# TRIAGE ATTENTION: CORE-PROTECTED SPARSE ATTENTION FOR LONG-VIDEO VISION-LANGUAGE MODELS

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

## ABSTRACT

Attention accounts for a substantial share of prefill computation in long-video vision-language models. Sparse attention can reduce this cost, but inappropriate block selection may discard critical information. Existing selection strategies typically apply a uniform filtering rule without explicitly distinguishing the decision requirements of a high-confidence core from those of uncertain candidates. We introduce TRIAGE ATTENTION, a training-free sparse attention method that assigns distinct roles to two estimates of block importance. A head-specific score identifies a protected core and a larger candidate set, while a group-level score filters only candidates outside the core. This asymmetric rule preserves core support while restricting the auxiliary estimator's authority to reject blocks, enabling a favorable quality–cost trade-off. Across three backbones evaluated on the complete Video-MME and LongVideoBench benchmarks, Triage achieves accuracy close to dense attention. On Video-MME with Qwen3-VL-8B, it achieves 69.56% accuracy versus 69.44% for dense attention, and on LongVideoBench with the same backbone it reduces attention computation by $4.65\times$ relative to dense attention. A prespecified sweep of 15 configurations further demonstrates a competitive quality–cost trade-off against the evaluated ProxyAttn and XAttention configurations.

## 1 INTRODUCTION

Vision-language models (VLMs) extend language-based interaction to visual content, enabling image description, visual question answering, and instruction-following assistants (Li et al., 2023; Liu et al., 2023). Their extension to video supports conversational understanding of events and activities (Lin et al., 2024), while long-video understanding requires retrieving relevant evidence and reasoning over temporally distributed content (Fu et al., 2024; Wu et al., 2024). These capabilities make efficient video-language inference important for interactive assistants and analysis of lengthy recordings. However, processing long videos can be slow: sampled frames, temporal markers, and text form long language-side input sequences, and dense self-attention incurs quadratic computation in sequence length during prefill, delaying the first generated response (Li et al., 2025). Block-sparse attention can reduce this cost by selecting which historical key blocks each query block attends to. The challenge is to preserve useful visual and temporal evidence without allowing selection itself to consume the computation it seeks to save.

Existing training-free sparse attention methods estimate attention structure from compressed query–key representations or sampled interactions. XAttention uses antidiagonal scoring to estimate block importance (Xu et al., 2025), whereas ProxyAttn combines representative-head scores with head-specific budgets (Wang et al., 2026b). FlexPrefill, FlashPrefill, and MInference adapt support through context-aware selection, pattern discovery, or dynamic sparse patterns (Lai et al., 2025; Fan et al., 2026; Jiang et al., 2024). However, detailed block-importance estimation can itself incur substantial computation, offsetting part of the savings from sparse attention execution. Coarser estimates reduce scoring cost but may discard head-specific information and underestimate useful blocks. This trade-off motivates reducing estimation complexity where simpler scores already yield similar retention decisions.



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.

We observe substantial agreement between XAttention and simple mean-pooled Q/K scoring in the blocks selected up to cumulative mass 0.5. On one 88,613-token input across five Qwen3-VL layers, pooling recovers an average of 92.9% of the blocks selected by XAttention, while XAttention covers 95.6% of the pooling-selected blocks, without forced structural keeps or a minimum-count floor (Table 1). At stride/pooling width 16, XAttention's antidiagonal dot-product scoring requires approximately $16\times$ the arithmetic of pooled-Q/K dot-product scoring, excluding pooling, softmax, and selection overheads. This agreement suggests that a simpler selector can recover much of the mass-0.5 support identified by a more expensive estimator. Appendix A provides the experimental protocol and threshold sweep.

Table 1: Core-set coverage at mass 0.5 (%). X: XAttention; P: pooled Q/K. One 88,613-token input, all 32 heads; no forced keeps or floor. Row filtering and averaging are specified in Appendix A.

| Layer | $\lvert X \cap P\rvert/\lvert X\rvert$ | $\lvert X \cap P\rvert/\lvert P\rvert$ |
|---|---|---|
| 6 | 90.7 | 97.3 |
| 12 | 94.3 | 96.6 |
| 18 | 92.7 | 92.8 |
| 24 | 92.6 | 94.2 |
| 30 | 94.5 | 96.9 |
| Mean | 92.9 | 95.6 |

We propose TRIAGE ATTENTION, a training-free sparse attention method that combines head-specific salience estimation with group-level boundary filtering. Mean-pooled Q/K scores define a protected core and a larger candidate set, while group-level scores filter only candidates outside the core. This asymmetric selection rule preserves core support and removes candidates that lack auxiliary agreement. Evaluations across three VLMs on the complete Video-MME and LongVideoBench benchmarks show accuracy close to dense attention with reduced attention computation. On Video-MME, Qwen3-VL-8B achieves 69.56% accuracy, matching dense attention within three questions, and on LongVideoBench the same backbone reduces attention computation by $4.65\times$ relative to dense attention. Prespecified parameter sweeps and full-model ablations further characterize the quality and computational cost of the selection rule.

Our contributions are:

- We propose TRIAGE ATTENTION, a training-free sparse attention method that accelerates attention computation during long-video VLM inference.
- We introduce a core-protection mechanism that uses low-cost head-specific scoring to distinguish a protected core from boundary candidates and selectively allocates computation to the latter through group-level filtering.
- Experiments across three VLMs on Video-MME and LongVideoBench demonstrate the lowest attention computation time among the evaluated baselines while maintaining accuracy close to dense attention.

## 2 RELATED WORK

**Vision-language models.** Recent VLMs integrate visual representations with language models to support visual question answering, document understanding, and video reasoning. BLIP-2 and LLaVA established effective visual-language alignment and instruction-tuning approaches (Li et al., 2023; Liu et al., 2023), while subsequent models extend these capabilities to higher-resolution inputs and longer temporal contexts. Qwen2.5-VL introduces dynamic-resolution processing and absolute temporal encoding for image and video understanding (Bai et al., 2025b). Qwen3-VL further develops multimodal perception, reasoning, and long-context video understanding (Bai et al., 2025a). MiniCPM-V 4.5 emphasizes efficient multimodal modeling through its architecture and training recipe, including spatiotemporal compression with a 3D resampler (Yu et al., 2025). Despite improvements in visual representation and compression, long videos can still produce lengthy language-side sequences. Dense attention over these sequences incurs quadratic prefill computation, making attention a persistent inference bottleneck as temporal coverage and retained visual detail increase (Li et al., 2025).

**Sparse attention methods.** Training-free sparse attention reduces prefill computation by estimating attention structure and restricting execution to selected interactions. MInference exploits dynamic sparse patterns, while MMInference adapts sparse computation to multimodal structure (Jiang

et al., 2024; Li et al., 2025). XAttention estimates block importance through antidiagonal scoring (Xu et al., 2025); ProxyAttn uses representative heads and head-specific budgets (Wang et al., 2026b). FlexPrefill adapts selection to the input context, FlashPrefill combines pattern discovery with thresholding, and Prism incorporates spectral information into block selection (Lai et al., 2025; Fan et al., 2026; Wang et al., 2026a). These approaches share a dependence on approximate importance estimates or attention patterns to allocate sparse computation. Their effectiveness is constrained by the trade-off between estimation cost and selection fidelity: detailed scoring can offset kernel savings, whereas coarse estimates can omit useful interactions. Larger retention budgets reduce the extent of pruning but increase execution cost. This work addresses that trade-off through low-cost head-specific scoring and selective use of auxiliary evidence, rather than applying auxiliary filtering indiscriminately across all candidates.

## 3 Problem Setup

Block-sparse prefill selects a subset of causally visible key blocks for each query block. The selection objective balances computational cost with retention of task-relevant evidence in long multimodal sequences.

Let $Q, K, V$ be the query, key, and value tensors, $T$ the sequence length, and $d$ the head dimension. Selection uses Q/K after normalization and RoPE. Tokens are partitioned into blocks of $B = 128$, giving $N = \lceil T/B \rceil$ blocks. For query block $r$, visible key blocks form $\mathcal{V} = \{0, \ldots, r\}$. Indices start at zero. Grouped-query attention uses the associated KV head, with the mapping specified in Appendix B.

Below we fix one layer, one query head, and one query block, omitting these shared indices. Thus $a_j$ scores key block $j$, and $M \subseteq \mathcal{V}$ is the set of retained blocks. The procedure runs independently at each language layer without changing model weights.

## 4 Triage Attention

Triage separates core identification from boundary filtering. Figure 1 presents an overview of the method. A head-specific score defines a protected core and a larger candidate set, while a group-level score and a per-head budget define auxiliary acceptance. The selection rule retains the core unconditionally and filters only the non-core candidates according to auxiliary acceptance. The attention kernel then computes the output using the original Q/K/V over the selected support. The following subsections detail the implementation of each component.

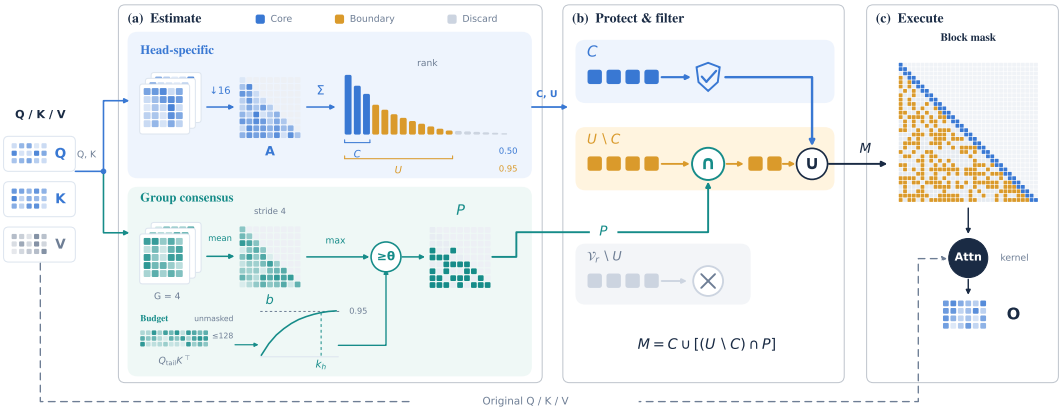


Figure 1: Triage pipeline. (a) Head-specific scores define core $C$ and candidates $U$; group-consensus scores and a per-head budget define acceptance set $P$. (b) Core blocks always survive, boundary blocks require acceptance by $P$, and blocks outside $U$ are discarded. (c) The attention kernel uses original Q/K/V over the selected support to produce $O$. This is a schematic, not a measured attention map.

### 4.1 HEAD-SPECIFIC SALIENCE ESTIMATION

The primary estimator computes a separate key-block ranking for each query head and each query block. Q and K are independently averaged over groups of 16 consecutive tokens to obtain pooled tensors $\bar{Q}$ and $\bar{K}$. The final group is zero-padded and retains a denominator of 16. For each query head and its associated KV head, scaled dot products between the pooled tensors are normalized by softmax over causally visible pooled key positions, yielding the low-resolution attention matrix $A$. Its entry $A_{uv}$ is the attention weight from pooled query position $u$ to pooled key position $v$.

Each complete 128-token block contains eight pooled positions. Let $I_r$ denote the pooled query positions belonging to query block $r$, and $I_j$ the pooled key positions belonging to key block $j$; only valid pooled positions are included in a partial final block. For a fixed head, the importance score of key block $j$ relative to query block $r$ is

$$a_{rj} = \sum_{u \in I_r} \sum_{v \in I_j} A_{uv}. \tag{1}$$

Each score therefore corresponds to a two-dimensional attention block linking a query block to a key block. Scores are ranked along the key-block dimension independently for every query block and every head. All query positions are retained; sparsification restricts the key/value blocks accessed to compute their outputs. These pooled scores estimate block importance rather than exact dense-attention mass.

**Core and Candidate Set Construction.** For each query head and query block $r$, causally visible key blocks are ranked by decreasing importance score $a_{rj}$, with ties resolved by increasing block index. Two prefixes of this ordering are selected using cumulative score fractions $c = 0.5$ and $u = 0.95$. Each prefix includes the first block that reaches or exceeds its target fraction of the total score over visible key blocks. The smaller prefix identifies the mass-based core, while the larger prefix defines the initial candidate set.

Both prefixes are augmented with a ranked floor $L$, consisting of the top $\min(16, r+1)$ key blocks, and structural support $F$, consisting of the first and current key blocks. For the final two query blocks, $F$ instead contains all causally visible key blocks. The augmented sets are denoted by $C$ for the protected core and $U$ for the candidates, satisfying

$$C \subseteq U \subseteq \mathcal{V}. \tag{2}$$

Here $\mathcal{V}$ is the visible key-block set for the fixed query block. The boundary $U \setminus C$ contains non-core candidates subject to subsequent group-level filtering; the tail $\mathcal{T} = \mathcal{V} \setminus U$ is excluded from final selection. The ranked floor retains blocks by importance rather than temporal proximity, and all visible blocks are retained when fewer than 16 are available. This construction is applied separately to every head and query block. Appendix B provides the general prefix definition and exact boundary conventions.

### 4.2 GROUP-CONSENSUS BOUNDARY ESTIMATION

The Group consensus and Budget branches in Figure 1(a) jointly construct the auxiliary acceptance set $P$. Group-level scores determine a shared block ranking, while a head-specific budget determines the acceptance range. The resulting set provides the filtering criterion for non-core candidates without redefining the protected core.

**Group-Level Block Scoring.** Query heads and unrepeated KV heads are each partitioned into four equal, contiguous groups. Within each group, Q and K are averaged across heads and sampled every fourth token along the sequence. Scaled dot products followed by softmax over causally visible sampled keys produce a coarse attention map. For each query block $r$ and key block $j$, the maximum attention weight over their corresponding two-dimensional region defines the block score. Scores are normalized across key blocks within each query-block row, yielding $b_j$ for the fixed group and query block; blocks without valid causal entries receive zero scores. Unlike head-specific salience, which sums attention weights within each region, this estimator uses the maximum to emphasize strong sampled interactions. The ranking is computed separately for every query block and shared by the heads in its group. Exact grouping and numerical conventions are specified in Appendix B.

**Head-Specific Retention Budget.** For each query head, a trailing-query probe computes attention from its final at most 128 query positions to all keys of the associated KV head. The weights are averaged across the probe queries and then averaged within each key block to form a block-level distribution. A cumulative mass threshold $\gamma = 0.95$, together with a minimum budget of 16 blocks capped by the available block count, determines the requested count $k$. This budget is estimated separately for each head and reused across its query blocks. The probe is an unmasked scoring operation; the attention output remains causally masked. The exact budget construction and rounding conventions are given in Appendix C.

**Auxiliary Acceptance Set.** For each head and query block, the budget $k$ is applied to the corresponding group-level scores. Let $\theta$ denote the $k$-th largest score in the full query-block row, with causally invalid blocks assigned zero scores. The accepted visible key blocks form

$$P = \{j \in \mathcal{V} : b_j \geq \theta\}, \tag{3}$$

where $\mathcal{V}$ is the visible key-block set for the current query block. All blocks tied at the cutoff are retained, so the accepted count may exceed $k$. Heads in the same group share scores but can obtain different acceptance sets because their budgets differ. The notation $P_\gamma$ makes the budget threshold explicit when needed. The next subsection combines this acceptance set with the protected core and candidate set, as shown in Figure 1(b).

### 4.3 Protected-Boundary Selection and Attention Computation

For a fixed query head and query block, the protected core $C$, candidate set $U$, and auxiliary acceptance set $P$ determine the final support shown in Figure 1(b):

$$M = C \cup \big((U \setminus C) \cap P\big). \tag{4}$$

All core blocks are retained, while boundary candidates are retained only if accepted by $P$. Blocks outside $U$ are excluded, yielding $C \subseteq M \subseteq U$. This rule is applied independently to each head and query block.

As illustrated in Figure 1(c), attention outputs are computed from the original Q/K/V over causally visible key tokens within the blocks selected by $M$, with attention weights renormalized over the retained positions. All query positions are preserved, and the remaining model computation is unchanged. Appendix B gives the output and cost definitions; Appendix K visualizes the executed support.

## 5 Experiments

### 5.1 Setup

**Models and benchmarks.** We evaluate Qwen3-VL-8B-Instruct (Bai et al., 2025a), Qwen2.5-VL-7B-Instruct-ForVideo (Bai et al., 2025b), and MiniCPM-V-4.5 (Yu et al., 2025), preserving each backbone's native visual preprocessing, temporal packing, and multimodal position encoding. For Video-MME (Fu et al., 2024), we sample video frames at 0.5 fps and evaluate all 2,700 questions by exact match. For LongVideoBench (Wu et al., 2024), we evaluate the 1,337-question validation split using visual inputs at 1 fps with a maximum of 512 frames, following the integrated Prism evaluation configuration (Wang et al., 2026a).

**Baselines.** We compare dense attention with XAttention (Xu et al., 2025), ProxyAttn (Wang et al., 2026b), FlexPrefill (Lai et al., 2025), and Prism (Wang et al., 2026a). All methods replace only language-side causal prefill attention. Decoding and the remaining model computation are unchanged. We report the final materialized local configurations rather than constructor defaults.

**Metrics.** Task quality is measured by exact-match accuracy. The efficiency tables report accumulated CUDA selector and attention-kernel spans across all language layers, averaged over samples. These spans exclude the visual encoder, QKV projection, MLPs, decoding, and host preprocessing; they are not end-to-end latency. Full benchmark results use all questions. Controlled ablations use a frozen 540-question Video-MME subset with 180 videos, balanced over short, medium, and long durations. Unless explicitly stated, differences of a few questions are treated as point estimates rather than statistically significant gains.

Table 2: Video-MME results. Accuracy (%, ↑) and attention S+K time (s/question, ↓) on 2,700 questions. Bold denotes the best displayed value per column.

| | Qwen3-VL | | Qwen2.5-VL | | MiniCPM-V | |
|---|---|---|---|---|---|---|
| Method | Acc. | Time | Acc. | Time | Acc. | Time |
| Dense | 69.44 | 21.148 | 62.85 | 2.252 | 68.52 | 0.324 |
| XAttention | 69.37 | 9.009 | 61.52 | 1.459 | 67.81 | 0.245 |
| ProxyAttn | 69.41 | 7.462 | 63.00 | 1.375 | **68.48** | 0.267 |
| FlexPrefill | 68.81 | 6.014 | 62.81 | 1.208 | 68.37 | 0.337 |
| Prism | 69.52 | 7.233 | 62.89 | 1.196 | 68.44 | 0.288 |
| **Triage (ours)** | **69.56** | **4.302** | **63.04** | **0.859** | **68.48** | **0.183** |

Table 3: LongVideoBench results. Accuracy (%, ↑) and attention S+K time (s/question, ↓) on 1,337 questions. Bold denotes the best displayed value per column.

| | Qwen3-VL | | Qwen2.5-VL | | MiniCPM-V | |
|---|---|---|---|---|---|---|
| Method | Acc. | Time | Acc. | Time | Acc. | Time |
| Dense | 63.65 | 12.854 | 57.07 | 14.965 | 60.66 | 0.341 |
| XAttention | 63.72 | 5.722 | 57.22 | 8.707 | 60.81 | 0.255 |
| ProxyAttn | 63.65 | 4.716 | 57.04 | 8.702 | 60.88 | 0.279 |
| FlexPrefill | 63.28 | 3.860 | 57.14 | 7.068 | 60.81 | 0.354 |
| Prism | 63.05 | 4.753 | 56.39 | 6.973 | 60.88 | 0.370 |
| **Triage (ours)** | **64.17** | **2.763** | **57.52** | **5.128** | **61.03** | **0.189** |

**Implementation.** Measurements use NVIDIA H20 GPUs and BF16 attention; the current evaluation stack uses PyTorch 2.10 with CUDA 12.9 and Transformers 5.14.1. Models are not fine-tuned. Appendix D specifies the measurement boundaries and the scope of timing comparisons.

### 5.2 Main Results

Tables 2 and 3 report complete Video-MME and LongVideoBench evaluations across three VLMs. On Video-MME with Qwen3-VL-8B, Triage Attention achieves 69.56% accuracy, close to dense attention's 69.44%, while reducing attention time from 21.148 to 4.302 seconds per question, a $4.92\times$ speedup. On LongVideoBench, the same backbone reduces attention time from 12.854 to 2.763 seconds per question, a $4.65\times$ speedup. On Qwen3-VL, Triage attains the highest observed accuracy and the lowest attention computation time among the evaluated sparse attention methods on both benchmarks. These results demonstrate a favorable balance between accuracy and computational efficiency on this backbone.

The separate selector/kernel breakdown is provided in Figure 5, as detailed in Appendix D; the main tables report total attention time only.

### 5.3 Quality–Cost Frontier

Figure 2 compares five prespecified configurations per method on the same 540-question Video-MME subset using Qwen3-VL, yielding 15 evaluated configurations. Triage and ProxyAttn vary the auxiliary-budget threshold $\gamma$, while XAttention varies the selection threshold $\tau$ at stride 16. Triage's core and candidate thresholds and minimum-count floor remain fixed. Open circles identify the prespecified default configurations.

Multiple Triage configurations lie on the quality–cost frontier of the evaluated points. Reducing its auxiliary-budget threshold from 0.95 to 0.90 preserves 377 correct answers out of 540 while reducing attention computation time from 4.420 to 3.330 seconds per question, a 24.7% reduction. These results indicate that Triage offers competitive accuracy at reduced attention cost across multiple evaluated budget settings, rather than at a single operating point. Accuracy does not vary monotonically with the threshold; Appendix E reports all configurations.

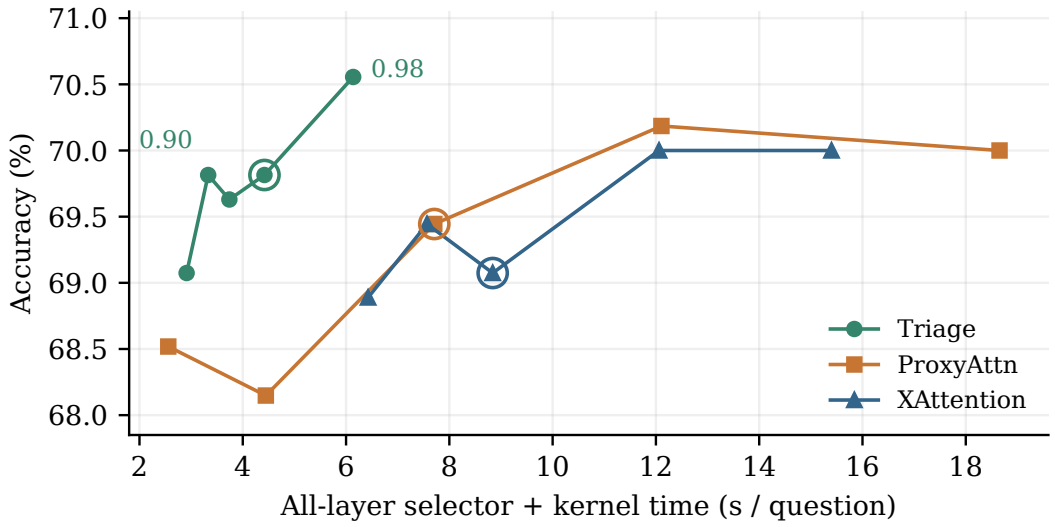


Figure 2: Quality–cost trade-off on Qwen3-VL. All 15 configurations use the same 540 Video-MME questions. Open circles mark the prespecified defaults: $\gamma = .95$ for Triage and ProxyAttn, and $\tau = .89$ for XAttention. Lines connect measured points, not interpolated results. The horizontal axis is attention S+K, not TTFT.

Table 4: Core-protection comparison on Qwen3-VL, all 2,700 Video-MME questions. Direct intersection varies the head-specific prefix threshold without additional core protection. Triage fixes candidate threshold $u = .95$ and auxiliary budget threshold $\gamma = .95$, varying only core threshold $c$. S+K is seconds per question under the same executor within this experiment; natural budgets are not matched.

| Selection | Correct | Accuracy (%) | S+K (s) |
|---|---|---|---|
| Direct intersection $R_{.80} \cap P_{.95}$ | 1860 | 68.89 | 3.785 |
| Direct intersection $R_{.95} \cap P_{.95}$ | 1868 | 69.19 | 5.551 |
| Triage $c = .50$ (ours) | 1878 | 69.56 | 5.913 |
| Triage $c = .60$ | 1878 | 69.56 | 6.145 |
| Triage $c = .70$ | 1875 | 69.44 | 6.542 |
| Triage $c = .80$ | 1876 | 69.48 | 7.297 |

### 5.4 Protection on the Complete Benchmark

Table 4 evaluates explicit core protection on all 2,700 Video-MME questions using Qwen3-VL and the same executor within this experiment. For a fixed head and query block, $R_t$ denotes the head-specific prefix reaching cumulative score fraction $t$, augmented with the ranked floor $L$ and structural support $F$ defined in Section 4. Thus, $C = R_c$ and $U = R_u$. The auxiliary acceptance set $P_\gamma$ uses the head-specific budget threshold $\gamma$. Direct intersection selects $R_t \cap P_{.95}$ with $t = .80$ or .95, retaining structural support but providing no additional core protection. Triage instead retains $R_c$ unconditionally and filters $R_u \setminus R_c$ through $P_{.95}$. Its four configurations fix $u = .95$ and $\gamma = .95$ and vary only $c$. The direct-intersection threshold $t$ therefore controls the entire head-specific set being filtered, whereas $c$ controls the protected portion of Triage's fixed candidate set.

At the same candidate threshold of .95, introducing core protection with $c = .50$ increases the correct-answer count from 1,868 to 1,878, while attention time increases from 5.551 to 5.913 seconds per question. Raising $c$ to .60, .70, or .80 provides no accuracy improvement over .50 and increases computation. These results characterize the quality and cost of core protection under natural budgets, rather than an advantage at matched computation. Appendix F complements this task-level comparison with local core/boundary deletion diagnostics. The historical score-file verification issue for the $R_{.95} \cap P_{.95}$ run is documented in Appendix J.

Table 5: Full-model selection ablations on Qwen3-VL using the same 540 Video-MME questions. Attention time is S+K in seconds per question. Natural budgets are not matched.

| Policy | Correct / 540 | Attention time |
|---|---|---|
| **Triage (ours)** | 381 | 4.433 |
| Head-specific .85 | 377 | 4.629 |
| Head-specific .95 | 381 | 7.535 |
| Group score only | 376 | 5.244 |

### 5.5 Full-Model Component Ablations

Table 5 evaluates the combination of head-specific scoring and group-level filtering across all language layers of Qwen3-VL on 540 Video-MME questions. The alternatives use head-specific selection at cumulative mass thresholds of 0.85 and 0.95 with the ranked floor and structural support retained, or group-only selection with support $P \cup F$. Triage achieves 381 correct answers, matching head-specific .95 while reducing attention computation time from 7.535 to 4.433 seconds per question, a 41.2% reduction. It also exceeds head-specific .85 and group-only selection by four and five correct answers, respectively, at lower attention cost. These results support the accuracy and efficiency of the combined selection rule under the evaluated natural-budget configurations. Appendix G reports additional policy ablations on the LongVideoBench transfer subset, while Appendix H compares rankings under frozen per-row block budgets.

### 5.6 Parameter Sensitivity

Figure 3 examines the core mass threshold $c$, candidate mass threshold $u$, and ranked retention floor on 540 Video-MME questions using Qwen3-VL. Each configuration changes one parameter from the default setting of $c = .5$, $u = .95$, and a 16-block floor. The default accuracy is taken from a separate rerun, while the other points retain their historical measurements; the comparison therefore describes observed configurations rather than isolating parameter effects under a common execution environment.

The displayed core- and candidate-threshold settings differ by at most three correct answers, whereas the floor settings span seven. The 16-block default attains 381 correct answers, and increasing the floor to 32 yields the same count at greater recorded attention cost. Among the evaluated configurations, the larger floor therefore provides no observed accuracy advantage over the default. Appendix I provides the complete accuracy and timing values in Table 10, together with the provenance of the default measurement.

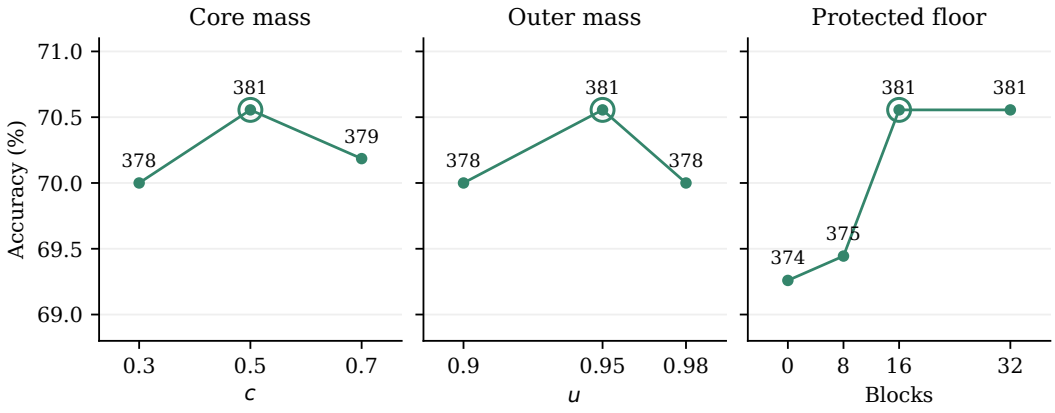


Figure 3: Hyperparameter sensitivity on 540 Video-MME questions. Each panel compares settings of one parameter with the others fixed at their default values. The common vertical scale shows absolute accuracy; labels give correct-answer counts. Rings identify the default configuration. Lines connect measured settings, not fitted trends. No repeated-run uncertainty is estimated.

### 5.7 Practical End-to-End Efficiency

Figure 4 evaluates the translation of attention savings into first-token latency on Qwen3-VL. Dense, XAttention, ProxyAttn, and Triage are measured on the same 24-video Video-MME subset using one H20. Host time to first token includes video preprocessing and the surrounding model computation. Each sample is measured once from an empty KV cache without warm-up.

Triage achieves the lowest median host TTFT among the four methods, reducing latency from 64.21 seconds for Dense to 36.06 seconds, a 1.78× speedup. It also provides 1.24× and 1.12× speedups over XAttention and ProxyAttn, respectively. These results demonstrate an observed first-response benefit beyond the attention kernel on this workload. The single-pass protocol does not estimate repeated-run variability. Appendix D distinguishes the attention-span measurements from this host-level latency metric.

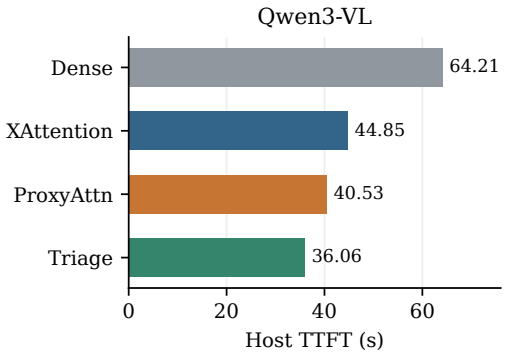


Figure 4: Median host TTFT on Qwen3-VL for 24 Video-MME videos. Values are seconds; lower is better. Each sample is measured once without warm-up.

## 6 Discussion and Limitations

**Design implications.** Triage distinguishes the identification of core support from the filtering of non-core candidates. Low-cost head-specific scoring defines the support to preserve, while group-level evidence constrains the remaining computation. The task evaluations and parameter studies support this asymmetric design as a practical approach to balancing accuracy and attention cost. Its benefit is assessed at the model level rather than by requiring fewer retained blocks in every head or layer. End-to-end gains additionally depend on the share of attention in the overall workload.

**Model scale and future evaluation.** Our evaluation covers three VLMs with approximately 7 to 8 billion parameters; substantially larger backbones have not yet been tested. Further experiments are needed to assess whether the observed accuracy and efficiency gains extend to larger models and whether their attention patterns require different core thresholds or retention budgets. Such evaluation would establish the applicability of the selection rule beyond the model scales studied here.

## 7 Conclusion

We introduced Triage Attention, a training-free sparse attention method for long-video VLMs. The method combines low-cost head-specific core identification with group-level filtering of non-core candidates, preserving core support while reducing attention computation. Evaluations on Video-MME and LongVideoBench show accuracy close to dense attention across three backbones. On Qwen3-VL-8B, Triage achieves 69.56% accuracy on Video-MME and a $4.65\times$ attention-computation reduction over dense attention on LongVideoBench, with a separate 24-video system evaluation demonstrating a $1.78\times$ median host-TTFT speedup. These results support core protection and selective boundary filtering as a practical design for efficient long-video inference.

The central contribution is the asymmetric use of importance estimates. Head-specific scoring determines the support that is retained unconditionally, while group-level evidence constrains additional candidates without overriding that support. The resulting rule separates core preservation from boundary pruning rather than requiring either estimator to determine every retention decision alone. The component comparisons and parameter studies characterize the accuracy and computation associated with this combination, while the prespecified budget sweep demonstrates competitive performance at multiple operating points.

## AI USE STATEMENT

We used ChatGPT to assist with implementing and debugging and to improve the clarity of the manuscript. No generative AI tools were used for other research tasks. The authors take full responsibility for the manuscript, code, and reported results.

## REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report, 2025a.

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, 2025b.

Qihang Fan, Huaibo Huang, Zhiying Wu, Juqiu Wang, Bingning Wang, and Ran He. FlashPrefill: Instantaneous pattern discovery and thresholding for ultra-fast long-context prefilling, 2026.

Chaoyou Fu, Yuhan Dai, Yondong 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, Rongrong Ji, and Xing Sun. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis, 2024.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. MInference 1.0: Accelerating pre-filling for long-context LLMs via dynamic sparse attention. In *Advances in Neural Information Processing Systems*, 2024.

Xunhao Lai, Jianqiao Lu, Yao Luo, Yiyuan Ma, and Xun Zhou. FlexPrefill: A context-aware sparse attention mechanism for efficient long-sequence inference. In *International Conference on Learning Representations*, 2025.

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 *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742, 2023. URL https://proceedings.mlr.press/v202/li23q.html.

Yucheng Li, Huiqiang Jiang, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Amir H. Abdi, Dongsheng Li, Jianfeng Gao, Yuqing Yang, and Lili Qiu. MMInference: Accelerating pre-filling for long-context VLMs via modality-aware permutation sparse attention, 2025.

Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-LLaVA: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024. URL https://aclanthology.org/2024.emnlp-main.342/.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2304.08485.

Xinghao Wang, Pengyu Wang, Xiaoran Liu, Fangxu Liu, Jason Chu, Kai Song, and Xipeng Qiu. Prism: Spectral-aware block-sparse attention, 2026a.

Yixuan Wang, Huang He, Siqi Bao, Hua Wu, Haifeng Wang, Qingfu Zhu, and Wanxiang Che. ProxyAttn: Guided sparse attention via representative heads, 2026b.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding, 2024.

Ruyi Xu, Guangxuan Xiao, Haofeng Huang, Junxian Guo, and Song Han. XAttention: Block sparse attention with antidiagonal scoring, 2025.

Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Yuanqian Zhao, Bokai Xu, Junbo Cui, Yingjing Xu, Liqing Ruan, Luoyuan Zhang, Hanyu Liu, Jingkun Tang, Hongyuan Liu, Qining Guo, Wenhao Hu, Bingxiang He, Jie Zhou, Jie Cai, Ji Qi, Zonghao Guo, Chi Chen, Guoyang Zeng, Yuxuan Li, Ganqu Cui, Ning Ding, Xu Han, Yuan Yao, Zhiyuan Liu, and Maosong Sun. MiniCPM-V 4.5: Cooking efficient MLLMs via architecture, data, and training recipe, 2025.

# A CORE-SET AGREEMENT PILOT

This appendix documents the core-set coverage observation in Table 1, including estimator settings, aggregation, and comparisons at additional cumulative-mass thresholds.

We reuse the archived Qwen3-VL QKV for Video-MME index 2225 (88,613 tokens), at zero-based layers 6, 12, 18, 24, and 30. The archived capture includes sample and layer labels but lacks fully certified checkpoint revision and input fingerprints. XAttention uses stride-16 antidiagonal scoring; the simpler estimator averages Q and K separately over consecutive groups of 16 tokens, applies reduced-resolution causal softmax, and sums probabilities into 128-token block scores. Both receive the same Q/K at each layer. We discard the production selectors' returned masks and independently construct mass-only prefixes from their raw scores. Scores are converted from BF16 to FP64 for cumulative sums; ties use increasing block index. The crossing block is included, and no floor, sink, diagonal, or final-row protection is forced.

For threshold $t$, let $X_t$ and $P_t$ denote the two prefix sets in this appendix, with $P_t$ referring to pooling rather than the method's group acceptance set. We compute $|X_t \cap P_t|/|X_t \cup P_t|$ and both directional coverages per head–query-block pair. Table 1 excludes rows 0–30 and the final partial block, leaving 21,152 pairs per layer and 105,760 in total. These pairs share one input and are not independent video-level observations. Prefixes are checked for nesting, threshold attainment, and minimality under the specified ordering.

Table 6: Threshold sweep for the same core-agreement pilot. Means over the five layers' interior head–query-block pairs (%). Equality is the fraction of exactly identical sets, not Jaccard overlap.

| Mass $t$ | Jaccard | X covered by pooling | Pooling covered by X | Equality |
|---|---|---|---|---|
| 0.10 | 87.1 | 90.5 | 91.0 | 72.9 |
| 0.30 | 88.6 | 92.5 | 94.5 | 51.0 |
| 0.50 | 89.2 | 92.9 | 95.6 | 32.8 |
| 0.70 | 90.5 | 93.9 | 96.2 | 15.5 |
| 0.90 | 94.3 | 96.5 | 97.6 | 4.9 |
| 0.95 | 96.0 | 97.7 | 98.2 | 4.8 |

The pilot supports substantial agreement at mass 0.5, not exact equality. Jaccard increases at higher thresholds while the exact-equality fraction decreases; growing sets also affect overlap. Thus, this sweep does not establish that relative disagreement is concentrated at the boundary, or that 0.5 is a uniquely justified cutoff. Multi-input validation and task-quality controls are needed for stronger conclusions.

# B EXACT SELECTOR DEFINITIONS

This appendix gives the fully indexed construction underlying Section 4, including grouped-query head mapping, pooled scoring, prefix selection, and the attention output. The main-text score $a_{rj}$ corresponds to $a_{hrj}$ for head $h$; with query block $r$ fixed, $a_j$ denotes the same score. The auxiliary score $b_j$ corresponds to $b_{g(h),r,j}$, where $g(h)$ identifies the head group. The sets $C, U, P, M$ correspond to their $hr$-indexed forms. The pooled position sets $I_r$ and $I_j$ are written $\bar{I}_r$ and $\bar{I}_j$ below.

Consider one causal self-attention layer with post-normalization, post-RoPE queries $Q \in \mathbb{R}^{H_q \times T \times d}$, keys $K \in \mathbb{R}^{H_{kv} \times T \times d}$, and values $V$ of matching key shape. We divide the sequence into blocks of $B = 128$ tokens and write $N = \lceil T/B \rceil$. Query block $r$ may attend to the visible key-block set $\mathcal{V}_r = \{0, \ldots, r\}$. For grouped-query attention, query head $h$ maps to key/value head

$$\kappa(h) = \left\lfloor \frac{h}{H_q/H_{kv}} \right\rfloor . \tag{5}$$

Indices are zero-based: $h \in \{0, \ldots, H_q-1\}$, $r, b \in \{0, \ldots, N-1\}$; $d$ is the head dimension and $H_{kv}$ divides $H_q$. The selector produces a binary mask $M_{hrb}$, with support set $M_{hr} = \{b : M_{hrb} = 1\}$. The sparse executor evaluates the original $Q, K, V$ tensors over all selected keys with one causal softmax; the selector never changes the model weights and requires no training data.

Layer and batch indices are omitted. Each prefill layer applies the construction to its own current Q/K representations without updating model weights.

### B.1 Head-Specific Salience

**A compact per-head view.** HSSE preserves differences between query heads while reducing the token resolution of scoring. It retains the rank-one/DC component of the within-group representation underlying XAttention's residue-based scoring (Xu et al., 2025). For pooling width $s = 16$, let $\bar{T} = \lceil T/s \rceil$ and form

$$\bar{q}_{hu} = \frac{1}{s}\sum_{t=0}^{s-1} \mathbf{1}[su + t < T] q_{h,su+t}, \qquad \bar{k}_{kv} = \frac{1}{s}\sum_{t=0}^{s-1} \mathbf{1}[sv + t < T] k_{k,sv+t}. \tag{6}$$

Pooling follows Q/K normalization and RoPE, so the score uses the position-dependent representations seen by attention. Out-of-range tokens contribute zero; even a partial tail group is divided by $s$. With $m = B/s = 8$ pooled positions per parent block, define $\bar{I}_r = \{mr, \ldots, \min(m(r+1), \bar{T}) - 1\}$. The score is

$$\bar{p}_{huv} = \frac{\mathbf{1}[v \le u] \exp(\bar{q}_{hu}^\top \bar{k}_{\kappa(h),v}/\sqrt{d})}{\sum_{w \le u} \exp(\bar{q}_{hu}^\top \bar{k}_{\kappa(h),w}/\sqrt{d})}, \qquad a_{hrb} = \sum_{u \in \bar{I}_r} \sum_{v \in \bar{I}_b} \bar{p}_{huv}. \tag{7}$$

Softmax is computed per pooled query *before* summing within blocks. Thus, a complete query-block row has total estimated mass eight, not one. This is a coarse ordering statistic, not the exact dense block mass or a calibrated probability of task relevance; pooling also approximates causality within each 16-token group.

**Nested salience prefixes.** Sort visible blocks by decreasing $a_{hrb}$, breaking ties by increasing block index, to obtain $\pi_{hr}$. Writing $\pi_j$ for the $j$-th block in this ordering ($j = 0, \ldots, r$), for a mass fraction $t \in [0, 1]$ retain the exclusive-CDF prefix

$$A_t^0(h, r) = \left\{ \pi_j : \sum_{i<j} a_{hr\pi_i} < t \sum_{b \in \mathcal{V}_r} a_{hrb} \right\}. \tag{8}$$

For positive total mass and $t > 0$, the crossing block is included; at $t = 0$ or zero total mass this mass-only set is empty, before floor and structural protection are added. Two thresholds on the same ordering yield nested sets without comparing different estimators' numerical scales. A smaller threshold identifies the mass-defined core; a larger threshold supplies candidates whose execution can still be restricted by group evidence.

### B.2 Group-Consensus Boundary Estimation

**Shared spatial evidence.** ProxyAttn motivates exploiting group-wise similarities between head attention patterns (Wang et al., 2026b). Our boundary module adapts its grouped proxy scoring and per-head budget mechanism; the new decision rule restricts where this evidence can act. With $G = 4$ contiguous groups and token stride $s_g = 4$, average query heads in group $\mathcal{H}_g$ and unrepeated key heads in group $\mathcal{K}_g$:

$$\widehat{q}_{gt} = \frac{1}{|\mathcal{H}_g|}\sum_{h \in \mathcal{H}_g} q_{h,s_g t}, \qquad \widehat{k}_{gv} = \frac{1}{|\mathcal{K}_g|}\sum_{k \in \mathcal{K}_g} k_{k,s_g v}. \tag{9}$$

Require $G$ to divide $H_q$ and $H_{kv}$. For $g = 0, \ldots, G-1$, define $\mathcal{H}_g = \{h : gH_q/G \le h < (g+1)H_q/G\}$ and analogously $\mathcal{K}_g$ using $H_{kv}$; $g(h) = \lfloor h/(H_q/G) \rfloor$. These groups are index-based, not learned clusters. Here $t, v$ index sampled positions, $0 \le t, v < \lceil T/s_g \rceil$, and $J_r = \{t : rB \le s_g t < \min((r+1)B, T)\}$. Define $\widehat{p}_{gtv} = \mathbf{1}[v \le t] \exp(\widehat{q}_{gt}^\top \widehat{k}_{gv}/\sqrt{d}) / \sum_{w=0}^{t} \exp(\widehat{q}_{gt}^\top \widehat{k}_{gw}/\sqrt{d})$. Unlike HSSE, the proxy map uses a blockwise maximum followed by row normalization:

$$\widetilde{b}_{grb} = \max_{t \in J_r,\, v \in J_b,\, v \le t} \widehat{p}_{gtv}, \qquad b_{grb} = \frac{\widetilde{b}_{grb}}{\sum_{j=0}^{N-1} \widetilde{b}_{grj} + 10^{-6}}. \tag{10}$$

Empty causal blocks have score zero. The maximum emphasizes a strong sampled interaction rather than accumulated block mass; averaging heads can lose head-specific evidence, which motivates withholding the proxy's authority over the protected core.

**Head-specific acceptance.** A separate trailing-query probe produces a requested count $k_h$ for each query head. It averages attention from the final at most 128 queries to all keys and pools this vector over key blocks. This probe is a scoring heuristic without a causal mask; the executed attention remains strictly causal. Its cumulative threshold $\gamma = 0.95$ and a 16-block budget floor determine $k_h$; the exact rounding convention is given in Appendix C. For head $h$, take the $k_h$-th largest score in the full, zero-masked row $b_{g(h),r,:}$ as cutoff $\theta_{hr}$ and set

$$P_{hr} = \{b \in \mathcal{V}_r : b_{g(h),rb} \geq \theta_{hr}\}. \tag{11}$$

Cutoff ties are all retained, so $P_{hr}$ need not contain exactly $k_h$ blocks. The spatial score is group-shared, but the count is head-specific and shared across that head's query rows. Neither the auxiliary mask nor the final support must therefore be identical within a group.

### B.3 Protected-Boundary Selection

**Protected support.** Let $F_r = \{0, r\}$ for ordinary rows and $F_r = \mathcal{V}_r$ for the final two parent-block rows. These are structural sink, diagonal, and final-row protections, not semantic labels. Let $L_{hr}$ contain the first $\min(16, r+1)$ blocks in the salience ordering: it is a *ranked* floor, not the earliest 16 blocks or a local window. With $c = 0.5$ and $u = 0.95$,

$$C_{hr} = A_c^0(h, r) \cup L_{hr} \cup F_r, \qquad U_{hr} = A_u^0(h, r) \cup L_{hr} \cup F_r. \tag{12}$$

CDFs are evaluated over all visible blocks before adding structural protection. The floor prevents a sharply concentrated estimated distribution from leaving too few candidates; the mass component adds input-dependent protection beyond that floor. Their distinct contributions are tested separately rather than assuming both are necessary. Throughout, $R_t = A_t^0 \cup L \cup F$ denotes a protected head-specific prefix, $P_\gamma$ the auxiliary acceptance set at probe threshold $\gamma$, and $\mathcal{T} = \mathcal{V} \setminus U$ the primary tail; head/row indices are suppressed when clear. Thus $C = R_c$ and $U = R_u$. These same sets define the task ablations: head-specific .85 uses $R_{.85}$; head-specific .95 uses $U$; group-only uses $P \cup F$; no explicit core uses $(U \cap P) \cup F$; floor-only uses $(U \cap P) \cup L \cup F$; global union uses $U \cup P \cup F$; core-only uses $C$. Removing explicit core protection therefore does not remove structural support or the floor from the outer candidate set.

**Asymmetric selection rule.** The final support is

$$M_{hr} = C_{hr} \cup \big((U_{hr} \setminus C_{hr}) \cap P_{hr}\big) = C_{hr} \cup (U_{hr} \cap P_{hr}). \tag{13}$$

Consequently, $C_{hr} \subseteq M_{hr} \subseteq U_{hr}$: the auxiliary signal cannot remove a protected block or rescue a block outside the outer set. For example, an auxiliary-rejected block survives if it belongs to $C$, but is removed if it belongs only to $U \setminus C$. A global intersection lacks the first guarantee, and a global union lacks the second. These are set-containment guarantees, not bounds on task error. Triage neither averages the two scores nor refills a fixed global budget after pruning. Both scorers are currently computed over their required domains; only the auxiliary *decision*, not its computation, is boundary-limited.

### B.4 Execution and Cost

Selected parent blocks are compiled into maximal contiguous key ranges. For an original query token $i$ in row $r$, define $\mathcal{S}_{hi} = \{j \leq i : \lfloor j/B \rfloor \in M_{hr}\}$. The executor computes

$$O_{hi} = \frac{\sum_{j \in \mathcal{S}_{hi}} \exp(q_{hi}^\top k_{\kappa(h),j}/\sqrt{d}) v_{\kappa(h),j}}{\sum_{j \in \mathcal{S}_{hi}} \exp(q_{hi}^\top k_{\kappa(h),j}/\sqrt{d})}. \tag{14}$$

Original Q/K/V, rather than pooled representatives, determine the output, with one normalization over all selected keys. Projection, RoPE, cache updates, output projection, and decoding are unchanged. At fixed head dimension, scoring uses approximately $H_q(T/16)^2$ head-specific and $G(T/4)^2$ grouped interactions, plus the trailing probe and block sorting; it is not linear-time attention. Sparse execution scales with selected token-pair area. We therefore measure selector and kernel costs separately and distinguish execution optimizations from the selection rule.

## C Auxiliary Budget Convention

This appendix specifies the trailing-query budget used by the Group consensus and Budget branches in Figure 1(a). It expands the head-specific count $k$ in Section 4 into $k_h$ and records the pooling and rounding operations needed to reproduce the acceptance set.

For trailing queries $\mathcal{J} = \{\max(0, T-128), \dots, T-1\}$, the probe forms

$$z_{hj} = \frac{1}{|\mathcal{J}|} \sum_{i \in \mathcal{J}} \frac{\exp(q_{hi}^\top k_{\kappa(h),j}/\sqrt{d})}{\sum_{t=0}^{T-1} \exp(q_{hi}^\top k_{\kappa(h),t}/\sqrt{d})}.$$

Average pooling over 128-key blocks gives $w_{hb}$; a partial final block is averaged over its valid entries. Write $w_h^\downarrow$ for its descending ordering. The implementation preserves the following two increments and the floating-point divide/multiply before integer conversion:

$$\rho_h = \frac{1 + \sum_{j=1}^{N} \mathbf{1}[\sum_{i=1}^{j} w_{hi}^\downarrow \le \gamma \sum_{i=1}^{N} w_{hi}^\downarrow]}{N}, \qquad k_h = \min\{N, \max\{16, \mathrm{int64}(\rho_h N) + 1\}\}.$$

These details follow the reused proxy budget path rather than an idealized minimum-prefix rule. The group-score cutoff includes ties; causal visibility and structural support are enforced in the final mask. No future value vectors enter the sparse attention output.

## D Execution and Measurement

This appendix expands the measurement definitions in Section 5.1 and decomposes the attention times in Tables 2 and 3. Those tables aggregate selector and attention-kernel spans, whereas Figure 4 measures host time to first token, including preprocessing and the surrounding model computation. The two boundaries should not be combined or interpreted as equivalent speedup estimates.

Figure 5 separates the two components behind the total times in Tables 2 and 3. On Qwen3, Triage spends more on selection than ProxyAttn but less on the attention kernel in both workloads. On Qwen2.5 LongVideoBench, the same trade-off appears: selector time rises from 0.346 to 0.619 seconds, while kernel time falls from 8.356 to 4.509 seconds. This decomposition explains why selector cost alone does not determine total attention cost; it does not measure end-to-end speedup.

The selection block is 128 tokens. The main benchmarks and historical partition sweep use different kernel configurations; timing differences between these experiments do not isolate the contribution of the selection rule.

The cost tables sum CUDA selector and kernel spans over language layers and average over questions. They exclude the post-kernel output-layout conversion as well as QKV projections, the visual encoder, MLPs and decoding. The means include successful recorded requests rather than a separately repeated steady-state microbenchmark; compilation and startup effects are not removed by a matched repeated-run protocol. Consequently, these are measured workload costs, not hardware-independent FLOP estimates or TTFT. Reported absence of fallback warnings is not by itself a proof that no internal range-overflow fallback occurred.

## E Complete Quality–Cost Sweep

This table reports the original 15-configuration sweep used in Figure 2. Its default Triage measurement is 377/540 with 4.420 seconds per question. The later default rerun reported in Table 5 and Table 10 yields 381/540 with 4.433 seconds. Separate selector and kernel times for that rerun have not been provided, so its total is not combined with the historical components below.

## F Deletion-Cost Definition and Protocol

These local diagnostics supplement the core-protection comparison in Table 4 by measuring attention-output changes after block deletion. They use sampled queries at one layer rather than

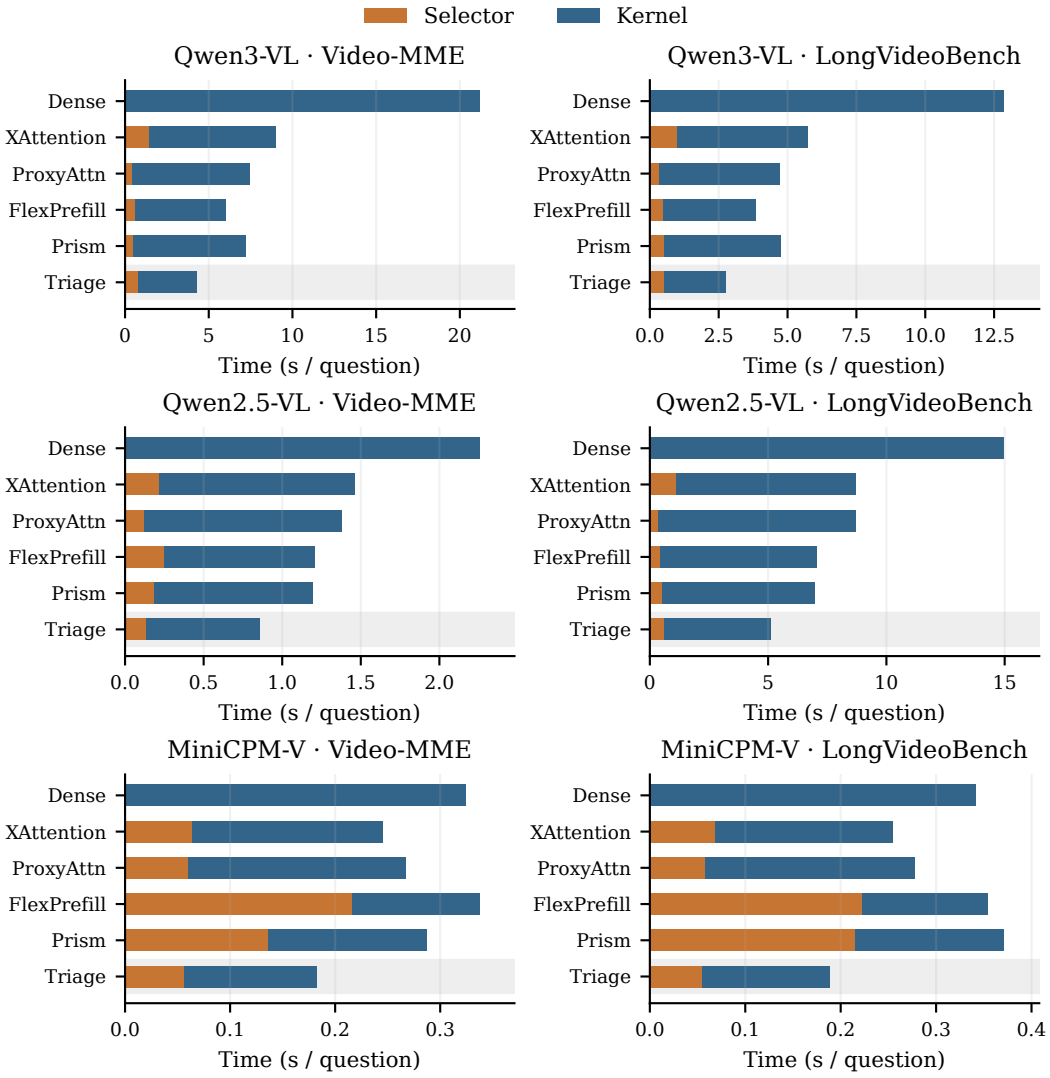


Figure 5: Attention time breakdown. Selector and kernel CUDA spans, averaged over all questions in each benchmark. The Triage row is shaded. Each panel has its own scale because model workloads differ. Dense bars contain kernel time only. Video-MME Dense means for all three models use raw timing records with complete sample and layer coverage; other components use rounded experiment-ledger values.

Table 7: All configurations from the original quality–cost sweep. Parameter is auxiliary-budget $\gamma$ for Triage/ProxyAttn and selection threshold $\tau$ for XAttention. Triage fixes $c = .5, u = .95$ and floor 16. Selector/Kernel times are seconds/question.

| Method | Parameter | Correct / 540 | Selector | Kernel |
|---|---|---|---|---|
| ProxyAttn | 0.80 | 370 | 0.458 | 2.100 |
| ProxyAttn | 0.90 | 368 | 0.464 | 3.982 |
| ProxyAttn | 0.95 | 375 | 0.466 | 7.242 |
| ProxyAttn | 0.975 | 379 | 0.467 | 11.640 |
| ProxyAttn | 0.99 | 378 | 0.470 | 18.184 |
| Triage | 0.85 | 373 | 0.862 | 2.050 |
| Triage | 0.90 | 377 | 0.858 | 2.472 |
| Triage | 0.925 | 376 | 0.860 | 2.881 |
| Triage | 0.95 | 377 | 0.859 | 3.561 |
| Triage | 0.98 | 381 | 0.866 | 5.272 |
| XAttention | 0.80 | 372 | 1.585 | 4.842 |
| XAttention | 0.85 | 375 | 1.587 | 5.981 |
| XAttention | 0.89 | 373 | 1.586 | 7.254 |
| XAttention | 0.95 | 378 | 1.586 | 10.474 |
| XAttention | 0.98 | 378 | 1.593 | 13.807 |

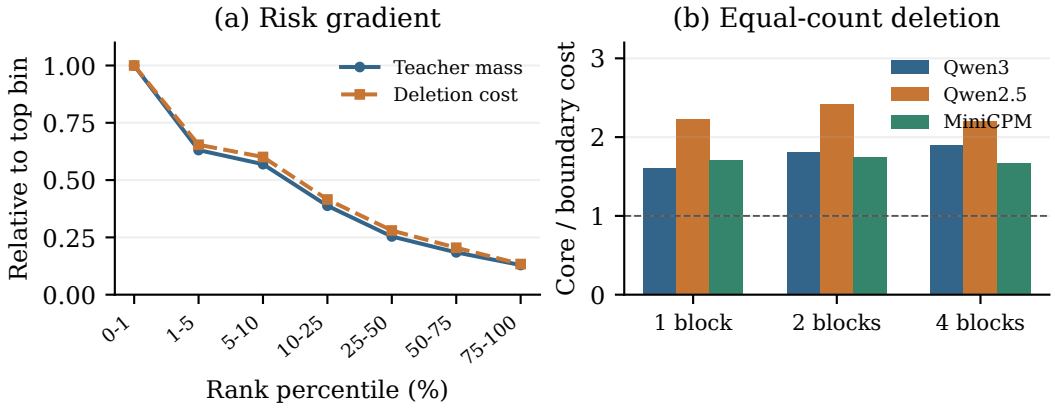


Figure 6: Local deletion diagnostics. (a) Qwen3 teacher mass and deletion cost, each normalized to its own top-rank bin. (b) Matched core/boundary deletion-cost ratios from ten videos per model; the Qwen2.5 four-block comparison has nine feasible videos. Bars are point estimates. These are local output effects, not task-level accuracy changes.

full-model answer generation and therefore do not establish task preservation. Figure 6 summarizes the rank-dependent and equal-count deletion measurements.

Figure 6 uses ten videos per model at one intermediate layer: layer 20 for Qwen3 and MiniCPM, and layer 15 for Qwen2.5. We sample eight heads and four active query-block rows, with at most eight queries per row; each teacher query attends to all causally visible keys. For a reference block support $S$, deletion set $D \subseteq S$, and valid sampled queries $\mathcal{Q}$, define

$$\delta(S, D) = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \|O_S(q) - O_{S \setminus D}(q)\|_2.$$

Here $O_S(q)$ is the attention output using original values and weights renormalized over causally visible keys in $S$; the norm spans output channels. Queries with vanishing retained mass are excluded. This metric is an unsquared, energy-unnormalized L2 change.

The rank-gradient test uses $S = \mathcal{V}$ and a single-block deletion, with percentile bins defined on non-structural visible blocks. On Qwen3, teacher mass decreases from 0.0634 to 0.0082 and deletion cost from 0.1914 to 0.0256 across the extreme bins. The matched test instead uses $S = U$ and samples

equally many blocks from $(C \setminus F) \setminus P$ and $(U \setminus C) \setminus P$. It averages five draws per feasible head-row for each deletion size 1, 2, or 4. Infeasible cases are excluded. Core/boundary cost ratios are formed within each video before averaging videos; one-block ratios are 1.61, 2.23, and 1.71 for Qwen3, Qwen2.5, and MiniCPM. These are local omission effects, not task-level guarantees. Auxiliary acceptance in the primary tail $\mathcal{T}$ is 57.46%, 48.79%, and 59.41%, respectively, so agreement does not define a universally safe tail.

## G Transfer Ablations

These experiments complement the Qwen3-VL component comparison in Table 5 with four selection policies on Qwen2.5-VL and MiniCPM-V. They use a separate 321-question LongVideoBench subset rather than the 540-question Video-MME subset in the main table. Full Triage selects $C \cup ((U \setminus C) \cap P)$; head-specific selection uses $R_{.85}$; no explicit core selects $(U \cap P) \cup F$; and floor-only selection uses $(U \cap P) \cup L \cup F$. These sets follow Appendix B, and each policy uses its natural support budget. The two models exhibit different accuracy orderings, so these transfer results are reported separately from the main-table comparison.

Table 8: Natural-budget policy ablations on 321 LongVideoBench questions per model. Selector and kernel times are seconds per question.

| Model | Policy | Correct / 321 | Selector | Kernel |
|---|---|---|---|---|
| Qwen2.5-VL | Full Triage | 184 | 0.647 | 4.479 |
| Qwen2.5-VL | Head-specific only | 183 | 0.373 | 4.412 |
| Qwen2.5-VL | No explicit core | 187 | 0.648 | 4.228 |
| Qwen2.5-VL | Floor only | 182 | 0.685 | 4.265 |
| MiniCPM-V | Full Triage | 207 | 0.076 | 0.130 |
| MiniCPM-V | Head-specific only | 208 | 0.043 | 0.125 |
| MiniCPM-V | No explicit core | 205 | 0.076 | 0.116 |
| MiniCPM-V | Floor only | 205 | 0.076 | 0.130 |

## H Frozen-Budget Task Diagnostic

This diagnostic complements the natural-budget comparison in Table 5 by fixing the retained block count for every layer, head, and query row. It evaluates ranking policies under a shared support budget rather than measuring deployable end-to-end speedups.

On a frozen 60-video, 180-question Video-MME subset, a separate dense-trajectory reference run records Triage's target block count for each layer, head, and query row. Each subsequent policy ranks its own current Q/K representations while retaining exactly these counts and structural support. Triage-priority orders core blocks first, accepted boundary blocks second, and remaining visible blocks last, with head-specific score and block-index tie breaking. It is a diagnostic policy, not the production Triage mask. Rank fusion equally weights normalized A/B ranks. Random selection samples non-structural visible blocks with seed 0; four additional planned seeds were omitted for compute/time constraints, so no random-seed variance is estimated.

Table 9: Frozen-budget full-model diagnostic. All methods use the same 180 questions and a shared attention executor. Times are CUDA S+K seconds/question, excluding the separate dense-reference budget acquisition; these are not deployable speedups.

| Policy | Correct / 180 | Selector | Kernel | Total |
|---|---|---|---|---|
| Head-specific ranking | 122 | 1.285 | 3.252 | 4.536 |
| Group ranking | **124** | 1.288 | 3.295 | 4.583 |
| Rank fusion | 123 | 1.382 | 3.273 | 4.655 |
| Triage-priority | 123 | 1.324 | 3.252 | 4.576 |
| Random (seed 0) | 118 | 1.286 | 3.349 | 4.636 |

Each run contains 6,480 layer-level audit records with matching selected/target block totals and no reported fallback; the runtime also checks per-row counts. A research range compiler allocates the theoretical maximum number of disjoint ranges rather than expanding fragmented masks or falling back to dense. An earlier capacity-failure attempt is excluded. Matching block budgets does not make selector computation or memory-access patterns identical. The one- and two-question differences among informed rankings do not establish significant superiority.

## I Complete Parameter Sensitivity

Table 10 supplies the numerical values underlying Figure 3 and adds attention time for each setting. The default row uses the separately reported rerun with 381 correct answers and 4.433 seconds per question, consistent with Table 5. The seven non-default settings retain their historical results. This table is distinct from the auxiliary-budget sweep in Appendix E, whose original default measurement is retained for that experiment.

Table 10: Parameter sensitivity. All seven non-default points on 540 questions; each varies one parameter from $c = .5$, $u = .95$, floor=16. Time is attention S+K in seconds/question.

| Axis | Value | Correct / 540 | Attention time |
|---|---|---|---|
| Triage default | – | 381 | 4.433 s |
| Core mass | 0.30 | 378 | 4.320 s |
| Core mass | 0.70 | 379 | 4.833 s |
| Outer mass | 0.90 | 378 | 3.868 s |
| Outer mass | 0.98 | 378 | 4.980 s |
| Protected floor | 0 | 374 | 4.356 s |
| Protected floor | 8 | 375 | 4.366 s |
| Protected floor | 32 | 381 | 4.613 s |

## J Experimental Boundaries

The conservative $R_{.95} \cap P_{.95}$ run in Table 4 produced two copies of the same complete 2,700-row score table, each with 1,868 correct answers. The campaign verifier rejected the ambiguous duplicate paths. We report the recovered scientific result as a historical comparison, not as an unqualified verified campaign. The other protected configurations belong to the original full-set partition sweep.

The full benchmark tables use complete task sets. The quality–cost, component, and sensitivity experiments use pre-frozen subsets and should not be interpreted as independent test sets. Natural-budget policies are not equal-FLOP comparisons because their selected support differs. The fixed-QKV analyses estimate local attention-output effects and do not substitute for full-model accuracy. We retain all failed, stopped, and quarantined runs in the experiment ledger but exclude them from formal numerical claims.

## K ATTENTION SUPPORT VISUALIZATIONS

These visualizations supplement the selection rule in Section 4. All methods use identical captured Q/K/V from one 88,613-token Qwen3-VL input, Video-MME index 2225. Layers 6, 12, 18, 24, and 30 are shown with fixed heads 0, 8, 16, and 24, followed by statistics over all 32 heads. Indices are zero-based. The maps show binary execution support, not attention weights. Capture provenance limitations are stated in Appendix A. No range overflow occurs in these replays; instrumentation preserves attention outputs.

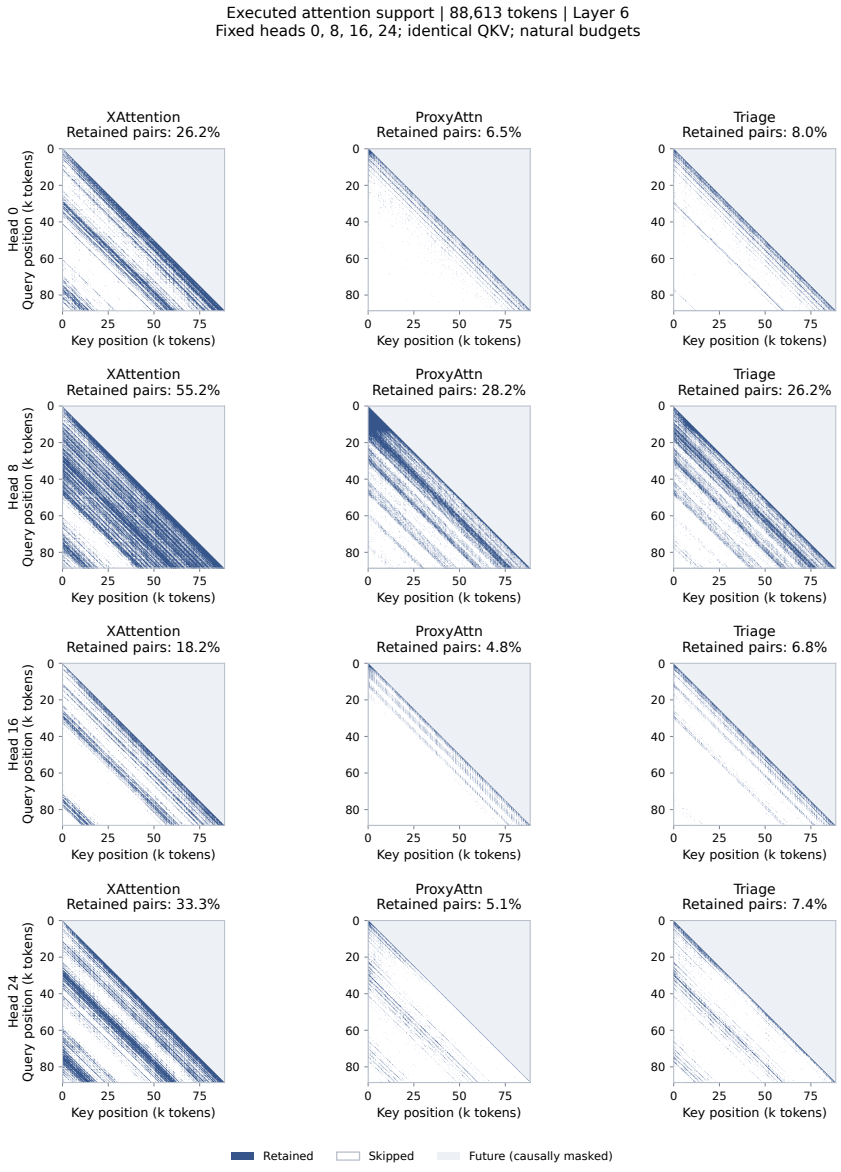


Figure 7: Executed attention support at layer 6. Rows show heads 0, 8, 16, and 24; columns show XAttention, ProxyAttn, and Triage. Blue denotes retained block pairs, white skipped pairs, and gray future blocks. Percentages measure retained causal token pairs, accounting for partial blocks. Natural budgets are not matched.

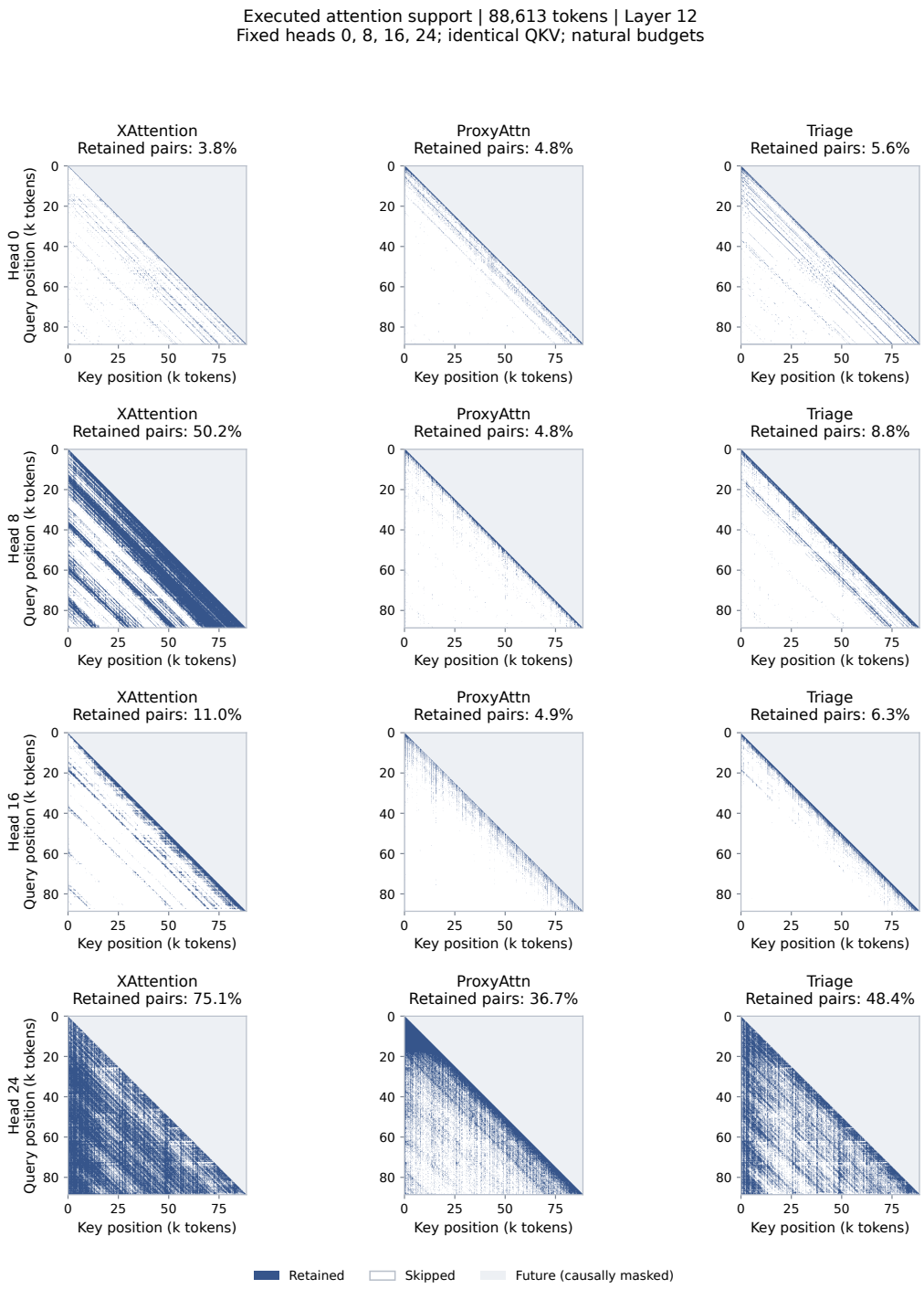


Figure 8: Executed attention support at layer 12, with the same input, methods, fixed heads, and display conventions as Figure 7.

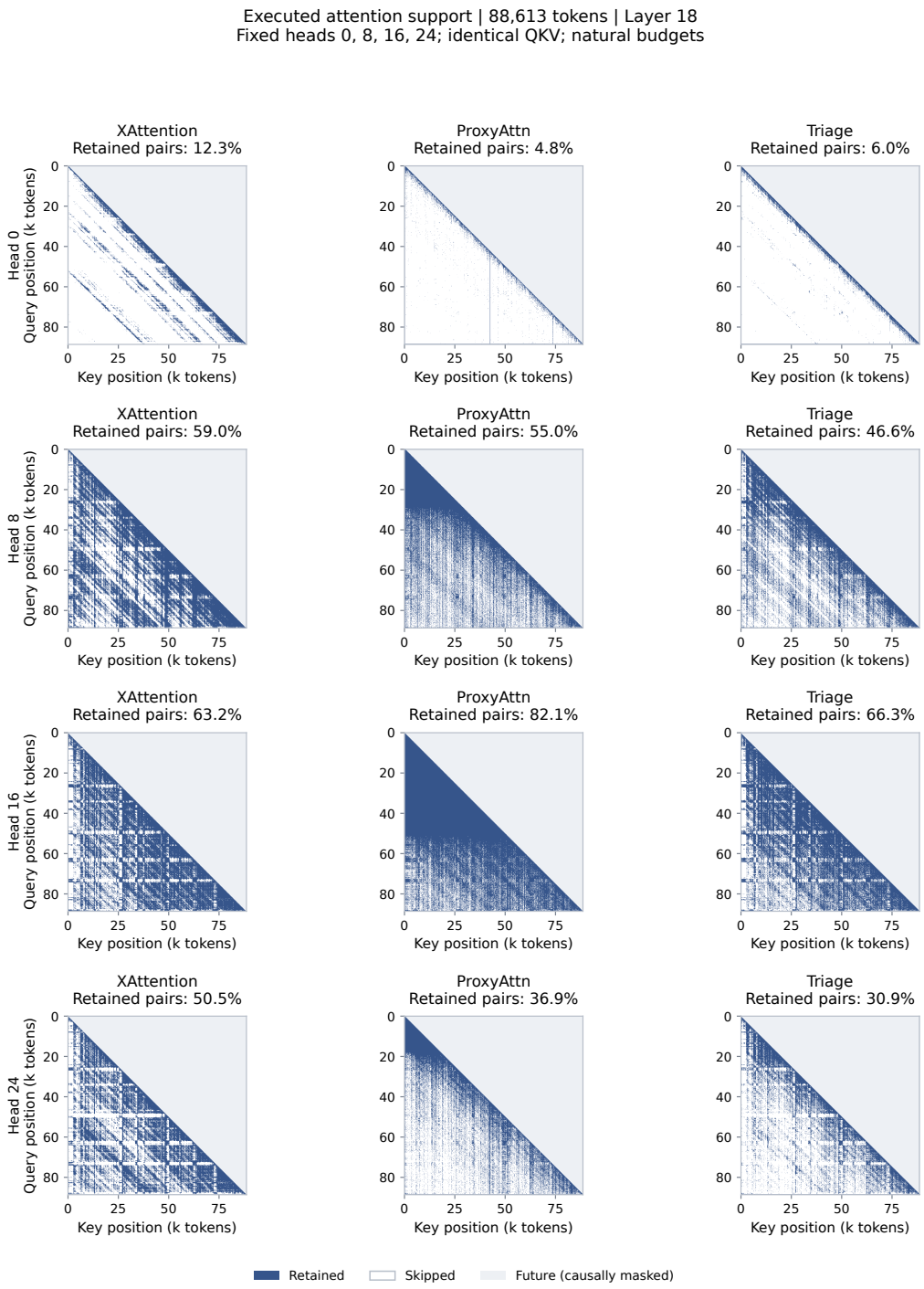


Figure 9: Executed attention support at layer 18, using the conventions of Figure 7. Core protection permits Triage to retain more support than the auxiliary selector in individual heads.

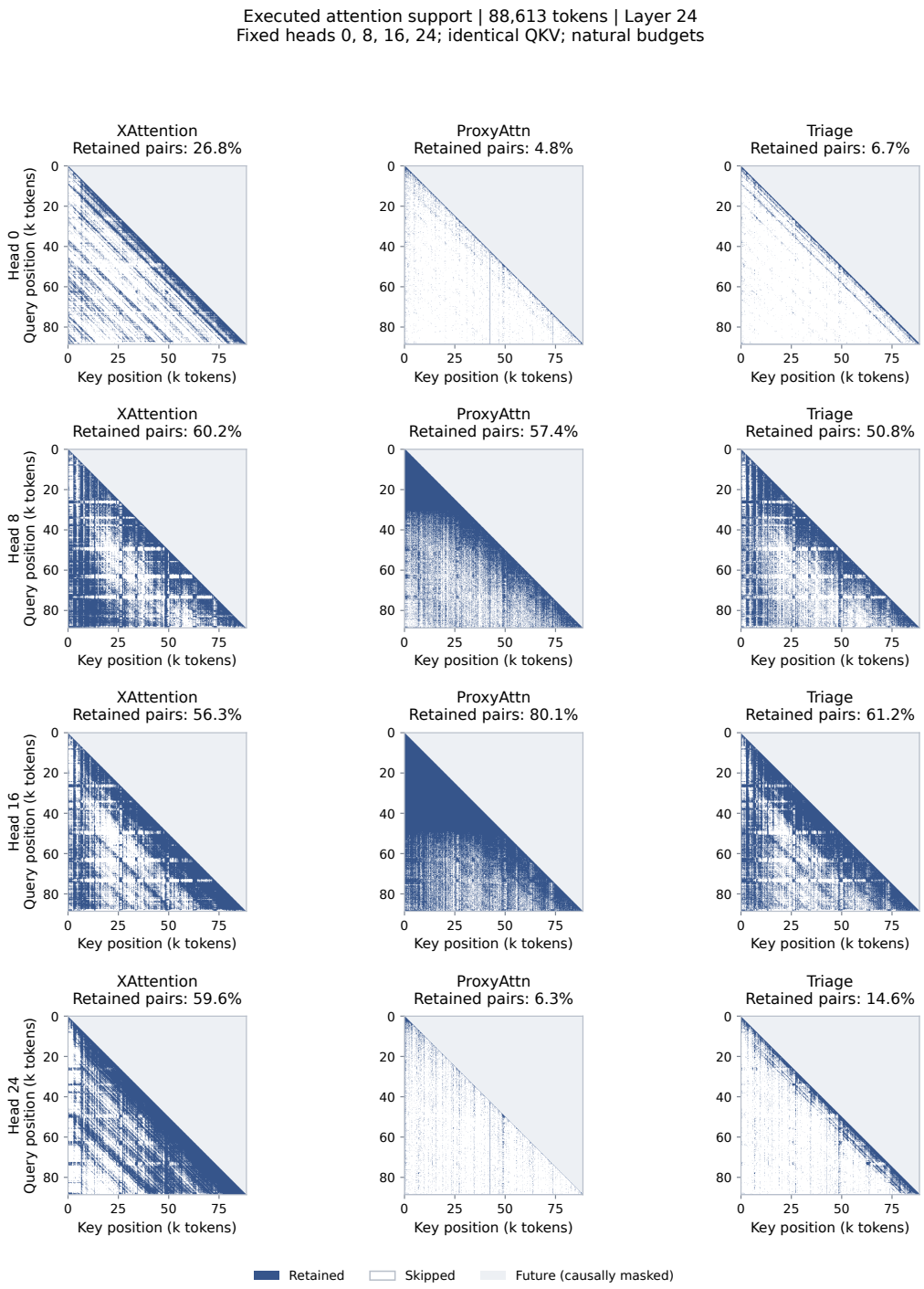


Figure 10: Executed attention support at layer 24, with the same methods, fixed heads, and display conventions as Figure 7.

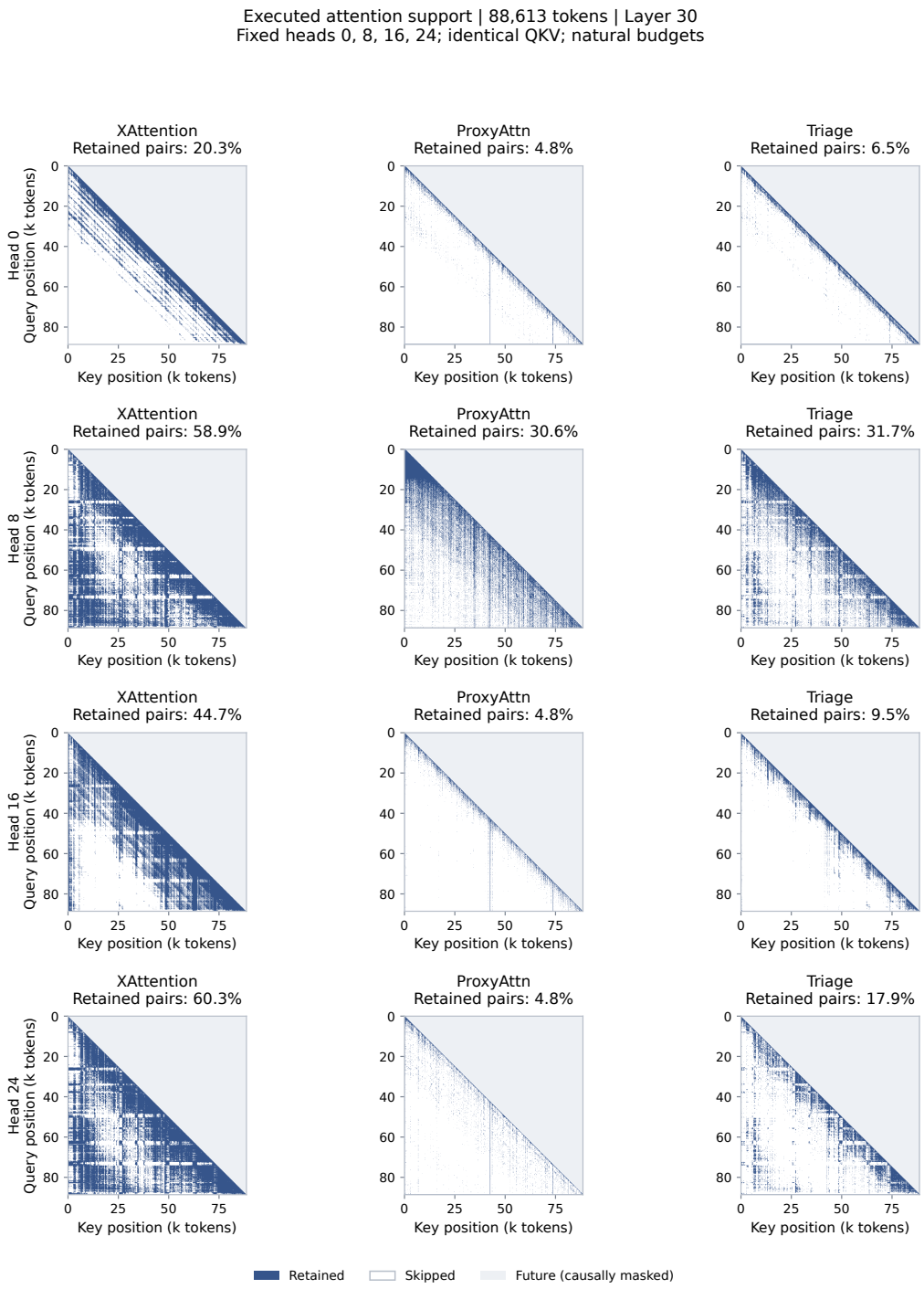


Figure 11: Executed attention support at layer 30, using the conventions of Figure 7. These local support maps do not measure task accuracy or end-to-end latency.

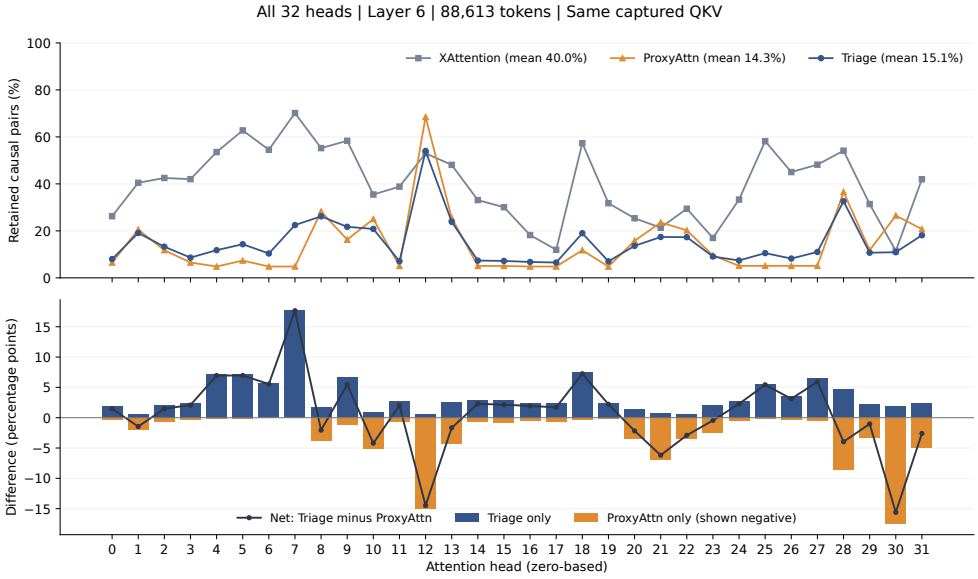


Figure 12: All-head support statistics at layer 6. Top: retained causal token-pair percentages for all 32 heads. Bottom: Triage-only support, ProxyAttn-only support displayed as negative bars, and their net difference in percentage points. Legend means weight heads equally.

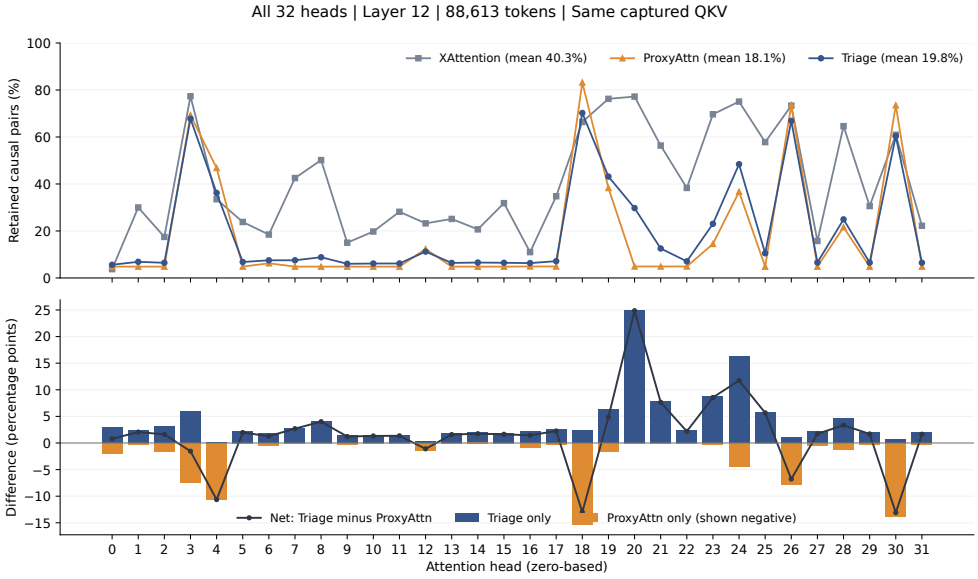


Figure 13: All-head support statistics at layer 12, using the definitions of Figure 12.

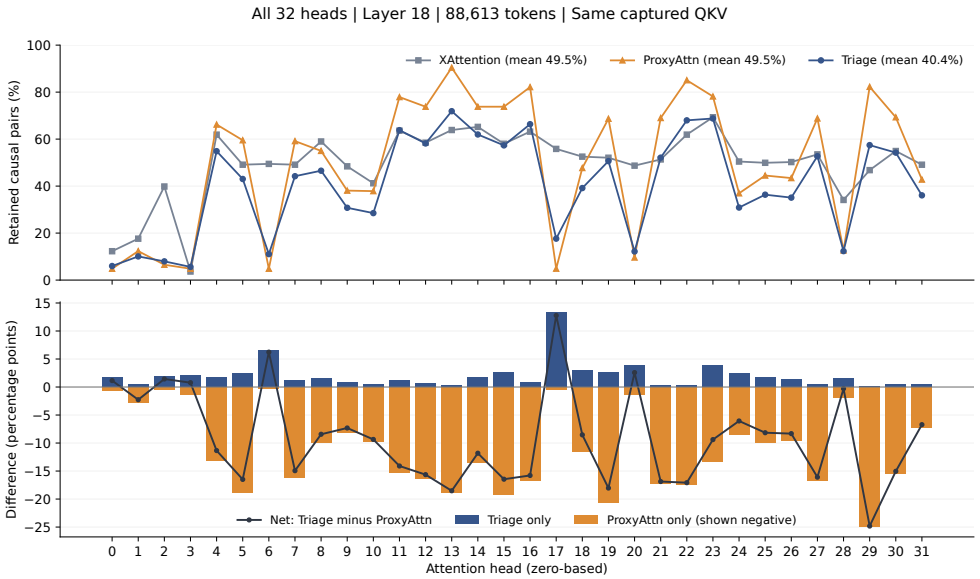


Figure 14: All-head support statistics at layer 18, using the definitions of Figure 12.

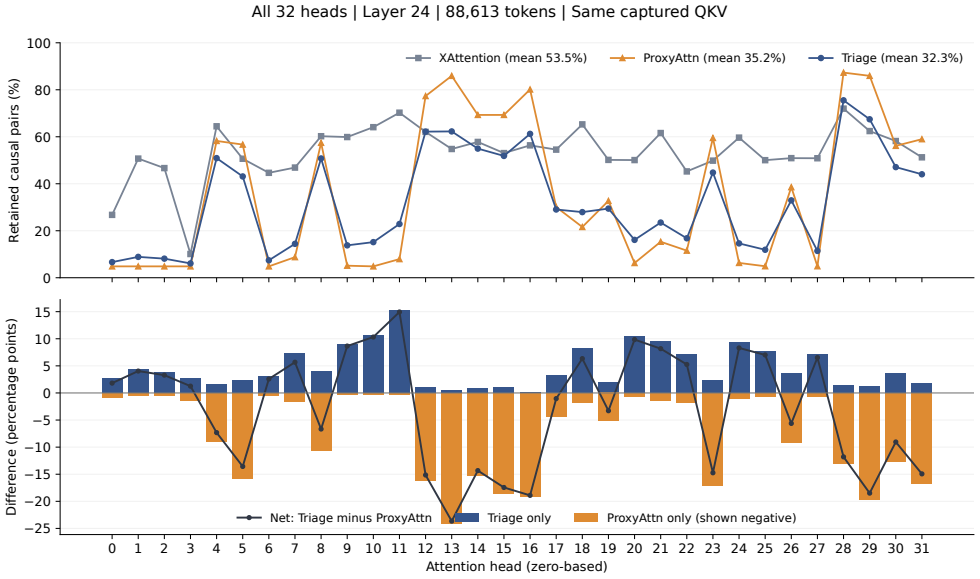


Figure 15: All-head support statistics at layer 24, using the definitions of Figure 12.

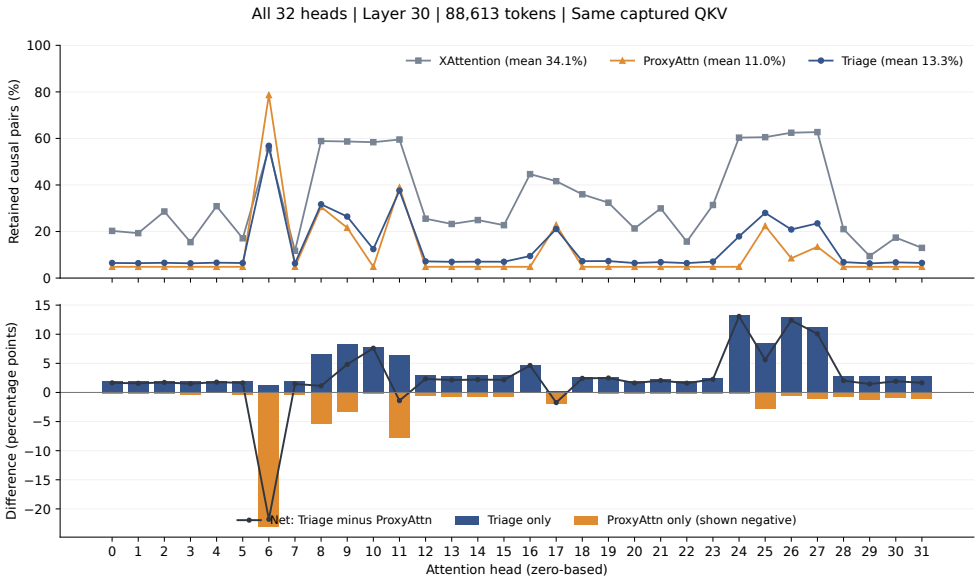


Figure 16: All-head support statistics at layer 30, using the definitions of Figure 12. The displayed five layers characterize one input rather than the full benchmark.