ell+1}} \mathcal{B}_j\right). \tag{6}$$

Gathering preserves positional identifiers. The first decoder layer ($\ell = 1$) uses Router scores alone. Feedback reallocates computation without removing visual states.

Router Learning via Token Reweighting. Inspired by MoD-style token reweighting (Raposo et al., 2024; Zhang et al., 2025a), we couple continuous scores to visual-state computation so the task loss can supervise the Router, while Top-K assignments remain discrete. With SRP causal self-attention $\mathrm{A}_{\mathrm{SRP}}^{\ell}$ and feed-forward operator F^{ℓ}, selected states in the active sequence $\mathbf{H}_{\mathrm{S}}^{\ell}$ satisfy

$$\mathbf{u}_i^{\ell} = \mathbf{v}_i^{\ell} + [\mathrm{A}_{\mathrm{SRP}}^{\ell}(\mathbf{H}_{\mathrm{S}}^{\ell})]_i, \qquad \mathbf{f}_i^{\ell} = \mathrm{F}^{\ell}(\mathbf{u}_i^{\ell}). \tag{7}$$

Scores modulate selected transformations and remaining residual states:

$$\begin{aligned} \mathbf{v}_i^{\ell+1} &= \mathbf{u}_i^{\ell} + \mathbf{f}_i^{\ell} + s_i^{\ell}\mathbf{f}_i^{\ell}, \quad i \in \mathcal{V}_{\mathrm{S}}^{\ell}, \\ \mathbf{v}_i^{\ell+1} &= \mathbf{v}_i^{\ell} + s_i^{\ell}\mathbf{v}_i^{\ell}, \qquad\quad i \in \mathcal{V}_{\mathrm{Q}}^{\ell}. \end{aligned} \tag{8}$$

Block members share discrete routing decisions but retain token-specific modulation scores; learning does not differentiate through Top-K.

3.2 Selective Refinement Path

The Selective Refinement Path gathers $\mathbf{H}_{\mathrm{S}}^{\ell} = \mathrm{Gather}(\mathbf{H}^{\ell}, \mathcal{V}_{\mathrm{S}}^{\ell} \cup \mathcal{T})$ in original order with unchanged positional identifiers. Blocks are routing units, not attention windows: the active sequence uses dense causal attention compatible with FlashAttention. Visual updates follow Eq. 8 and are written back to their original positions in the unified sequence.

3.3 Query-Guided Access Path

QGA gathers the remaining visual states $\mathbf{H}_{\mathrm{QGA}}^{\ell} = \mathrm{Gather}(\mathbf{H}^{\ell}, \mathcal{V}_{\mathrm{Q}}^{\ell})$. Text queries use layer-input states $\mathbf{H}_{\mathcal{T}_{\mathrm{Q}}}^{\ell}$ at valid post-video positions $\mathcal{T}_{\mathrm{Q}} \subseteq \mathcal{T}$. Both paths reuse Q/K/V projected once from all layer-input states.

Let $\mathrm{A}_{\mathrm{QGA}}^{\ell}$ denote text-to-visual attention with learnable per-head gates $g_h^{\ell} = \tanh(\gamma_h^{\ell})$ and an additional output projection. The SRP attention-residual text state is $\mathbf{U}_{\mathcal{T}_{\mathrm{Q}}}^{\ell} = \mathbf{H}_{\mathcal{T}_{\mathrm{Q}}}^{\ell} + [\mathrm{A}_{\mathrm{SRP}}^{\ell}(\mathbf{H}_{\mathrm{S}}^{\ell})]_{\mathcal{T}_{\mathrm{Q}}}$. We fuse retrieved evidence before a single text FFN:

$$\begin{aligned} \widehat{\mathbf{H}}_{\mathcal{T}_{\mathrm{Q}}}^{\ell} &= \mathbf{U}_{\mathcal{T}_{\mathrm{Q}}}^{\ell} + \mathrm{A}_{\mathrm{QGA}}^{\ell}\Big(\mathbf{H}_{\mathcal{T}_{\mathrm{Q}}}^{\ell}, \mathbf{H}_{\mathrm{QGA}}^{\ell}; \mathbf{g}^{\ell}\Big), \\ \mathbf{H}_{\mathcal{T}_{\mathrm{Q}}}^{\ell+1} &= \widehat{\mathbf{H}}_{\mathcal{T}_{\mathrm{Q}}}^{\ell} + \mathrm{F}^{\ell}\Big(\widehat{\mathbf{H}}_{\mathcal{T}_{\mathrm{Q}}}^{\ell}\Big). \end{aligned} \tag{9}$$

For $n_q = |\mathcal{T}_{\mathrm{Q}}|$ text queries, $n_v = |\mathcal{V}_{\mathrm{Q}}^{\ell}|$ remaining visual tokens and model width d, the additional attention and output projection cost is $O(n_q n_v d + n_q d^2)$. Shared full-sequence Q/K/V projection is counted separately in total FLOPs. When $n_q \ll n_v$, QGA accesses remaining visual states without their full self-attention or FFN updates.

4 Experiments

In this section, we present the experimental setup, details of the datasets, and comparisons with state-of-the-art efficient methods across multiple video benchmarks.

4.1 EXPERIMENTAL SETUP

Implementation Details. We implement AdaDPC on Qwen2.5-VL-3B-Instruct (Bai et al., 2025), hereafter Qwen2.5-VL-3B. Dense denotes the original checkpoint without our additional post-training, while Dense + SFT denotes the same dense architecture fine-tuned with our post-training data and optimization settings. During supervised fine-tuning, video inputs are sampled at 2 frames per second with native spatial resolution. We use a 32K visual-token context for the main experiments, efficiency profiling, and architectural comparison, and an 8K visual-token context for the routing-granularity and path-allocation ablations. These budgets count visual tokens at the decoder input, excluding text and template tokens. All decoder layers are routed using 8×4 spatial blocks (32 tokens), a shifted cosine refinement schedule with $\beta = 0.5$ and $r_{\min} = 0.2$, and a cross-layer feedback ratio of $\lambda = 0.25$. Appendix C provides further routing details and training hyperparameters; Appendix D specifies benchmark-dependent frame-sampling settings.

Training Data. Our training methodology incorporates a mixed dataset, centrally featuring the LLaVA-Video-178K dataset (Zhang et al., 2025c), which serves as a foundational corpus for general video understanding. In addition, we augment the training data with 77K timestamp-grounded samples from DiDeMo (Anne Hendricks et al., 2017) and ActivityNet Captions (Krishna et al., 2017) to strengthen the model's ability to capture temporally localized information. We further include 118K samples from VideoMarathon (Lin et al., 2025) to enhance the model's capability for long-video understanding and reasoning over extended temporal contexts.

Benchmarks and Compared Models. We evaluate AdaDPC on eight video-language benchmarks covering four capability axes: long-form video understanding with MLVU Dev (Zhou et al., 2025), VideoMME (Fu et al., 2025a), LongVideoBench (Wu et al., 2024a), LVBench (Wang et al., 2025), and TimeScope; short-form video understanding with MVBench (Li et al., 2024); spatial perception with VSI-Bench (Yang et al., 2025a); and temporal reasoning with TOMATO (Shangguan et al., 2025). We re-implement the efficiency-oriented baselines on the same Qwen2.5-VL-3B backbone using the configurations recommended in their original publications and evaluate them alongside AdaDPC with the same `lmms-eval` pipeline. Table 1 additionally reports representative Video-LLMs, including InternVL2.5 (Chen et al., 2025b), Ovis2 (Lu et al., 2024), and LongVU (Shen et al., 2025b), for broader reference.

4.2 MAIN RESULTS

As shown in Table 1, AdaDPC improves over Dense across all eight benchmarks. The gains are particularly pronounced on long-form video understanding: AdaDPC achieves 69.7 on MLVU Dev and 81.9 on TimeScope, exceeding Dense by 4.4 and 5.6 percentage points, respectively. It also improves LongVideoBench from 54.2 to 56.4 and VideoMME from 61.5 to 62.6, while achieving consistent gains on LVBench, MVBench, VSI-Bench, and TOMATO. AdaDPC further surpasses the Dense + SFT baseline on all eight benchmarks, indicating that the gains extend beyond the effect of additional post-training.

Compared with the re-implemented efficiency-oriented methods, AdaDPC achieves strong performance across all evaluated capability axes. It exceeds Sparsity Forcing by 2.5 points on MLVU Dev and 1.6 points on VideoMME, and outperforms FrameFusion across all eight shared benchmarks. AdaDPC also reaches 66.9 on MVBench, 33.9 on VSI-Bench, and 26.1 on TOMATO, showing that the benefits extend beyond long-form understanding to short-form, spatial, and temporal reasoning tasks. Overall, these results demonstrate that AdaDPC substantially reduces repeated visual computation while maintaining strong and broadly distributed video understanding capabilities.

Appendix B provides complete breakdowns for the seven benchmarks with available category-level results. AdaDPC achieves higher scores on 51 of 73 subtasks and lower scores on the remaining 22. Consistent with our motivation, the gains are pronounced on tasks requiring repeated evidence and temporal integration, including MLVU Count (+30.08) and Order (+11.84), VideoMME Temporal Reasoning (+6.25), LongVideoBench O3O (+11.15), and MVBench Action Localization (+13.85). The complete breakdown also reveals task-dependent regressions rather than uniform improvements.

Table 1: **Performance on zero-shot video-language benchmarks.** Using a 32K visual-token context, we evaluate AdaDPC on 5 long-form video benchmarks, 1 short-form video benchmark, 1 spatial perception benchmark, and 1 temporal reasoning benchmark. Dense and Dense + SFT denote Qwen2.5-VL-3B before and after our additional supervised fine-tuning, respectively. The results in **bold** and underline indicate the best and second-best results in each column, respectively. Subscripts in the AdaDPC row report absolute gains over Dense in percentage points.

	Long-form					Short-form	Spatial	Temporal
Models	*MLVU Dev*	*VideoMME*	*LongVideoBench*	*LVBench*	*TimeScope*	*MVBench*	*VSI-Bench*	*TOMATO*
InternVL2.5-2B	61.4	51.9	52.0	37.9	–	**68.8**	25.8	20.0
Ovis2-2B	68.6	57.2	50.9	39.0	–	64.9	26.7	21.4
LongVU-3B	55.9	51.5	46.7	36.6	–	60.9	–	25.7
Dense	65.3	61.5	54.2	43.3	76.3	65.7	32.4	23.5
Dense + SFT	65.2	60.3	53.6	43.1	77.4	64.9	33.1	24.2
+ FastV	65.1	62.4	55.9	43.3	73.9	65.6	29.9	23.6
+ ToMe	65.4	62.2	55.8	42.4	76.1	64.5	29.4	23.5
+ MoBA	63.2	58.4	51.2	39.4	74.7	64.0	28.9	21.4
+ FrameFusion	65.0	62.4	56.0	43.4	79.5	65.4	33.6	22.2
+ Sparsity Forcing	67.2	61.0	–	–	–	–	–	–
+ CaCoVID	59.9	57.5	53.3	–	–	–	–	–
+ StreamMem	62.3	59.5	–	–	–	–	–	–
+ AdaDPC (Ours)	**69.7** $_{+4.4}$	**62.6** $_{+1.1}$	**56.4** $_{+2.2}$	**44.0** $_{+0.7}$	**81.9** $_{+5.6}$	66.9 $_{+1.2}$	**33.9** $_{+1.5}$	**26.1** $_{+2.6}$

Table 2: **Efficiency and architectural ablation.** The comparison uses a 32K visual-token context. SRP and QGA denote the Selective Refinement Path and Query-Guided Access Path, respectively; SRP + QGA denotes AdaDPC. Red. denotes the LLM TFLOPs reduction relative to Dense, reported as a positive saving.

	Efficiency		Performance							
Configuration	**TFLOPs**	**Red.**	*LVBench*	*LongVideoBench*	*MLVU Dev*	*VideoMME*	*MVBench*	*TOMATO*	*TimeScope*	*VSI-Bench*
Dense	3,175	–	43.3	54.2	65.3	61.5	65.7	23.5	76.3	32.4
Dense + SFT	3,175	–	43.1	53.6	65.2	60.3	64.9	24.2	77.4	33.1
SRP	1,407	55.7%	43.4	53.8	65.9	60.7	62.6	26.0	79.1	32.8
SRP + QGA	**1,412**	55.5%	**44.0**	**56.4**	**69.7**	**62.6**	**66.9**	**26.1**	**81.9**	**33.9**

4.3 Efficiency Analysis

Following prior work on efficient long-video understanding (Fu et al., 2025b), we report LLM inference FLOPs excluding the vision encoder. AdaDPC reduces LLM computation by 55.5% relative to Dense while retaining full-sequence Q/K/V projection. We further profile decoder layer 20 with 32K visual tokens using BF16, FlashAttention 2 (Dao, 2024), and batch size 1. As shown in Table 3, the reduced computation translates to substantial kernel-level acceleration and lower FFN memory usage.

Table 3: **Efficiency profile.** FLOPs and peak-memory reductions are relative to Dense; profiling uses 32K visual tokens.

Metric	**Result**
LLM FLOPs reduction	55.5%
Self-attention speedup	5.24×
FFN speedup	2.65×
FFN peak-memory reduction	62.8%

4.4 Ablation Study

We study the 3B model under the 32K visual-token setting for Table 2 and the 8K visual-token setting for Tables 4 and 5. For the controlled architectural comparison, Dense + SFT, SRP, and SRP + QGA share post-training data and optimization settings, including training steps, learning rate, and global batch size, differing only in computation architecture.

Effect of Dual-Path Computation. Table 2 shows gains from SRP + QGA over SRP on all eight benchmarks, including 4.3 percentage points on MVBench, 3.8 on MLVU Dev, 2.8 on TimeScope, and 2.6 on LongVideoBench. AdaDPC also outperforms Dense and Dense + SFT throughout. These gains support selective refinement with question-guided evidence access over shared visual states, consistent with our premise that computational redundancy does not imply task redundancy.

Table 4: **Access gating and routing granularity.** All variants use 8K visual tokens. Blocks are width $\times$ height. MLVU = MLVU Dev; VMME = VideoMME; TOM. = TOMATO.

Config.	MLVU	VMME	TOM.
Access Gating			
w/o Gate	59.8	58.5	22.8
w/ Gate	**62.5**	**61.5**	**24.3**
Routing Granularity			
Token	59.6	55.7	23.8
Block 16 (4×4)	60.7	57.7	23.5
Block 32 (8×4)	**62.5**	**61.5**	**24.3**
Block 64 (8×8)	59.8	59.8	23.7

Table 5: **Computation allocation and priority feedback.** Fixed ratios denote SRP:QGA. PF denotes Cross-Layer Priority Feedback. Parentheses report aggregate LLM FLOPs reduction over the three benchmarks relative to Dense. All λ variants use the Cosine Decay budget.

Config.	MLVU	VMME	TOM.
Path Allocation Strategy			
Top-p (42.7%)	60.5	56.1	23.7
Fixed 50:50 (43.7%)	61.0	56.7	23.5
Fixed 25:75 (61.9%)	57.6	53.7	21.1
Cosine Decay (45.1%)	**62.5**	**61.5**	**24.3**
Priority Feedback Ratio			
$\lambda = 0$ (w/o PF)	61.1	57.7	21.5
$\lambda = 0.1$	62.2	58.3	23.7
$\boldsymbol{\lambda = 0.25}$	**62.5**	**61.5**	**24.3**
$\lambda = 0.5$	61.9	57.8	22.5

Effect of Access Gating and Routing Granularity. In Table 4, removing the access gate lowers MLVU Dev, VideoMME, and TOMATO by 2.7, 3.0, and 1.5 points, respectively, supporting gating to regulate evidence injection into text representations. Among token-wise routing and 2D blocks of 16, 32, and 64 tokens, Block 32 performs best on all three benchmarks, exceeding token-wise routing by 2.9, 5.8, and 0.5 points, respectively. Performance declines again with 64-token blocks. This suggests that intermediate granularity balances local grouping and routing flexibility: smaller units provide less spatial grouping, whereas larger blocks constrain selection.

Effect of Computation Allocation and Priority Feedback. Table 5 shows that Fixed 25:75 maximizes LLM FLOPs savings at 61.9% but yields the weakest scores. Cosine decay performs best on all three benchmarks while saving 45.1%, compared with 42.7% for Top-p and 43.7% for Fixed 50:50. Its accuracy gains therefore do not require additional computation.

With the cosine schedule fixed, λ changes block priorities, not the refinement budget. Disabling feedback at $\lambda = 0$ lowers MLVU Dev, VideoMME, and TOMATO by 1.4, 3.8, and 2.8 points relative to $\lambda = 0.25$. A ratio of 0.1 improves over no feedback, while 0.25 performs best. Increasing to 0.5 lowers all three scores, suggesting that moderate feedback transfers useful cross-layer relevance while preserving current-layer routing flexibility.

Together, the ablations favor gated evidence access, intermediate spatial granularity, depth-wise budget scheduling, and moderate feedback for adaptive visual computation.

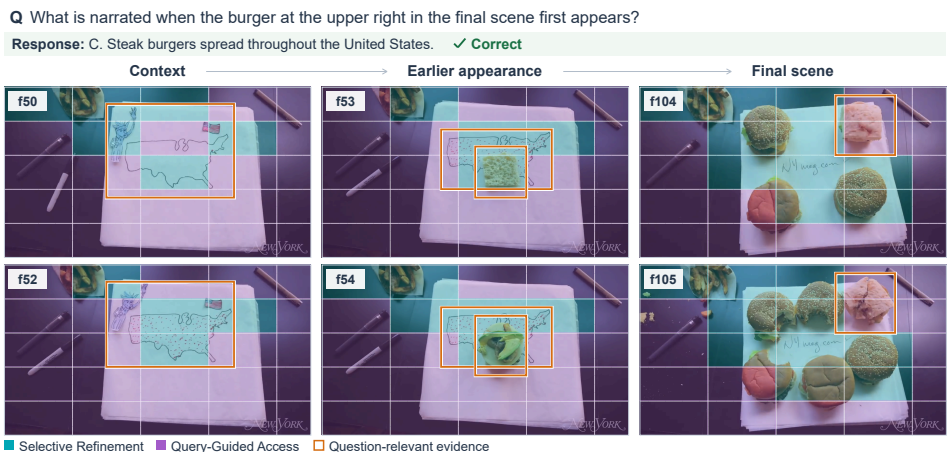

Figure 3: **Visualization of AdaDPC's adaptive computation behavior.**

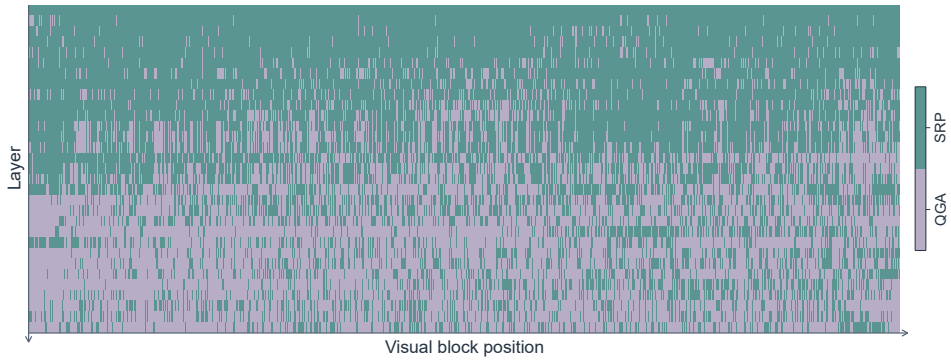

Figure 4: **Layer-wise block routing in AdaDPC.** Teal and purple indicate assignment to SRP and QGA, respectively.

4.5 Visualization Analysis

Figure 3 shows that question-relevant evidence is distributed across both computation paths. While SRP selectively refines salient regions, some evidence associated with the queried object is captured through QGA rather than full refinement, yet remains available to support the final answer. Figure 4 further shows that block assignments vary across visual positions and layers, with individual blocks switching between SRP and QGA as computation progresses. Together, these results show that AdaDPC preserves useful visual evidence without intensively updating every relevant region.

5 Conclusion

We presented AdaDPC for efficient long-video understanding, motivated by the distinction between visual-state refinement and visual-evidence access. AdaDPC selectively applies expensive attention and FFN updates to high-priority visual blocks while preserving lightweight question-guided access to the remaining visual states, with cross-layer feedback further adapting computation across depth. Experiments across eight video understanding benchmarks show consistent improvements over the matched Dense + SFT baseline while reducing LLM inference FLOPs by 55.5%. These results suggest that efficient long-video modeling does not require uniformly refining all useful visual content, as long as broadly distributed evidence remains accessible to the language model.

ETHICS STATEMENT

This work studies efficient long-video understanding using existing models and publicly available datasets, without new data collection. AdaDPC inherits potential biases and privacy risks from its backbone and training data; reducing inference costs does not resolve these limitations. More efficient video understanding may also facilitate misuse, including visual surveillance. Deployments should assess these risks and respect the applicable dataset and model licences.

REPRODUCIBILITY STATEMENT

Appendix C describes the training configuration, routed-layer pseudocode, and FLOPs accounting. Appendix D reports benchmark-specific evaluation and frame-sampling settings. Appendix A specifies the token-level motivation analysis, including score definitions and normalization, and Appendix B provides complete category-level results. The main text identifies the backbone, post-training data, visual-token budgets, and comparison settings used in our experiments.

THE USE OF LARGE LANGUAGE MODELS (LLMS)

Large language models were used to assist with manuscript editing, code development, and related-work organization. They were not used to generate experimental measurements. The authors reviewed the assisted content and take responsibility for the reported results and conclusions.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a Visual Language Model for Few-Shot Learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 23716–23736. Curran Associates, Inc., 2022. doi: 10.52202/068431-1723. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/960a172bc7fbf0177ccccbb411a7d800-Paper-Conference.pdf.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing Moments in Video With Natural Language. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL Technical Report, 2025. URL https://arxiv.org/abs/2502.13923.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token Merging: Your ViT But Faster. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=JroZRaRw7Eu.

Qingqing Cao, Bhargavi Paranjape, and Hannaneh Hajishirzi. PuMer: Pruning and Merging Tokens for Efficient Vision Language Models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12890–12903, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.721. URL https://aclanthology.org/2023.acl-long.721/.

Feng Chen, YEFEI HE, Lequan Lin, Jing Liu, Chenhui Gou, Bohan Zhuang, and Qi Wu. Sparsity Forcing: Reinforcing Token Sparsity of MLLMs. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 155748–155762, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/fc034d186280f55370b6aca7a3285a65-Paper-Conference.pdf.

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 Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 19–35, Cham, 2025a. Springer Nature Switzerland. ISBN 978-3-031-73004-7.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yiming Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang, Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Huipeng Deng, Jiaye Ge, Kai Chen, Kaipeng Zhang, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Expanding Performance Boundaries of Open-Source Multimodal Models with Model, Data, and Test-Time Scaling, 2025b. URL https://arxiv.org/abs/2412.05271.

Tri Dao. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 35549–35562, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/98ed250b203d1ac6b24bbcf263e3d4a7-Paper-Conference.pdf.

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast Networks for Video Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

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, June 2025a.

Tianyu Fu, Tengxuan Liu, Qinghao Han, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. FrameFusion: Combining Similarity and Importance for Video Token Reduction on Large Vision Language Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22654–22663, October 2025b.

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, June 2024.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-Captioning Events in Videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Handong Li, Zikang Liu, Longteng Guo, Tongtian Yue, Yepeng Tang, Xinxin Zhu, Chuanyang Zheng, Ziming Wang, Zhibin Wang, Jun Song, Cheng Yu, Bo Zheng, and Jing Liu. AdaSpark: Adaptive Sparsity for Efficient Long-Video Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 40548–40558, June 2026.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of

Proceedings of Machine Learning Research, pp. 19730–19742. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/li23q.html.

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, June 2024.

Yanwei Li, Chengyao Wang, and Jiaya Jia. LLaMA-VID: An Image is Worth 2 Tokens in Large Language Models. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 323–340, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-72952-2.

Jingyang Lin, Jialian Wu, Ximeng Sun, Ze Wang, Jiang Liu, Yusheng Su, Xiaodong Yu, Hao Chen, Jiebo Luo, Zicheng Liu, and Emad Barsoum. Unleashing Hour-Scale Video Training for Long Video-Language Understanding. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 17523–17552. Curran Associates, Inc., 2025. doi: 10.52202/085713-0591. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/19741c617c87a25627682e5714af1501-Paper-Conference.pdf.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916. Curran Associates, Inc., 2023. doi: 10.52202/075280-1516. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf.

Enzhe Lu, Zhejun Jiang, Jingyuan Liu, Yulun Du, Tao Jiang, Chao Hong, Shaowei Liu, Weiran He, Enming Yuan, Yuzhi Wang, Zhiqi Huang, Huan Yuan, Suting Xu, Xinran Xu, Guokun Lai, Yanru Chen, Huabin Zheng, Junjie Yan, Jianlin Su, Yuxin Wu, Yutao Zhang, Zhilin Yang, Xinyu Zhou, Mingxing Zhang, and Jiezhong Qiu. MoBA: Mixture of Block Attention for Long-Context LLMs. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 17790–17815. Curran Associates, Inc., 2025. doi: 10.52202/085713-0602. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/19eae75beed66321d62272e794a9c2ac-Paper-Conference.pdf.

Shiyin Lu, Yang Li, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Han-Jia Ye. Ovis: Structural Embedding Alignment for Multimodal Large Language Model, 2024. URL https://arxiv.org/abs/2405.20797.

Yaxin Luo, Gen Luo, Jiayi Ji, Yiyi Zhou, Xiaoshuai Sun, Zhiqiang Shen, and Rongrong Ji. γ-MoD: Exploring Mixture-of-Depth Adaptation for Multimodal Large Language Models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 72170–72183, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/b3847cda0c8cc0cfcdacf462dc122214-Paper-Conference.pdf.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 13937–13949. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/747d3443e319a22747fbb873e8b2f9f2-Paper.pdf.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-Depths: Dynamically allocating compute in transformer-based language models, 2024. URL https://arxiv.org/abs/2404.02258.

Ziyao Shangguan, Chuhan Li, Yuxuan Ding, Yanan Zheng, Yilun Zhao, Tesca Fitzgerald, and Arman Cohan. TOMATO: Assessing Visual Temporal Reasoning Capabilities

in Multimodal Foundation Models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 7593–7734, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/16ba99f25a235f1100a4014d71d34ad8-Paper-Conference.pdf.

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 D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 123553–123581. Curran Associates, Inc., 2025a. doi: 10.52202/085713-4118. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/b2e63e36c57e153b9015fece2352a9f9-Paper-Conference.pdf.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, Zhuang Liu, Hu Xu, Hyunwoo J. Kim, Bilge Soran, Raghuraman Krishnamoorthi, Mohamed Elhoseiny, and Vikas Chandra. LongVU: Spatiotemporal Adaptive Compression for Long Video-Language Understanding. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 54582–54599. PMLR, 13–19 Jul 2025b. URL https://proceedings.mlr.press/v267/shen25j.html.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang. MovieChat: From Dense Token to Sparse Memory for Long Video Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18221–18232, June 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, June 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

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, October 2025.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A Benchmark for Long-context Interleaved Video-Language Understanding. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 28828–28857. Curran Associates, Inc., 2024a. doi: 10.52202/079017-0907. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/329ad516cf7a6ac306f29882e9c77558-Paper-Datasets_and_Benchmarks_Track.pdf.

Shiwei Wu, Joya Chen, Kevin Qinghong Lin, Qimeng Wang, Yan Gao, Qianli Xu, Tong Xu, Yao Hu, Enhong Chen, and Mike Zheng Shou. VideoLLM-MoD: Efficient Video-Language Streaming with Mixture-of-Depths Vision Computation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 109922–109947. Curran Associates, Inc., 2024b. doi: 10.52202/079017-3488. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/c6a79e139ec4f371701ea8cc9e06018e-Paper-Conference.pdf.

Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. SlowFast-LLaVA: A Strong Training-Free Baseline for Video Large Language Models, 2024. URL https://arxiv.org/abs/2407.15841.

Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in Space: How Multimodal Large Language Models See, Remember, and Recall Spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10632–10643, June 2025a.

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, June 2025b.

Jun Zhang, Desen Meng, Zhengming Zhang, Zhenpeng Huang, Tao Wu, and Limin Wang. p-MoD: Building Mixture-of-Depths MLLMs via Progressive Ratio Decay. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3705–3715, October 2025a.

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 Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 74840–74857. PMLR, 13–19 Jul 2025b. URL https://proceedings.mlr.press/v267/zhang25s.html.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun MA, Ziwei Liu, and Chunyuan Li. LLaVA-Video: Video Instruction Tuning With Synthetic Data. *Transactions on Machine Learning Research*, 2025c. ISSN 2835-8856. URL https://openreview.net/forum?id=EElFGvt39K.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. MLVU: Benchmarking Multi-task Long Video Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13691–13701, June 2025.

A Details of Motivation Analysis

Data. For the token-level analysis in Figure 1(a), we randomly sample 320 video-QA examples from the eight benchmarks used in our evaluation. We collect visual-token activations and question-conditioned attention from Qwen2.5-VL-3B-Instruct during dense prefill.

Update and access scores. Let $\mathcal{L}_{\text{obs}}$ denote the decoder layers included in the analysis. For visual token i, the update score measures the relative change between the input and output of a complete decoder layer, including both attention and FFN residual updates:

$$U_i = \frac{1}{|\mathcal{L}_{\text{obs}}|} \sum_{\ell \in \mathcal{L}_{\text{obs}}} \frac{\|\mathbf{h}^{\ell}_{i,\text{out}} - \mathbf{h}^{\ell}_{i,\text{in}}\|_2}{\|\mathbf{h}^{\ell}_{i,\text{in}}\|_2 + \epsilon}, \qquad \epsilon = 10^{-8}. \tag{10}$$

The access score is the attention probability assigned to that visual token, averaged over the observed layers, all H query heads, and the question-token positions $\mathcal{Q}_{\text{ques}}$:

$$A_i = \frac{1}{|\mathcal{L}_{\text{obs}}| H |\mathcal{Q}_{\text{ques}}|} \sum_{\ell \in \mathcal{L}_{\text{obs}}} \sum_{h=1}^{H} \sum_{q \in \mathcal{Q}_{\text{ques}}} p^{\ell,h}_{qi}. \tag{11}$$

Question tokens are identified by their overlap with the question text in the tokenized prompt, excluding answer options and template instructions from the query set. Here $p^{\ell,h}_{qi}$ is the causal attention probability from question query q to visual key i, normalized over all causally visible keys, including text keys.

Normalization for visualization. After averaging over the observed layers, we normalize the scores separately within each example x. Let $\mathcal{V}_x$ denote its visual-token set. The plotted coordinates are

$$\widetilde{U}_i = \frac{U_i}{\text{median}_{j \in \mathcal{V}_x} U_j}, \qquad \widetilde{A}_i = \frac{A_i}{\text{Q}_{0.75}(\{A_j : j \in \mathcal{V}_x\})}, \tag{12}$$

where $\text{Q}_{0.75}$ denotes the 75th percentile. The vertical axis uses a logarithmic scale. Although the raw attention score A_i lies in $[0, 1]$, $\widetilde{A}_i$ is a relative score rather than a probability and may exceed one. The median-update and upper-access-quartile thresholds therefore correspond to $\widetilde{U} = 1$ and $\widetilde{A} = 1$, respectively. Only after this within-example normalization are the coordinates pooled for visualization.

Within-example thresholds and aggregation. Using the normalized scores, we define the low-update and high-access sets as

$$\mathcal{V}^{\text{low}}_x = \{i \in \mathcal{V}_x : \widetilde{U}_i < 1\}, \qquad \mathcal{V}^{\text{high}}_x = \{i \in \mathcal{V}_x : \widetilde{A}_i \geq 1\}. \tag{13}$$

Thus, the lower-update half and top access quartile are defined within each example, rather than by pooling scores across examples. The reported overlap is the fraction of high-access tokens that also belong to the low-update set, computed from the full visual-token sets:

$$\text{Overlap} = \frac{\sum_x |\mathcal{V}^{\text{low}}_x \cap \mathcal{V}^{\text{high}}_x|}{\sum_x |\mathcal{V}^{\text{high}}_x|}. \tag{14}$$

This is a token-weighted conditional proportion, rather than an unweighted average of per-example percentages.

B Complete Subtask Results

For completeness, we report all available category-level results for seven benchmarks: MLVU Dev, VideoMME, TOMATO, LongVideoBench, MVBench, VSI-Bench, and TimeScope. Both improvements and regressions are retained. Δ denotes AdaDPC minus Dense + SFT in percentage points. For each benchmark, the Overall score is computed according to that dataset's aggregation protocol.

Table 6: **Complete subtask results across seven benchmarks.** All available category-level results are reported without filtering.

MLVU Dev

Subtask	Dense+SFT	AdaDPC	Δ
Count	30.12	**60.20**	+30.08
Order	53.76	**65.60**	+11.84
Topic Reasoning	**82.55**	81.70	−0.85
Plot QA	70.65	**72.40**	+1.75
Anomaly Recognition	**70.62**	59.00	−11.62
Ego	57.42	**61.90**	+4.48
Needle	77.47	**78.90**	+1.43
Overall	65.20	**69.70**	**+4.50**

TOMATO

Subtask	Dense+SFT	AdaDPC	Δ
Direction	22.02	**22.08**	+0.06
Count	34.24	**34.59**	+0.35
Rotation	21.66	**24.13**	+2.47
Shape & Trend	26.24	**27.80**	+1.56
Velocity & Frequency	13.06	**20.48**	+7.42
Visual Cues	32.11	**34.29**	+2.18
Overall	24.20	**26.14**	**+1.94**

LongVideoBench

Subtask	Dense+SFT	AdaDPC	Δ
S2E	**63.54**	60.22	−3.32
S2O	**54.89**	54.17	−0.72
S2A	59.31	**68.18**	+8.87
E2O	65.64	**67.84**	+2.20
O2E	63.47	**64.27**	+0.80
T2E	**57.95**	54.89	−3.06
T2O	**57.16**	53.68	−3.48
T2A	49.85	**56.40**	+6.55
E3E	56.46	**69.15**	+12.69
O3O	46.43	**57.58**	+11.15
SSS	35.06	**41.24**	+6.18
T3E	47.27	**54.79**	+7.52
T3O	53.36	**58.11**	+4.75
SOS	**63.38**	58.02	−5.36
TOS	**40.42**	39.73	−0.69
SAA	45.17	**50.00**	+4.83
TAA	**51.61**	48.78	−2.83
Overall	53.60	**56.40**	**+2.80**

VSI-Bench

Subtask	Dense+SFT	AdaDPC	Δ
Object Counting	24.80	**36.91**	+12.11
Absolute Distance	**29.07**	24.65	−4.42
Object Size	21.11	**37.01**	+15.90
Room Size	**30.57**	4.54	−26.03
Relative Distance	41.47	**42.47**	+1.00
Relative Direction	47.20	**47.57**	+0.37
Appearance Order	35.36	**43.38**	+8.02
Route Planning	**35.22**	34.67	−0.55
Overall	33.10	**33.90**	**+0.80**

VideoMME

Subtask	Dense+SFT	AdaDPC	Δ
Temporal Perception	61.90	**66.02**	+4.12
Spatial Perception	66.78	**70.93**	+4.15
Attribute Perception	73.04	**74.89**	+1.85
Action Recognition	59.29	**62.54**	+3.25
Object Recognition	63.80	**66.10**	+2.30
OCR	73.08	**74.66**	+1.58
Counting	37.07	**43.10**	+6.03
Temporal Reasoning	40.64	**46.89**	+6.25
Spatial Reasoning	78.62	**79.13**	+0.51
Action Reasoning	**54.05**	52.49	−1.56
Object Reasoning	57.73	**60.03**	+2.30
Information Synopsis	**77.83**	77.65	−0.18
Overall	60.30	**62.60**	**+2.30**

TimeScope

Subtask	Dense+SFT	AdaDPC	Δ
Temporal	70.13	**76.20**	+6.07
QA	79.53	**83.40**	+3.87
OCR	82.53	**86.20**	+3.67
Overall	77.40	**81.93**	**+4.53**

MVBench

Subtask	Dense+SFT	AdaDPC	Δ
Action Antonym	**81.15**	79.00	−2.15
Action Count	57.15	**65.50**	+8.35
Action Localization	44.15	**58.00**	+13.85
Action Prediction	**69.15**	64.00	−5.15
Action Sequence	**76.15**	72.50	−3.65
Character Order	**77.15**	73.50	−3.65
Counterfactual Inference	**74.65**	74.00	−0.65
Egocentric Navigation	38.15	**43.00**	+4.85
Episodic Reasoning	**54.65**	52.00	−2.65
Fine-grained Action	45.65	**46.00**	+0.35
Fine-grained Pose	57.15	**62.50**	+5.35
Moving Attribute	90.15	**93.00**	+2.85
Moving Count	64.15	**73.50**	+9.35
Moving Direction	**47.15**	41.00	−6.15
Object Existence	86.15	**92.00**	+5.85
Object Interaction	75.15	**78.00**	+2.85
Object Shuffle	33.15	**39.50**	+6.35
Scene Transition	90.15	**92.50**	+2.35
State Change	55.65	**57.00**	+1.35
Unexpected Action	**81.15**	80.50	−0.65
Overall	64.90	**66.85**	**+1.95**

C Implementation Details

C.1 Training Configuration

Table 7 summarizes the configuration of AdaDPC. We use Qwen2.5-VL-3B as the backbone and organize visual tokens into 2D spatial blocks of width 8 and height 4. The refinement budget follows a shifted cosine schedule with $\beta = 0.5$ and a minimum ratio of 0.2. Cross-Layer Priority Feedback uses a feedback ratio of 0.25 relative to the next layer's refinement budget. Both Router priorities and feedback relevance use maximum reduction within each block.

Our post-training data combines LLaVA-Video-178K, 77K timestamp-grounded samples from DiDeMo and ActivityNet Captions, and 118K samples from VideoMarathon. These sources support general video understanding, temporally localized interpretation, and long-video reasoning, respectively. We use a learning rate of 2×10^{-6} and a global batch size of 256. For the controlled architectural comparison, Dense + SFT, SRP, and SRP + QGA share the same post-training data, optimization steps, learning rate, and global batch size.

C.2 Pseudocode

Algorithm 1 summarizes computation at a routed decoder layer $\ell \in \{1, \ldots, L\}$. The position sets follow the main text. CausalAttn and GatedCrossAttn implement $\mathrm{A}^{\ell}_{\mathrm{SRP}}$ and $\mathrm{A}^{\ell}_{\mathrm{QGA}}$, respectively. Attention operators include their output projections, and all gathers preserve the original sequence order. Top-K returns block indices after mapping to the fixed spatial partition. Incoming feedback is empty at $\ell = 1$; outgoing feedback is used only for $\ell < L$.

Table 7: **Configuration of AdaDPC.** Block dimensions are specified as width × height in visual token-grid units.

Hyperparameter	Value
Model and Routing	
Backbone	Qwen2.5-VL-3B
Spatial block shape	8×4
Tokens per block	32
Routed decoder layers	All L layers
Refinement budget schedule	Cosine decay
Shift factor β	0.5
Minimum refinement ratio	0.2
Priority feedback ratio λ	0.25
Router token-to-block reduction	Maximum
Feedback token-to-block reduction	Maximum
Training Configuration	
Post-training data scale	$\sim$ 373K
Global batch size	256
Learning rate	2×10^{-6}
Training input FPS	2
Visual Context Length	
Main and architectural comparisons	32K visual tokens
Routing and allocation ablations	8K visual tokens

Table 8: **Frame-sampling settings for evaluation.**

Benchmark	FPS	Max Frames
MLVU Dev	2	512
VideoMME	2	512
LongVideoBench	2	512
LVBench	2	512
TimeScope	2	512
MVBench	4	128
VSI-Bench	2	256
TOMATO	2	256

Algorithm 1 AdaDPC Routed-Layer Computation

Require: Layer input $\mathbf{H}^{\ell}$, spatial blocks $\{\mathcal{B}_j\}_{j=1}^{N_b}$
Require: Position sets $\mathcal{V}, \mathcal{T}, \mathcal{T}_{\mathrm{Q}}, \mathcal{Q}_{\mathrm{fb}}$; budgets $K_{\ell}, K_{\ell+1}$ when applicable
Require: Incoming priority blocks $\mathcal{P}^{\ell-1}$
Ensure: Updated sequence $\mathbf{H}^{\ell+1}$ and priority blocks $\mathcal{P}^{\ell}$

Project $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ once from $\mathbf{H}^{\ell}$; apply the original positional encoding
$\rho_i^{\ell} \leftarrow \mathrm{Router}^{\ell}(\mathbf{H}_i^{\ell}), s_i^{\ell} \leftarrow \tanh(\rho_i^{\ell})$ for $i \in \mathcal{V}$
$R_j^{\ell} \leftarrow \max_{i \in \mathcal{B}_j} \rho_i^{\ell}$ for each block j
$\mathcal{G}_{\mathrm{S}}^{\ell} \leftarrow \mathcal{P}^{\ell-1} \cup \mathrm{TopK}_{j \notin \mathcal{P}^{\ell-1}}(R_j^{\ell}, K_{\ell} - |\mathcal{P}^{\ell-1}|)$
$\mathcal{G}_{\mathrm{Q}}^{\ell} \leftarrow \{1, \ldots, N_b\} \setminus \mathcal{G}_{\mathrm{S}}^{\ell}$
$\mathcal{V}_{\mathrm{S}}^{\ell} \leftarrow \bigcup_{j \in \mathcal{G}_{\mathrm{S}}^{\ell}} \mathcal{B}_j, \mathcal{V}_{\mathrm{Q}}^{\ell} \leftarrow \mathcal{V} \setminus \mathcal{V}_{\mathrm{S}}^{\ell}$
$\mathcal{I}_{\mathrm{S}}^{\ell} \leftarrow \mathrm{Sort}(\mathcal{V}_{\mathrm{S}}^{\ell} \cup \mathcal{T})$
$\mathbf{U}^{\ell}_{\mathcal{I}_{\mathrm{S}}^{\ell}} \leftarrow \mathbf{H}^{\ell}_{\mathcal{I}_{\mathrm{S}}^{\ell}} + \mathrm{CausalAttn}(\mathbf{Q}_{\mathcal{I}_{\mathrm{S}}^{\ell}}, \mathbf{K}_{\mathcal{I}_{\mathrm{S}}^{\ell}}, \mathbf{V}_{\mathcal{I}_{\mathrm{S}}^{\ell}})$
$\mathbf{C}^{\ell} \leftarrow \mathrm{GatedCrossAttn}(\mathbf{Q}_{\mathcal{T}_{\mathrm{Q}}}, \mathbf{K}_{\mathcal{V}_{\mathrm{Q}}^{\ell}}, \mathbf{V}_{\mathcal{V}_{\mathrm{Q}}^{\ell}}; \mathbf{g}^{\ell})$
$\widehat{\mathbf{H}}^{\ell}_{\mathcal{I}_{\mathrm{S}}^{\ell}} \leftarrow \mathbf{U}^{\ell}_{\mathcal{I}_{\mathrm{S}}^{\ell}}$
$\widehat{\mathbf{H}}^{\ell}_{\mathcal{T}_{\mathrm{Q}}} \leftarrow \mathbf{U}^{\ell}_{\mathcal{T}_{\mathrm{Q}}} + \mathbf{C}^{\ell}$
$\mathbf{f}_i^{\ell} \leftarrow \mathrm{F}^{\ell}(\widehat{\mathbf{H}}_i^{\ell})$ for $i \in \mathcal{I}_{\mathrm{S}}^{\ell}$
$\mathbf{H}_i^{\ell+1} \leftarrow \widehat{\mathbf{H}}_i^{\ell} + (1 + s_i^{\ell})\mathbf{f}_i^{\ell}$ for $i \in \mathcal{V}_{\mathrm{S}}^{\ell}$
$\mathbf{H}_i^{\ell+1} \leftarrow (1 + s_i^{\ell})\mathbf{H}_i^{\ell}$ for $i \in \mathcal{V}_{\mathrm{Q}}^{\ell}$
$\mathbf{H}_i^{\ell+1} \leftarrow \widehat{\mathbf{H}}_i^{\ell} + \mathbf{f}_i^{\ell}$ for $i \in \mathcal{T}$
$\mathcal{P}^{\ell} \leftarrow \varnothing$
if $\ell < L$ **then**
    Average QGA attention over heads and $\mathcal{Q}_{\mathrm{fb}}$ to obtain token scores a_i^{ℓ}
    $\eta_j^{\ell} \leftarrow \max_{i \in \mathcal{B}_j} a_i^{\ell}$ for $j \in \mathcal{G}_{\mathrm{Q}}^{\ell}$
    $m_{\ell} \leftarrow \min\{\lceil \lambda K_{\ell+1} \rceil, |\mathcal{G}_{\mathrm{Q}}^{\ell}|\}$
    $\mathcal{P}^{\ell} \leftarrow \mathrm{TopK}_{j \in \mathcal{G}_{\mathrm{Q}}^{\ell}}(\eta_j^{\ell}, m_{\ell})$
end if
return $\mathbf{H}^{\ell+1}, \mathcal{P}^{\ell}$

C.3 Detail of FLOPs Calculation

We detail the main matrix multiplications in attention and the FFN. For the prefill computation of one routed layer, let S denote the complete sequence length and D the model dimension. Let $S_{\text{act}} = |\mathcal{V}_{\text{S}}^{\ell}| + |\mathcal{T}|$ denote the active sequence length, including selected visual tokens and all valid non-visual tokens. We use $S_{\text{rem}} = |\mathcal{V}_{\text{Q}}^{\ell}|$ for the remaining visual tokens and $S_q = |\mathcal{T}_{\text{Q}}|$ for the QGA queries.

FLOPs Analysis for the Selective Refinement Path. Standard dense causal attention computes Query-Key and Attention-Value products over $S(S+1)/2$ valid position pairs. Their combined cost is

$$\begin{aligned}\text{FLOPs}_{\text{Dense-Attn}} &= 2 \times \frac{S(S+1)}{2} \times D + 2 \times \frac{S(S+1)}{2} \times D \\ &= 2S(S+1)D \approx 2S^2D.\end{aligned} \tag{15}$$

The Selective Refinement Path applies causal attention to the active sequence, giving

$$\text{FLOPs}_{\text{SRP-Attn}} = 2S_{\text{act}}(S_{\text{act}}+1)D \approx 2S_{\text{act}}^2 D. \tag{16}$$

The corresponding output projection costs $2S_{\text{act}}D^2$, compared with $2SD^2$ for the dense sequence. Full-sequence Q/K/V projections are shared by the two paths. Reducing the active sequence therefore decreases the attention interaction cost and the number of outputs requiring projection, without constructing separate attention windows for individual blocks.

FLOPs Analysis for the Query-Guided Access Path. The Query-Guided Access Path computes attention between S_q text queries and S_{rem} visual keys and values. Its Query-Key and Attention-Value products cost

$$\begin{aligned}\text{FLOPs}_{\text{QGA-Attn}} &= 2S_qS_{\text{rem}}D + 2S_qS_{\text{rem}}D \\ &= 4S_qS_{\text{rem}}D.\end{aligned} \tag{17}$$

Including the additional output projection, the cost after shared Q/K/V projection is

$$\text{FLOPs}_{\text{QGA|QKV}} = 4S_qS_{\text{rem}}D + 2S_qD^2. \tag{18}$$

For a fixed number of text queries, this interaction grows linearly with the number of remaining visual tokens. When $S_q \ll S_{\text{rem}}$, it provides access to those visual states without computing visual-to-visual attention among them.

FLOPs Analysis for the FFN. Let D_{ff} denote the intermediate FFN dimension. The backbone uses a gated FFN with gate, up, and down projections. For the dense sequence, these three matrix multiplications cost

$$\begin{aligned}\text{FLOPs}_{\text{Dense-FFN}} &= 2SDD_{\text{ff}} + 2SDD_{\text{ff}} + 2SD_{\text{ff}}D \\ &= 6SDD_{\text{ff}}.\end{aligned} \tag{19}$$

AdaDPC applies this computation only to selected visual tokens and valid non-visual tokens. Its FFN cost is therefore

$$\text{FLOPs}_{\text{AdaDPC-FFN}} = 6S_{\text{act}}DD_{\text{ff}}. \tag{20}$$

Thus, the FFN matrix multiplication cost scales with S_{act}/S. The remaining visual tokens retain their states through score-aware propagation and contribute through query-guided access. Text states integrate the two attention outputs before a single FFN update, so QGA does not require an additional text FFN.

D Evaluation Settings

We evaluate AdaDPC on eight video-language benchmarks. Long-form video understanding is assessed using MLVU Dev, VideoMME, LongVideoBench, LVBench, and TimeScope. MVBench evaluates short-form video understanding, VSI-Bench evaluates spatial perception, and TOMATO evaluates temporal reasoning. The main benchmark evaluation and architectural comparison use a 32K visual-token context. The routing-granularity and path-allocation ablations use an 8K visual-token context.

Table 8 organizes the benchmark-specific sampling settings. FPS denotes the sampling rate, and the maximum frame count limits the number of frames provided for each video.