SPARSETRANSFER: DOES CALIBRATION-FREE SPARSE ATTENTION GENERALIZE ACROSS DOMAINS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Training-free sparse attention methods are increasingly evaluated as if they were universal: a mechanism validated in one domain is often assumed to transfer to others without recalibration. We test this assumption directly by porting two independently-published, training-free block selection mechanisms EntropyInfer, an entropy-guided adaptive budgeting method, and a cumulative-energy-based thresholding method into a shared attention framework (XAttention), configuring each to match XAttention's own tuned baseline operating density, and evaluating both against that baseline across two domains: long-context language modeling (RULER, Llama-3.1-8B, context lengths 4K–128K) and long-video question answering (Video-MME, Qwen2-VL-7B, 900 videos). We observe a striking, consistent asymmetry: both mechanisms transfer far more faithfully to video-understanding (97–101% of baseline accuracy) than to language (77–99.6% of baseline, with the largest gap at the longest context length), and this pattern holds across both tested mechanisms, with the cumulative-energy method consistently more robust than EntropyInfer in both domains. We verify our findings are not artifacts of implementation error through multiple independent checks, including direct inspection of the patched selection function and measured selection density. Our results suggest that "calibration-free" claims in the sparse attention literature are domain-contingent rather than universal, and that block-selection quality, not merely computational budget is a meaningful, underexplored axis of transfer failure.

1 INTRODUCTION

Serving large language models over long contexts is expensive, and much of that expense is attributable to the quadratic cost of full attention. A growing body of work addresses this with sparse attention: rather than attending to every token or block, these methods select a subset of the context believed to be most relevant, and compute attention only over that subset. Recent methods increasingly frame themselves as training-free and calibration-free requiring no fine-tuning and, in principle, no per-deployment tuning positioning this as a practical advantage over methods that must be recalibrated for each new model, context length, or modality.

This framing carries an implicit and rarely tested claim: that a method's selection mechanism generalizes beyond the exact setting in which it was originally validated. In practice, we observe a curious pattern in the literature. Methods that are evaluated across multiple domains or modalities (e.g., language and video) typically retune their sparsity parameters separately for each one. Methods that claim to avoid such retuning are, to our knowledge, only ever evaluated within a single domain. No published work has taken a method's actual selection mechanism and not just its high-level idea, but its exact, published specification, and tested whether it continues to function when moved into a different implementation, a different model family, or a different modality entirely, without modification.

This gap is not merely academic. If calibration-free claims hold only within the narrow setting in which a method was introduced, then practitioners adopting these methods in new settings are implicitly relying on an untested assumption. Conversely, if such methods do generalize, this would be a genuinely useful and underappreciated property worth understanding and exploiting further.

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 address this gap directly. We take two independently-published, training-free sparse attention mechanisms namely EntropyInfer, which adaptively allocates block budget using per-head attention entropy, and a cumulative-energy-based thresholding method and port each, following its published specification, into a different sparse attention framework's block-selection interface (XAttention). We then evaluate both mechanisms against that framework's own tuned baseline in two domains that differ substantially in their underlying data structure: long-context language modeling, where attention operates over sequential text tokens, and long-video question answering, where attention operates over spatiotemporal video patches. This design lets us hold the surrounding implementation fixed while varying only the selection mechanism and the domain, isolating the effect of each.

Our central finding is a striking asymmetry: both mechanisms transfer with only modest, but real, degradation into video-understanding, while both show substantially larger degradation in language, particularly at long context lengths. This asymmetry is most pronounced for EntropyInfer; for DynSparse the gap largely closes at matched context lengths ($\leq$32K), suggesting it reflects something general about how these selection strategies interact with domain structure, rather than being an artifact of either individual method. We further find that the two mechanisms differ meaningfully from each other in how well they tolerate this transfer, indicating that "training-free" is not a single property but spans a range of robustness.

In building this evaluation, we also address a practical gap: no working integration of block-sparse attention existed for the vision-language model we use for our video-understanding evaluation in the codebase we build on. We construct and validate this integration against the framework's own published methodology before using it for our experiments.

Our contributions are: (1) the first evaluation, to our knowledge, of whether published training-free sparse attention mechanisms retain their effectiveness when transferred across implementations and domains without modification; (2) a methodology for isolating selection quality from selection budget when comparing mechanisms across dissimilar interfaces; (3) an empirical finding that calibration-free transfer is domain-contingent rather than universal, with quality holding up far better in video-understanding than in language; and (4) We locate and validate an existing but undocumented integration.

2 Related work

Sparse attention for long-context inference. Reducing the quadratic cost of attention has motivated substantial work on identifying and computing only the most relevant portion of the attention matrix. Early work established both the practical necessity and the basic feasibility of sparse factorizations of attention for modeling very long sequences (Child et al., 2019), a direction that gained urgency as language models scaled and long-context capability became a first-class requirement (Brown et al., 2020). Streaming and windowed approaches showed that retaining a small set of consistently-important tokens can preserve much of dense attention's behavior over arbitrarily long sequences (Xiao et al., 2024), and architectures explicitly designed around structured sparsity have pushed feasible context lengths into the billions of tokens (Ding et al., 2023). More recent methods, including the framework we build on (XAttention), instead estimate block-level importance dynamically at inference time using cheap proxy statistics in XAttention's case, the sum of values along strided antidiagonals of each attention block (Xu et al., 2025) before pruning low-scoring blocks and computing full attention only within the retained set. Related dynamic approaches identify per-head sparsity patterns online (Jiang et al., 2024) or adaptively switch between pattern types based on context statistics (Lai et al., 2025). Our work builds directly on the block-selection abstraction common to this line of work, treating the underlying scoring-and-pruning framework as a fixed substrate into which different selection mechanisms can be substituted.

Training-free, adaptive selection mechanisms. Several recent methods argue that a single, globally fixed sparsity budget is a poor fit for attention heads and contexts that vary substantially in how concentrated or diffuse their attention is, and propose instead to determine the selection budget adaptively using a cheap online signal computed at inference time. EntropyInfer, one of the two mechanisms we evaluate, categorizes heads by whether their attention entropy is stable ("rigid") or fluctuates across segments ("dynamic"), and walks the selection budget for dynamic heads in proportion to observed entropy fluctuation (Xu et al., 2026). DynSparse, the second mechanism we evaluate, reaches a related conclusion via a different derivation: formulating token selection as a

dual-goal optimization problem, it shows that fixed top- k or fixed top- p selection cannot simultaneously satisfy accuracy and efficiency objectives, and instead selects the smallest set of blocks whose cumulative score exceeds a fixed recall target (Li et al., 2026). Both methods are explicitly motivated by, and evaluated within, a single domain long-context text and video generation, respectively and neither reports results outside the setting in which it was introduced.

Empirical surveys and cross-method comparisons. Closest in spirit to our work are large-scale empirical studies that harmonize evaluation across many sparse attention methods and model families to characterize where existing approaches succeed or fail (Nawrot et al., 2025), typically organizing the design space along axes such as fixed-versus-context-aware importance estimation and uniform-versus-adaptive budget allocation. Such studies are comprehensive within a single domain (language) and do not test whether a given method's selection mechanism as opposed to its high-level design philosophy continues to function correctly when moved into a different codebase or a different modality. Methods that are themselves evaluated across multiple domains, including XAttention (Xu et al., 2025), uniformly retune their sparsity parameters separately for each domain and model, rather than testing whether a single configuration transfers.

Positioning. To our knowledge, no prior work has taken a training-free, adaptive selection mechanism's published specification and tested it, unmodified, in a domain and implementation other than the one in which it was introduced. Our contribution is this specific test: holding the surrounding sparse attention framework fixed, we substitute two independently-developed selection mechanisms and measure how much of their reported effectiveness survives the transfer, in a setting video-understanding that, to our knowledge, neither original evaluation considered.

Additional sparse attention variants. A broader family of methods explores alternative sparsification strategies at the architecture or kernel level. Early fixed-pattern approaches restrict attention to local windows combined with limited global tokens (Beltagy et al., 2020; Zaheer et al., 2020), or use locality-sensitive hashing to approximate nearest-neighbor attention (Kitaev et al., 2020). Content-based routing methods cluster or route tokens to reduce effective attention span (Roy et al., 2021; Wang et al., 2022). More recent inference-time approaches identify important key-value pairs via learned or heuristic importance signals: SeerAttention and its successor learn intrinsic sparsity patterns directly (Gao et al., 2024; 2025); Quest and Double Sparsity identify query-aware or channel-based sparsity at decode time (Tang et al., 2024; Yang et al., 2024); and retrieval- and clustering-based approaches manage the KV cache directly to reduce memory and compute (Liu et al., 2024a;b; Chen et al., 2025). Native Sparse Attention and Mixture of Block Attention integrate sparsity directly into the training process rather than applying it post-hoc (Yuan et al., 2025; Lu et al., 2025), and SpargeAttn targets accurate, training-free sparsity applicable across model architectures (Zhang et al., 2025), closest in spirit to the training-free, post-hoc setting we study. Landmark Attention introduces learned landmark tokens for random-access long-context retrieval (Mohtashami & Jaggi, 2023), and Rectified Sparse Attention addresses accuracy degradation specific to sparse decoding (Sun et al., 2025). Efficient kernel implementations underlying much of this work, including FlashAttention-2 and the Block-Sparse-Attention library we build on, provide the hardware-aware primitives that make block-level sparsity practical at scale (Dao, 2024; Guo et al., 2024), and modern tensor compilers such as TileLang further streamline sparse and dense attention kernel generation (Wang et al., 2025). Most of these methods are evaluated within a single domain.

3 Experiment

3.1 Setup

Framework and mechanisms. We build on XAttention, a block-sparse attention framework that estimates block importance via strided antidiagonal scoring and prunes blocks below a threshold before computing attention only over the retained set (Xu et al., 2025). We evaluate two independently-published, training-free selection mechanisms by substituting them for XAttention's own threshold-based selection at the identical interface point: EntropyInfer (Xu et al., 2026), which adaptively walks a per-head, per-segment budget based on observed attention-entropy fluctuation, and DynSparse (Li et al., 2026), which selects the smallest set of blocks whose cumulative score exceeds a fixed recall target. Both mechanisms were originally specified with budget parameters calibrated for their own evaluation settings (an absolute token count and a fixed recall target, respectively) that do not translate directly into XAttention's block-based interface; we set each mecha-

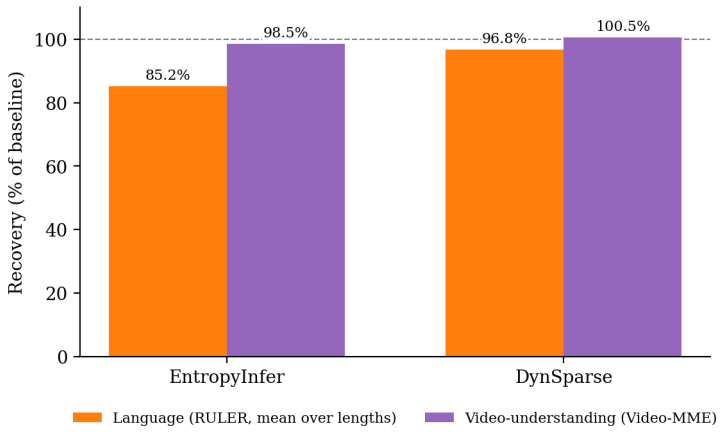


Figure 1: Both mechanisms recover substantially more of XAttention's baseline accuracy in video-understanding than in language, a pattern that holds most pronounced for EntropyInfer; for DynSparse the gap largely closes at matched context lengths ($\leq 32K$).

nism's operating density to match XAttention's own tuned baseline density (approximately 20–30%, depending on context length and domain), measured directly via the fraction of blocks retained per attention call. This isolates selection quality which blocks each mechanism retains from selection budget.

Domains and models. We evaluate in two domains that differ substantially in the structure of the underlying attention computation: long-context language modeling, using Llama-3.1-8B-Instruct (Grattafiori et al., 2024) on RULER (Hsieh et al., 2024) across six context lengths (4K, 8K, 16K, 32K, 64K, and 128K tokens, 13 tasks per length); and long-video question answering, using Qwen2-VL-7B-Instruct (Wang et al., 2024) on the full Video-MME benchmark (Fu et al., 2025) (900 videos, 2,700 multiple-choice questions, stratified across short/medium/long duration categories), scored via exact-match against the ground-truth answer choice.

Video-understanding integration. XAttention's public codebase includes a working block-sparse attention integration for Qwen2-VL that we located and validated (rather than reimplementing independently), applying sparse attention from the fourth transformer layer onward while retaining dense attention in the first three layers, matching the reference implementation exactly. Because of single-GPU memory constraints, we cap video frame sampling at 64 frames per video regardless of duration, rather than the uncapped, duration-scaled frame rate used in XAttention's own evaluation; this keeps all video sequences within Qwen2-VL's trained context window (32,768 tokens) and avoids out-of-memory failures we observed at higher frame counts. This is a real methodological deviation, and we return to its implications in Section 6.

Verification. Given that both mechanisms are ported into a codebase neither was originally evaluated in, we verify at each stage that the intended selection mechanism is genuinely active rather than silently falling back to baseline behavior: by direct inspection of the patched selection function's identity before and after patching, by confirming distinct, mechanism-specific timing signatures relative to baseline, and by measuring block-selection density directly on real inputs in both domains.

3.2 Language-domain results (RULER)

Table 1 reports accuracy for baseline XAttention and both ported mechanisms across all six context lengths.

Both mechanisms degrade relative to baseline at every context length, but by substantially different margins. EntropyInfer recovers 77–89% of baseline accuracy depending on length, with the largest gap at 128K (77% recovery). DynSparse is markedly more robust, recovering 91–99.6% of baseline at every length tested, degrading only modestly even at 128K (91.2% recovery). Both mechanisms

Table 1: RULER accuracy (Llama-3.1-8B-Instruct) across context lengths. EntropyInfer and DynSparse are ported into XAttention's block-selection interface with density matched to XAttention's own tuned baseline.

	Context length					
Method	4K	8K	16K	32K	64K	128K
XAttention (baseline)	96.12	94.14	93.36	90.71	82.90	70.60
EntropyInfer	81.53	83.64	82.29	79.65	69.91	54.33
DynSparse	95.68	93.72	91.65	89.38	77.98	64.39

Table 2: Video-MME accuracy (Qwen2-VL-7B-Instruct) on the full 900-video benchmark, by duration category.

Method	Short	Medium	Long	Overall
XAttention (baseline)	72.33	58.78	50.33	60.48
EntropyInfer	70.33	58.11	50.33	59.59
DynSparse	72.56	59.11	50.67	60.78

show their largest relative degradation at the longest context length, consistent with degradation compounding as the number of selection decisions per sequence grows.

3.3 Video-understanding results (Video-MME)

Table 2 reports accuracy across the full 900-video benchmark, broken down by duration category.

In sharp contrast to the language domain, both mechanisms transfer near-losslessly: EntropyInfer recovers 97–100% of baseline accuracy across all three duration categories (98.5% overall), and DynSparse matches or marginally exceeds baseline in every category (100.5% overall).

3.4 The domain asymmetry

Figure 1 summarizes the central pattern across both domains: mechanism recovery is consistently far higher in video-understanding than in language, and this ordering holds for both mechanisms independently. We measured block-selection density directly to check whether this asymmetry is attributable to a systematic density mismatch rather than selection quality: DynSparse operates at approximately 59% density on RULER but approximately 26% density on Video-MME the latter close to XAttention's own tuned range suggesting that the same fixed decision rule (a 90% cumulative-recall target) happens to land in a well-calibrated operating regime for video content but not for text, independent of any deliberate tuning on our part. This indicates the asymmetry reflects a genuine interaction between selection mechanism and domain structure, not merely an artifact of one domain being easier to achieve high density in.

4 Results

4.1 Language domain (RULER)

Table 1 reports accuracy across all six context lengths. Both ported mechanisms degrade relative to baseline at every length, but by markedly different margins. EntropyInfer recovers 77–89% of baseline accuracy; DynSparse recovers 91–99.6%. Both mechanisms show their largest relative degradation at 128K (EntropyInfer: 77% recovery; DynSparse: 91.2% recovery), consistent with degradation compounding as the number of per-sequence selection decisions grows.

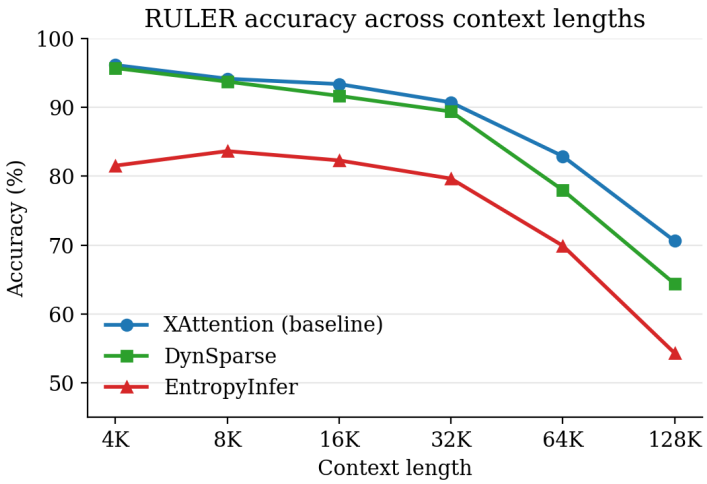


Figure 2: RULER accuracy across context lengths for XAttention (baseline), DynSparse, and EntropyInfer. Both ported mechanisms degrade relative to baseline at every length, with the gap widening substantially at longer contexts.

Table 3: RULER per-task accuracy at 128K context length.

Task	XAttention	EntropyInfer	DynSparse
niah_single_1	100.0	100.0	100.0
niah_single_2	98.0	66.0	95.0
niah_single_3	100.0	89.0	98.0
niah_multikey_1	92.0	57.0	79.0
niah_multikey_2	63.0	17.0	44.0
niah_multikey_3	34.0	4.0	20.0
niah_multivalue	92.75	83.5	88.25
niah_multiquery	97.75	80.5	93.75
vt	61.0	78.6	65.0
cwe	1.3	0.4	1.7
fwe	74.0	76.33	75.33
qa_1	65.0	27.0	41.0
qa_2	39.0	27.0	36.0
Average	**70.60**	**54.33**	**64.39**

Table 4: RULER per-task accuracy at 8K context length.

Task	XAttention	EntropyInfer	DynSparse
niah_single_1	100.0	100.0	100.0
niah_single_2	100.0	99.0	100.0
niah_single_3	100.0	97.0	100.0
niah_multikey_1	100.0	90.0	100.0
niah_multikey_2	100.0	45.0	99.0
niah_multikey_3	98.0	68.0	96.0
niah_multivalue	99.5	92.25	99.0
niah_multiquery	100.0	94.25	99.0
vt	98.8	96.4	99.2
cwe	94.9	96.8	93.2
fwe	87.67	87.67	89.0
qa_1	87.0	70.0	86.0
qa_2	58.0	51.0	58.0
Average	**94.14**	**83.64**	**93.72**

Table 5: Recovery (% of XAttention baseline accuracy) by mechanism and domain. Language recovery is the unweighted mean of per-context-length recovery ratios; the ≤32K column restricts to lengths comparable to the video setting. Video recovery is the overall accuracy ratio.

Method	Language (all lengths)	Language (≤32K)	Video (overall)
EntropyInfer	85.2%	87.4%	98.5%
DynSparse	96.8%	99.0%	100.5%

Tables 3 and 4 report per-task results at the longest and a representative shorter context length; full per-task results for all six lengths are provided in Appendix B.

4.2 Video-understanding domain (Video-MME)

Table 2 reports accuracy across the full 900-video benchmark. Both mechanisms transfer near-losslessly: EntropyInfer recovers 97–100% of baseline across all three duration categories (98.5% overall); DynSparse matches or marginally exceeds baseline in every category (100.5% overall).

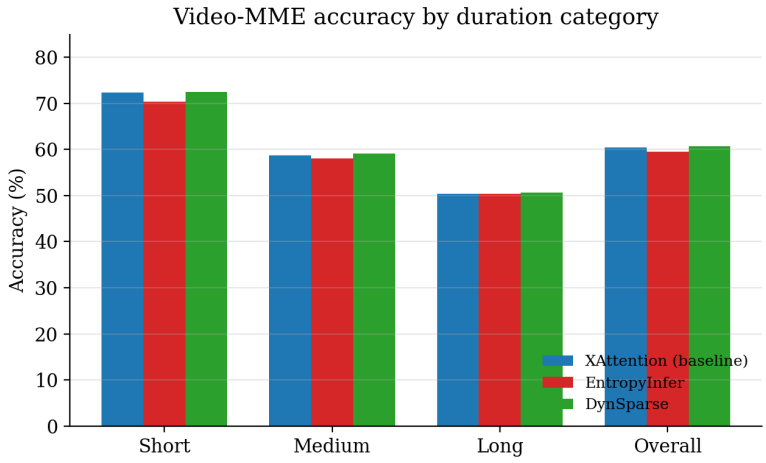


Figure 3: Video-MME accuracy by duration category. Unlike the language domain, all three methods perform nearly identically across short, medium, and long videos.

4.3 Cross-domain asymmetry

The central result is summarized in Table 5: mechanism recovery is consistently far higher in video-understanding than in language, for both mechanisms independently.

To rule out a trivial density explanation for this asymmetry, we measured block-selection density directly on real inputs. DynSparse operates at approximately 59% density on RULER but approximately 26% density on Video-MME the latter close to XAttention's own tuned baseline range (20–30%). The same fixed decision rule (90% cumulative-recall target) lands in a well-calibrated operating regime for video content but not for text, without any deliberate tuning on our part. This indicates the asymmetry reflects a genuine interaction between selection mechanism and domain structure, rather than an artifact of one domain being trivially easier to achieve high accuracy in at any density.

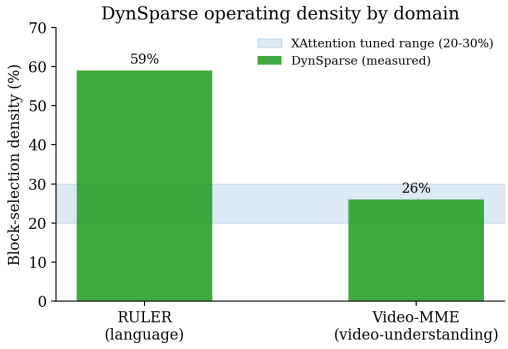


Figure 4: DynSparse's measured block-selection density differs substantially by domain, landing within XAttention's own tuned range for video-understanding but well above it for language.

5 Conclusion

We tested whether training-free sparse attention mechanisms, published and validated in a single domain, retain their effectiveness when transferred unmodified into a different implementation and a different domain entirely. Porting two independently-developed mechanisms, EntropyInfer and DynSparse, into XAttention's block-selection interface, we found a consistent and striking asymmetry: both mechanisms degrade substantially in long-context language modeling, particularly at long context lengths, while transferring near-losslessly to long-video question answering. This pattern held across both mechanisms, and direct measurement of block-selection density confirmed it is not an artifact of a systematic density mismatch, but reflects a genuine, domain-dependent difference in selection quality. Our results indicate that "calibration-free" claims in the sparse attention literature should be understood as domain-contingent rather than universal, and that the *quality* of what a mechanism selects not merely how much it retains is a meaningful and, until now, largely unmeasured axis along which these methods can fail to generalize.

6 Future work

Several directions follow directly from the limitations of this study. First, our video-understanding evaluation used a fixed 64-frame cap to remain within single-GPU memory constraints; because our language-domain results show recovery degrading with context length, disentangling the effect of sequence length from the effect of domain/modality is an open question, and would require evaluating video-understanding at longer, uncapped context lengths, likely necessitating a more memory-efficient block-scoring implementation than the direct one we build on. Second, our evaluation uses one model per domain (Llama-3.1-8B and Qwen2-VL-7B); testing whether the observed asymmetry holds across additional model families and scales would strengthen the generality of our claim. Third, we evaluate two selection mechanisms; extending this methodology to additional training-free mechanisms would clarify whether the language–video asymmetry we observe is a property of this mechanism class broadly or specific to the two we tested. Finally, our study is restricted to language and video-understanding; extending the same transfer test to video-generation a domain with non-causal attention structure that neither mechanism nor our current integration currently supports would complete the picture across the three domains XAttention itself was originally evaluated on. We view this work as establishing a methodology for testing calibration-free claims empirically, rather than as a final answer to when and why such transfer succeeds or fails; we hope it motivates more careful, domain-aware evaluation of future training-free sparse attention methods.

AI Use Statement

In this work, we used generative AI tools (Claude, Anthropic) to assist with writing and polish aid, and to help draft portions of the manuscript text, which were subsequently reviewed and revised by the authors. We have not used generative AI tools to generate experimental results, run experiments

autonomously, or make claims about our findings; all reported numbers were independently checked by the authors against the underlying log files and evaluation outputs before inclusion in this paper. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Ethics Statement

This work evaluates existing, publicly available sparse attention mechanisms and pretrained language models (Llama-3.1-8B-Instruct, Qwen2-VL-7B-Instruct) on publicly available benchmarks (RULER, Video-MME). We do not collect, use, or release any new data involving human subjects, and we do not introduce new capabilities for harm beyond what is already present in the underlying models and mechanisms we study. Our contribution is empirical and diagnostic in nature; we do not foresee direct negative societal applications specific to this work. We note that our findings that training-free sparse attention mechanisms may not transfer reliably across domains without validation have a positive implication for responsible deployment: practitioners adopting such methods in new settings should not assume calibration-free claims hold universally, and should validate mechanism behavior empirically before deployment in a new domain, rather than assuming an off-the-shelf configuration is safe or accurate.

Reproducibility Statement

We provide full experimental setup details, including hardware and software configuration, exact context lengths and sample sizes, model and benchmark configurations, and our selection-mechanism calibration procedure, in Appendix A. Complete per-task and per-category results underlying every summary statistic reported in the main text are provided in Section 4 and the appendix tables. Our verification procedure for confirming that ported mechanisms are genuinely active at inference time (rather than silently reverting to baseline behavior) is described in Appendix A. We intend to release our code, including the ported selection mechanisms, the patching infrastructure, and the vision-language model integration described in this work, upon publication.

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Zhuoming Chen, Ranajoy Sadhukhan, Zihao Ye, Yang Zhou, Jianyu Zhang, Niklas Nolte, Yuandong Tian, Matthijs Douze, Léon Bottou, Zhihao Jia, and Beidi Chen. MagicPIG: LSH sampling for efficient LLM generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*, 2024.

Jiayu Ding, Shuming Ma, Li Dong, Xingxing Zhang, Shaohan Huang, Wenhui Wang, Nanning Zheng, and Furu Wei. LongNet: Scaling transformers to 1,000,000,000 tokens. *arXiv preprint arXiv:2307.02486*, 2023.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. 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*, 2025.

Yizhao Gao, Zhichen Zeng, Dayou Du, Shijie Cao, Peiyuan Zhou, Jiaxing Qi, Junjie Lai, Hayden Kwok-Hay So, Ting Cao, Fan Yang, et al. Seerattention: Learning intrinsic sparse attention in your LLMs. *arXiv preprint arXiv:2410.13276*, 2024.

Yizhao Gao, Shuming Guo, Shijie Cao, Yuqing Xia, Yu Cheng, Lei Wang, Lingxiao Ma, Yutao Sun, Tianzhu Ye, Li Dong, Hayden Kwok-Hay So, Yu Hua, Ting Cao, Fan Yang, and Mao Yang. SeerAttention-R: Sparse attention adaptation for long reasoning. *arXiv preprint arXiv:2506.08889*, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Junxian Guo, Haotian Tang, Shang Yang, Zhekai Zhang, Zhijian Liu, and Song Han. Block sparse attention, 2024. `https://github.com/mit-han-lab/Block-Sparse-Attention`.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 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.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020.

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

Chunlu Li, Yixuan Pan, Bai Du, Zhenyuan Chen, Yanzhao Li, Hui Dong, Hui Wang, and Zhiqiang Zou. Training-free sparse attention based on cumulative energy filtering. *arXiv preprint arXiv:2606.16317*, 2026.

Di Liu, Meng Chen, Baotong Lu, Huiqiang Jiang, Zhenhua Han, Qianxi Zhang, Qi Chen, Chengruidong Zhang, Bailu Ding, Kai Zhang, Chen Chen, Fan Yang, Yuqing Yang, and Lili Qiu. Retrievalattention: Accelerating long-context LLM inference via vector retrieval. *arXiv preprint arXiv:2409.10516*, 2024a.

Guangda Liu, Chengwei Li, Jieru Zhao, Chenqi Zhang, and Minyi Guo. ClusterKV: Manipulating LLM KV cache in semantic space for recallable compression. *arXiv preprint arXiv:2412.03213*, 2024b.

Enzhe Lu, Zhejun Jiang, Jingyuan Liu, Yulun Du, Tao Jiang, Chao Hong, Shaowei Liu, Weiran He, Enming Yuan, Yuzhi Wang, et al. MoBA: Mixture of block attention for long-context LLMs. *arXiv preprint arXiv:2502.13189*, 2025.

Amirkeivan Mohtashami and Martin Jaggi. Random-access infinite context length for transformers. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Piotr Nawrot, Robert Li, Renjie Huang, Sebastian Ruder, Kelly Marchisio, and Edoardo M. Ponti. The sparse frontier: Sparse attention trade-offs in transformer LLMs. *arXiv preprint arXiv:2504.17768*, 2025.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021.

Yutao Sun, Tianzhu Ye, Li Dong, Yuqing Xia, Jian Chen, Yizhao Gao, Shijie Cao, Jianyong Wang, and Furu Wei. Rectified sparse attention. *arXiv preprint arXiv:2506.04108*, 2025.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. QUEST: Query-aware sparsity for efficient long-context LLM inference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*. PMLR, 2024.

Lei Wang, Yu Cheng, Yining Shi, Zhengju Tang, Zhiwen Mo, Wenhao Xie, Lingxiao Ma, Yuqing Xia, Jilong Xue, Fan Yang, and Zhi Yang. TileLang: A composable tiled programming model for AI systems. *arXiv preprint arXiv:2504.17577*, 2025.

Ningning Wang, Guobing Gan, Peng Zhang, Shuai Zhang, Junqiu Wei, Qun Liu, and Xin Jiang. ClusterFormer: Neural clustering attention for efficient and effective transformer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2390–2402, 2022.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, 2024.

Ruyi Xu, Guangxuan Xiao, Haofeng Huang, Junxian Guo, and Song Han. XAttention: Block sparse attention with antidiagonal scoring. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 69819–69831. PMLR, 2025.

Zhanchao Xu, Haoyang Li, Qingfa Xiao, Fei Teng, Chen Jason Zhang, Lei Chen, and Qing Li. From rigid to dynamic: Entropy-guided adaptive inference for long-context LLMs. *arXiv preprint arXiv:2606.09508*, 2026.

Shuo Yang, Ying Sheng, Joseph E. Gonzalez, Ion Stoica, and Lianmin Zheng. Post-training sparse attention with double sparsity. *arXiv preprint arXiv:2408.07092*, 2024.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Y. X. Wei, Lean Wang, Zhiping Xiao, Yuqing Wang, Chong Ruan, Ming Zhang, Wenfeng Liang, and Wangding Zeng. Native sparse attention: Hardware-aligned and natively trainable sparse attention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17283–17297, 2020.

Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. SpargeAttention: Accurate and training-free sparse attention accelerating any model inference. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*. PMLR, 2025.

A Experimental setup details

We report full reproducibility details here, complementing the summary given in Section 3.1.

A.1 Hardware and software

All experiments were run on a single NVIDIA A100 (80GB) GPU. We use PyTorch 2.4.0 (CUDA 12.1), FlashAttention 2.6.3, and Transformers 4.46.0. XAttention's block-sparse attention kernels are compiled against this environment; we additionally build MInference and Block-Sparse-Attention from source to support the underlying framework's baseline comparisons.

A.2 LANGUAGE DOMAIN: RULER

We evaluate Llama-3.1-8B-Instruct on the RULER benchmark's synthetic task suite (13 tasks: eight needle-in-a-haystack variants, variable tracking, common word extraction, frequency word extraction, and two question-answering tasks), at six context lengths: 4,096, 8,192, 16,384, 32,768, 65,536, and 131,072 tokens, with 100 samples per task per length. XAttention is configured with stride S=16; the block size is fixed at 128 tokens throughout. Generation uses greedy decoding (temperature 0.0) with a maximum of 128 new tokens.

A.3 VIDEO-UNDERSTANDING DOMAIN: VIDEO-MME

We evaluate Qwen2-VL-7B-Instruct (Wang et al., 2024) on the full Video-MME benchmark (900 videos, 2,700 multiple-choice questions, stratified across short/medium/long duration categories), scored via exact-match against the ground-truth answer choice, following the benchmark's own `exact_matching` evaluation protocol (no LLM judge). We locate and validate XAttention's own block-sparse attention integration for Qwen2-VL (present in the framework's codebase but not surfaced by its top-level evaluation scripts), which applies sparse attention from the fourth transformer layer onward (layer_idx $>$ 2) while retaining dense attention in the first three layers. Frame sampling is capped at 64 frames per video; this keeps all sequences within Qwen2-VL's trained context window of 32,768 tokens. Generation uses greedy decoding with a maximum of 8 new tokens, sufficient for a single multiple-choice letter response.

A.4 SELECTION MECHANISM CONFIGURATION

EntropyInfer's per-head, per-segment budget walk and DynSparse's cumulative-recall threshold are both implemented following their original published specifications, substituted directly for XAttention's threshold-based `find_blocks_chunked` selection function at the identical call site, so that both ported mechanisms inherit XAttention's surrounding kernel implementation (antidiagonal scoring, causal masking, block-sparse computation) unchanged. Neither mechanism's original budget parameter translates directly into XAttention's block-based interface (EntropyInfer specifies an absolute token count; DynSparse specifies a fixed recall target with no inherent scale dependence). We calibrate EntropyInfer's budget fraction so that its measured block-selection density matches XAttention's own tuned operating density (20–30%, depending on context length and domain); DynSparse's recall target (90%) is used unmodified, as its recall-based formulation is scale-invariant by construction and its measured density in both domains is reported directly in Section 4.3.

A.5 VERIFICATION PROCEDURE

To confirm that each ported mechanism is genuinely active at inference time, rather than silently reverting to baseline behavior, we verify at three levels: (i) direct inspection of the patched selection function's identity (memory address) before and after patching, confirmed consistent across both internal module namespaces the function is referenced from; (ii) confirmation of mechanism-specific timing signatures distinct from baseline, consistent with each mechanism's differing computational overhead; and (iii) direct measurement of block-selection density on real inputs in both domains, reported in Section 4.3.

B LANGUAGE DOMAIN (RULER)

We report accuracy for XAttention (baseline), EntropyInfer, and DynSparse across all six RULER context lengths, broken down by individual task. Tables 6–11 present these results in order of increasing context length, from 4K to 128K tokens, allowing task-level comparison both within and across lengths.

Table 6: RULER per-task accuracy at 4K context length.

Task	XAttention	EntropyInfer	DynSparse
niah_single_1	100.0	100.0	100.0
niah_single_2	100.0	98.0	100.0
niah_single_3	100.0	96.0	99.0
niah_multikey_1	100.0	63.0	100.0
niah_multikey_2	100.0	41.0	100.0
niah_multikey_3	100.0	63.0	99.0
niah_multivalue	99.5	88.0	99.0
niah_multiquery	100.0	94.0	100.0
vt	99.2	77.6	99.6
cwe	99.5	99.9	99.6
fwe	95.33	97.33	96.67
qa_1	92.0	81.0	88.0
qa_2	64.0	61.0	63.0
Average	**96.12**	**81.53**	**95.68**

Table 7: RULER per-task accuracy at 8K context length.

Task	XAttention	EntropyInfer	DynSparse
niah_single_1	100.0	100.0	100.0
niah_single_2	100.0	99.0	100.0
niah_single_3	100.0	97.0	100.0
niah_multikey_1	100.0	90.0	100.0
niah_multikey_2	100.0	45.0	99.0
niah_multikey_3	98.0	68.0	96.0
niah_multivalue	99.5	92.25	99.0
niah_multiquery	100.0	94.25	99.0
vt	98.8	96.4	99.2
cwe	94.9	96.8	93.2
fwe	87.67	87.67	89.0
qa_1	87.0	70.0	86.0
qa_2	58.0	51.0	58.0
Average	**94.14**	**83.64**	**93.72**

Table 8: RULER per-task accuracy at 16K context length.

Task	XAttention	EntropyInfer	DynSparse
niah_single_1	100.0	100.0	100.0
niah_single_2	100.0	99.0	100.0
niah_single_3	100.0	95.0	100.0
niah_multikey_1	100.0	91.0	100.0
niah_multikey_2	99.0	42.0	91.0
niah_multikey_3	100.0	63.0	92.0
niah_multivalue	98.5	95.5	98.75
niah_multiquery	99.5	95.25	99.0
vt	99.8	95.8	99.0
cwe	85.2	89.5	85.0
fwe	90.67	88.67	90.67
qa_1	86.0	65.0	80.0
qa_2	55.0	50.0	56.0
Average	**93.36**	**82.29**	**91.65**

Table 9: RULER per-task accuracy at 32K context length.

Task	XAttention	EntropyInfer	DynSparse
niah_single_1	100.0	100.0	100.0
niah_single_2	100.0	99.0	99.0
niah_single_3	100.0	100.0	100.0
niah_multikey_1	98.0	83.0	100.0
niah_multikey_2	97.0	32.0	86.0
niah_multikey_3	97.0	59.0	92.0
niah_multivalue	98.5	95.75	98.0
niah_multiquery	99.0	97.25	98.0
vt	99.4	97.0	98.0
cwe	60.7	70.8	60.3
fwe	93.67	93.67	93.67
qa_1	83.0	55.0	83.0
qa_2	53.0	53.0	54.0
Average	**90.71**	**79.65**	**89.38**

Table 10: RULER per-task accuracy at 64K context length.

Task	XAttention	EntropyInfer	DynSparse
niah_single_1	100.0	100.0	100.0
niah_single_2	98.0	90.0	97.0
niah_single_3	100.0	96.0	100.0
niah_multikey_1	100.0	85.0	97.0
niah_multikey_2	85.0	28.0	58.0
niah_multikey_3	90.0	44.0	75.0
niah_multivalue	94.25	90.75	92.25
niah_multiquery	99.5	96.5	98.0
vt	92.4	91.2	92.2
cwe	14.2	17.1	11.0
fwe	86.33	85.33	86.33
qa_1	69.0	44.0	60.0
qa_2	49.0	41.0	47.0
Average	**82.90**	**69.91**	**77.98**

Table 11: RULER per-task accuracy at 128K context length.

Task	XAttention	EntropyInfer	DynSparse
niah_single_1	100.0	100.0	100.0
niah_single_2	98.0	66.0	95.0
niah_single_3	100.0	89.0	98.0
niah_multikey_1	92.0	57.0	79.0
niah_multikey_2	63.0	17.0	44.0
niah_multikey_3	34.0	4.0	20.0
niah_multivalue	92.75	83.5	88.25
niah_multiquery	97.75	80.5	93.75
vt	61.0	78.6	65.0
cwe	1.3	0.4	1.7
fwe	74.0	76.33	75.33
qa_1	65.0	27.0	41.0
qa_2	39.0	27.0	36.0
Average	**70.60**	**54.33**	**64.39**

Table 6 through Table 11 show that both ported mechanisms degrade relative to baseline at every length, but by markedly different margins. EntropyInfer recovers 77–89% of baseline accuracy;

DynSparse recovers 91–99.6%. Both mechanisms show their largest relative degradation at 128K (EntropyInfer:77% recovery; DynSparse: 91.2% recovery), consistent with degradation compounding as the number of per-sequence selection decisions grows.

C SELECTION MECHANISM BUDGET CALIBRATION

This appendix details the process by which we resolved each mechanism's budget parameter into XAttention's block-based selection interface, referenced briefly in Appendix A. We include this level of detail for full transparency and reproducibility, since neither mechanism's original specification directly addresses how its budget should be interpreted at a different block granularity than the one it was introduced in.

C.1 ENTROPYINFER: AN INITIAL UNITS MISMATCH

EntropyInfer's published specification sets its base budget parameter as an absolute count of 2,048 tokens, calibrated for the language contexts (tens to hundreds of thousands of tokens) used in its original evaluation. Our first implementation translated this literally: we expressed the budget as a fixed fraction of available blocks, derived directly from this constant relative to a representative context length ($2048/100{,}000 \approx 2\%$).

Applying this fraction at RULER's actual block counts (256 to 1,024 blocks, depending on context length) produced a measured selection density of approximately 2% at every context length tested roughly an order of magnitude below XAttention's own tuned operating density (20–30%). We diagnosed this directly by instrumenting the block-selection function to record the fraction of blocks retained per call, confirming the discrepancy was a systematic under-provisioning of context rather than a per-input anomaly. At this density, we observed a substantial accuracy collapse relative to baseline (roughly a 50% relative drop, averaged across context lengths), with degradation concentrated in tasks requiring retrieval of specific, sparsely-distributed content.

Because a selection mechanism results conflate two distinct effects how much context is retained (budget) and which context is retained (selection quality) we judged that this initial result was not a meaningful test of EntropyInfer's selection mechanism on its own terms: at 2% density, the mechanism was effectively starved of context regardless of how well it chose among the blocks available to it. We therefore recalibrated the budget fraction (to approximately 20%) so that EntropyInfer's measured density matched XAttention's own tuned operating range, isolating selection quality as the variable of interest. This is the configuration used for all EntropyInfer results reported in the main text.

We note that this recalibration is itself a form of intervention beyond EntropyInfer's original, published specification, and is disclosed accordingly: our results characterize EntropyInfer's selection mechanism under a budget deliberately matched to a fair operating point, not the mechanism's behavior under a literal, unmodified translation of its published parameter, which we found to be non-viable.

C.2 DYNSPARSE: NO ANALOGOUS ADJUSTMENT WAS NEEDED

DynSparse's budget parameter (a fixed cumulative-recall target of 90%) is dimensionless by construction and required no comparable translation. We nonetheless measured its resulting density directly, in both domains, as an additional diagnostic. On RULER, DynSparse's measured density was approximately 59% across all context lengths tested notably higher than XAttention's own tuned range, though this over-provisioning did not require correction, since a scale-invariant recall target has no obvious "wrong" translation to adjust. On Video-MME, the same fixed recall target produced a substantially different measured density of approximately 26%, close to XAttention's own tuned range for that domain. We use DynSparse's recall target unmodified throughout; the density figures above are reported as a diagnostic explanation for the cross-domain asymmetry discussed in Section 4.3, not as evidence of any correction applied to the mechanism itself.