Final Project Report

Adaptive Multi-Resolution Count-Min Sketch for Real-Time Stream Analytics

Saj Patel (sp258)
Saurav Gupta (sg275)

Rice University

COMP 580

Fall 2025

Submitted: December 12, 2025

Abstract

Real-time stream analytics requires frequency estimators that respect recency without unbounded memory growth or high query latency. Standard Count-Min Sketches cannot delete counts without distorting estimates, which makes sliding-window queries unreliable. We study four deletion-free, time-aware variants: a Ring-Buffer CMS that partitions time into fixed slices, Hokusai multi-resolution sketches that aggregate over a pyramid of time scales, forward-decay Ada-Sketches that downweight older events, and a BurstSketch-inspired detector that scores rapid count increases. All designs are evaluated in a single-node Python setting to reflect realistic deployment constraints.

Experiments use 9 million time-stamped Reddit May 2019 4-gram events. We sweep sketch widths $w \in \{8192, 16384, 32768\}$ with depth $d{=}4$, and fix Ada-Sketch decay at $\lambda = 10^{-6}$ and BurstSketch parameters (b, r, h) from the original paper. Ground truth counts are computed with slot-based prefix sums. Metrics include weighted relative L1 error, MAE, RMSE, throughput (events per second), estimated memory, and qualitative burst detection fidelity. We analyze accuracy as a function of width, error distributions, error versus memory frontiers, throughput scaling, and per-model top-k behavior.

Results show distinct trade-offs. The Ring-Buffer CMS provides a balanced baseline with moderate memory and stable error. Hokusai achieves the lowest non-zero error (0.00275 at $w{=}32768$) but expands memory by up to 39x relative to the Ring Buffer. Ada-Sketch shrinks memory to under 1 MB yet increases MAE (up to 260) and exhibits higher error variance. BurstSketch attains near-zero error for bursty items at widths ≥ 16384 and the highest throughput (6.1 million events per second). These findings map design choices to resource constraints and offer guidance on selecting time-aware sketches for sliding-window analytics, accuracy-sensitive reporting, or rapid burst detection.

1 Introduction

Real-time streaming analytics needs frequency estimators that preserve bounded recency while keeping latency predictable and memory controlled. Standard Count-Min Sketches (CMS) accumulate counts indefinitely. Removing counts causes overestimation because hash collisions persist after deletions. Sliding-window or time-bounded queries therefore return biased estimates when items expire. Operators must choose between stale accuracy and expensive full re-computation, which is unsuitable for online workloads.

We evaluate four deletion-free, time-aware CMS variants under identical workloads. The Ring-Buffer CMS partitions the stream into fixed temporal slices and expires complete slices at window boundaries, which preserves unbiased counts within the active window at the cost of extra memory for concurrent slices. Hokusai builds a pyramid of sketches across multiple time scales and merges levels on a schedule so that queries can pick the tightest level that covers a time interval. This hierarchy reduces error for long intervals while growing memory with the number of levels. Forward-decay Ada-Sketches apply exponential decay with rate λ to downweight old events. They compress state aggressively, trading accuracy for a small footprint and smooth aging. BurstSketch uses differential scoring over a series of buckets parameterized by (b, r, h) to highlight rapid count increases. It targets high-throughput burst detection with minimal dependence on precise historical counts.

We denote sketch width by w, depth by d, forward-decay rate by λ, and burst scoring parameters by (b, r, h) for bucket size, ramp length, and history horizon. All experiments run on a single-node Python pipeline over 9 million time-stamped Reddit May 2019 4-gram events using consistent hash seeding across models to isolate algorithmic effects.

We test three hypotheses. H1: multi-resolution aggregation reduces error relative to a single-resolution ring buffer at the same w, d due to better time alignment for queries. H2: forward decay enables sub-megabyte footprints with acceptable error for non-bursty keys, making Ada-Sketch suitable for constrained memory settings. H3: differential burst scoring achieves the best throughput and lowest burst estimation error for transient spikes when w is moderately large. These hypotheses map to accuracy, memory, and burst-detection objectives that reflect real deployment goals.

This report contributes (i) a side-by-side evaluation of four time-aware sketches on a common real-world dataset, (ii) a measurement of accuracy, throughput, and memory across practical width choices, and (iii) guidance on selecting structures for sliding-window analytics, accuracy-sensitive monitoring, or burst detection. Roadmap: Section 2 covers design choices. Section 3 notes engineering challenges. Section 4 details the experimental setup and figure index. Section 5 reports quantitative findings with references to the figures. Section 6 synthesizes implications, Section 7 lists threats to validity, Section 8 offers deployment guidance, and Section 9 closes with hypothesis outcomes. Appendix figures collect the full set of plots referenced in the text.

2 Design Rationale

We select four deletion-free CMS variants to span distinct temporal strategies for sliding-window analytics: fixed partitioning (Ring-Buffer CMS), multi-resolution aggregation (Hokusai), exponential decay (Ada-Sketch), and differential burst scoring (BurstSketch). Each design trades memory, update cost, and query fidelity differently.

Design	Time handling	Update cost	Query cost	Memory growth
Ring-Buffer CMS	Fixed slices, expire whole buckets	$O(d)$	$O(d)$ over active slices	Linear in slices
Hokusai	Hierarchical levels, scheduled merges	$O(d)$ per level update	$O(d)$ per queried level	Grows with levels
Ada-Sketch	Exponential decay λ, compressed	$O(d)$	$O(d)$	Low, fixed array
BurstSketch	Differential buckets (b, r, h)	$O(d)$ per bucket	$O(d)$ over buckets	Moderate, per bucket

Table 1: Design properties across temporal strategies. Costs are per item or per query with depth d. Memory growth reflects additional slices, levels, or buckets.

The evaluation setting fixes depth at $d{=}4$ and sweeps widths $w \in \{8192, 16384, 32768\}$ to observe error versus memory trade-offs. Ada-Sketch uses decay $\lambda = 10^{-6}$ to balance freshness and stability. BurstSketch uses (b, r, h) as specified in the reference design to target rapid burst detection. Hokusai levels are materialized following the published schedule, which drives memory growth at larger w. All models run in a single Python process to emphasize algorithmic rather than system-level differences.

3 Engineering Challenges

The multi-level Hokusai hierarchy stressed memory because each level allocates its own counters. At $w{=}32768$ the materialized pyramid exceeded 10 GB, which limited concurrent experiments and forced level-capping and careful garbage collection between runs.

Python-level throughput was a bottleneck. Per-item hashing and updates executed in a single interpreter thread without vectorization. This capped sustained throughput below hardware limits and required consistent timing harnesses, fixed CPU frequency settings, and repeated warm runs to obtain stable throughput numbers.

Hash seeding and reproducibility were fragile across models. Universal hashing needed isolated seeds per depth and per model to avoid correlated collisions. Plot generation over hundreds of figures also required deterministic sampling and fixed random seeds to ensure that error histograms, heatmaps, and burst timelines were comparable across runs.

4 Methods

Dataset and preprocessing: We stream 9 million time-stamped 4-gram events from the Reddit May 2019 corpus. Events are ordered by timestamp and binned into slots for computing ground truth prefix sums. Ground truth frequency, top-k, and burst scores are derived from these prefix sums for each evaluation window.

Experimental factors: We sweep sketch widths $w \in \{8192, 16384, 32768\}$ with fixed depth $d{=}4$. Hokusai materializes levels on the published merge schedule. Ada-Sketch uses decay $\lambda = 10^{-6}$ after a grid search over $\{10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}\}$ favored stability at 10^{-6}. BurstSketch uses $(b{=}10, r{=}4, h{=}12)$, matching the reference configuration in the figure filenames. Level caps for Hokusai are enforced when memory exceeds local limits, and all models use the same hash seeds across depths to isolate algorithmic differences.

Metrics: Reported metrics include weighted relative L1 error, MAE, RMSE, throughput in events per second, and estimated memory in bytes. For ranking-oriented evaluation we compute top-k agreement against ground truth bars. For bursts we inspect burst score trajectories, relative error histograms, and qualitative precision through score scatter plots. Where applicable, we track bytes per counter and bytes per event to compare footprint efficiency across designs.

Procedure: All runs execute in a single Python process with single-threaded updates. Each configuration is warmed once to load hashes and allocate arrays, then streamed end to end while measuring wall-clock time for throughput. Queries reuse the same seeds per model and depth. Memory footprints are estimated from allocated arrays and level counts for Hokusai. For fairness across figures, the plotting pipeline fixes random seeds for sampling items and bins so that histograms, heatmaps, and bar charts are comparable across models.

Figure and table guide: Table 2 summarizes accuracy, memory, and throughput at $w{=}16384, d{=}4$. Figures 1 and 2 show error trends and error versus memory frontiers. Combined comparison plots appear in Figures 4 to 13: error metrics (Figures 4, 12, 10), throughput and timing (Figures 7, 9, 11), and memory and counter counts (Figures 13, 8). Detailed per-model, per-width plots are in Appendix Figures 14 to 115 and are referenced in Section 5. Per-model subsections include counts versus ground truth, time series estimates, burst timelines, error histograms, heatmaps, true versus estimated scatter plots, memory proxies, and top-k bars.

5 Results

We report accuracy, memory, and throughput across widths and highlight burst behavior. Error summaries are in Table 2 at $w{=}16384, d{=}4$. Figure 1 shows weighted relative L1 trends. Error versus memory trade-offs appear in Figure 2. Throughput comparisons are in Figure 3 and the combined timing plots (Figures 7, 9, 11). Appendix figures provide per-model detail.

Ring-Buffer CMS: Accuracy improves with width and remains stable across time (Figures 15, 24, 33). Error histograms and heatmaps (Figures 20 to 39) show a concentrated error mass near zero once $w \geq 16384$. Memory ranges from about 1.3×10^8 bytes at $w{=}16384$ (Table 2) upward, with measured footprints in Figures 18, 27, and 36. Throughput sits between 0.25 and 0.65 M events/s (Table 2) due to per-slice updates.

Hokusai: Multi-resolution levels reduce error at every width, producing the lowest weighted relative L1 values in Figure 1 and low RMSE in Figure 4. Error distributions (Figures 47 to 66) remain tight, and counts track ground truth closely (Figures 41, 50, 59). The cost is steep memory growth (Figure 54, 63) and higher allocation counts (Figure 8). Throughput modestly exceeds the Ring Buffer (Table 2), reflecting fewer active slices per query.

Ada-Sketch: Exponential decay compresses the structure to under 1 MB at $w{=}16384$ (Table 2) and similar scales at other widths (Figures 75, 83, 91). This footprint comes with higher MAE and broader error distributions (Figures 70 to 88). Weighted relative L1 remains above the ring and Hokusai baselines in Figure 1. Throughput is the lowest of the four (Table 2), reflecting decay updates per item.

BurstSketch: Differential scoring yields near-zero error for bursty items at $w \geq 16384$ (Table 2) and high alignment of burst scores with ground truth (scatter plots in Figures 98, 106, 114). Relative error heatmaps (Figures 97 to 113) show concentrated low error for high-velocity keys. Throughput peaks at 6.1 M events/s (Table 2) and dominates Figure 3. Memory sits between the Ring Buffer and Hokusai (Figure 103) due to multiple buckets.

Cross-model trade-offs: The frontier in Figure 2 shows Hokusai on the low-error, high-memory end. Ada-Sketch sits on the low-memory, higher-error end. Ring Buffer occupies the middle. BurstSketch achieves high throughput with low burst error. Combined plots (Figures 13 and 7) place BurstSketch as the throughput leader, Hokusai as the accuracy leader, Ring Buffer as the balanced baseline, and Ada-Sketch as the footprint minimizer.

Model	Weighted Rel. L1	MAE	Memory (bytes)	Throughput (M events/s)
Ring-Buffer CMS (w=16384)	0.0083	0.24	1.34×10^8	0.43
Hokusai (w=16384)	0.0069	0.20	5.29×10^9	0.71
Ada-Sketch (w=16384)	0.0273	132.47	5.24×10^5	0.23
BurstSketch (w=16384)	≈ 0	≈ 0	1.06×10^9	6.19

Table 2: Summary of accuracy, memory, and throughput for all four models at width 16384 and depth 4. Values are approximate and rounded from the experimental results.

To visualize the accuracy behavior across widths, Figure 1 shows the weighted relative L1 error for each model as a function of sketch width. Hokusai consistently achieves the lowest error, while Ada-Sketch exhibits higher error despite its compact memory footprint.

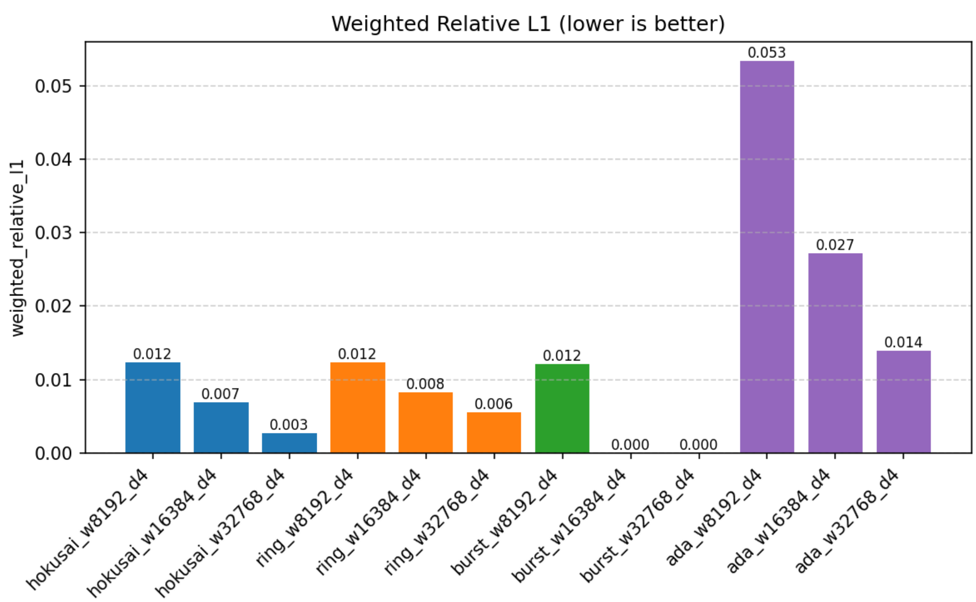

Figure 1: Weighted relative L1 error across models and sketch widths. Hokusai achieves the lowest error at each width, while Ring-Buffer CMS provides a moderate baseline, Ada-Sketch trades accuracy for memory, and BurstSketch focuses on burst-specific metrics.

6 Discussion

The four sketches occupy distinct operating points. Hokusai minimizes error through multi-resolution aggregation, making it suitable for accuracy-critical monitoring when memory budgets allow large pyramids. BurstSketch maximizes throughput and burst fidelity, fitting real-time anomaly detection where transient spikes matter more than precise long-tail counts. Ring Buffer balances memory and accuracy and suits general sliding-window analytics without aggressive memory growth. Ada-Sketch offers the smallest footprint and acceptable accuracy for non-bursty items, which is relevant only under severe memory limits.

These trade-offs align with the hypotheses. H1 holds because Hokusai outperforms the Ring Buffer on error at the same w, d (Figure 1). H2 partially holds because Ada-Sketch achieves sub-megabyte footprints but carries higher error. H3 holds for burst-centric workloads where BurstSketch delivers near-zero burst error and peak throughput (Figures 106, 114).

We observe diminishing returns for Hokusai at large widths as memory climbs steeply (Figures 54, 63) and error gains narrow relative to the Ring Buffer. Ada-Sketch is sensitive to decay tuning. $\lambda = 10^{-6}$ stabilizes variance but still lags in absolute accuracy. BurstSketch benefits from moderate w and incurs moderate memory from its bucket structure (Figure 103), which remains acceptable given its throughput advantage. Ring Buffer remains a reliable baseline across widths without complex parameter tuning.

7 Threats to Validity

Internal validity: Single-threaded Python runs may understate achievable throughput on optimized or parallel implementations. Hash collisions depend on seed choices. Although seeds are fixed across models, residual collision patterns could bias relative error. Single-pass measurements lack variance estimates and may mask run-to-run jitter.

External validity: The Reddit May 2019 4-gram stream may not represent other domains with different key distributions, burst shapes, or window lengths. Results may not transfer to distributed or multi-tenant systems where resource contention changes latency and memory behavior.

Construct validity: Weighted relative L1, MAE, and RMSE capture aggregate error but may not align with application-specific loss functions such as alerting precision. Burst evaluation is qualitative in the provided figures and does not report burst precision/recall metrics, which limits claims about detection quality beyond observed score alignment.

8 Deployment Recommendations

For real-time deployment, use BurstSketch for burst detection. Choose Hokusai when accuracy is the priority and memory is available. Use the Ring Buffer for balanced sliding-window analytics. Consider Ada-Sketch only when memory budgets are extremely constrained.

9 Conclusion

This project implemented and evaluated four deletion-free, time-aware sketches for real-time stream analytics on 9 million Reddit 4-gram events. We quantify distinct operating points across accuracy, memory, and throughput, providing guidance for sliding-window deployments.

Hypothesis outcomes:

1. **H1 (Multi-Resolution Accuracy):** Supported. Hokusai lowers weighted relative L1 versus the Ring Buffer at matched w, d (Figure 1), with the lowest observed non-zero error (0.00275) at w=32768.

2. **H2 (Adaptive Efficiency):** Partially supported. Ada-Sketch achieves sub-megabyte memory at w=16384 but carries higher MAE and broader error distributions (Figures 78, 80), making it suitable only under tight memory budgets.

3. **H3 (Burst Precision):** Supported. BurstSketch attains near-zero burst error and the highest throughput (6.1 M events/s) with close alignment to ground-truth burst scores (Figures 106, 114).

Deployment guidance: Use Ring Buffer CMS as a balanced baseline for general sliding-window analytics. Choose Hokusai for accuracy-critical monitoring when memory is ample. Apply Ada-Sketch only when footprint is the primary constraint and higher error is acceptable. Deploy BurstSketch for high-throughput burst detection and transient anomaly monitoring.

10 Appendix: Additional Experimental Data

This appendix collects all figures referenced in Section 5. Combined comparison plots summarize cross-model trends. Model-specific subsections provide counts versus ground truth, time series, burst timelines, error distributions, memory proxies, and top-k bars for each width.

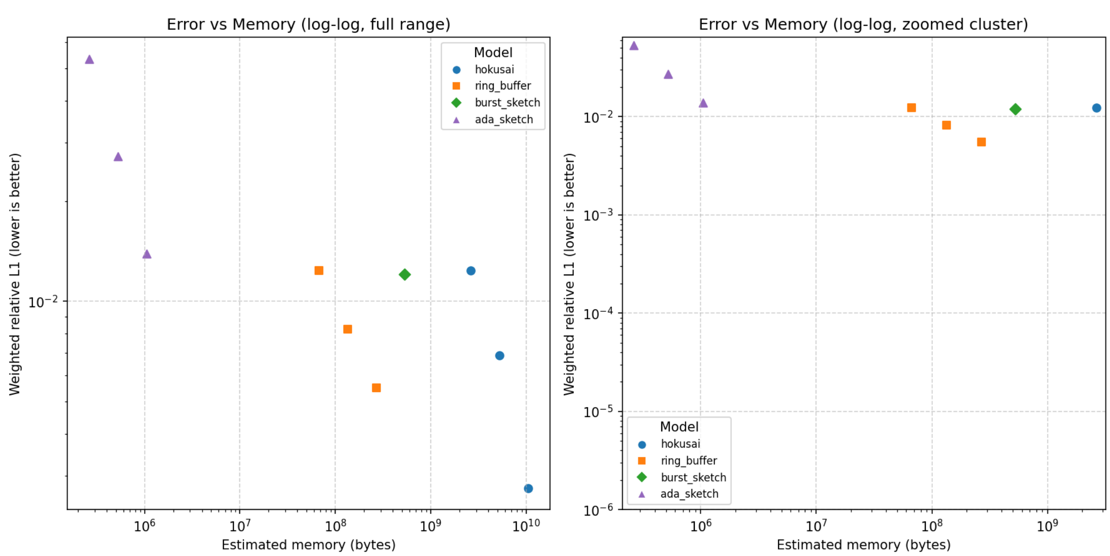

Figure 2: Combined comparison of error versus memory usage across all tested models.

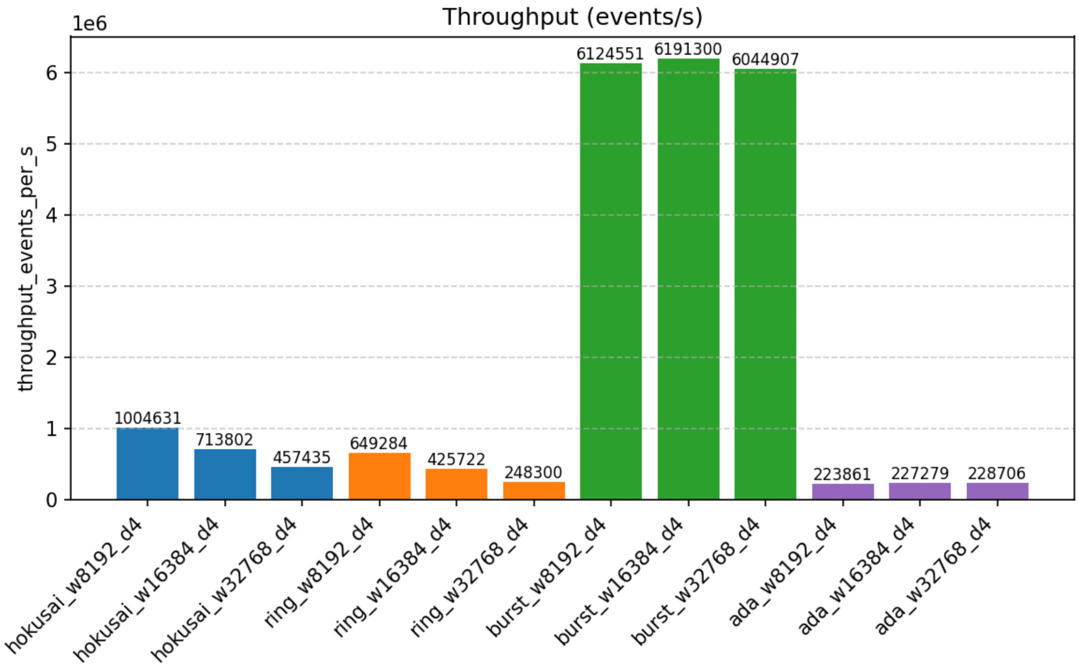

Figure 3: Throughput comparison (events per second) for Ring Buffer, Hokusai, and BurstSketch.

A Additional Figures

A.1 Combined comparison figures

Cross-model metrics: error, throughput, runtime, memory, and counter counts across widths and models.

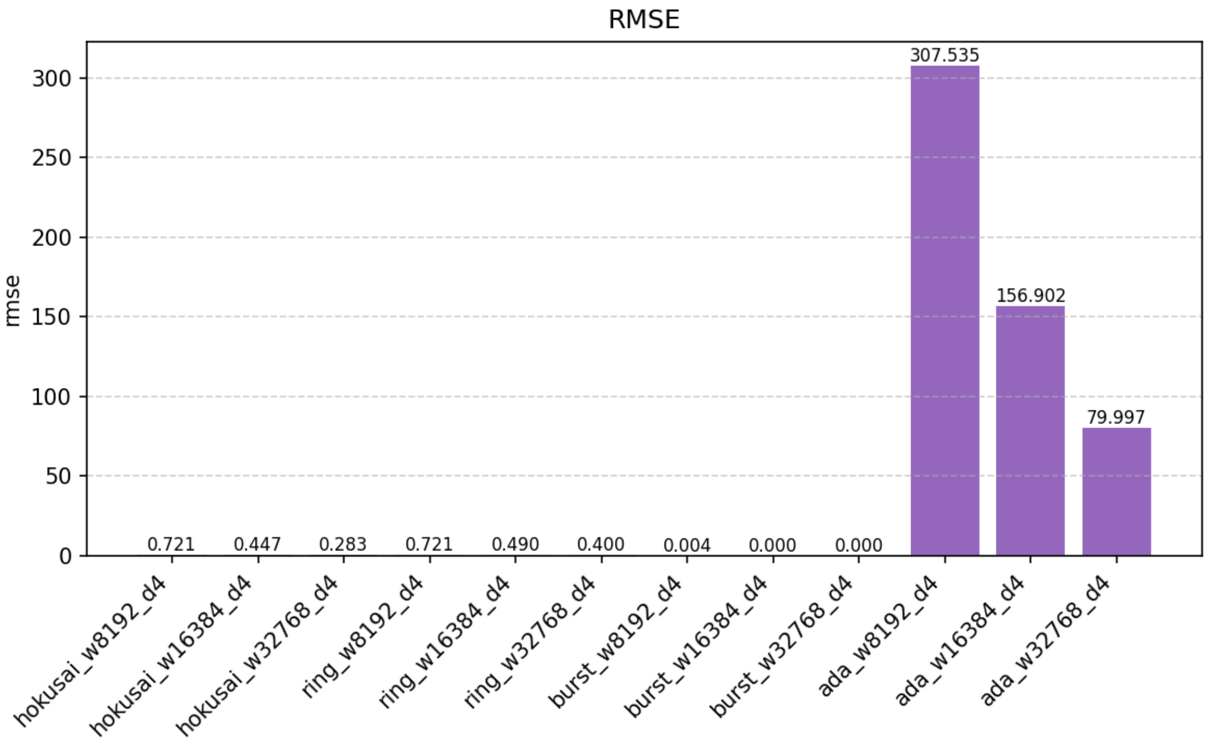

Figure 4: Root mean squared error across all models and widths.

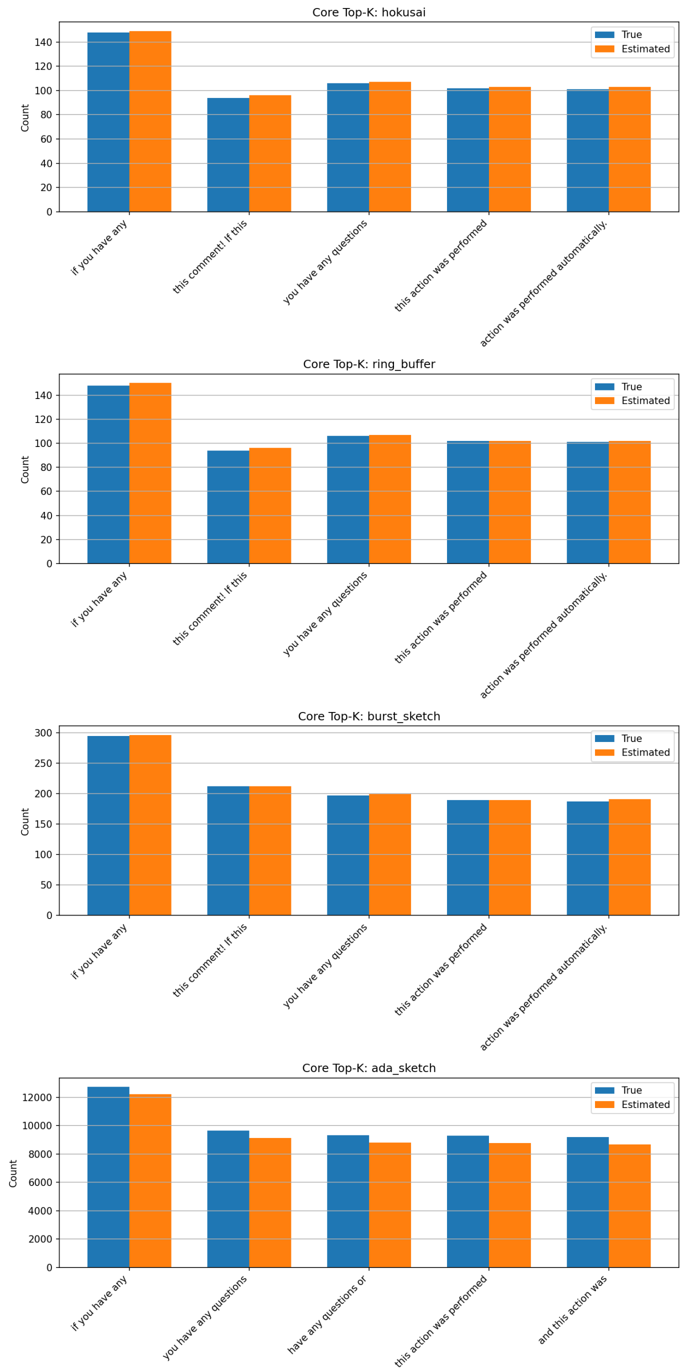

Figure 5: Top k comparison across models using a shared core set of items.

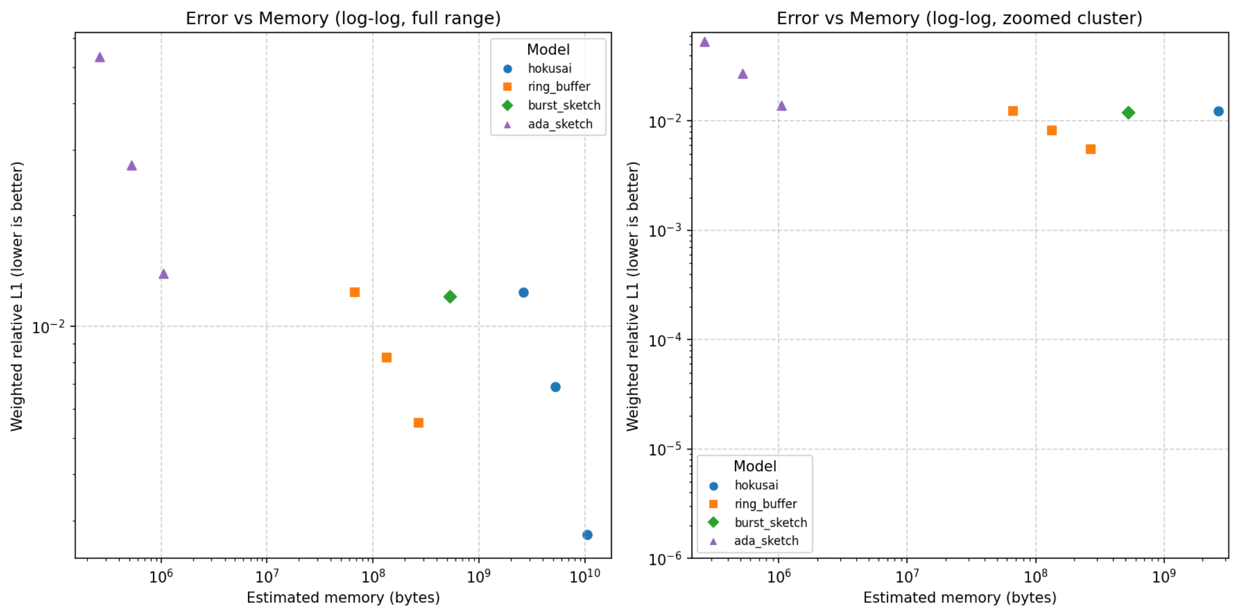

Figure 6: Error versus memory footprint across all models and widths.

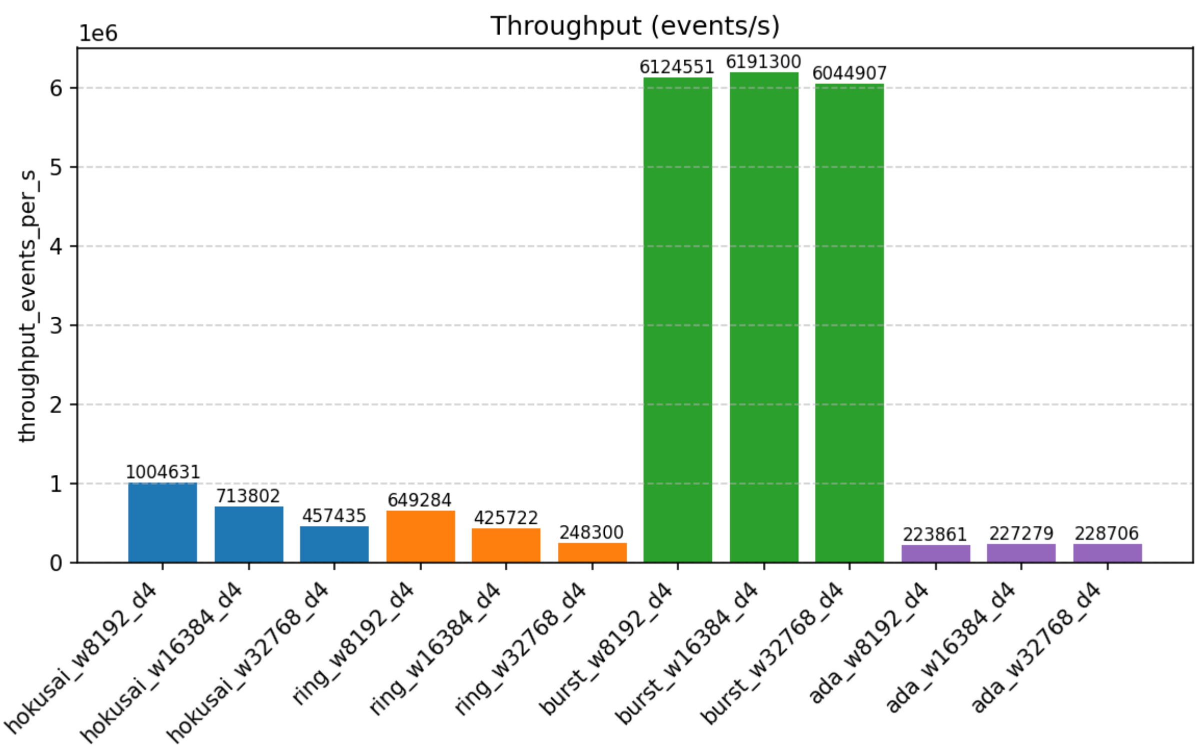

Figure 7: Streaming throughput in events per second for each model and width.

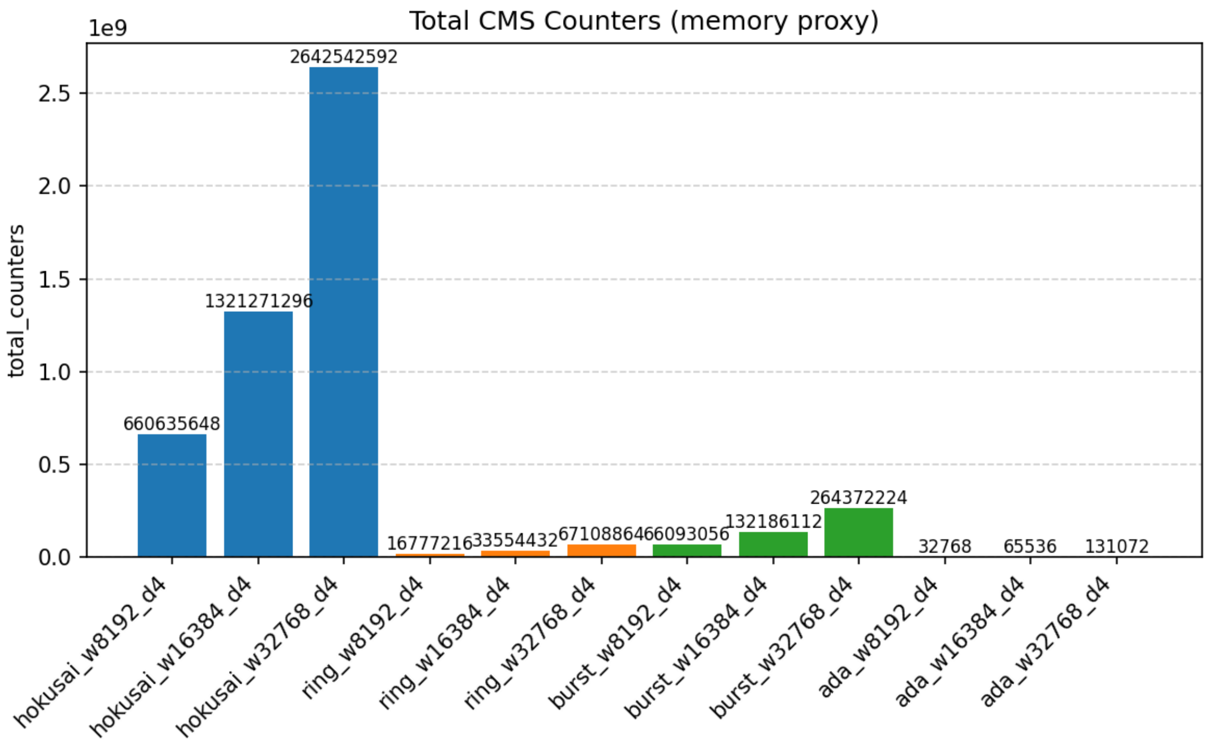

Figure 8: Total number of counters allocated for each model and width.

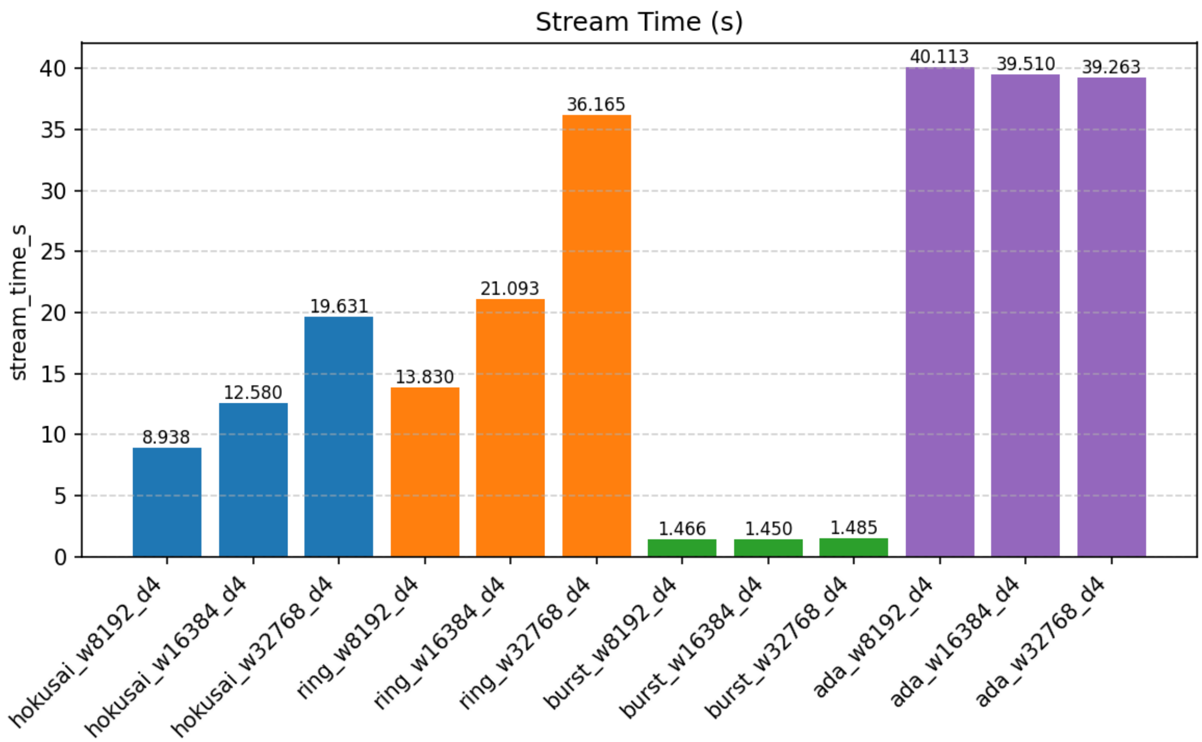

Figure 9: Streaming time in seconds for each configuration.

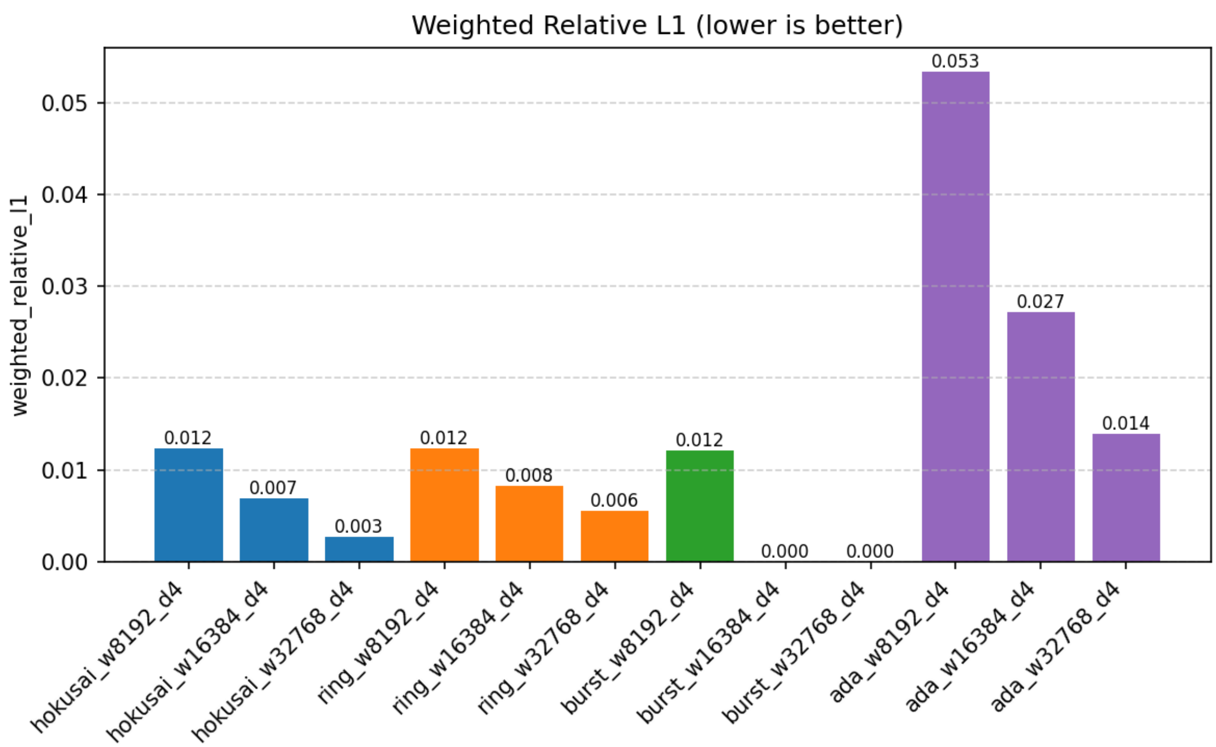

Figure 10: Weighted relative L1 error across all models and widths.

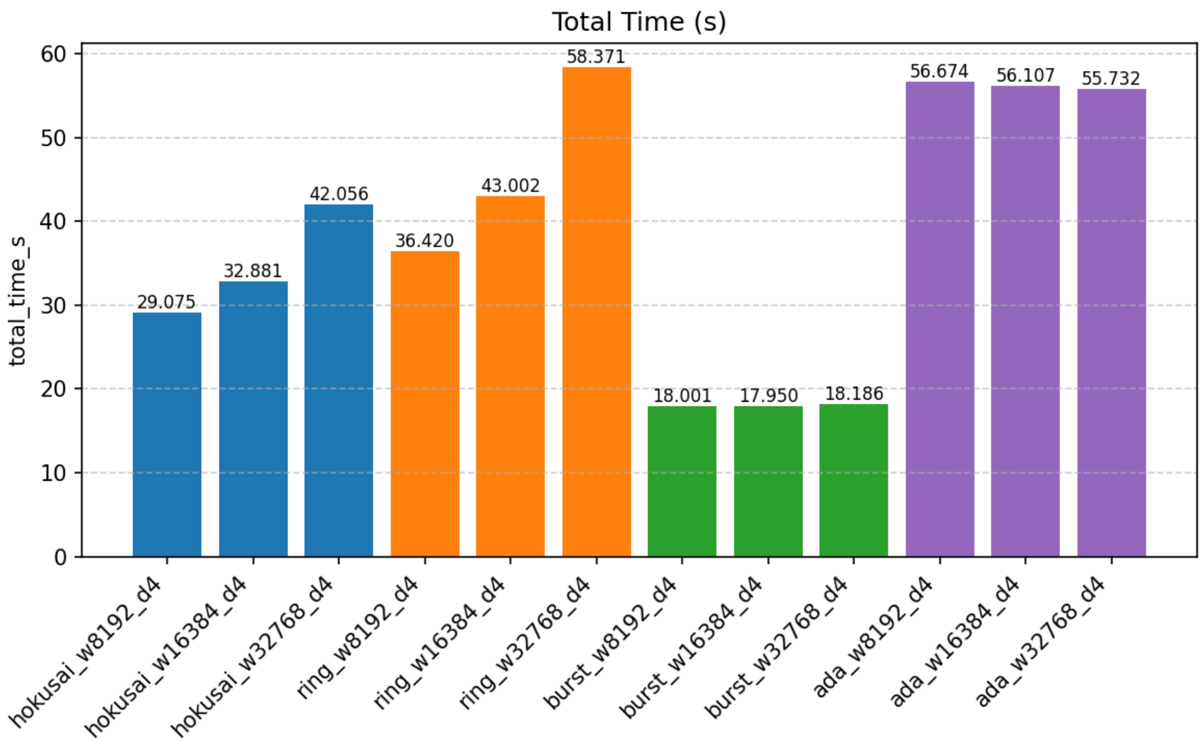

Figure 11: Total runtime in seconds including load, stream, and evaluation.

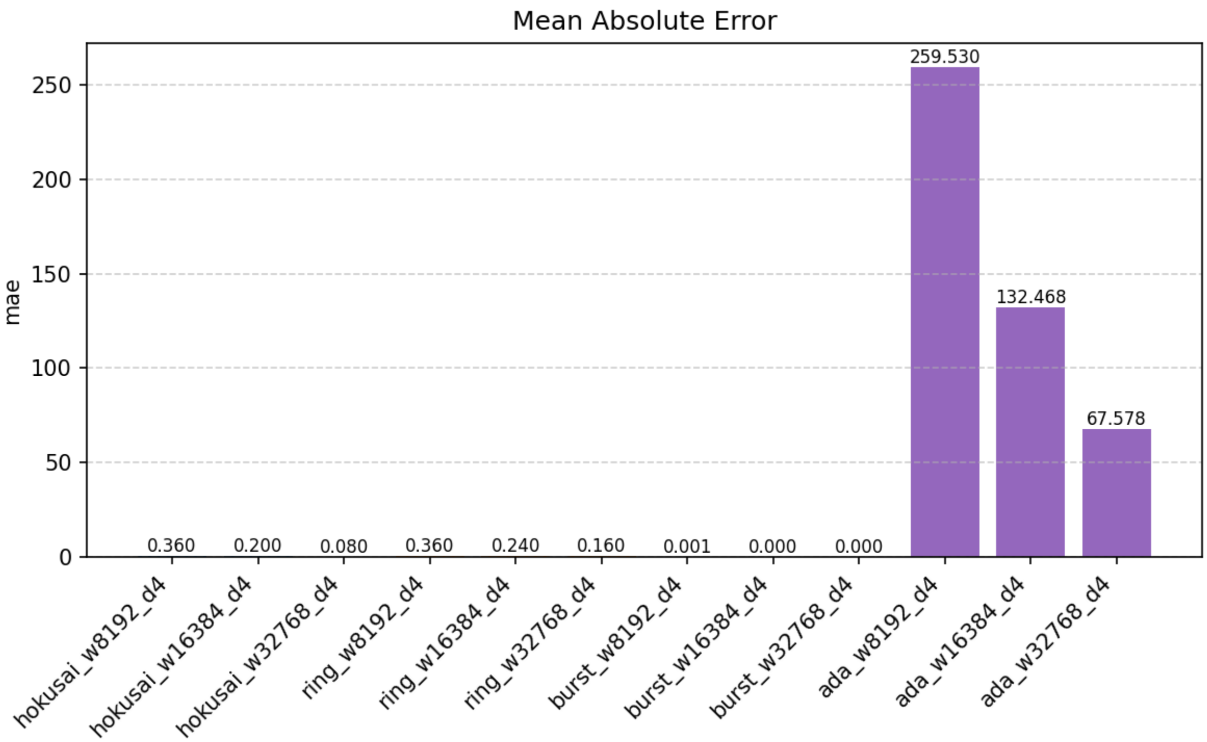

Figure 12: Mean absolute error across all models and widths.

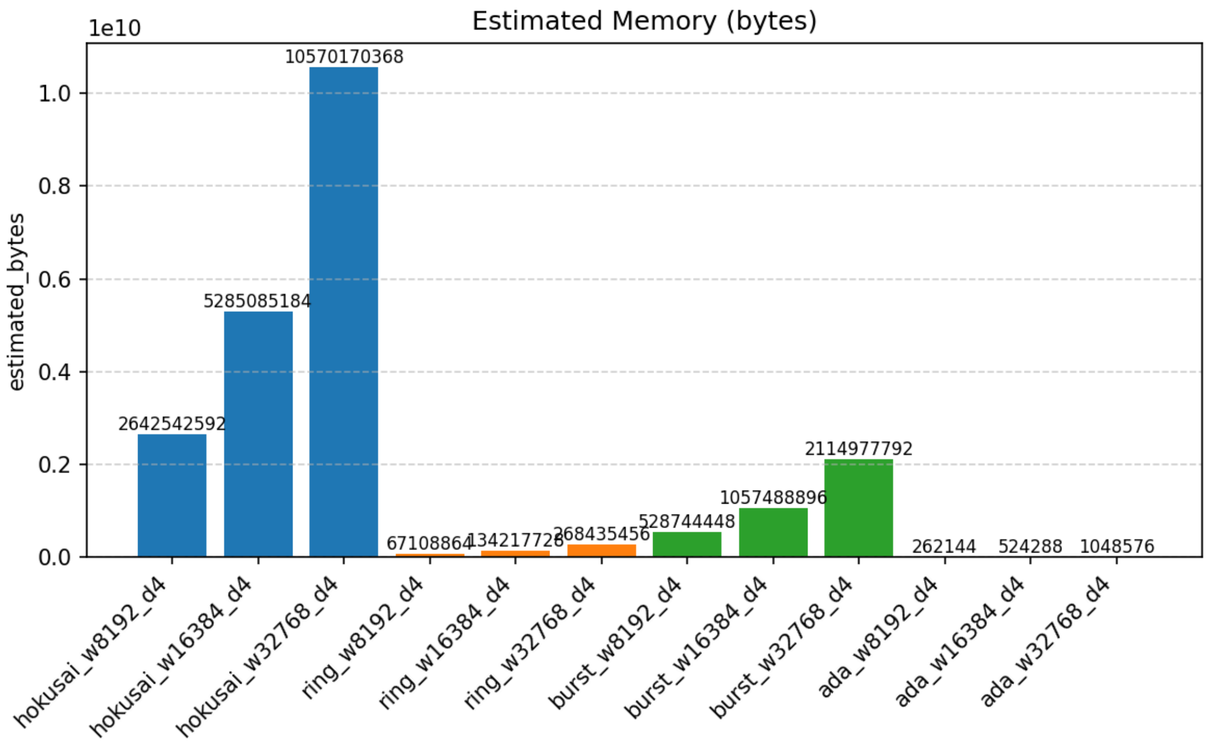

Figure 13: Estimated memory footprint in bytes for each configuration.

A.2 Ring buffer results

Per-width Ring Buffer plots: counts and time series versus ground truth, burst scores, error distributions, memory footprint, top-k, and scatter views.

A.2.1 Width 8192 depth four

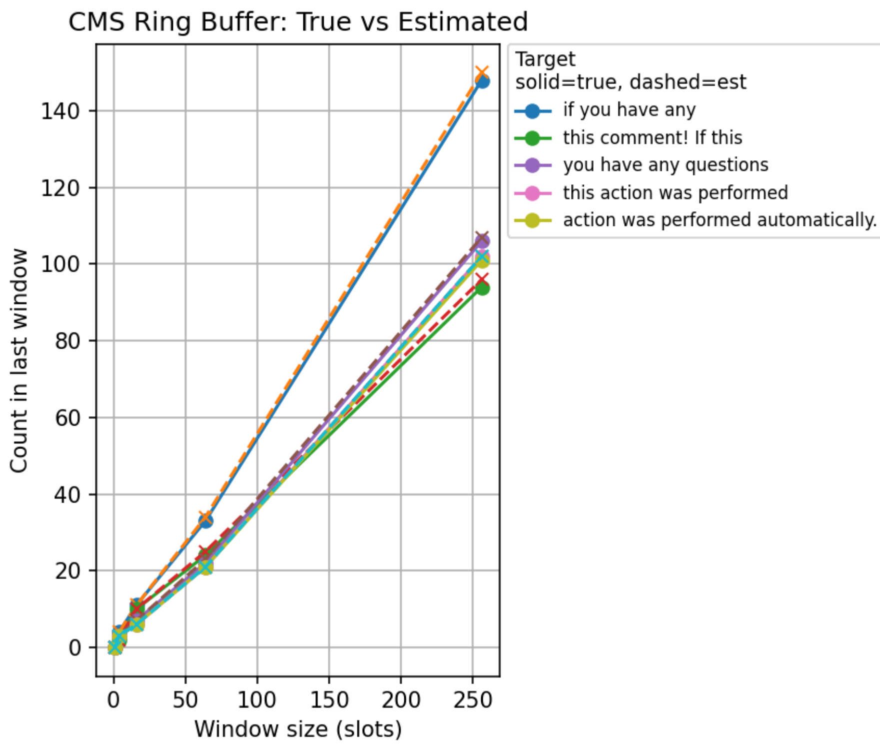

Figure 14: Ring buffer counts versus ground truth for width 8192 and depth four.

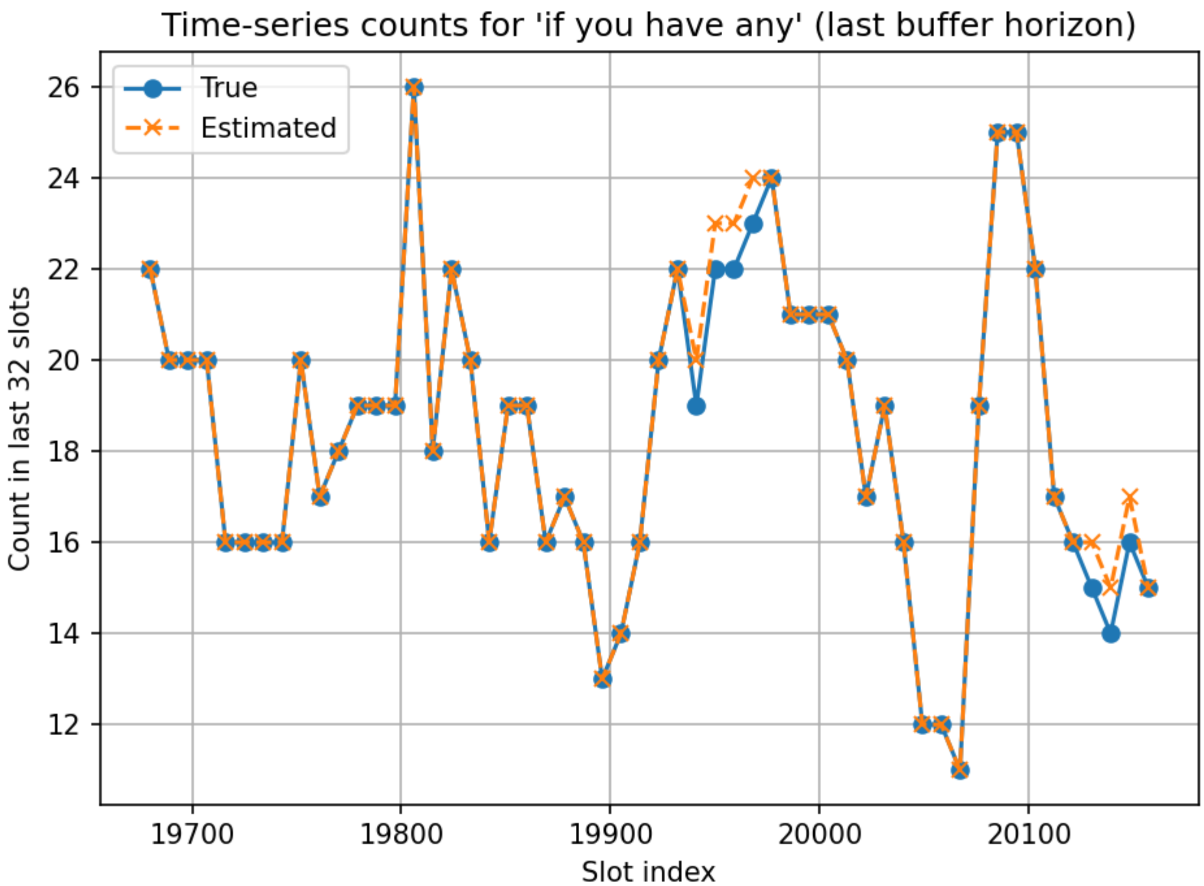

Figure 15: Ring buffer time series estimates for width 8192 and depth four.

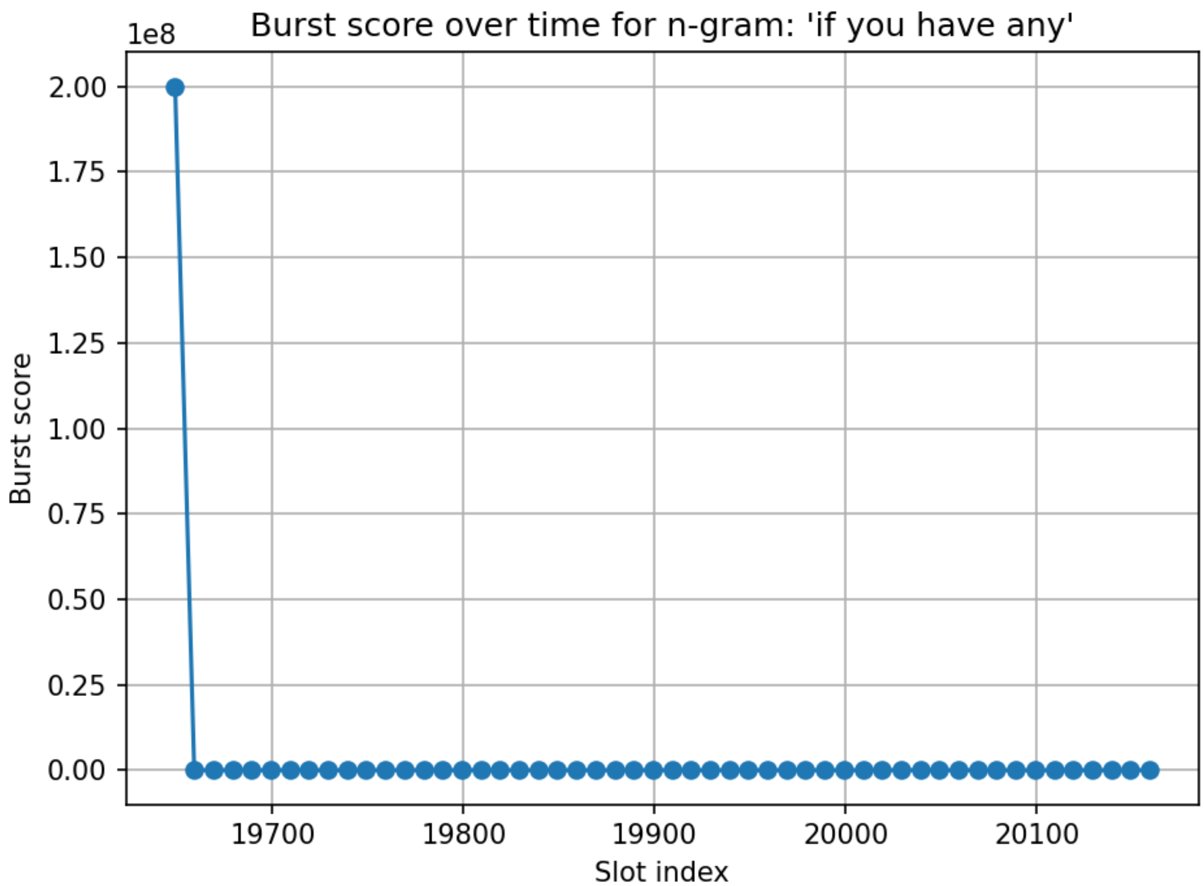

Figure 16: Ring buffer burst scores over time for width 8192 and depth four.

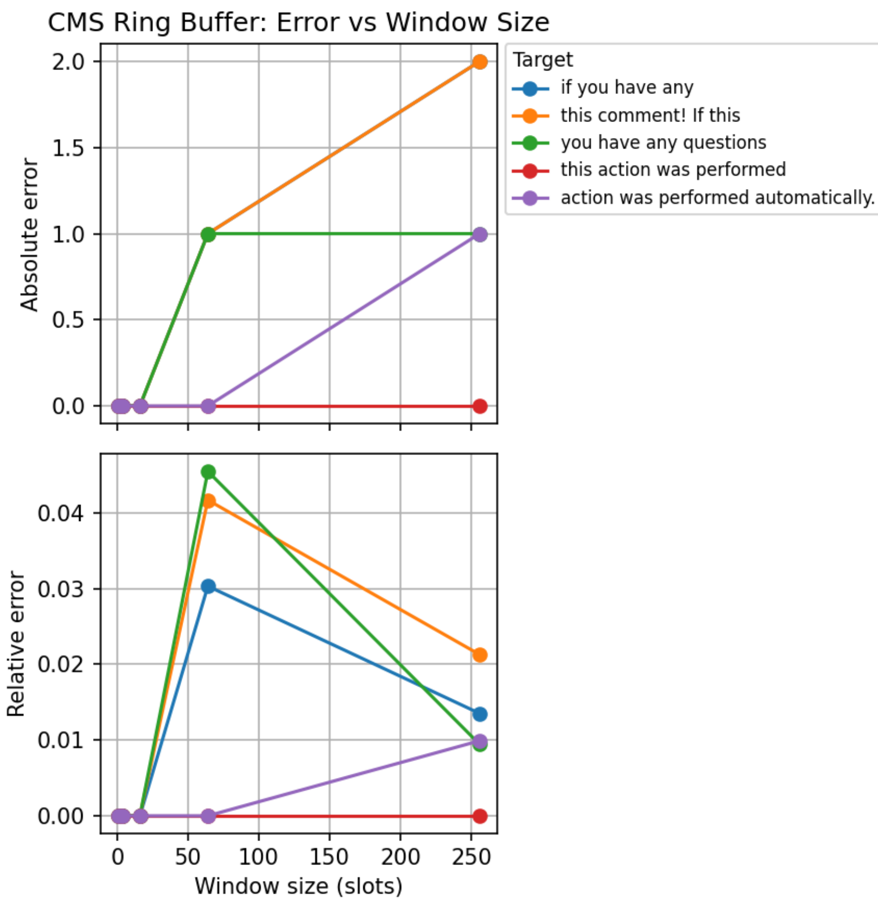

Figure 17: Ring buffer error traces for width 8192 and depth four.

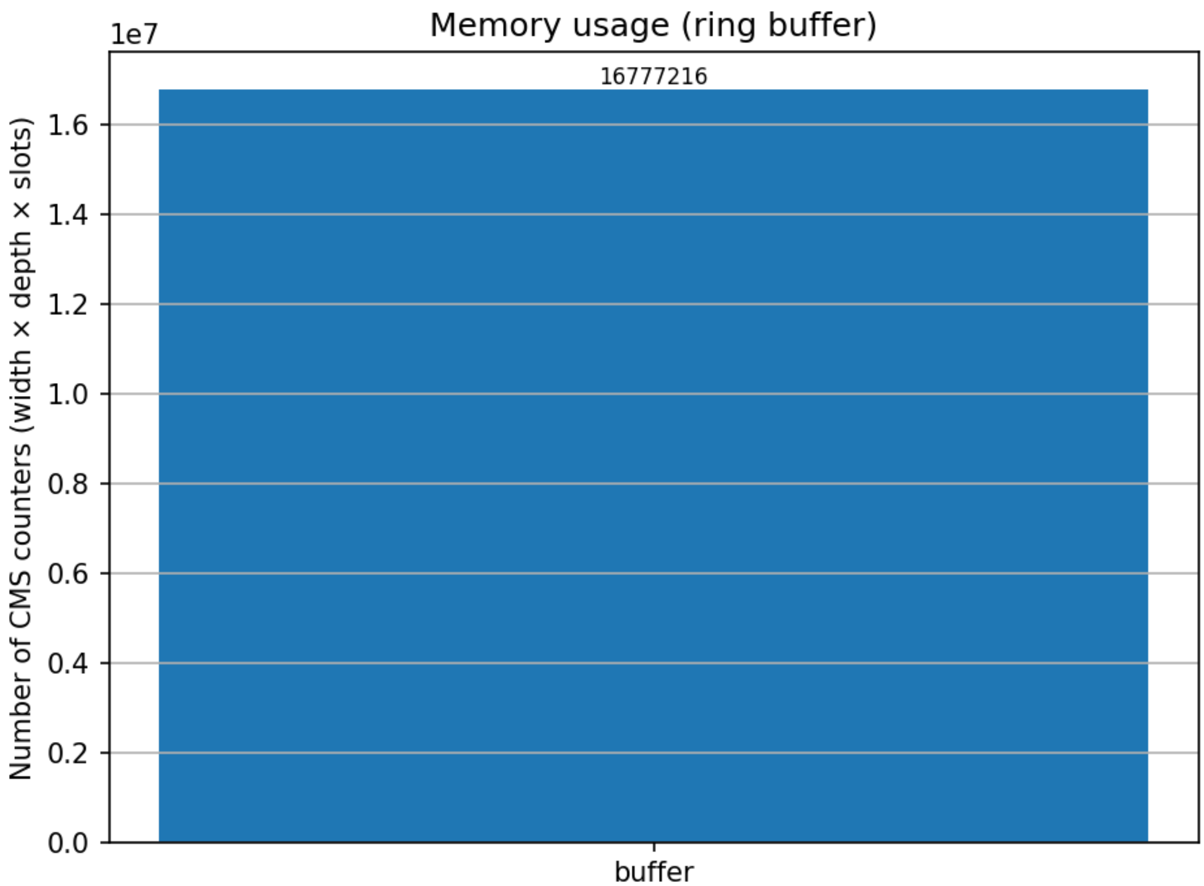

Figure 18: Ring buffer memory footprint for width 8192 and depth four.

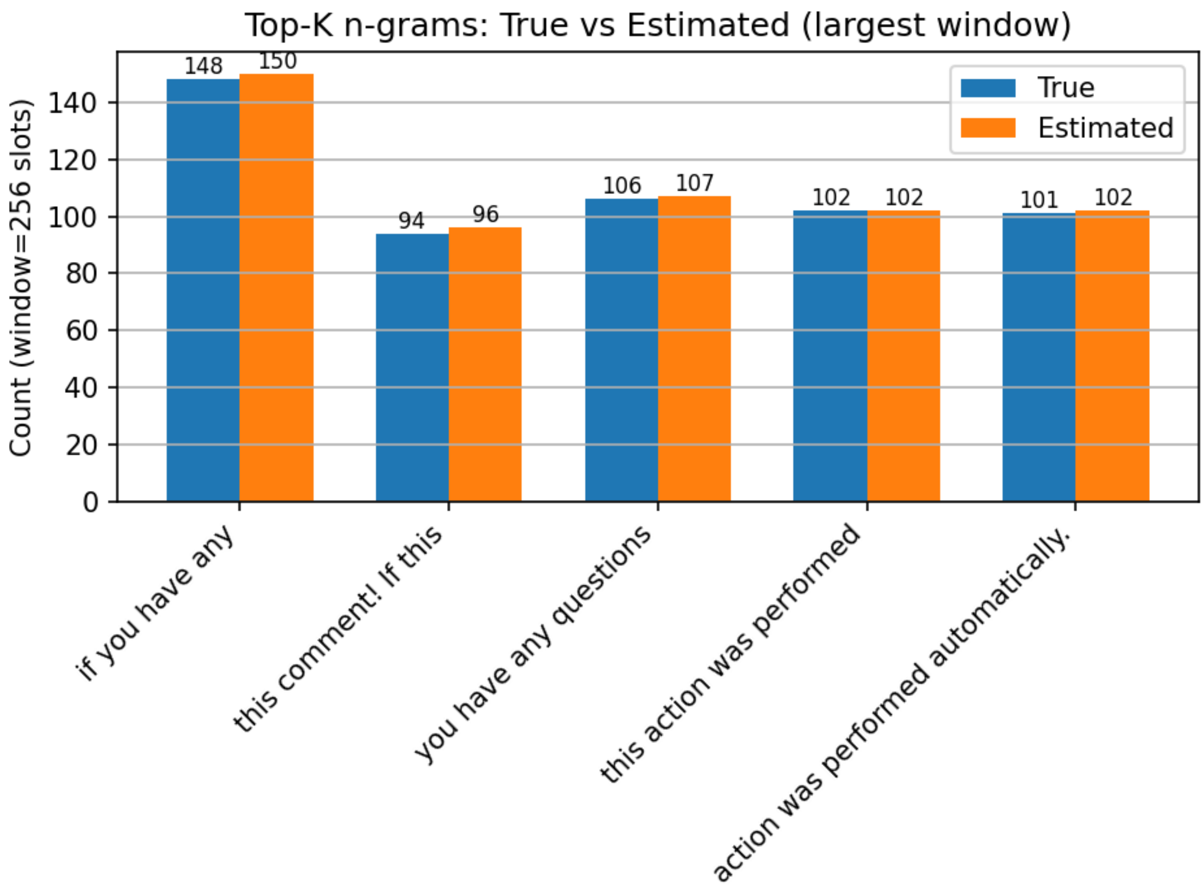

Figure 19: Ring buffer top k counts for width 8192 and depth four.

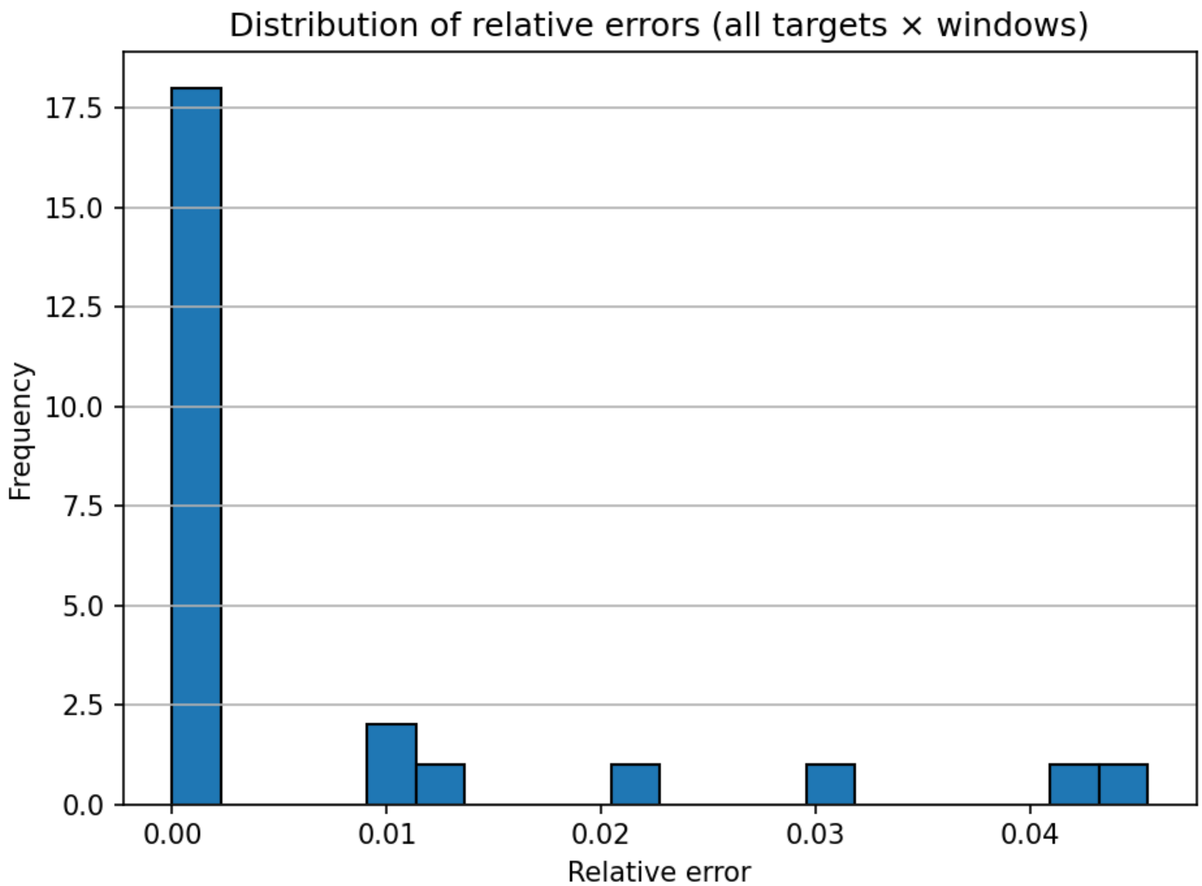

Figure 20: Histogram of relative error for the ring buffer at width 8192 and depth four.

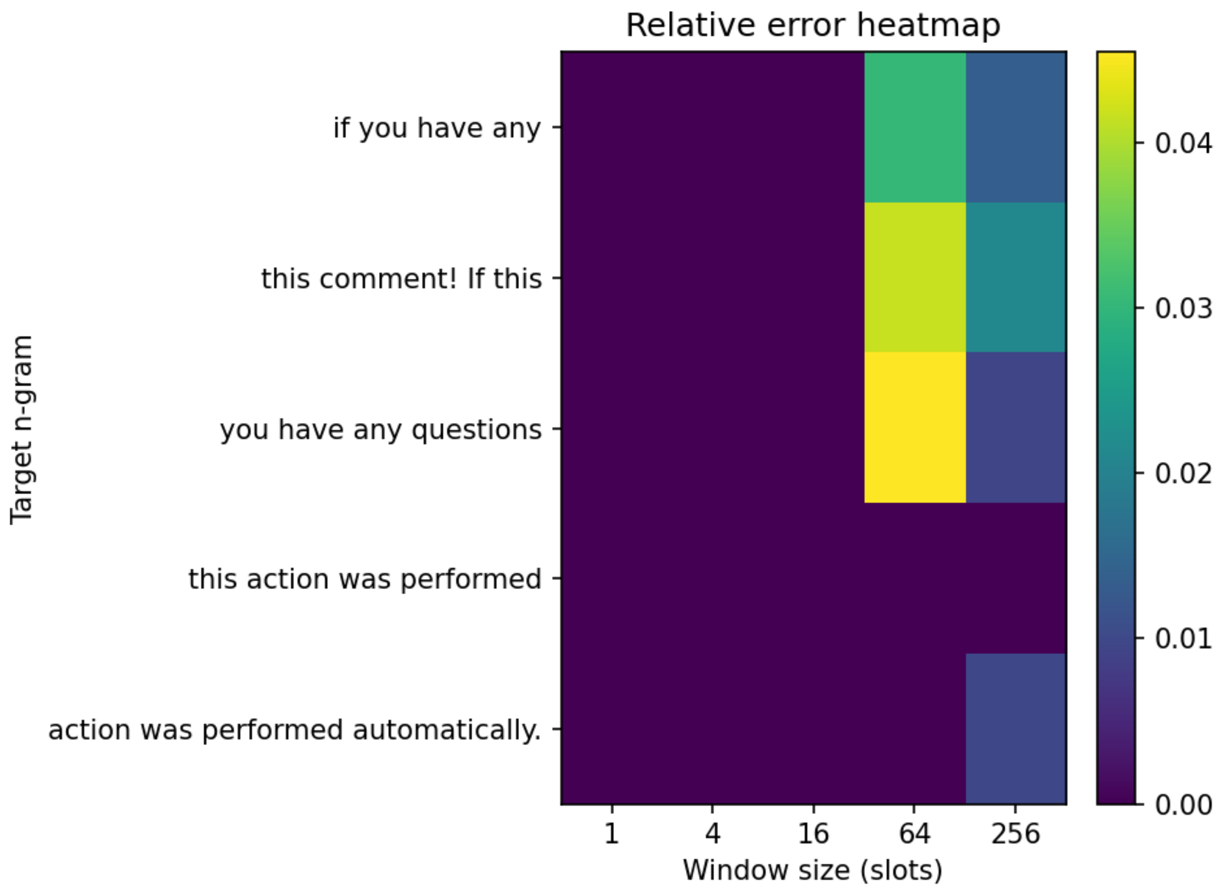

Figure 21: Heatmap of relative error for the ring buffer at width 8192 and depth four.

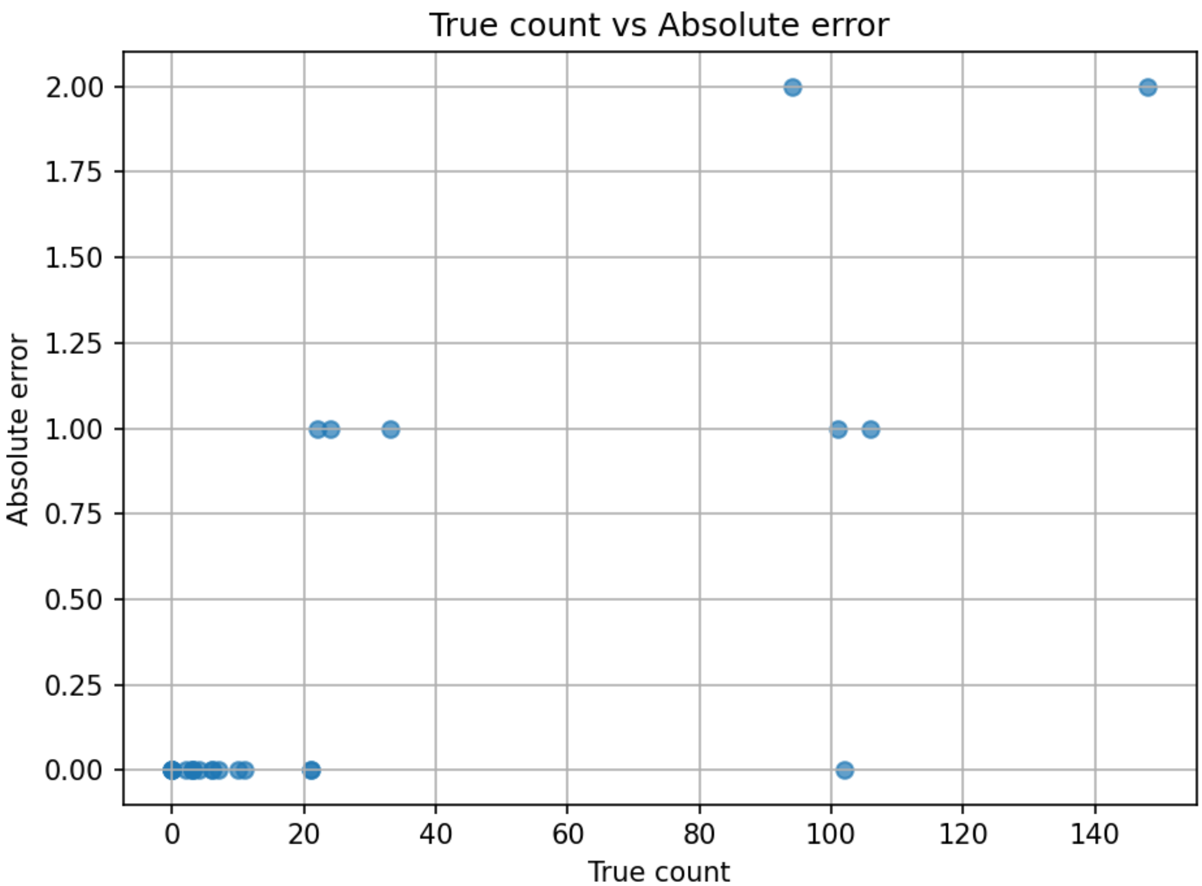

Figure 22: True counts versus absolute error for the ring buffer at width 8192 and depth four.

A.2.2 Width 16384 depth four

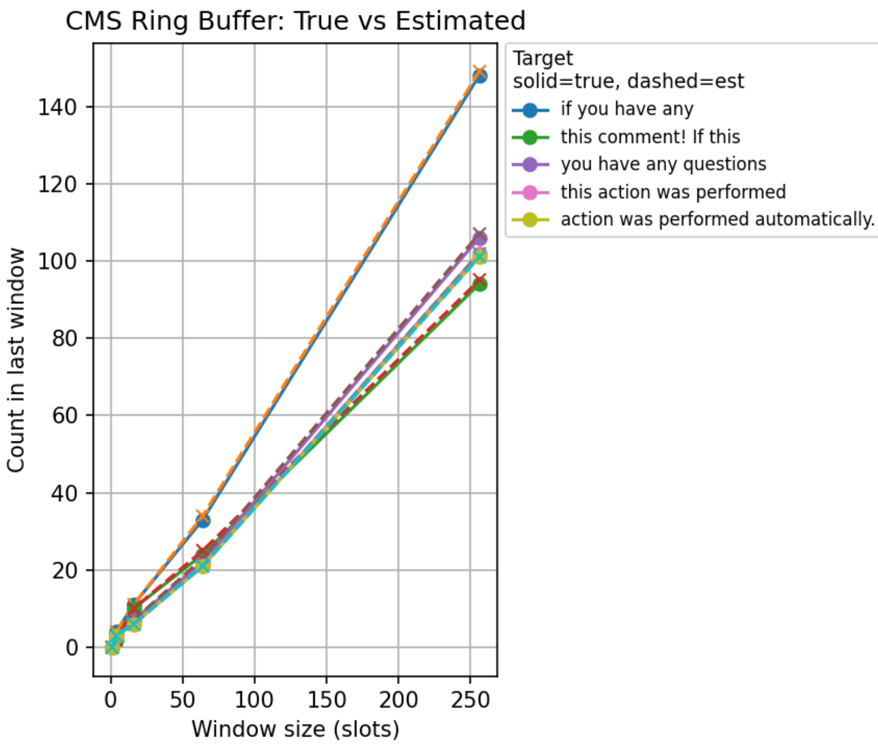

Figure 23: Ring buffer counts versus ground truth for width 16384 and depth four.

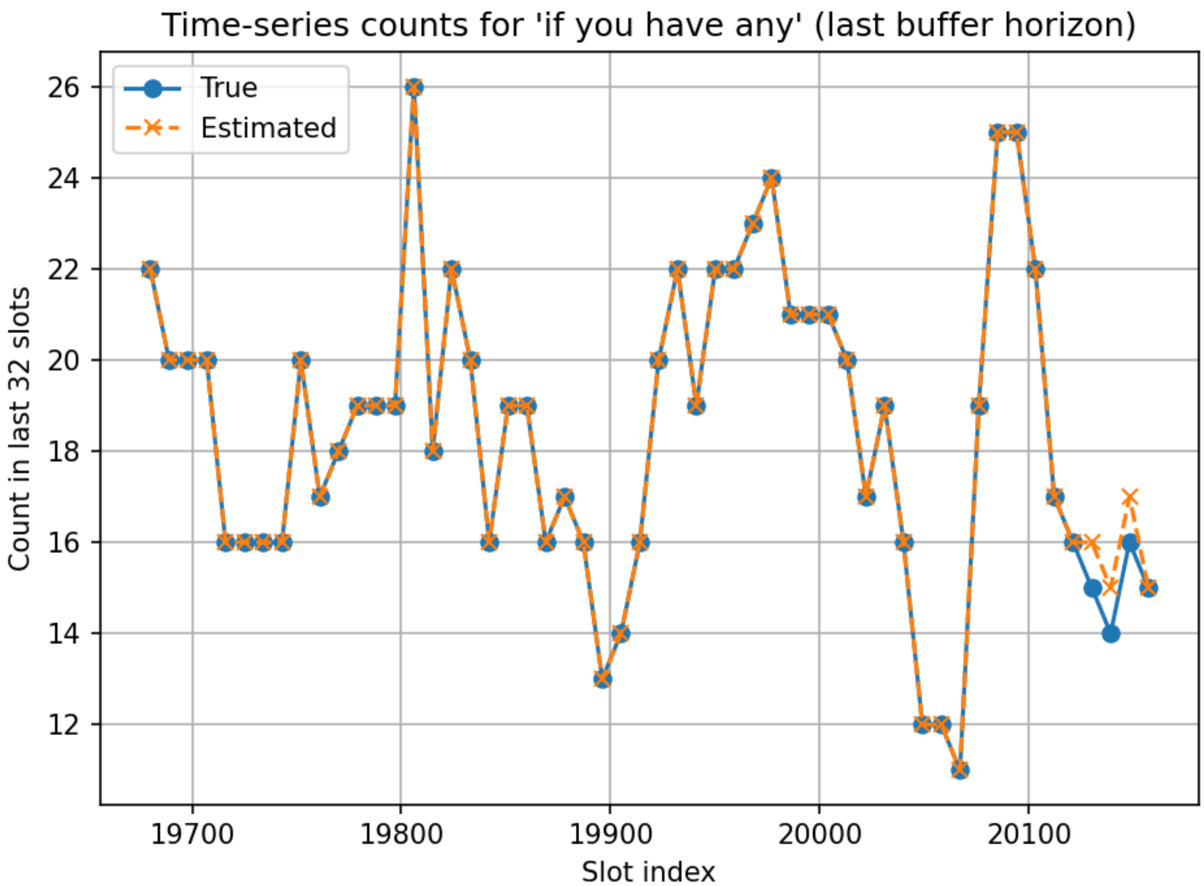

Figure 24: Ring buffer time series estimates for width 16384 and depth four.

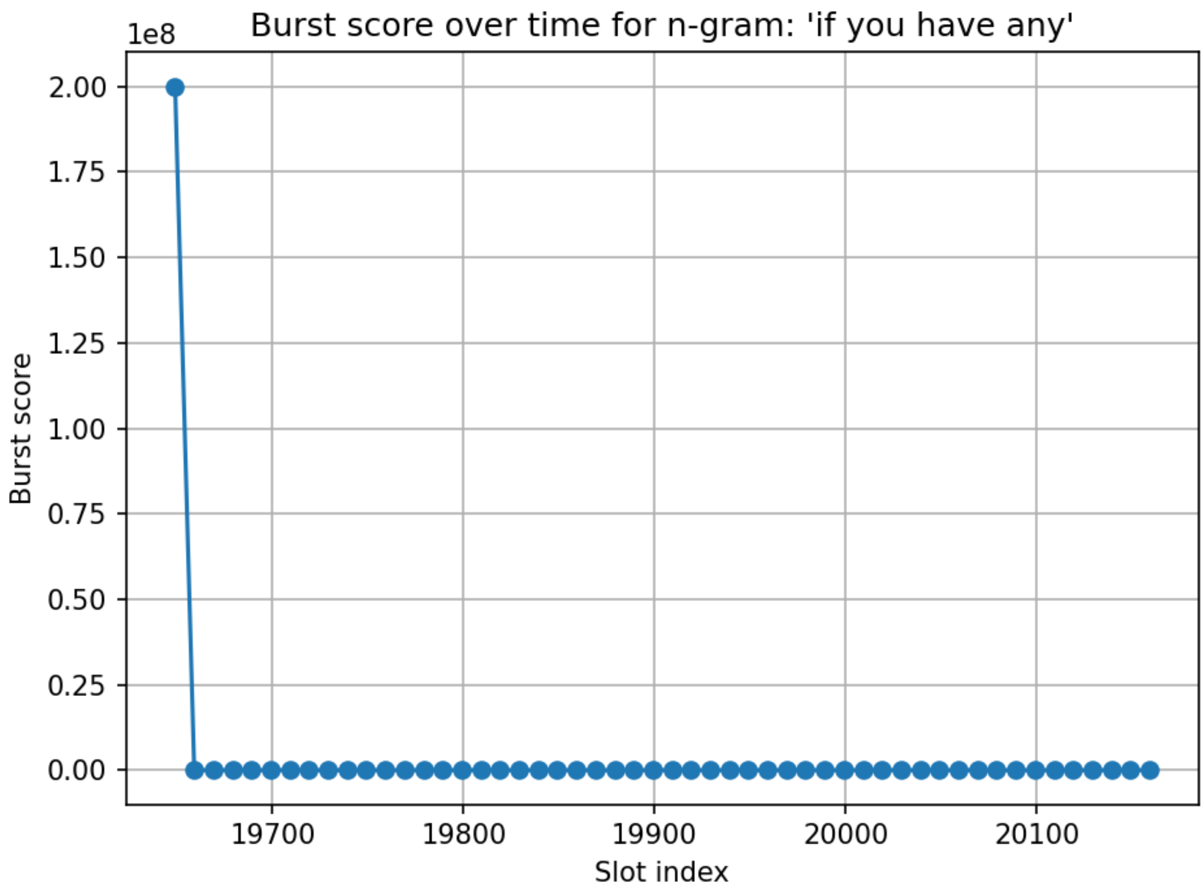

Figure 25: Ring buffer burst scores over time for width 16384 and depth four.

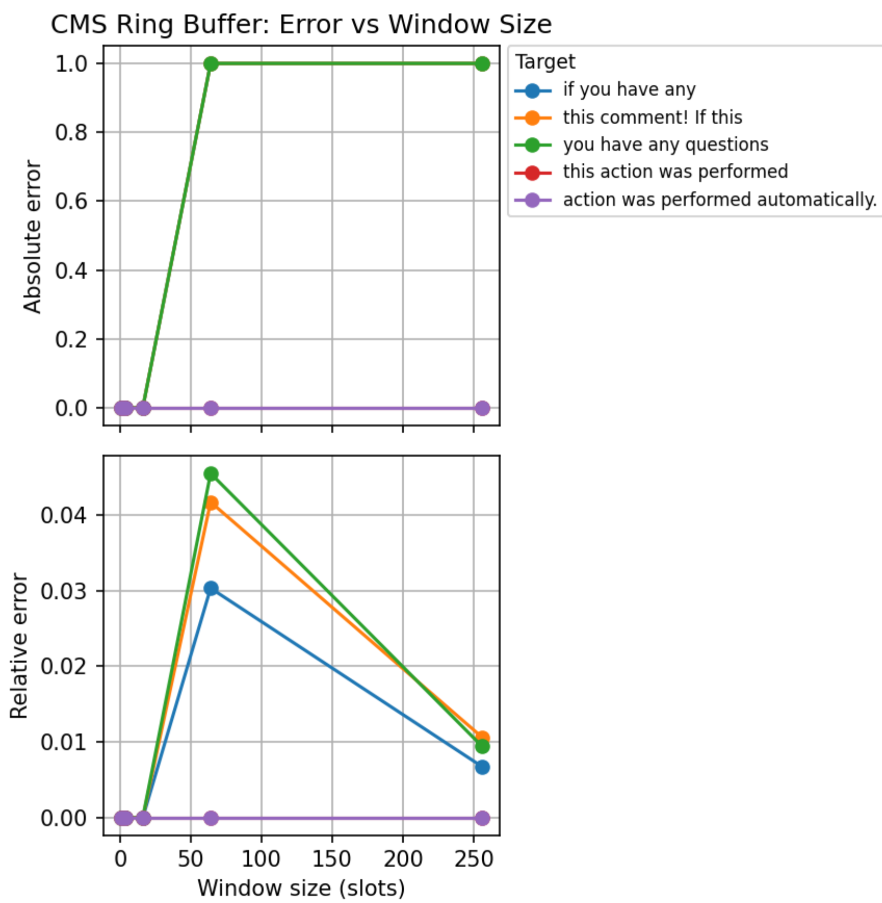

Figure 26: Ring buffer error traces for width 16384 and depth four.

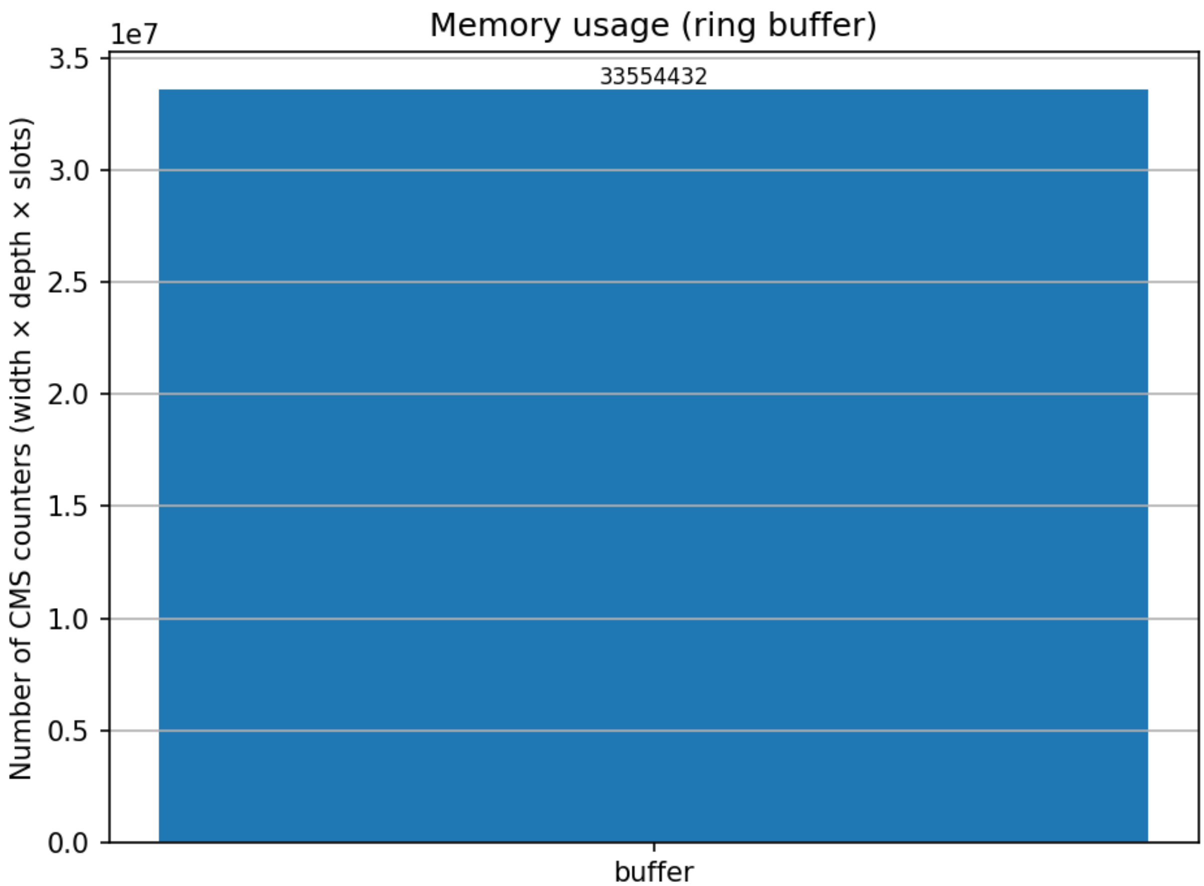

Figure 27: Ring buffer memory footprint for width 16384 and depth four.

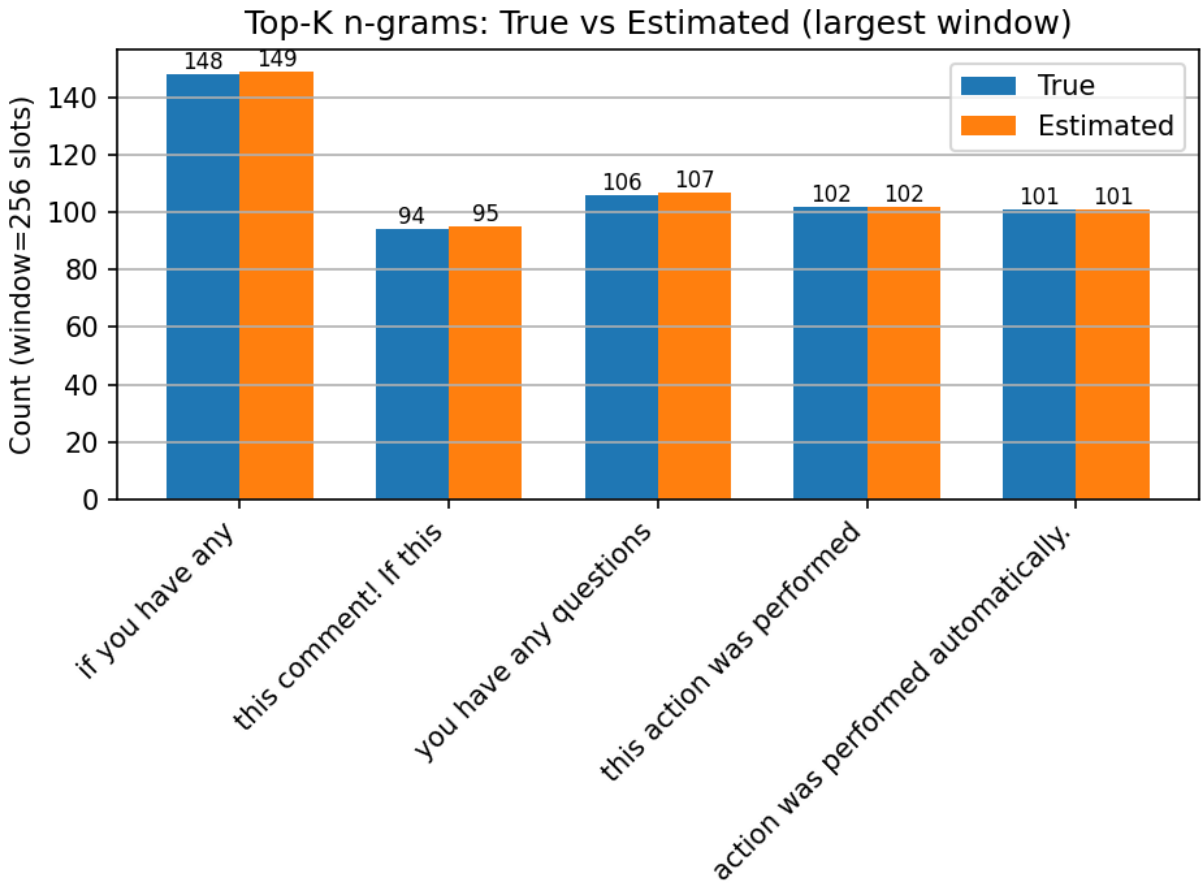

Figure 28: Ring buffer top k counts for width 16384 and depth four.

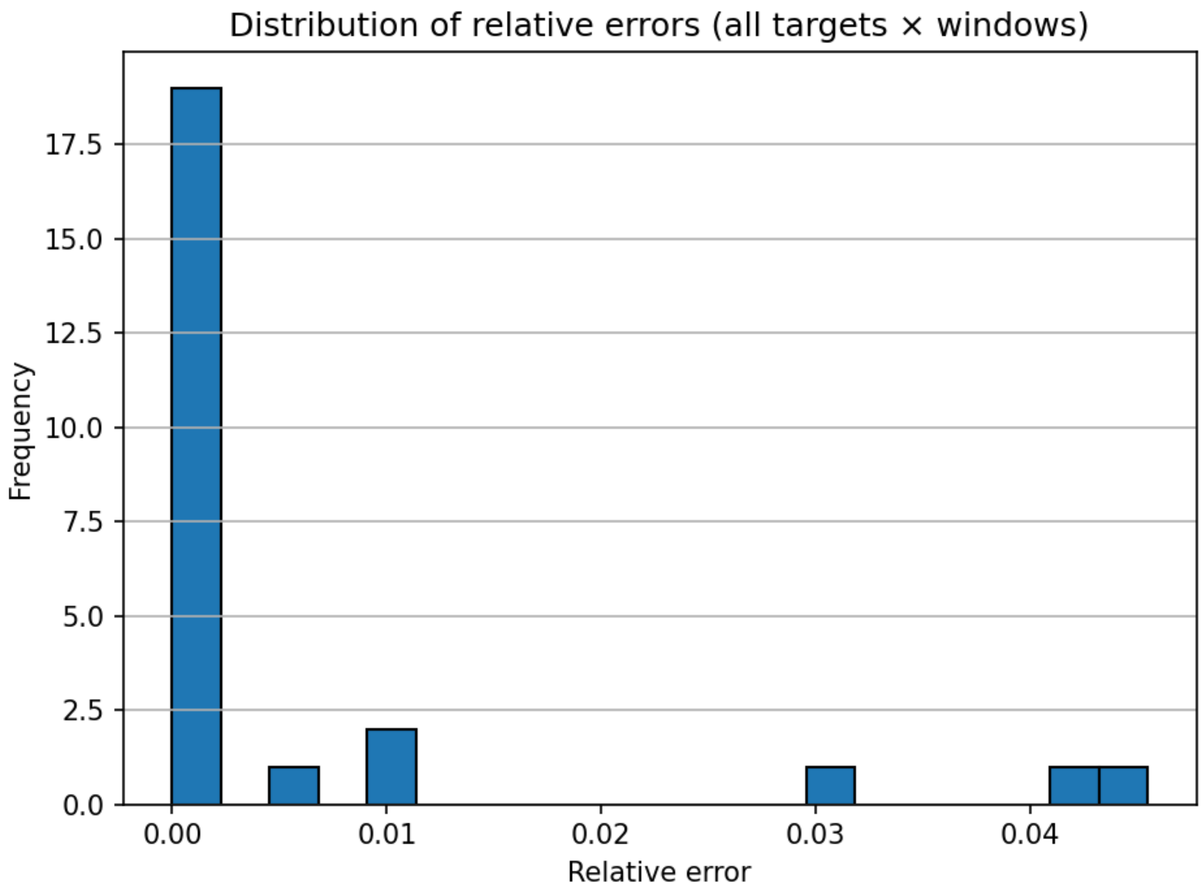

Figure 29: Histogram of relative error for the ring buffer at width 16384 and depth four.

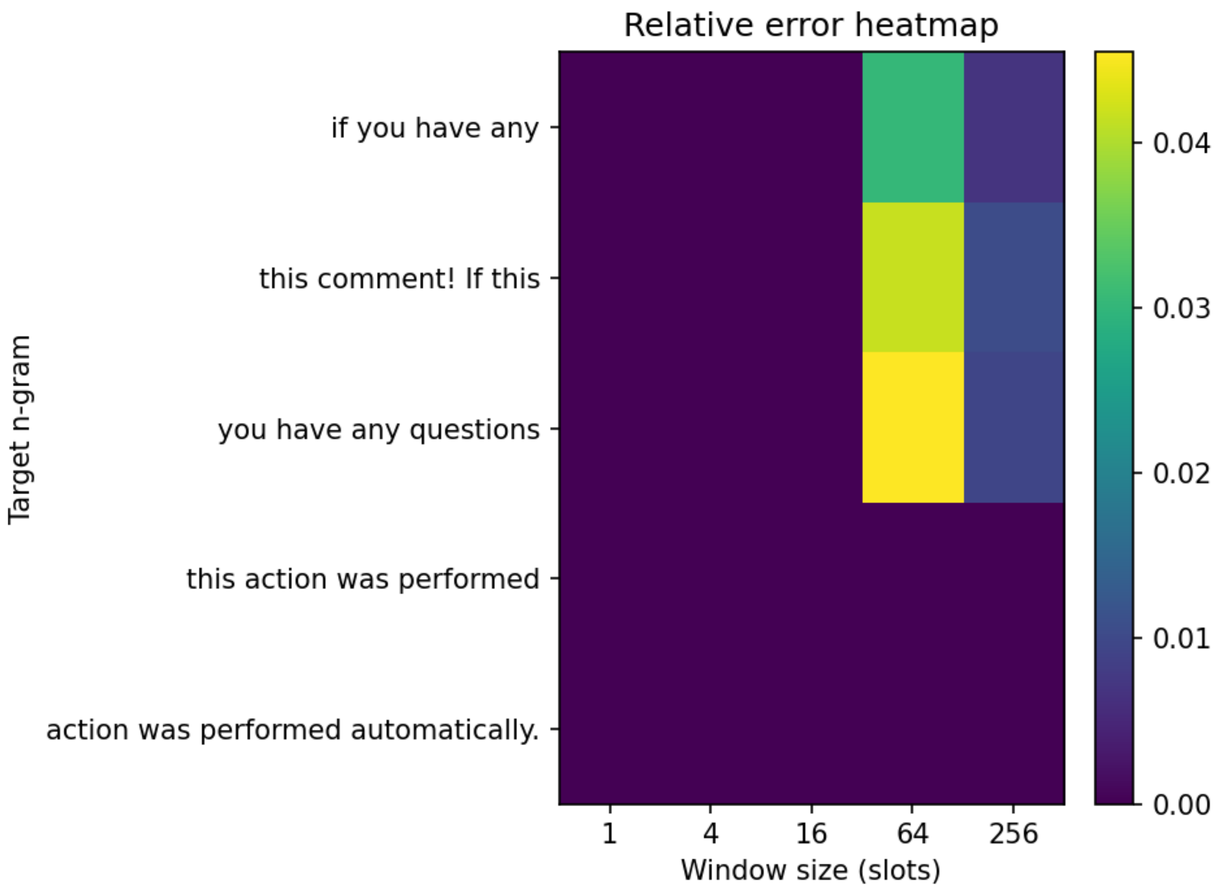

Figure 30: Heatmap of relative error for the ring buffer at width 16384 and depth four.

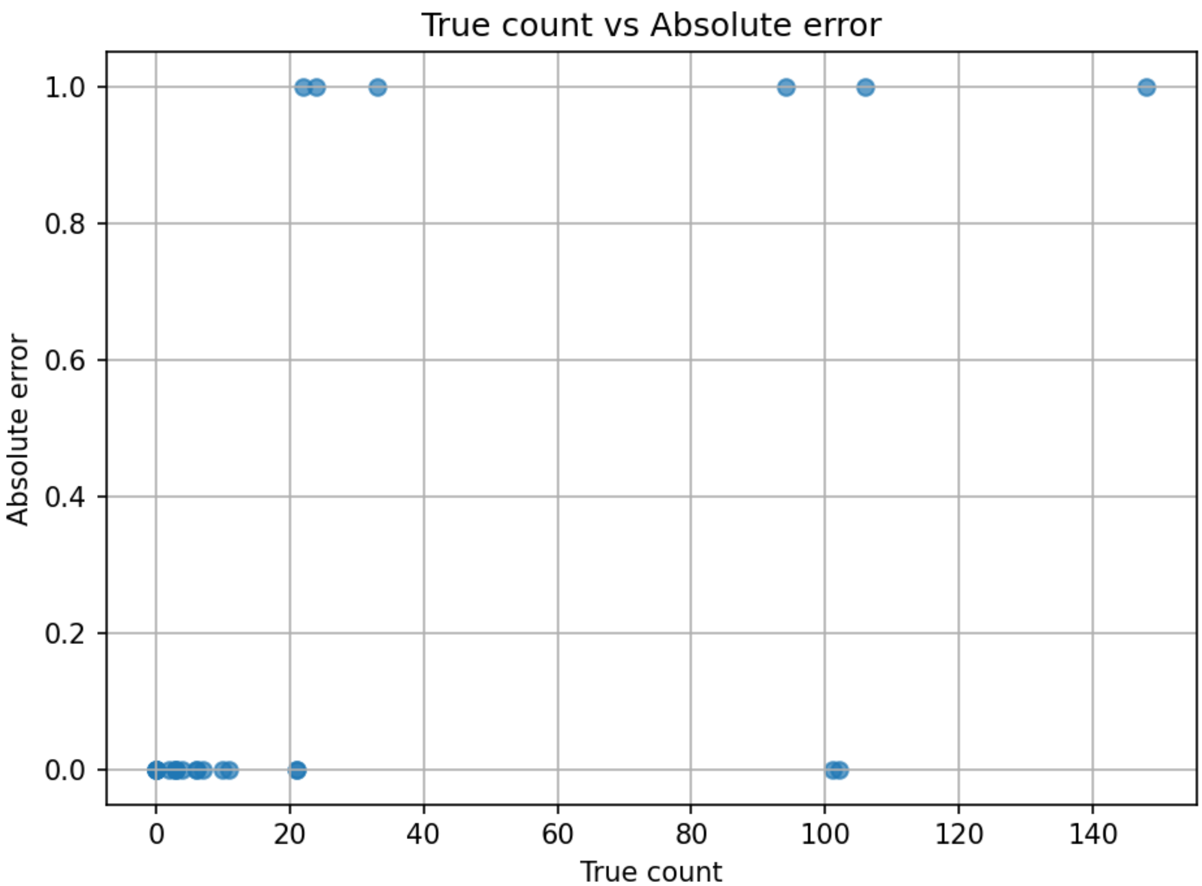

Figure 31: True counts versus absolute error for the ring buffer at width 16384 and depth four.

A.2.3 Width 32768 depth four

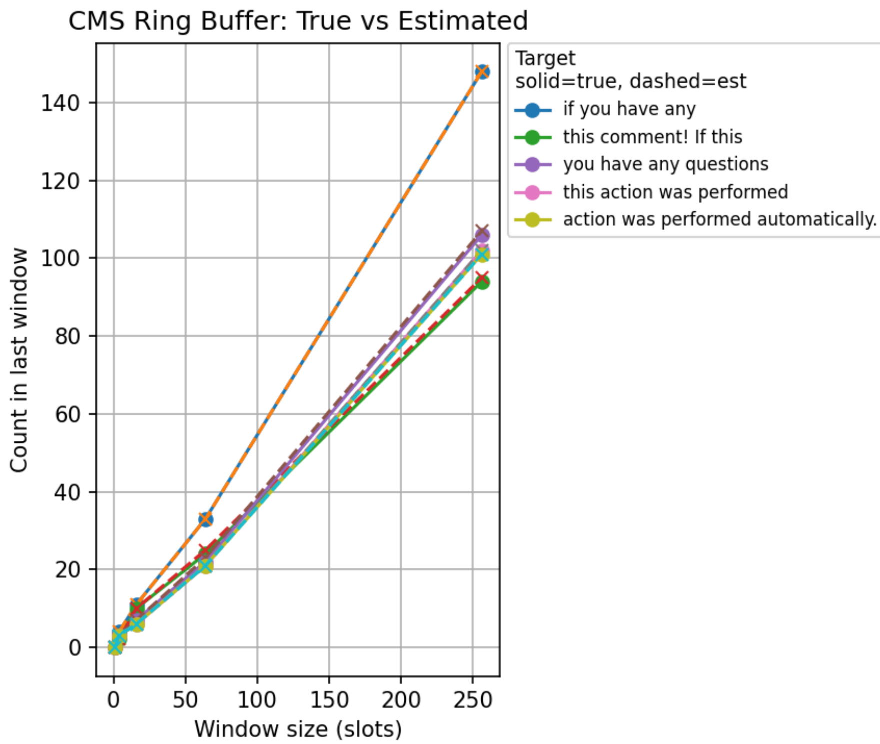

Figure 32: Ring buffer counts versus ground truth for width 32768 and depth four.

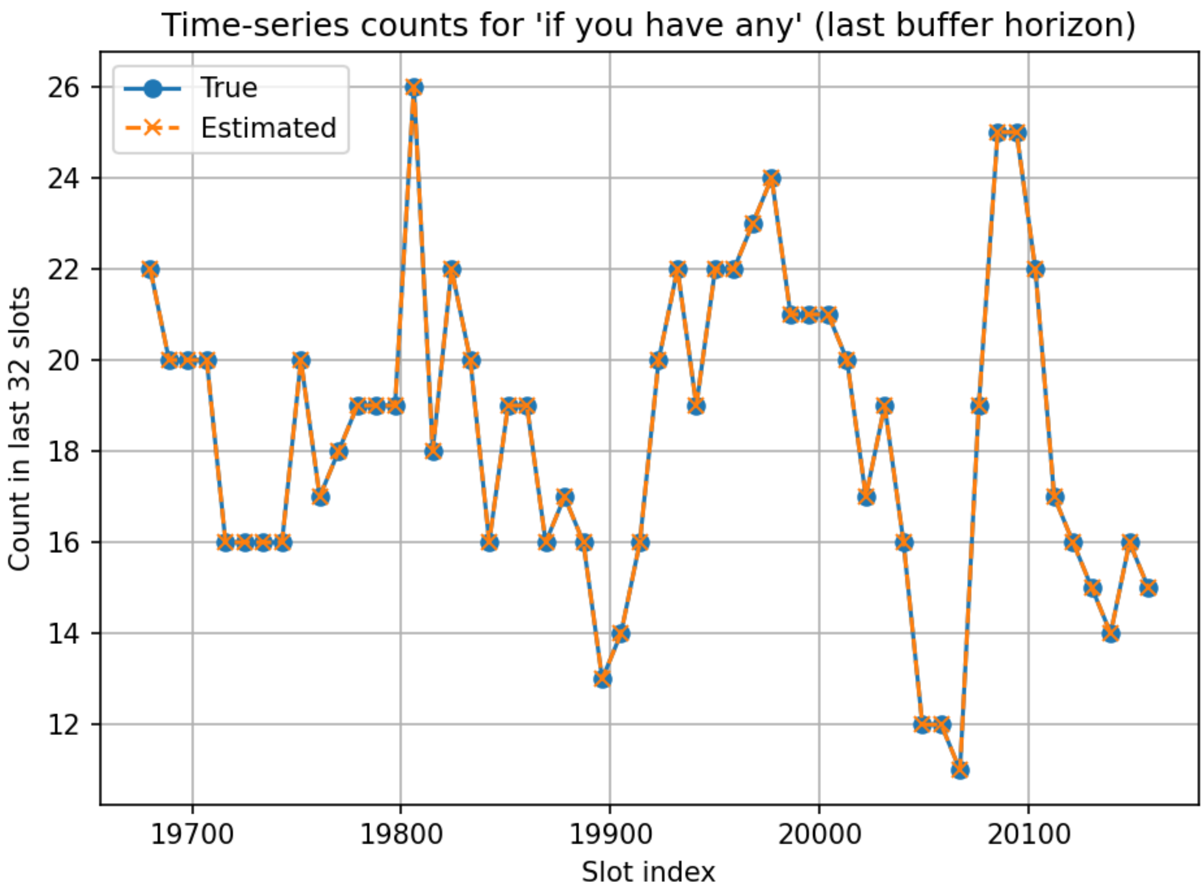

Figure 33: Ring buffer time series estimates for width 32768 and depth four.

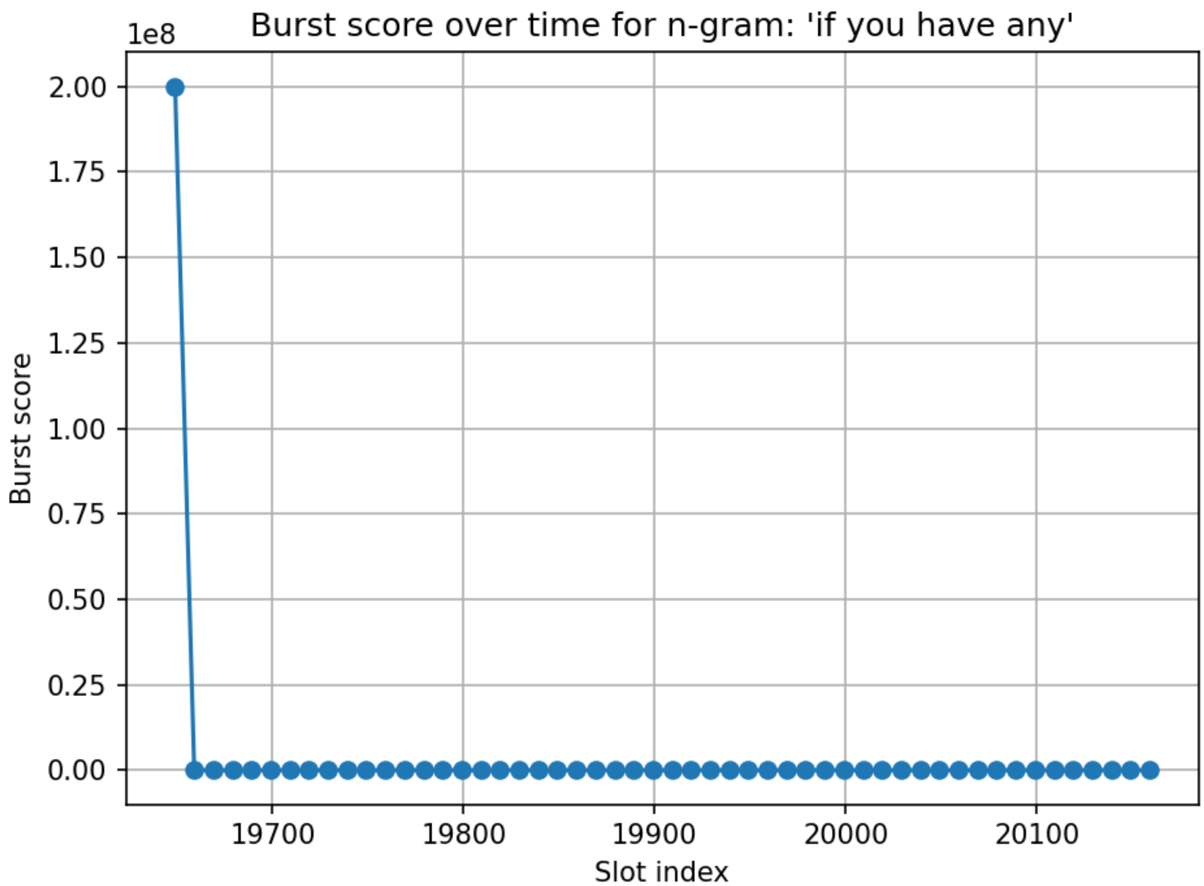

Figure 34: Ring buffer burst scores over time for width 32768 and depth four.

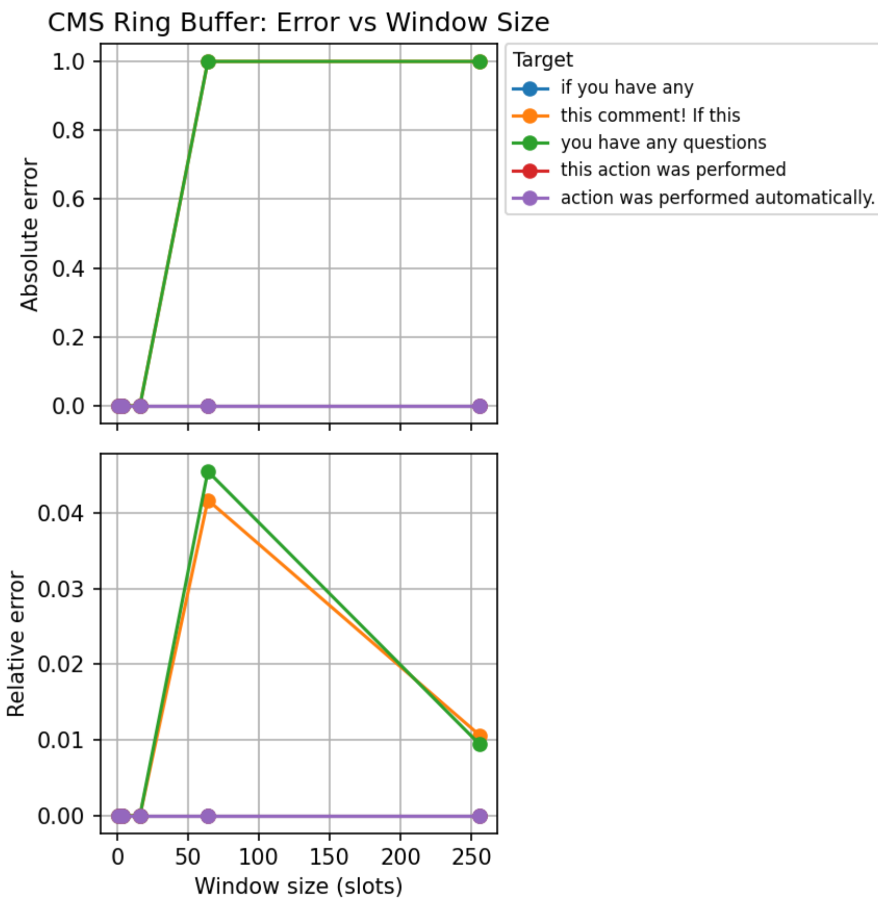

Figure 35: Ring buffer error traces for width 32768 and depth four.

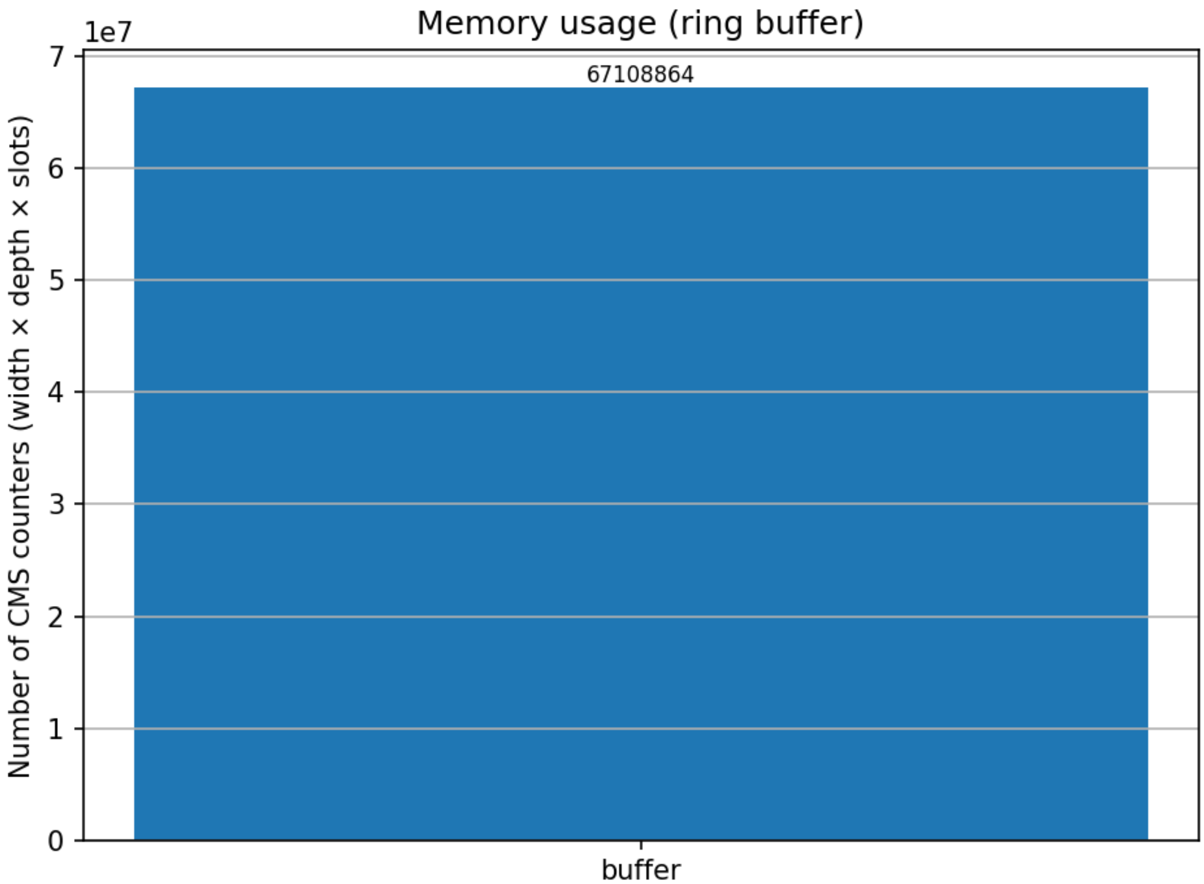

Figure 36: Ring buffer memory footprint for width 32768 and depth four.

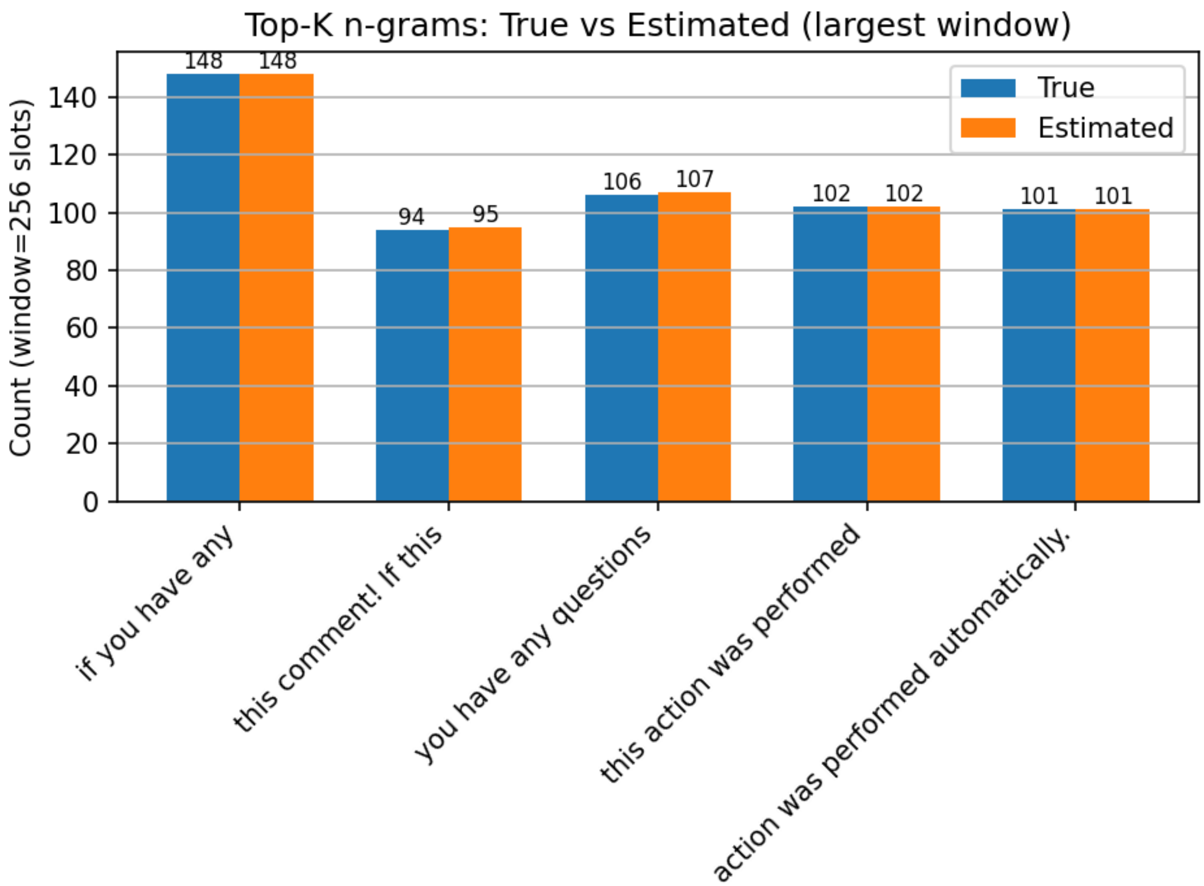

Figure 37: Ring buffer top k counts for width 32768 and depth four.

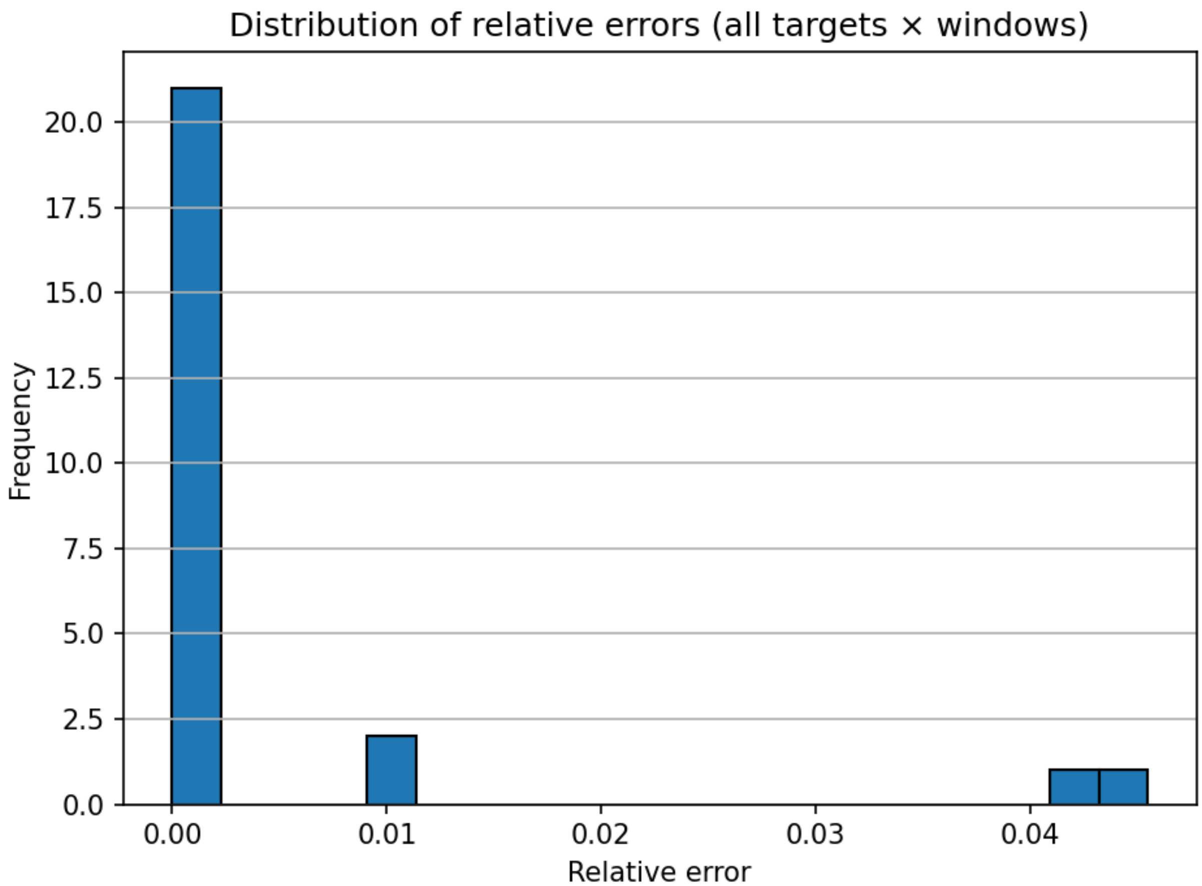

Figure 38: Histogram of relative error for the ring buffer at width 32768 and depth four.

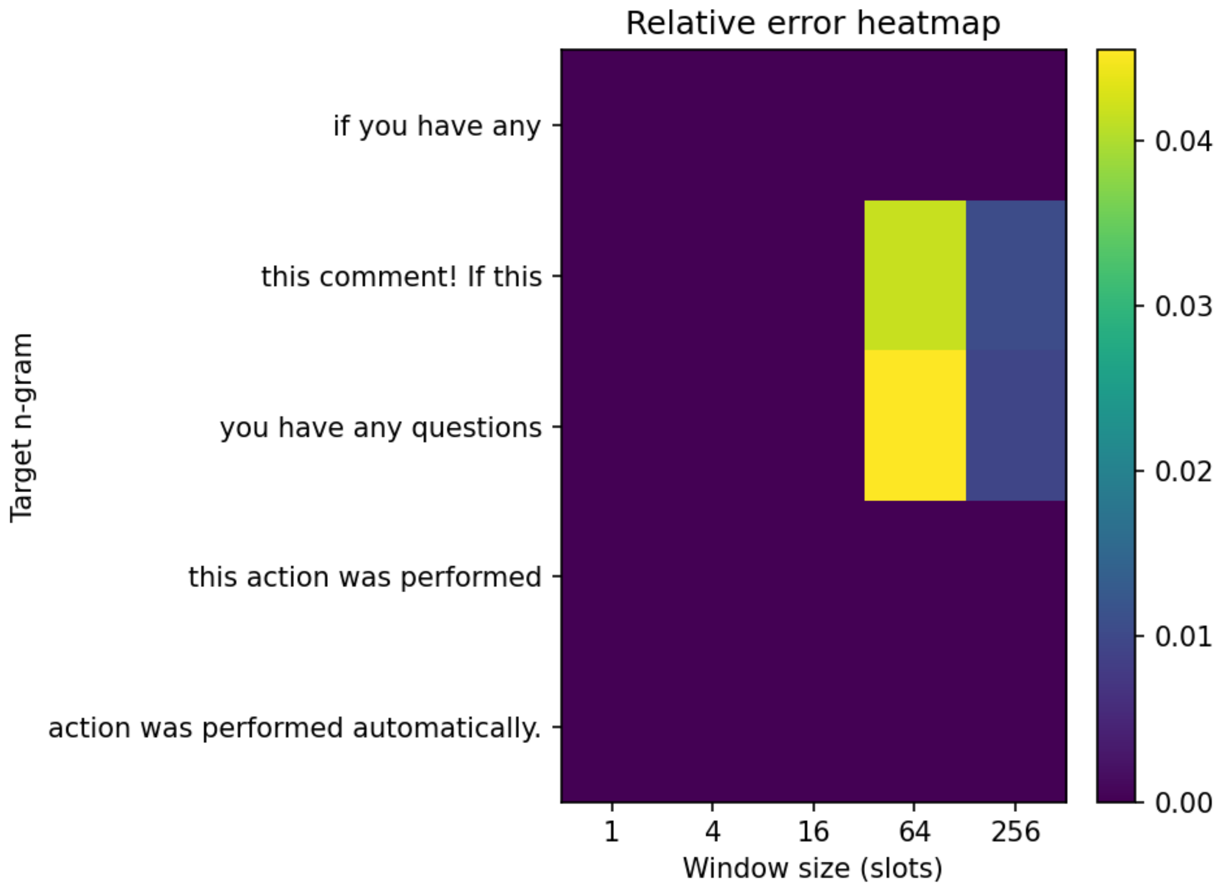

Figure 39: Heatmap of relative error for the ring buffer at width 32768 and depth four.

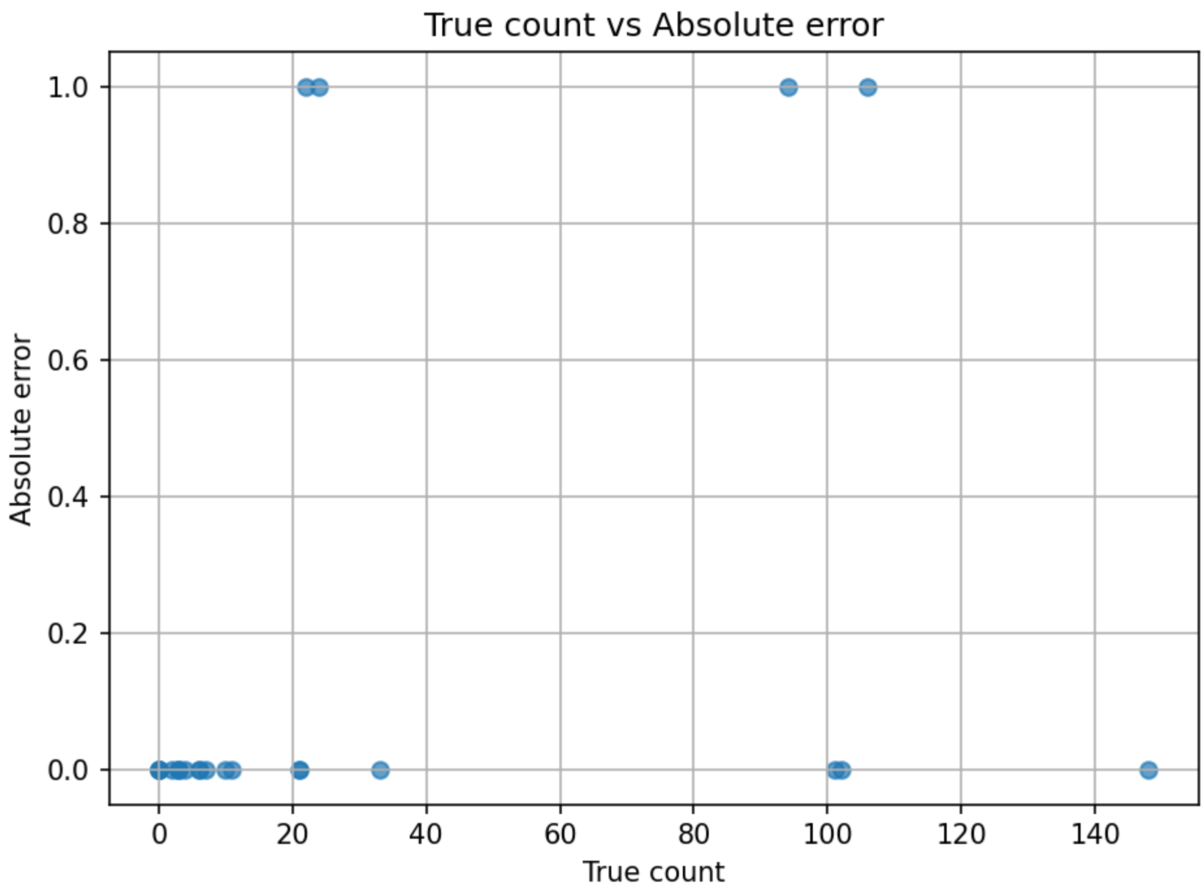

Figure 40: True counts versus absolute error for the ring buffer at width 32768 and depth four.

A.3 Hokusai hierarchy results

Per-width Hokusai plots: counts and time series versus ground truth, burst scores, error distributions, memory per level, top-k, and scatter views.

A.3.1 Width 8192 depth four

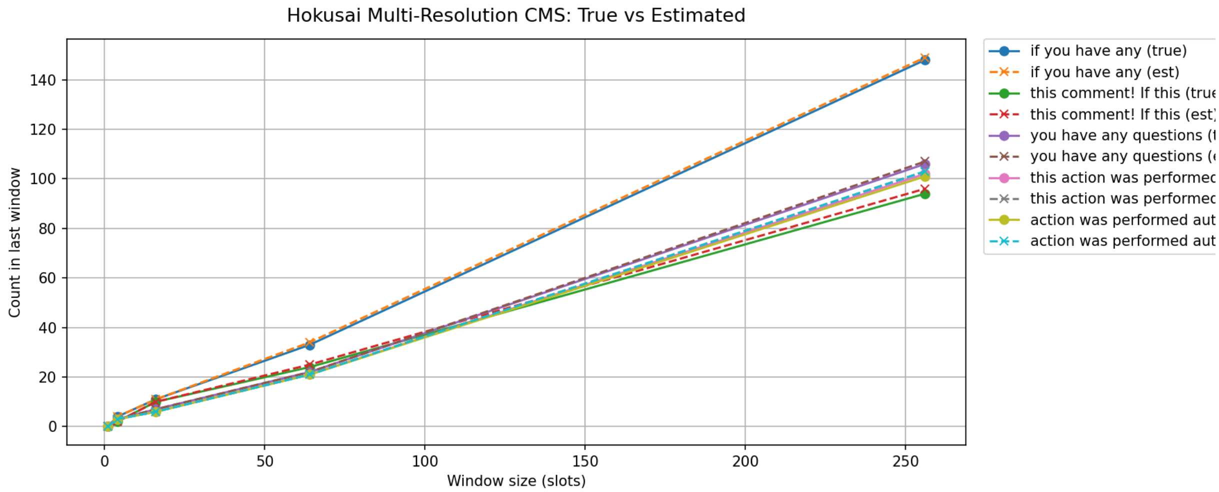

Figure 41: Hokusai counts versus ground truth for width 8192 and depth four.

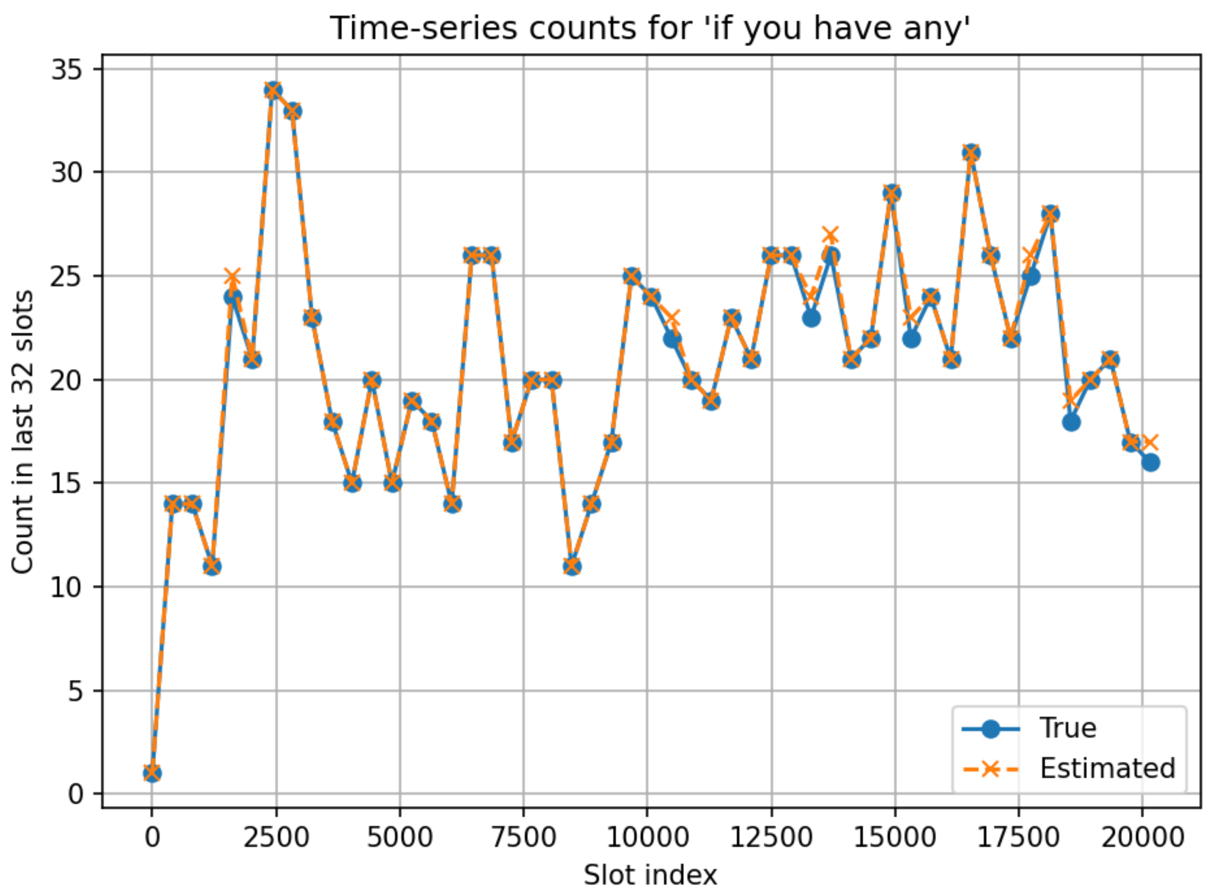

Figure 42: Hokusai time series estimates for width 8192 and depth four.

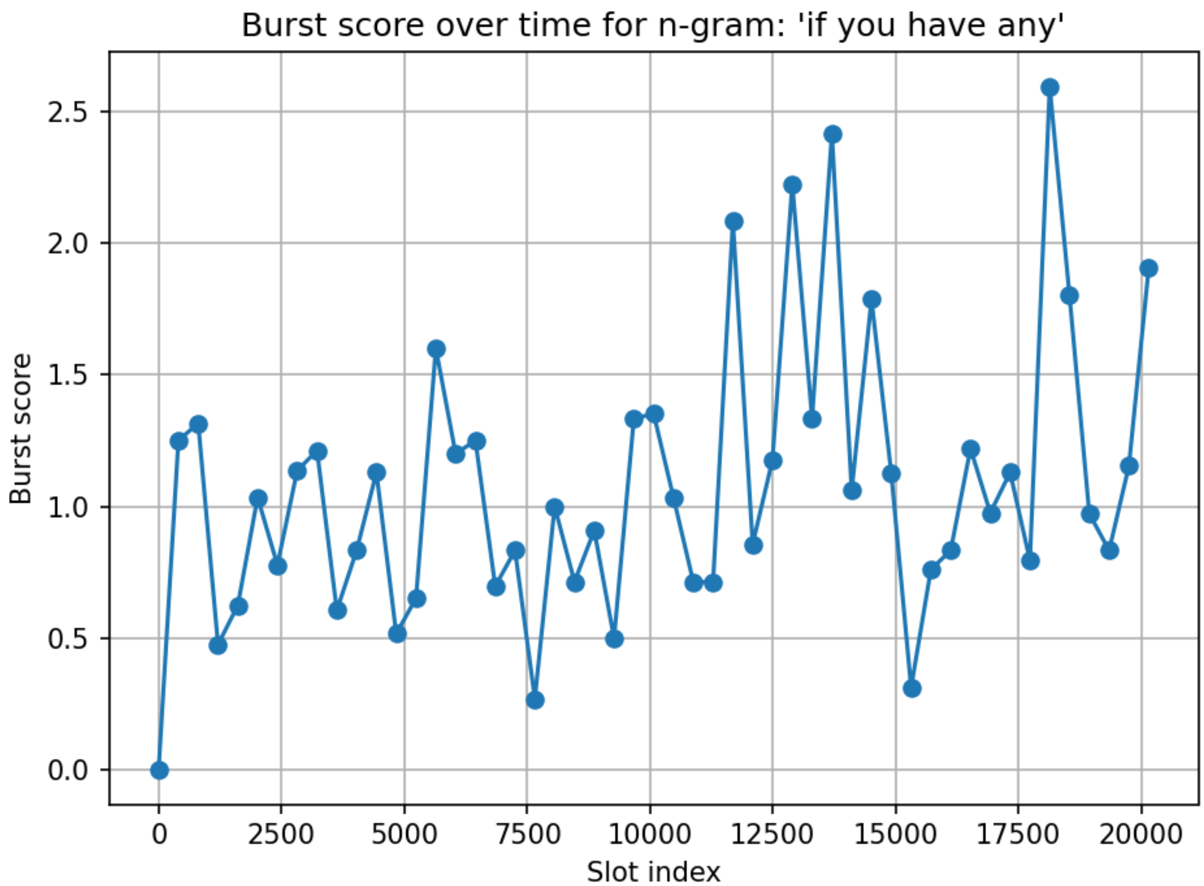

Figure 43: Hokusai burst scores over time for width 8192 and depth four.

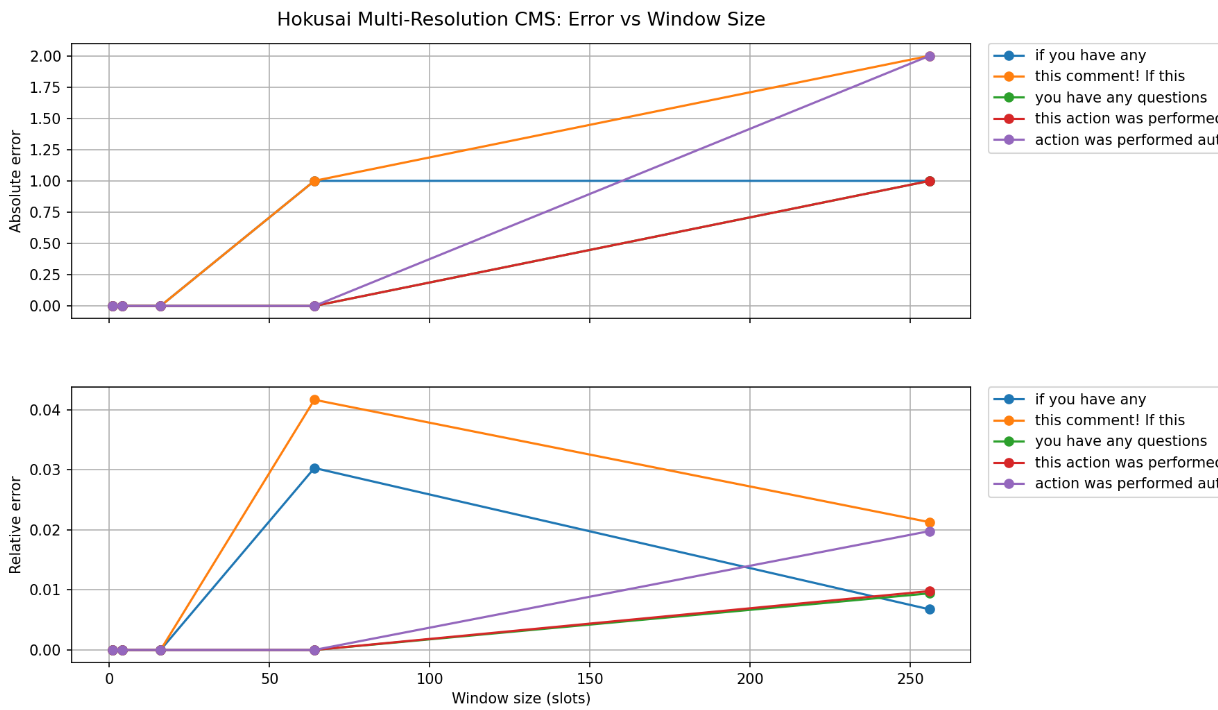

Figure 44: Hokusai error traces for width 8192 and depth four.

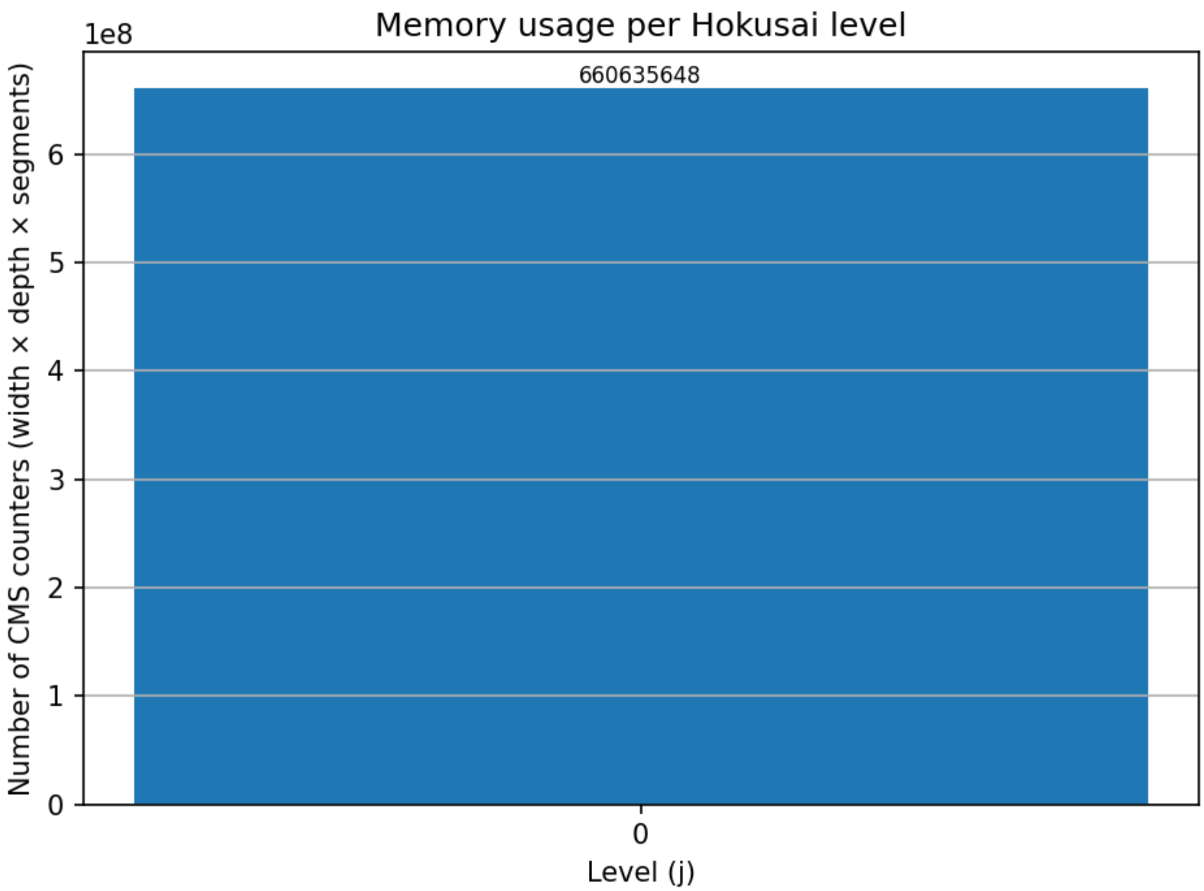

Figure 45: Hokusai memory footprint per level for width 8192 and depth four.

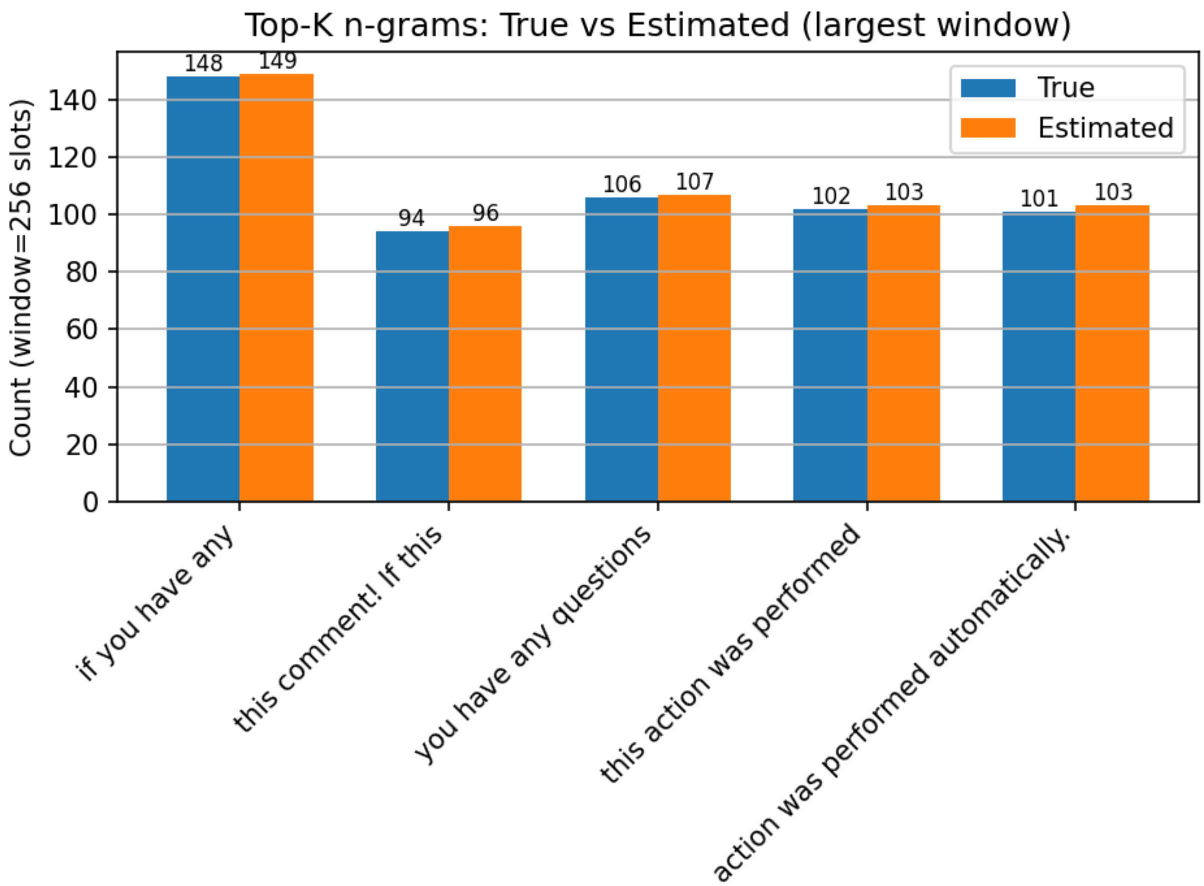

Figure 46: Hokusai top k counts for width 8192 and depth four.

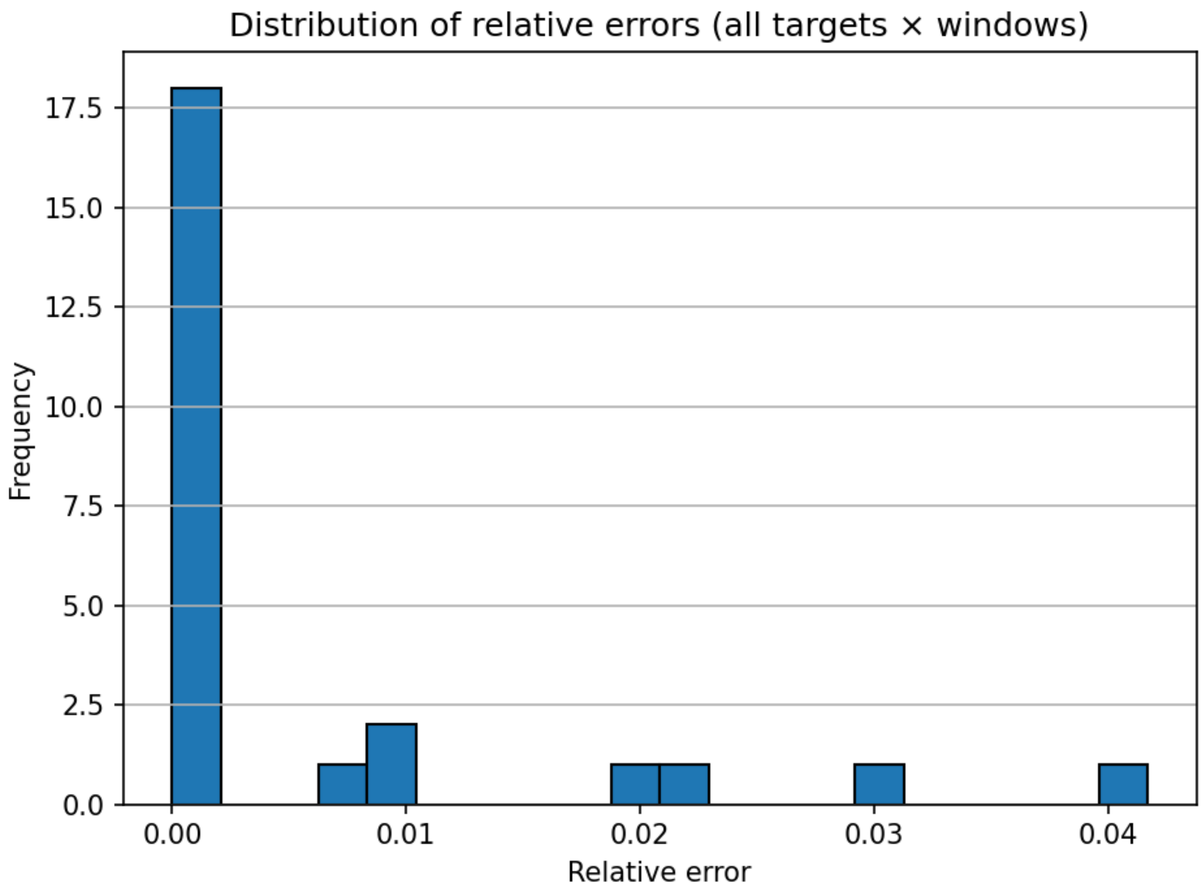

Figure 47: Histogram of relative error for Hokusai at width 8192 and depth four.

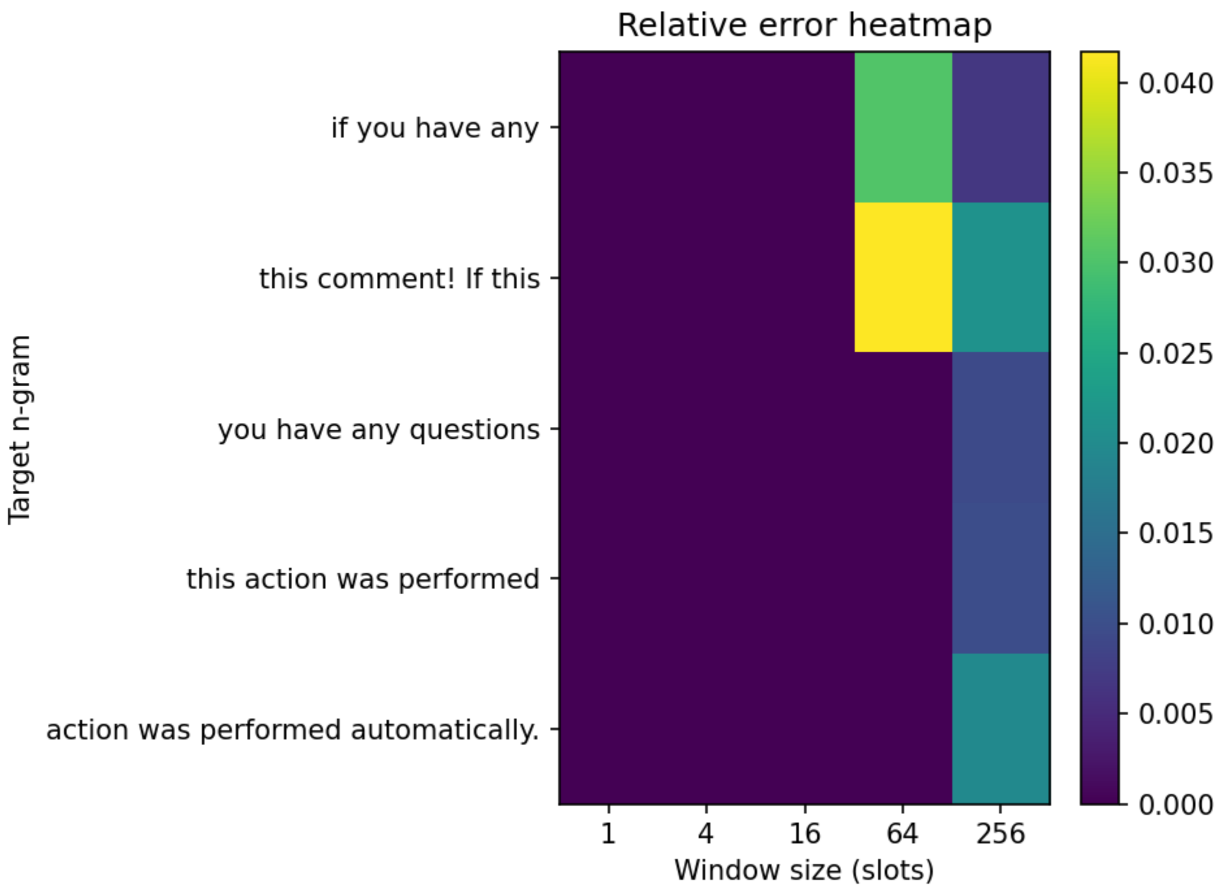

Figure 48: Heatmap of relative error for Hokusai at width 8192 and depth four.

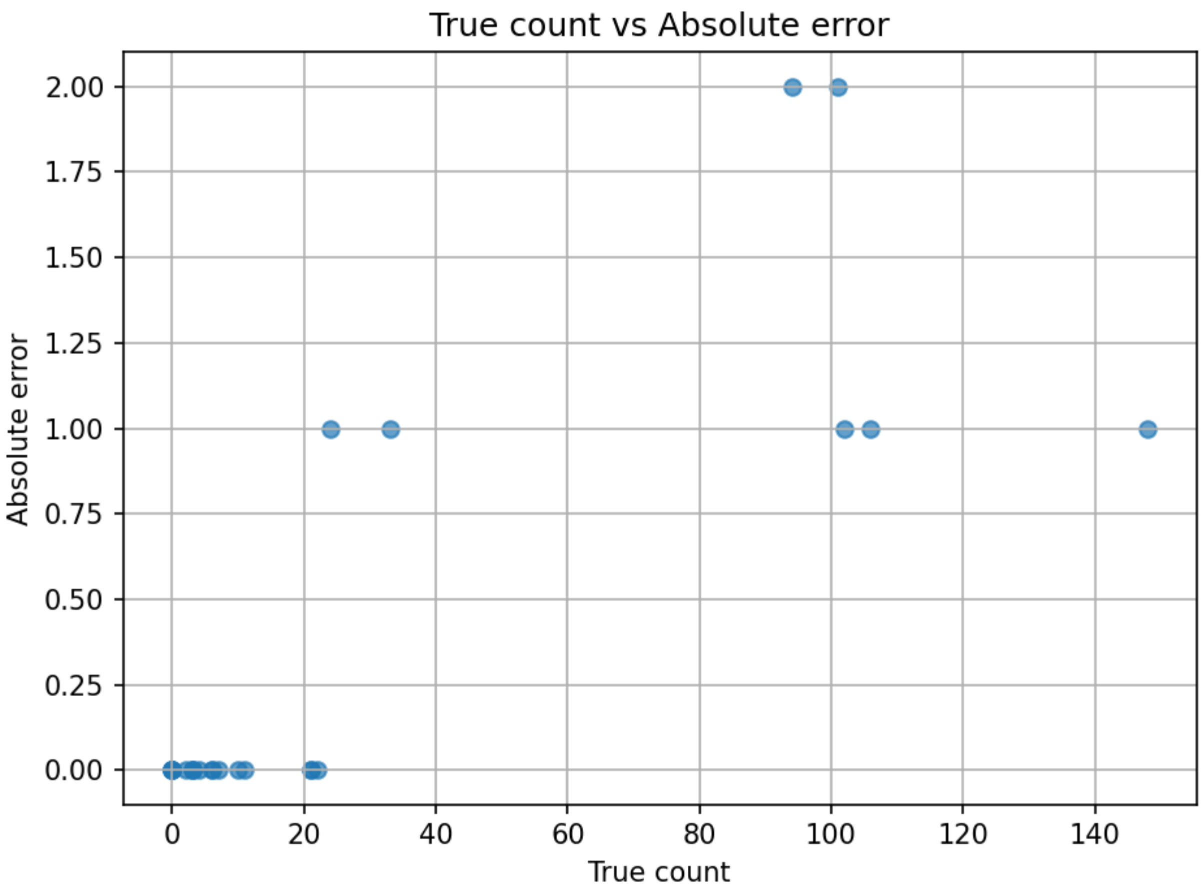

Figure 49: True counts versus absolute error for Hokusai at width 8192 and depth four.

A.3.2 Width 16384 depth four

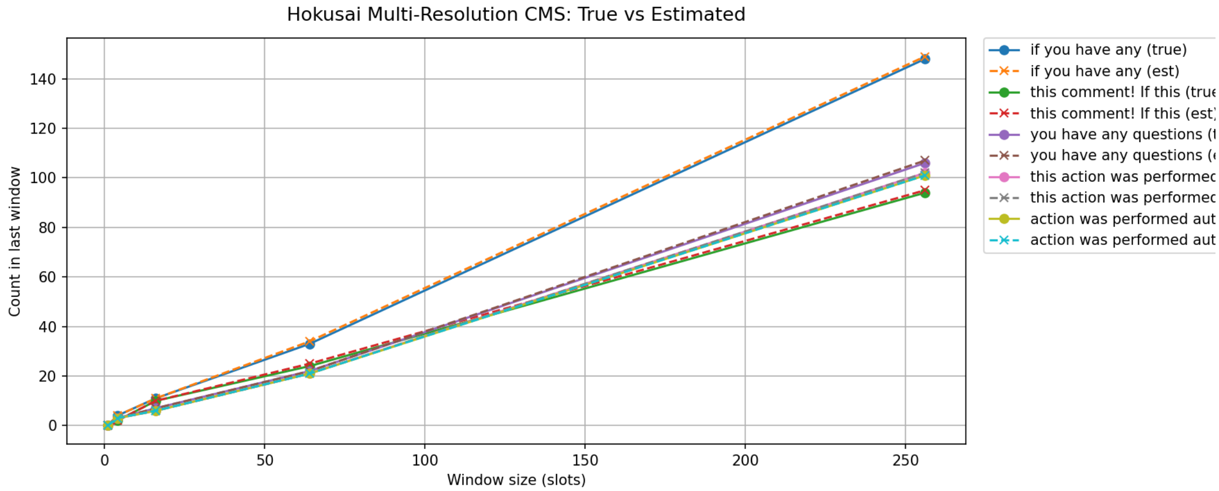

Figure 50: Hokusai counts versus ground truth for width 16384 and depth four.

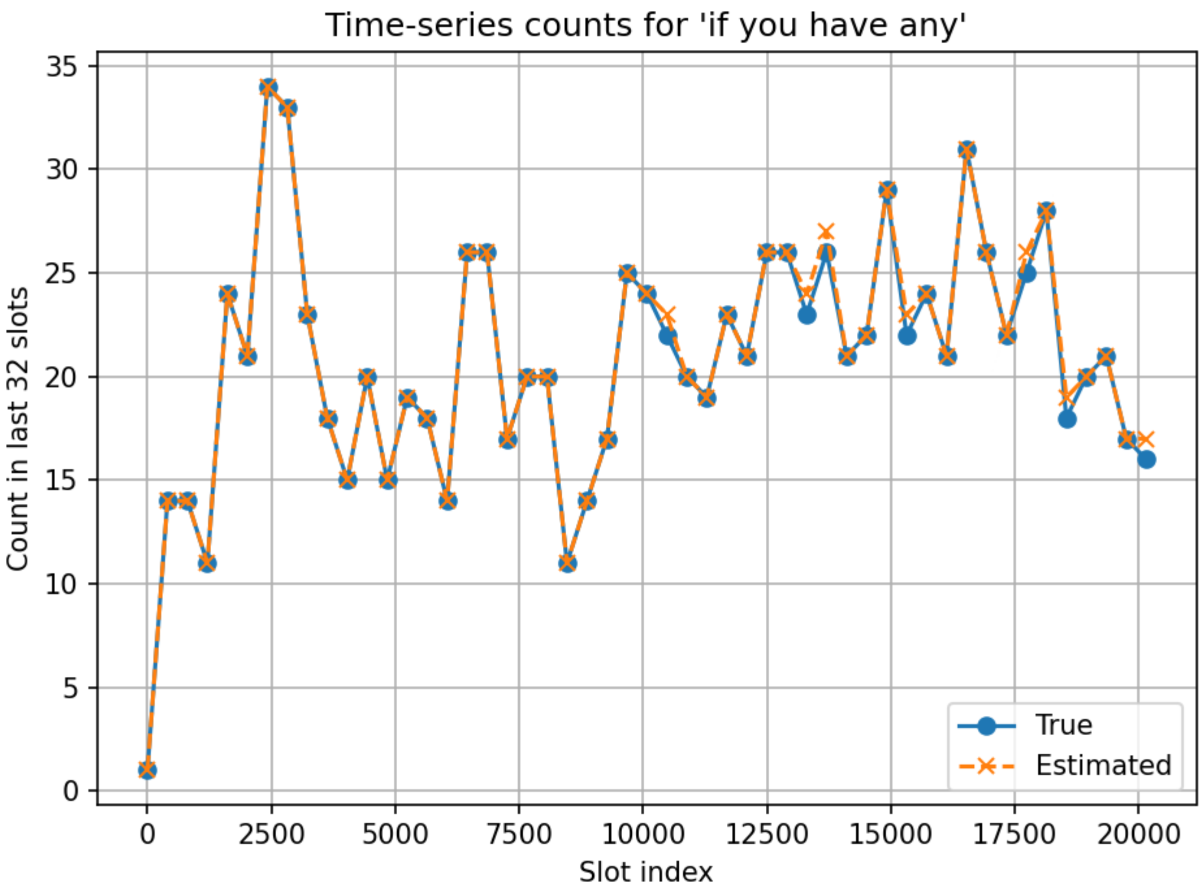

Figure 51: Hokusai time series estimates for width 16384 and depth four.

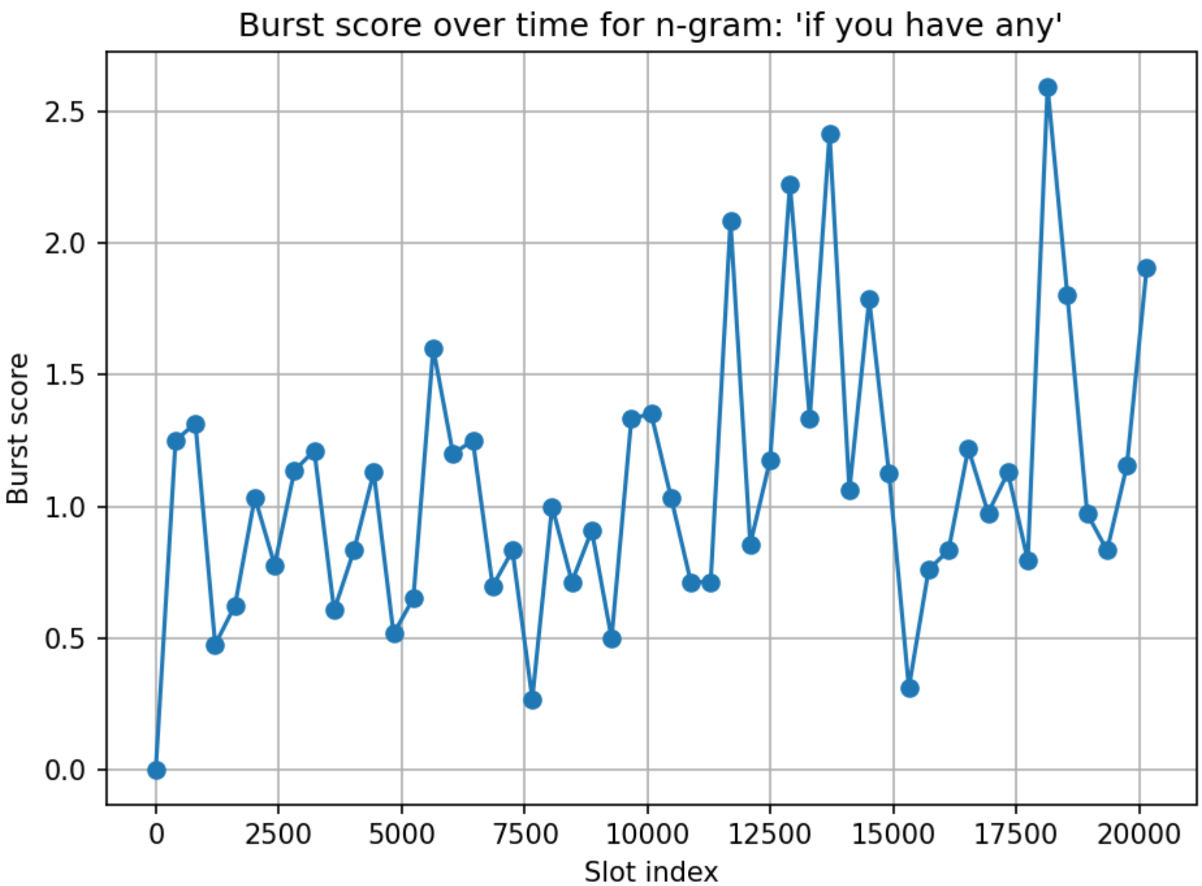

Figure 52: Hokusai burst scores over time for width 16384 and depth four.

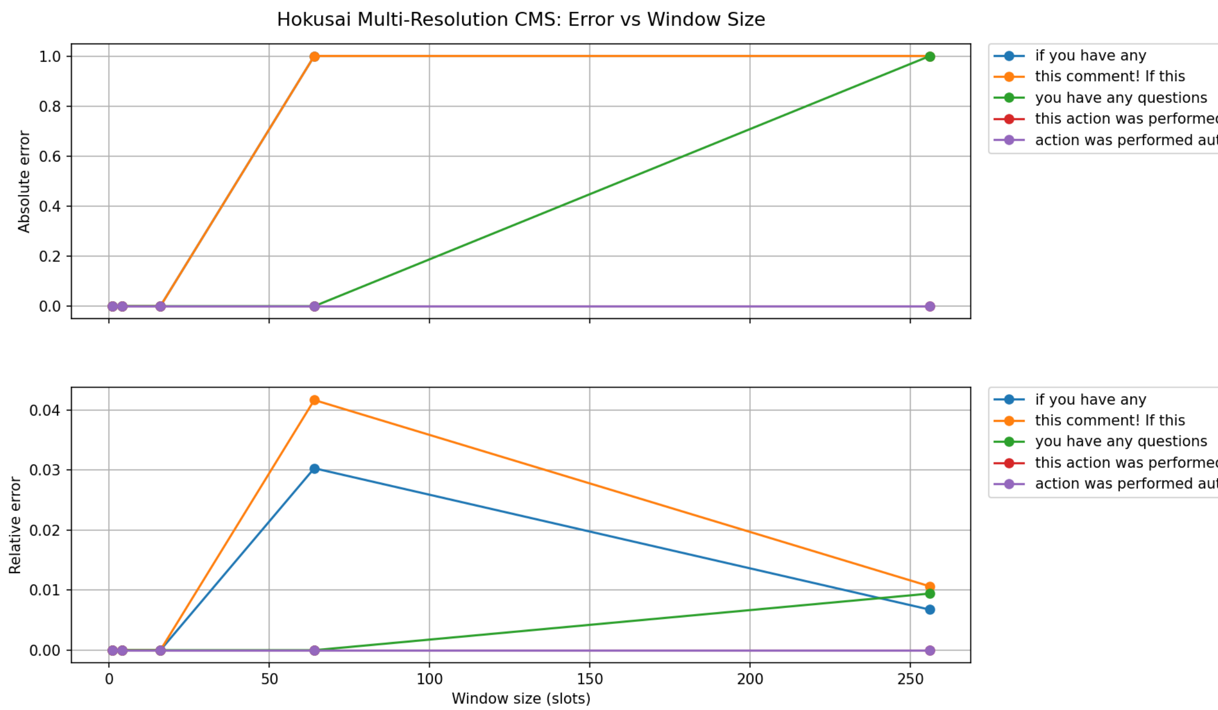

Figure 53: Hokusai error traces for width 16384 and depth four.

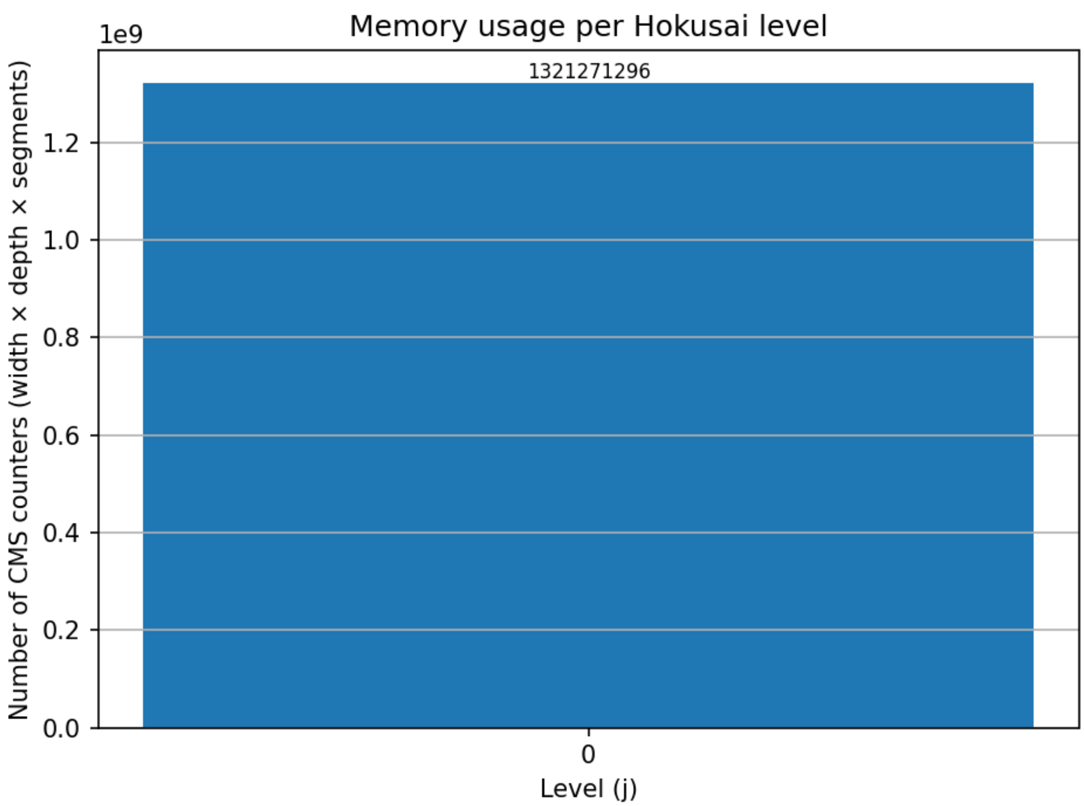

Figure 54: Hokusai memory footprint per level for width 16384 and depth four.

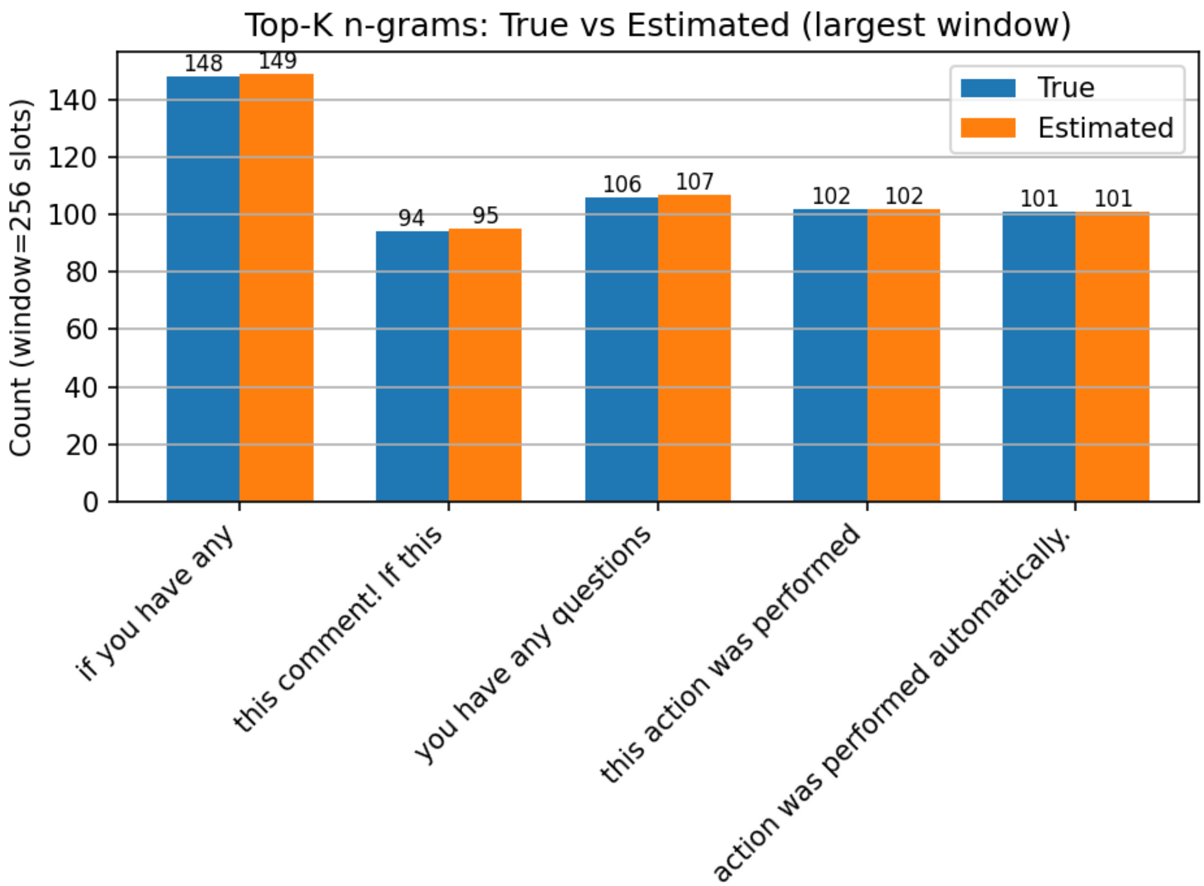

Figure 55: Hokusai top k counts for width 16384 and depth four.

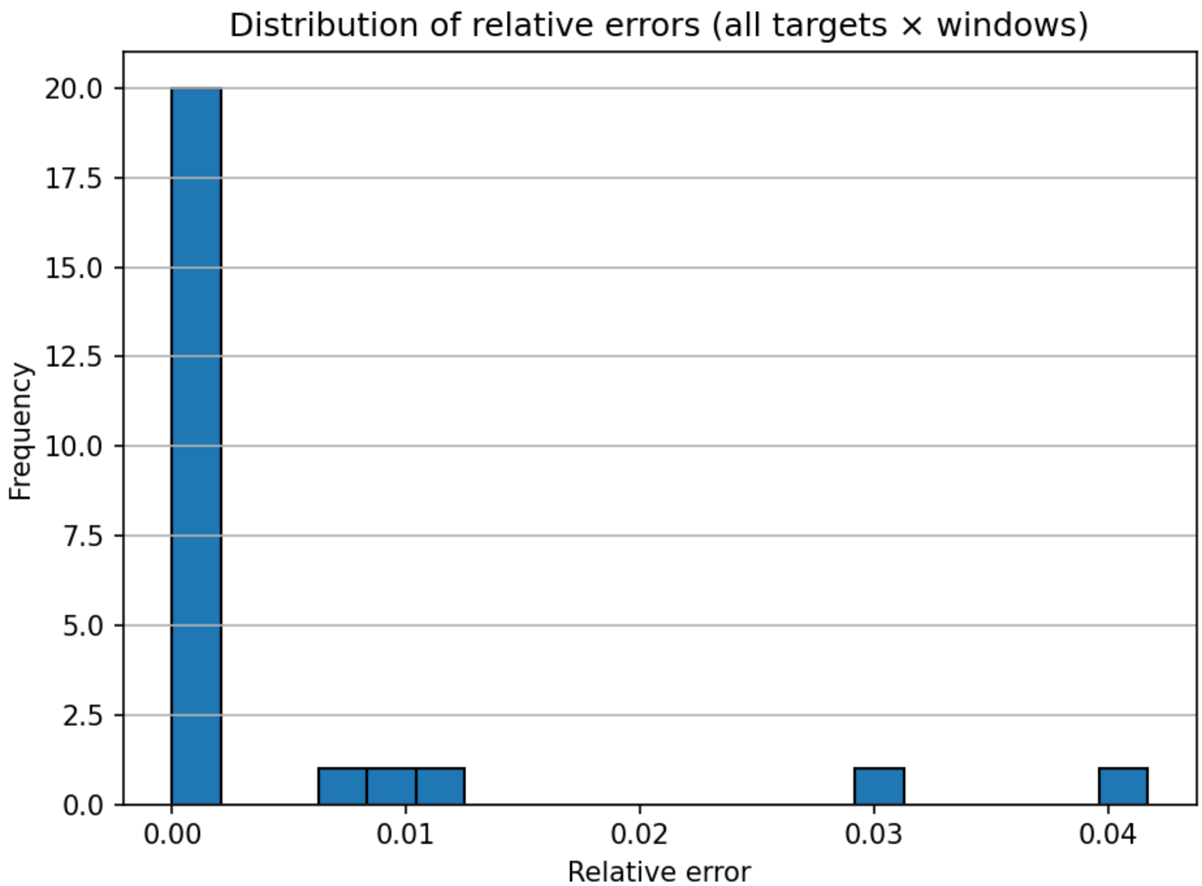

Figure 56: Histogram of relative error for Hokusai at width 16384 and depth four.

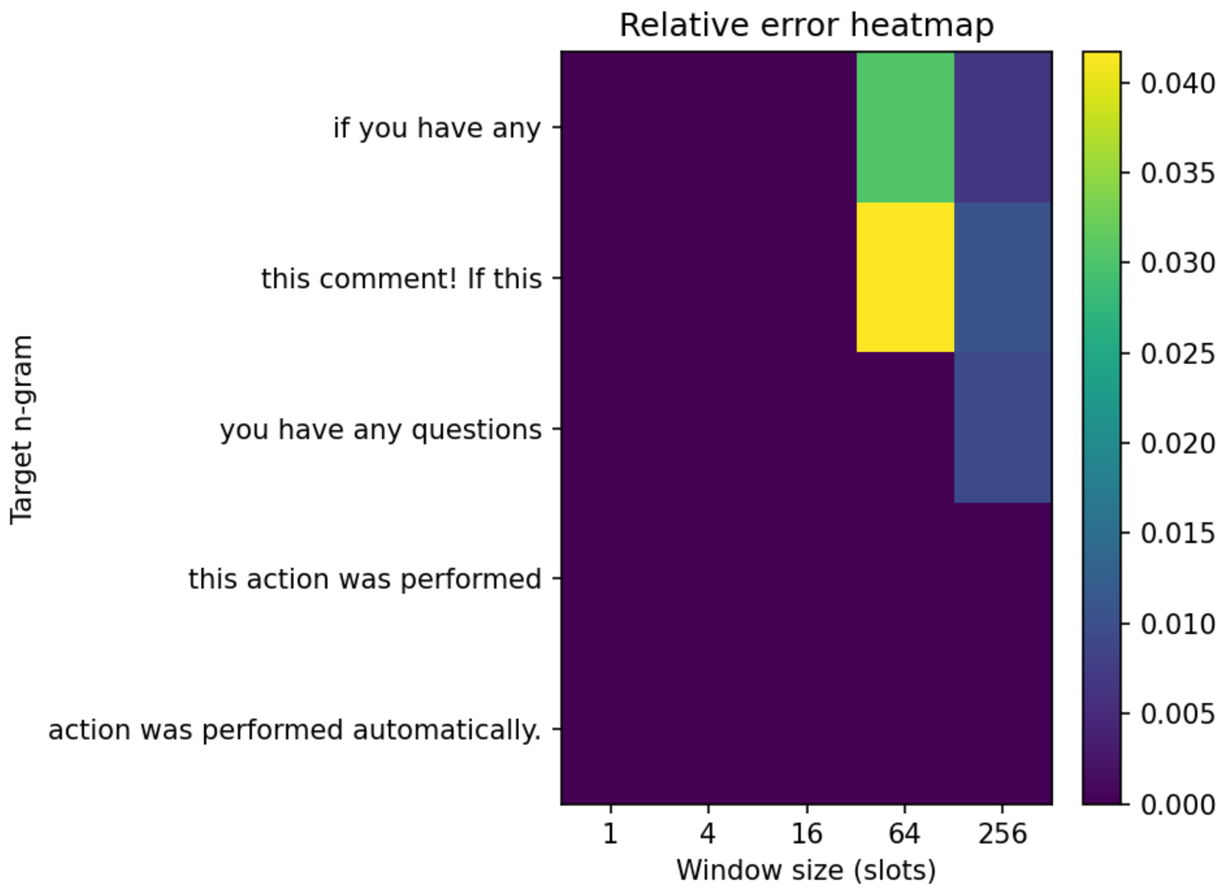

Figure 57: Heatmap of relative error for Hokusai at width 16384 and depth four.

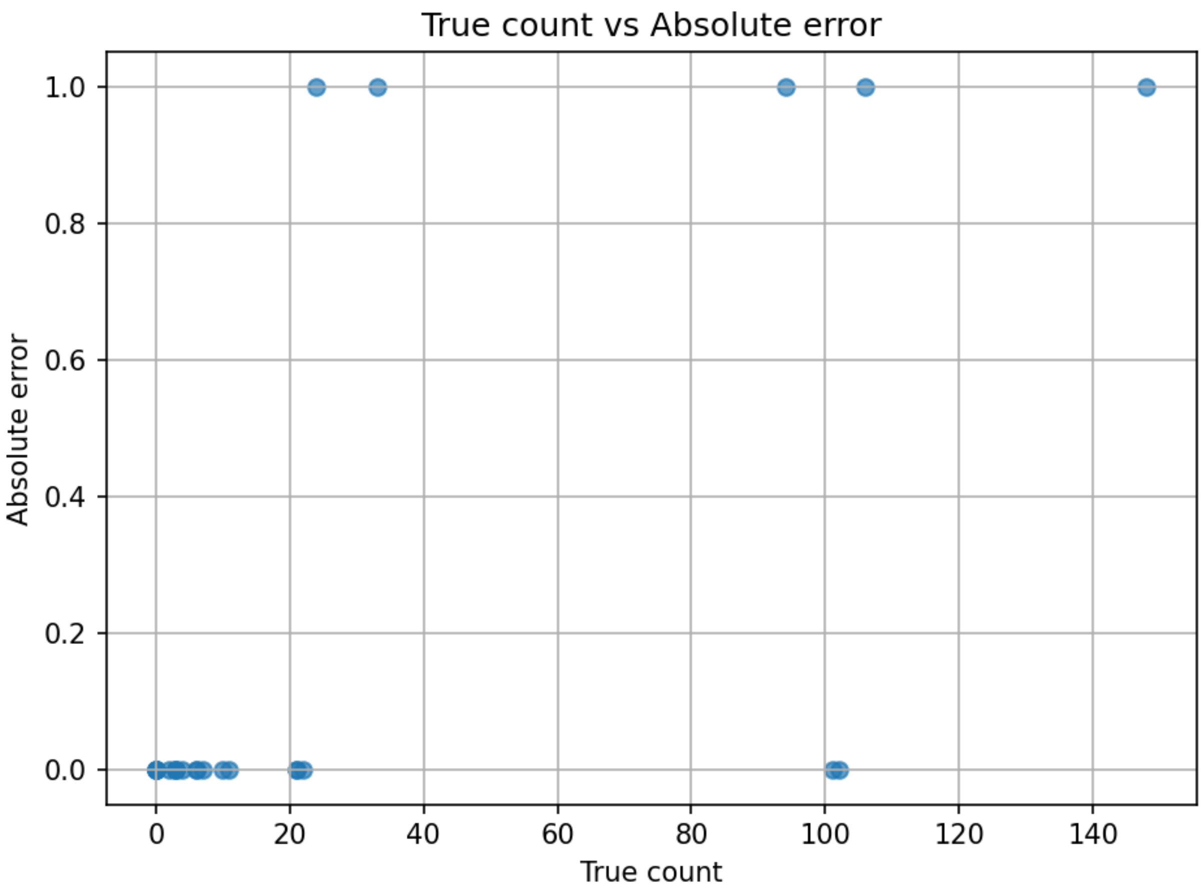

Figure 58: True counts versus absolute error for Hokusai at width 16384 and depth four.

A.3.3 Width 32768 depth four

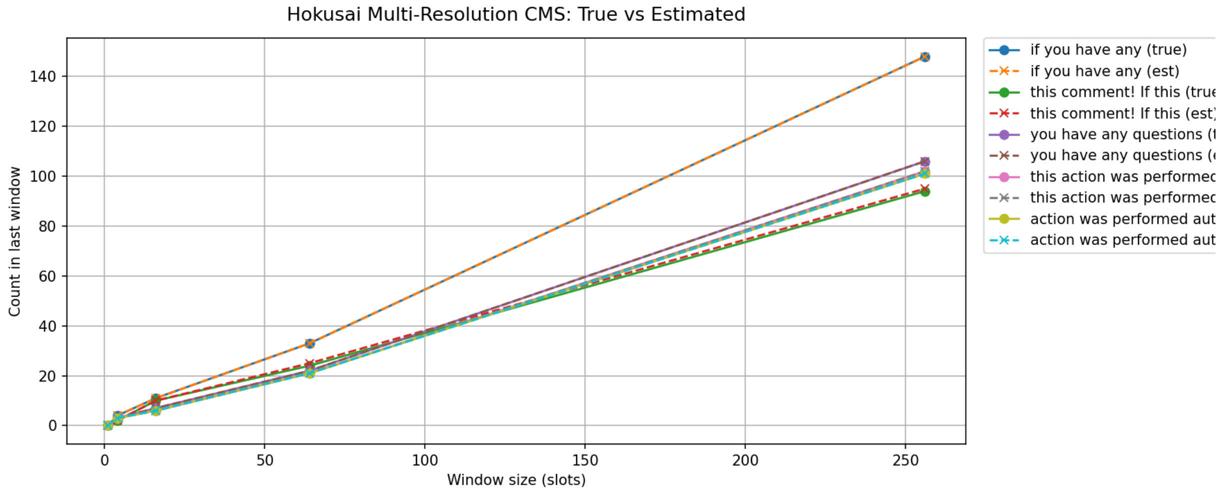

Figure 59: Hokusai counts versus ground truth for width 32768 and depth four.

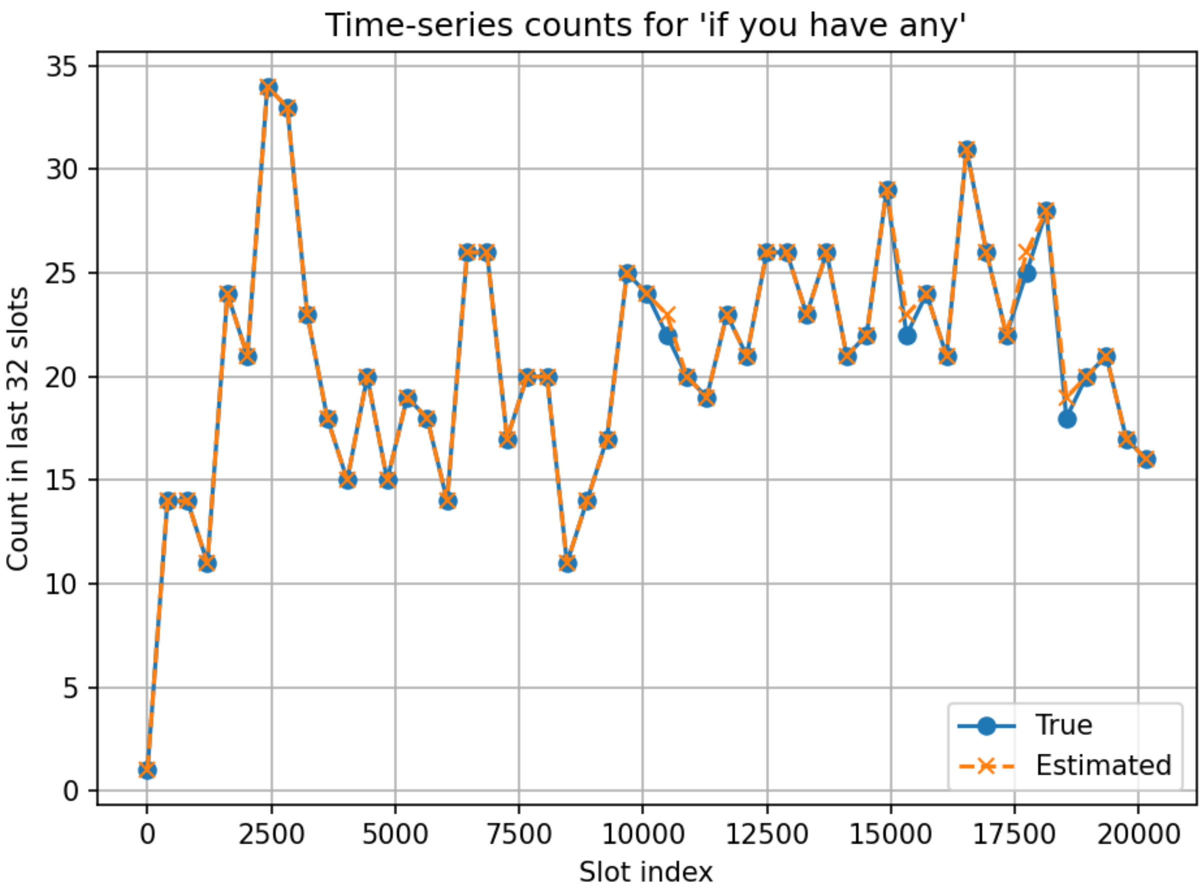

Figure 60: Hokusai time series estimates for width 32768 and depth four.

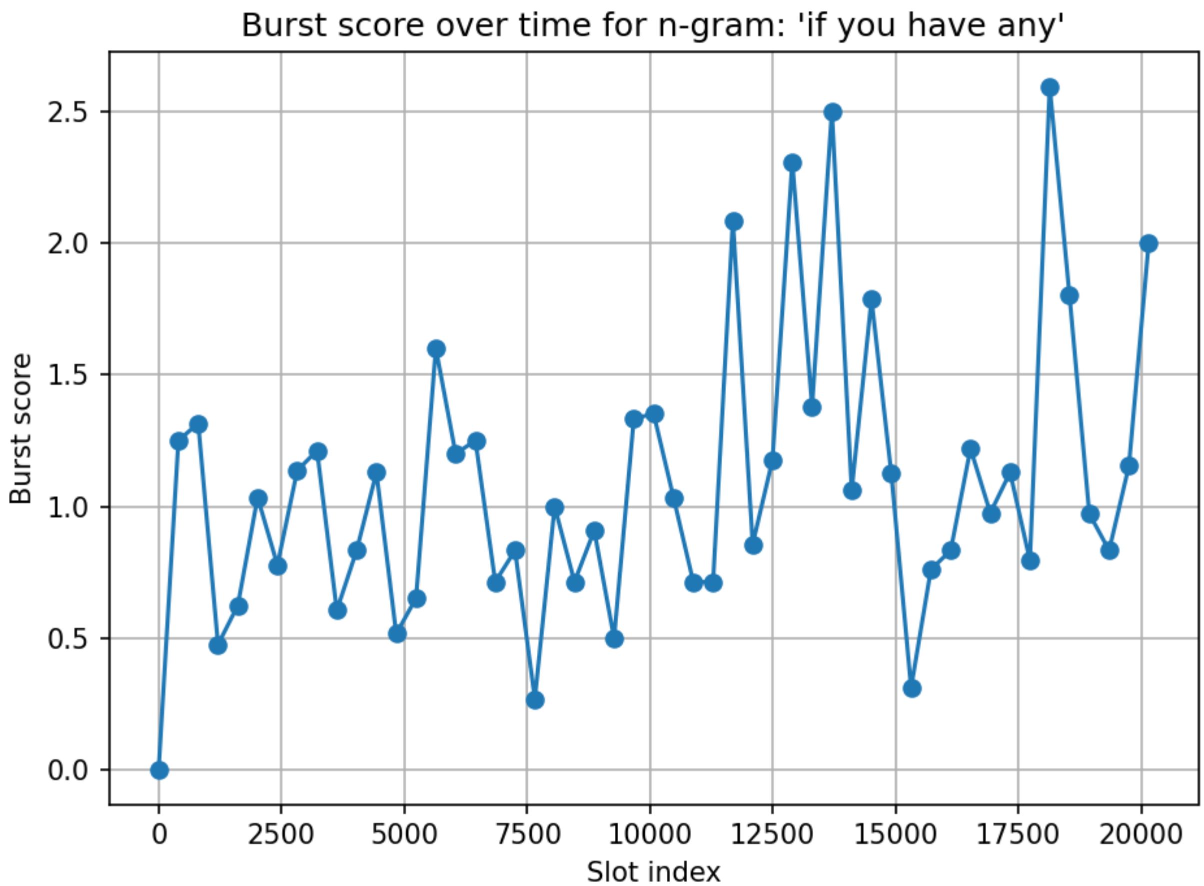

Figure 61: Hokusai burst scores over time for width 32768 and depth four.

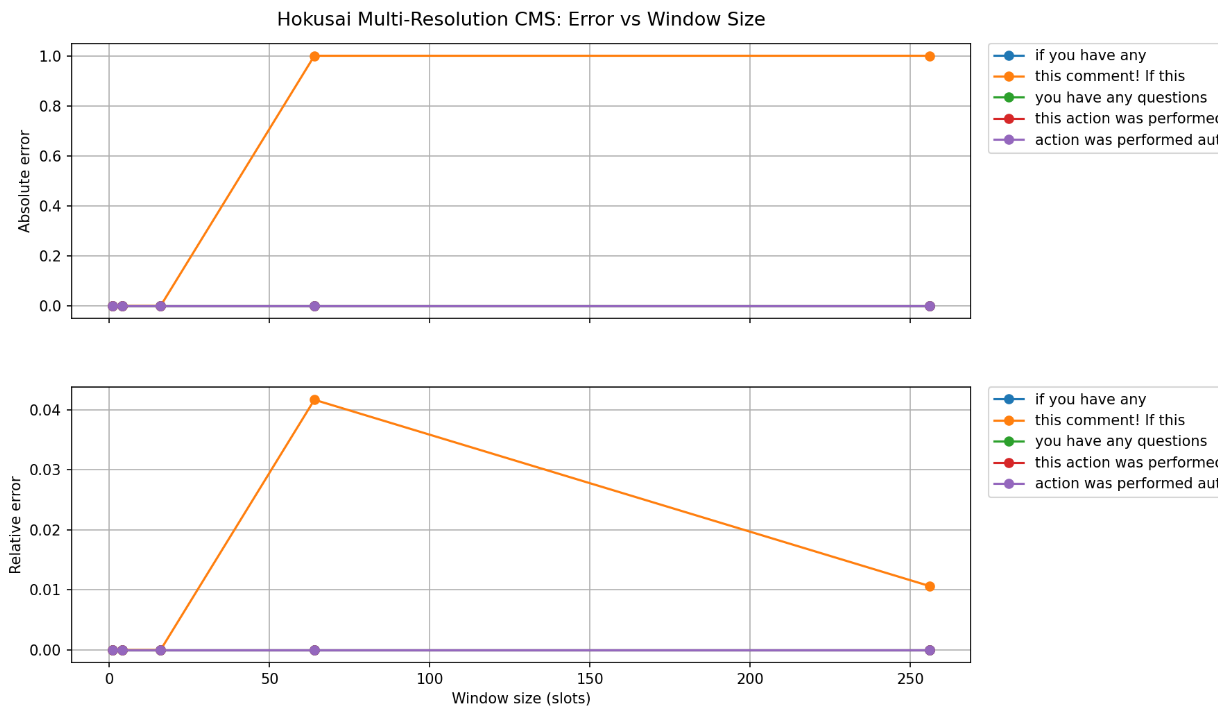

Figure 62: Hokusai error traces for width 32768 and depth four.

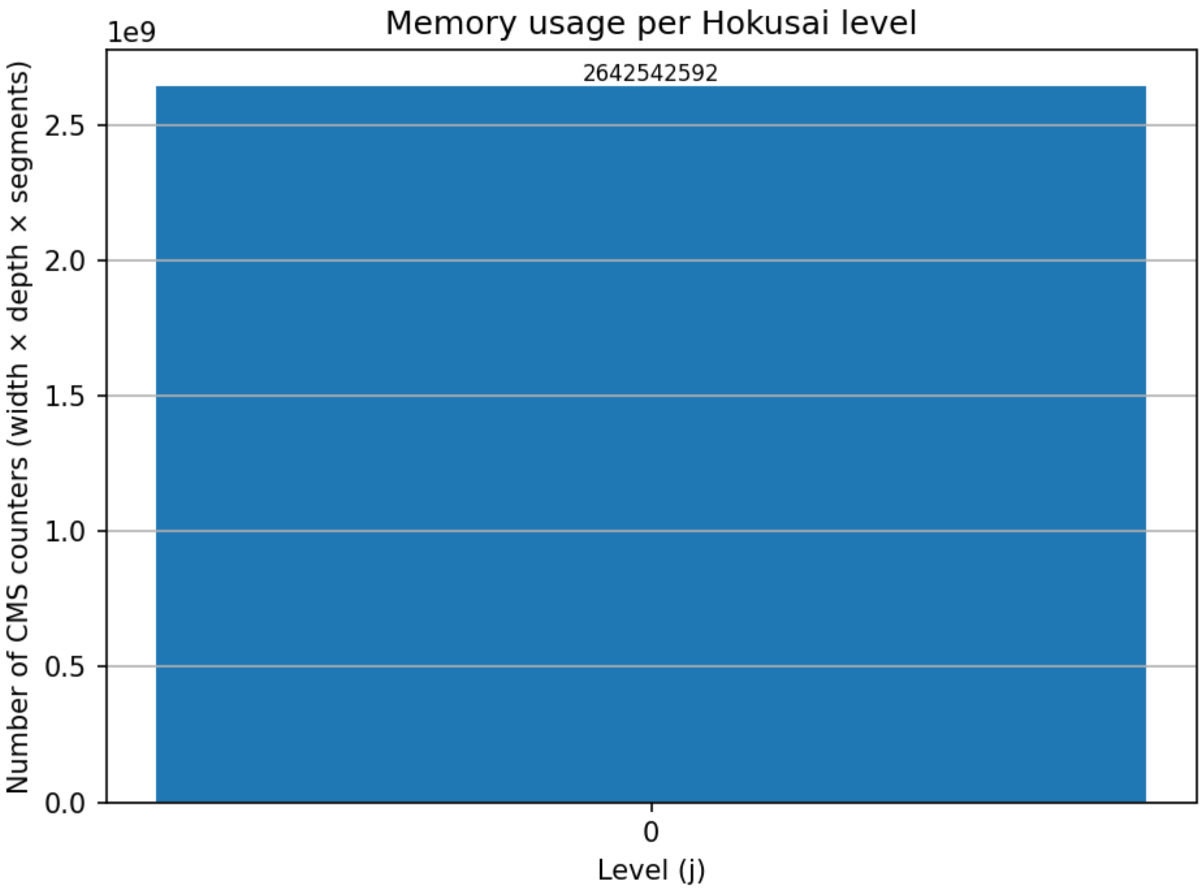

Figure 63: Hokusai memory footprint per level for width 32768 and depth four.

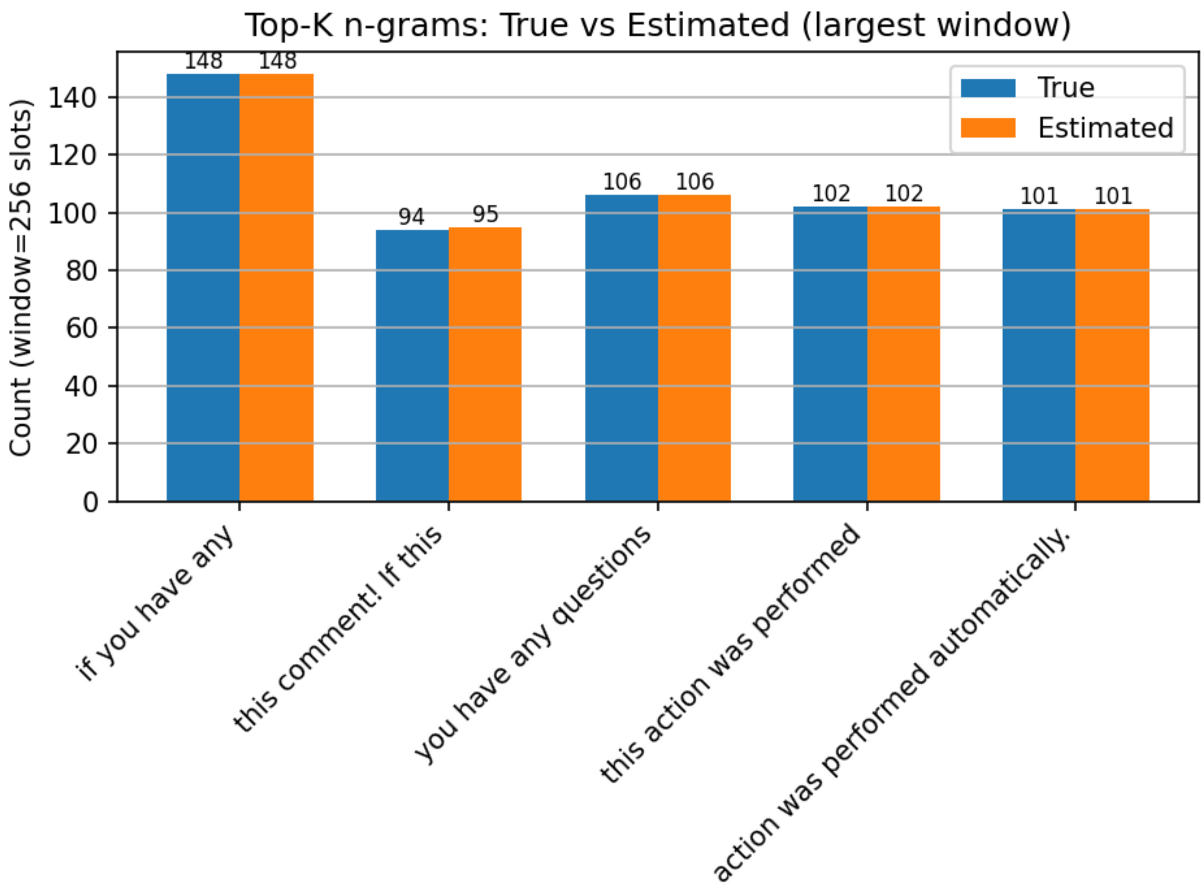

Figure 64: Hokusai top k counts for width 32768 and depth four.

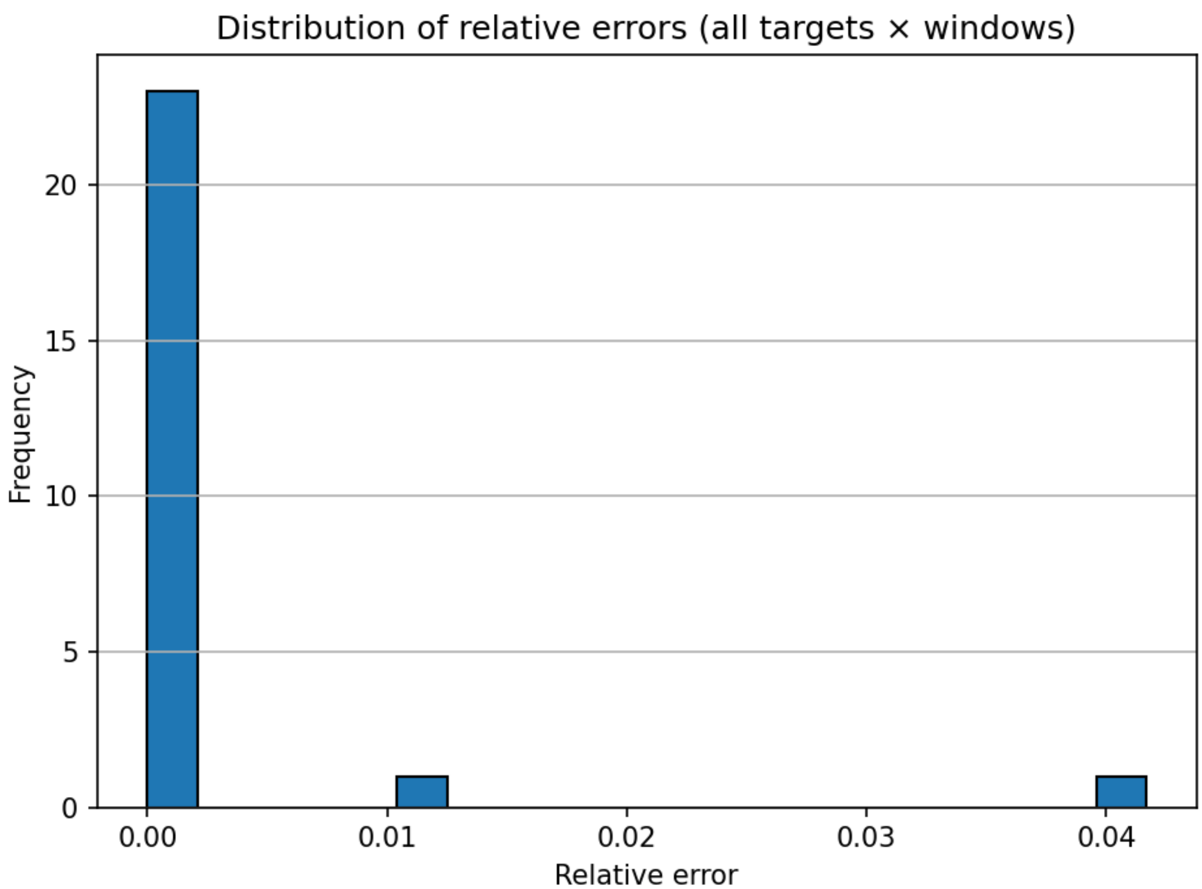

Figure 65: Histogram of relative error for Hokusai at width 32768 and depth four.

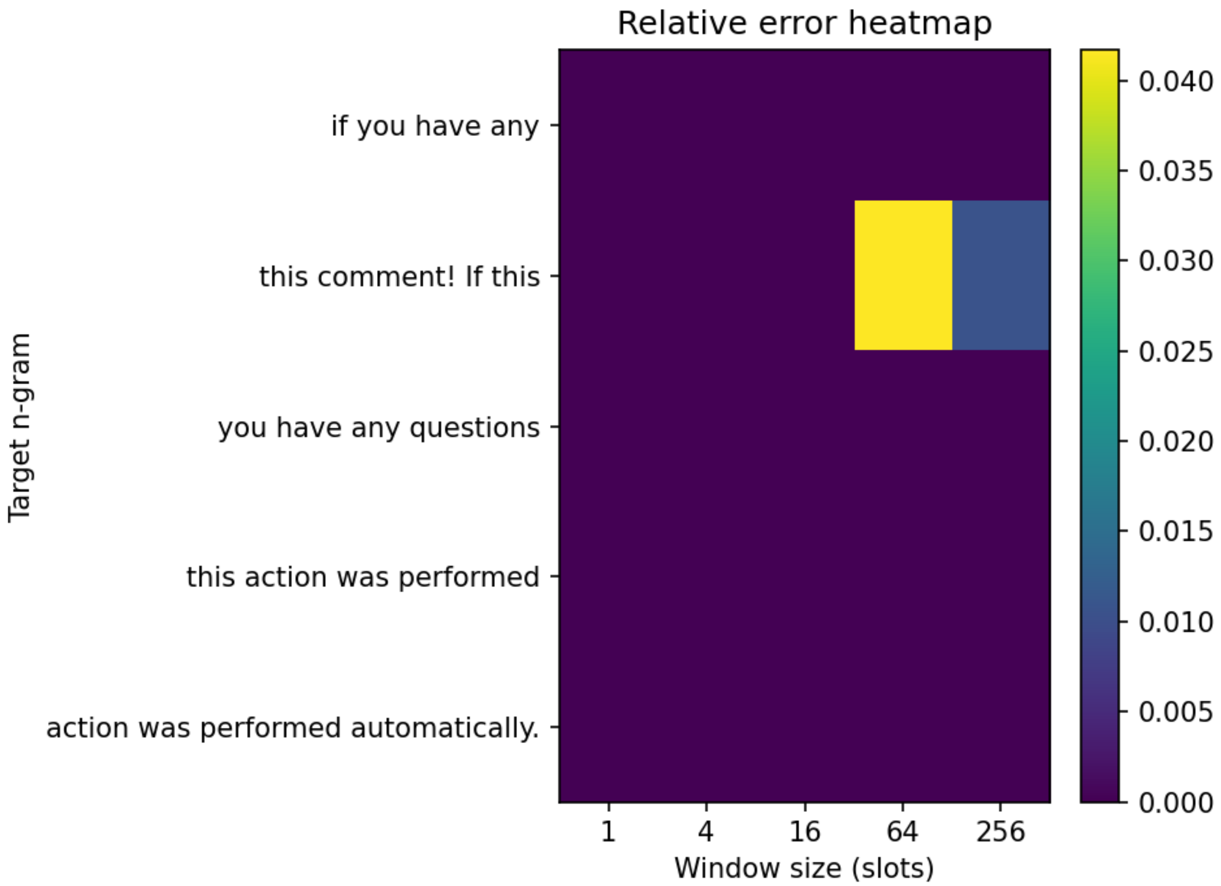

Figure 66: Heatmap of relative error for Hokusai at width 32768 and depth four.

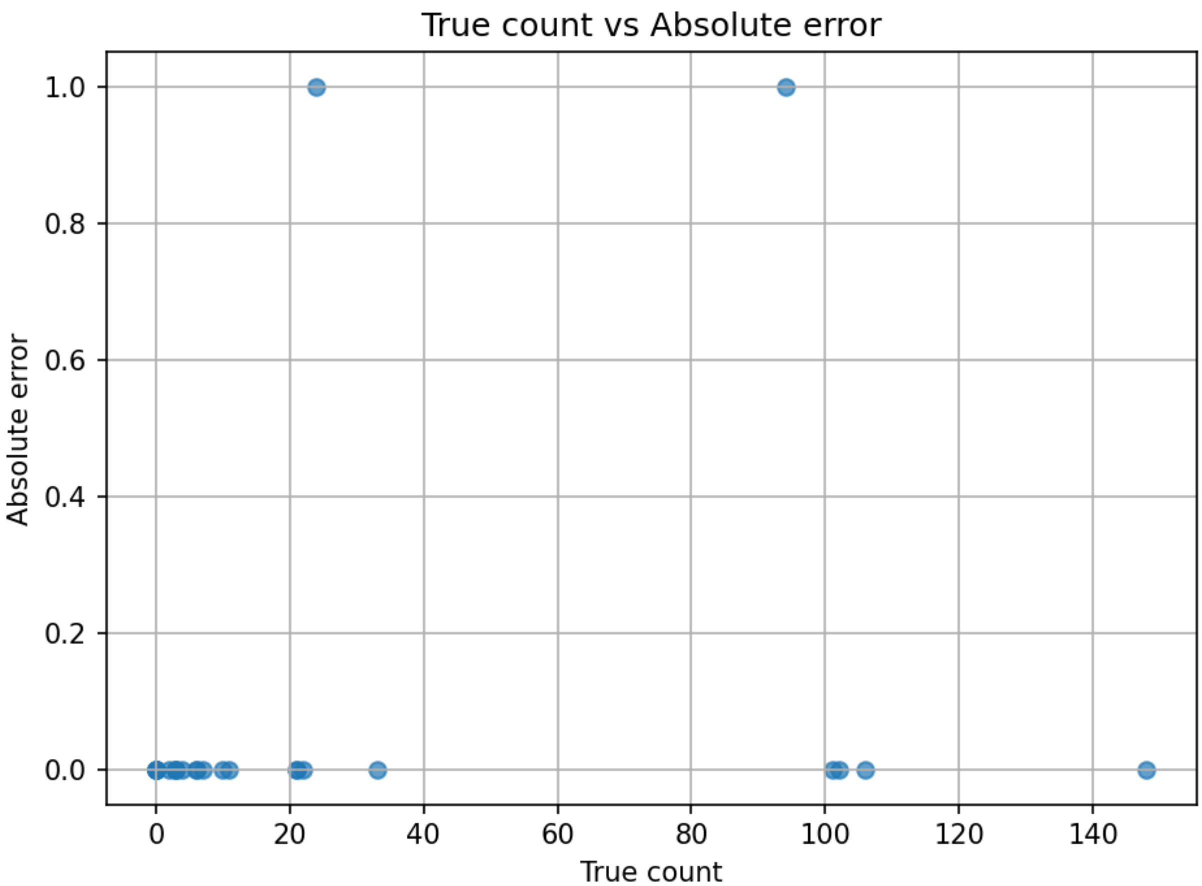

Figure 67: True counts versus absolute error for Hokusai at width 32768 and depth four.

A.4 Ada Sketch results

Per-width Ada-Sketch plots: time series and counts versus ground truth, error distributions, memory proxies, top-k, and scatter views.

A.4.1 Width 8192 depth four

Figure 68: Ada Sketch time series estimates for width 8192 and depth four.

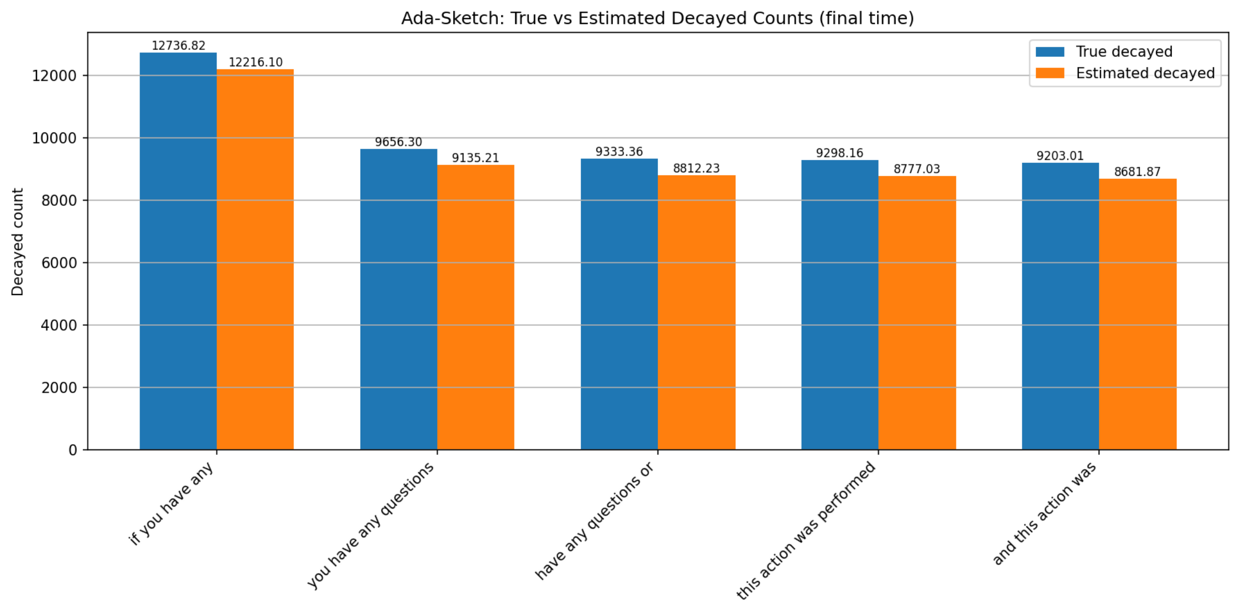

Figure 69: Ada Sketch counts versus ground truth for width 8192 and depth four.

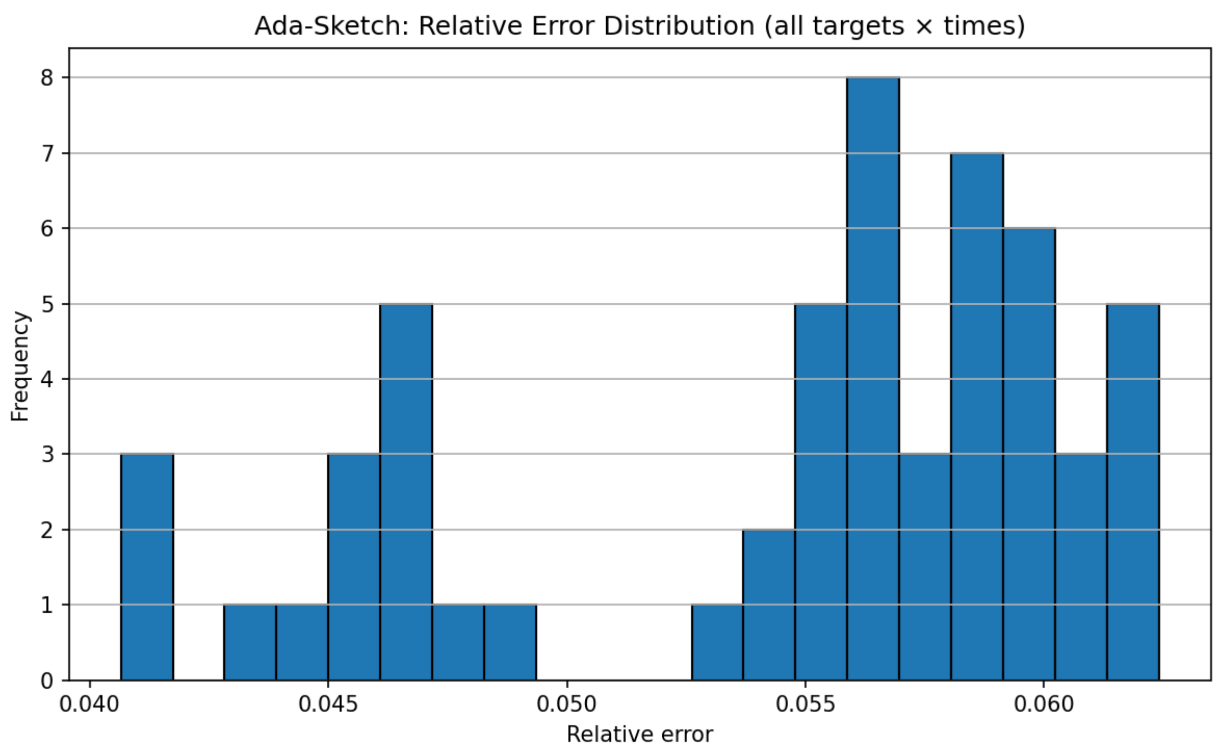

Figure 70: Histogram of relative error for Ada Sketch at width 8192 and depth four.

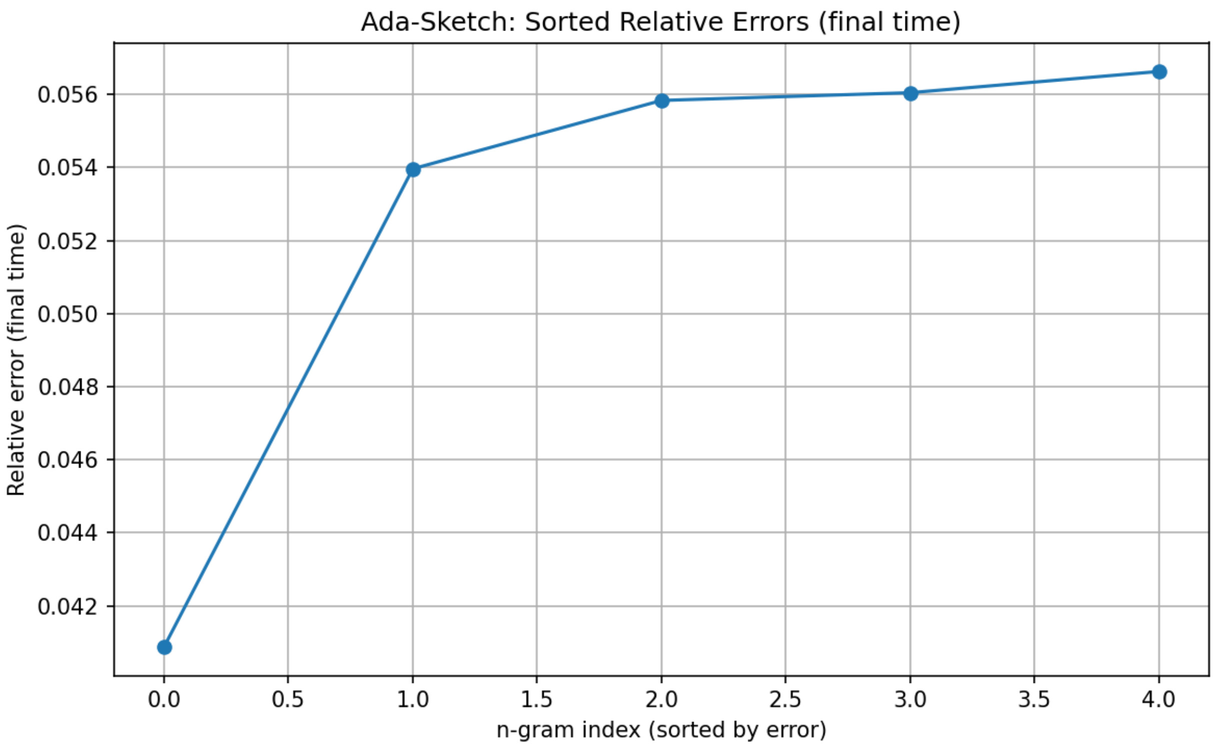

Figure 71: Sorted relative error for Ada Sketch at width 8192 and depth four.

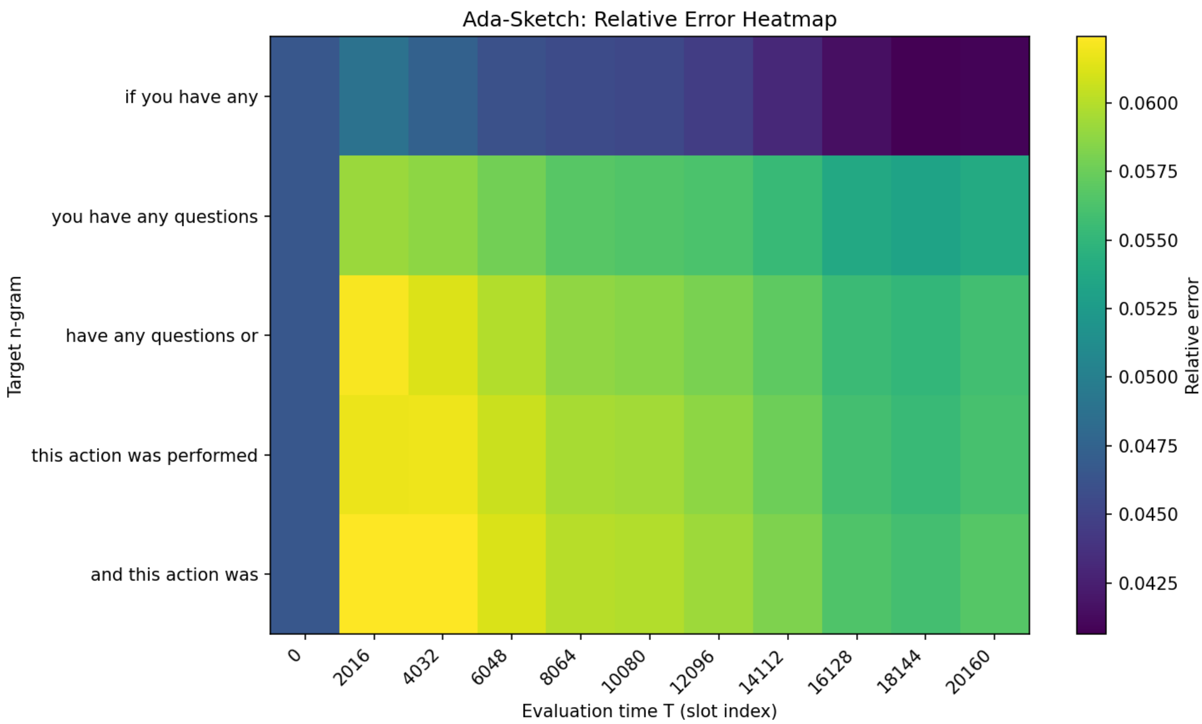

Figure 72: Heatmap of relative error for Ada Sketch at width 8192 and depth four.

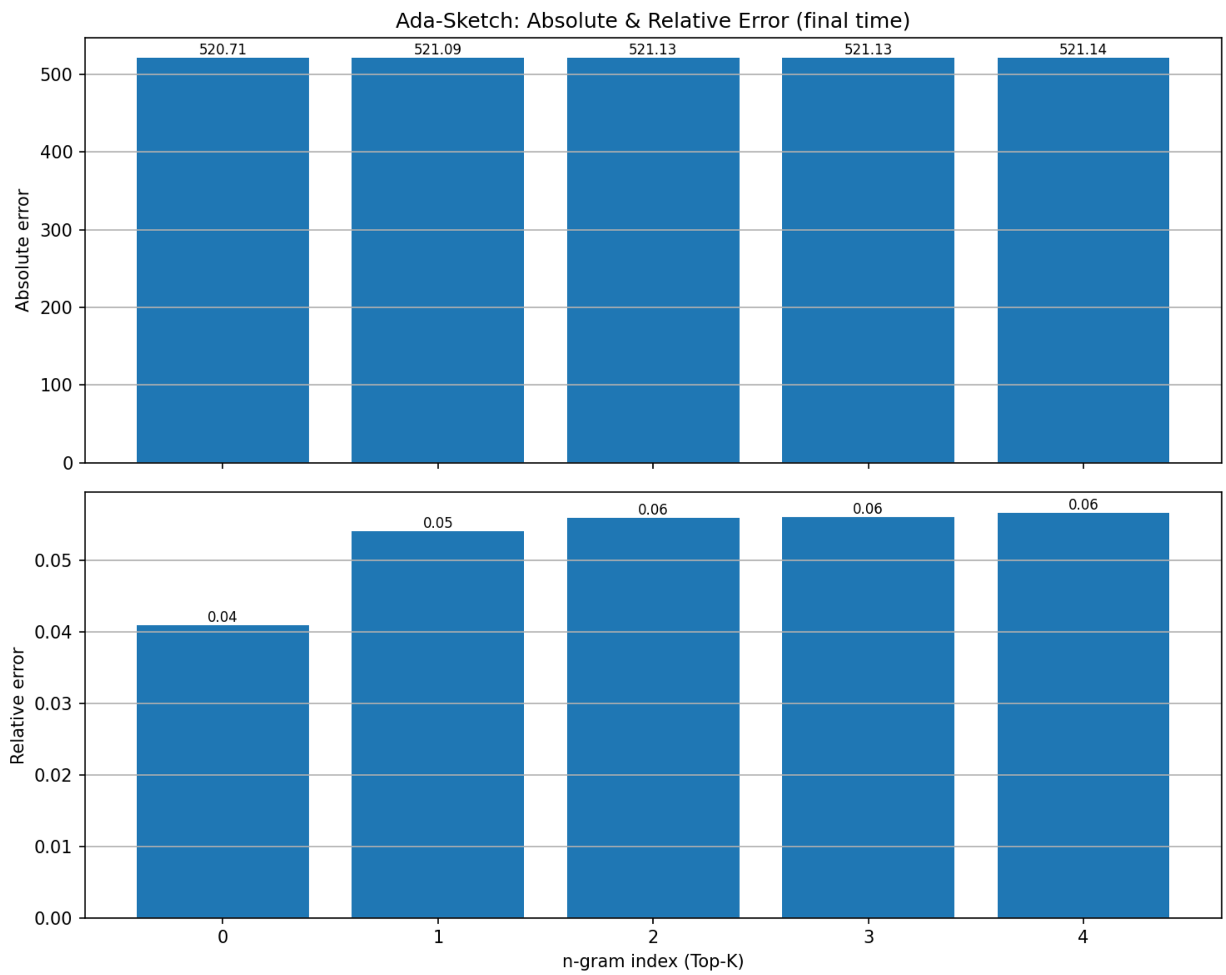

Figure 73: Absolute relative error for Ada Sketch at width 8192 and depth four.

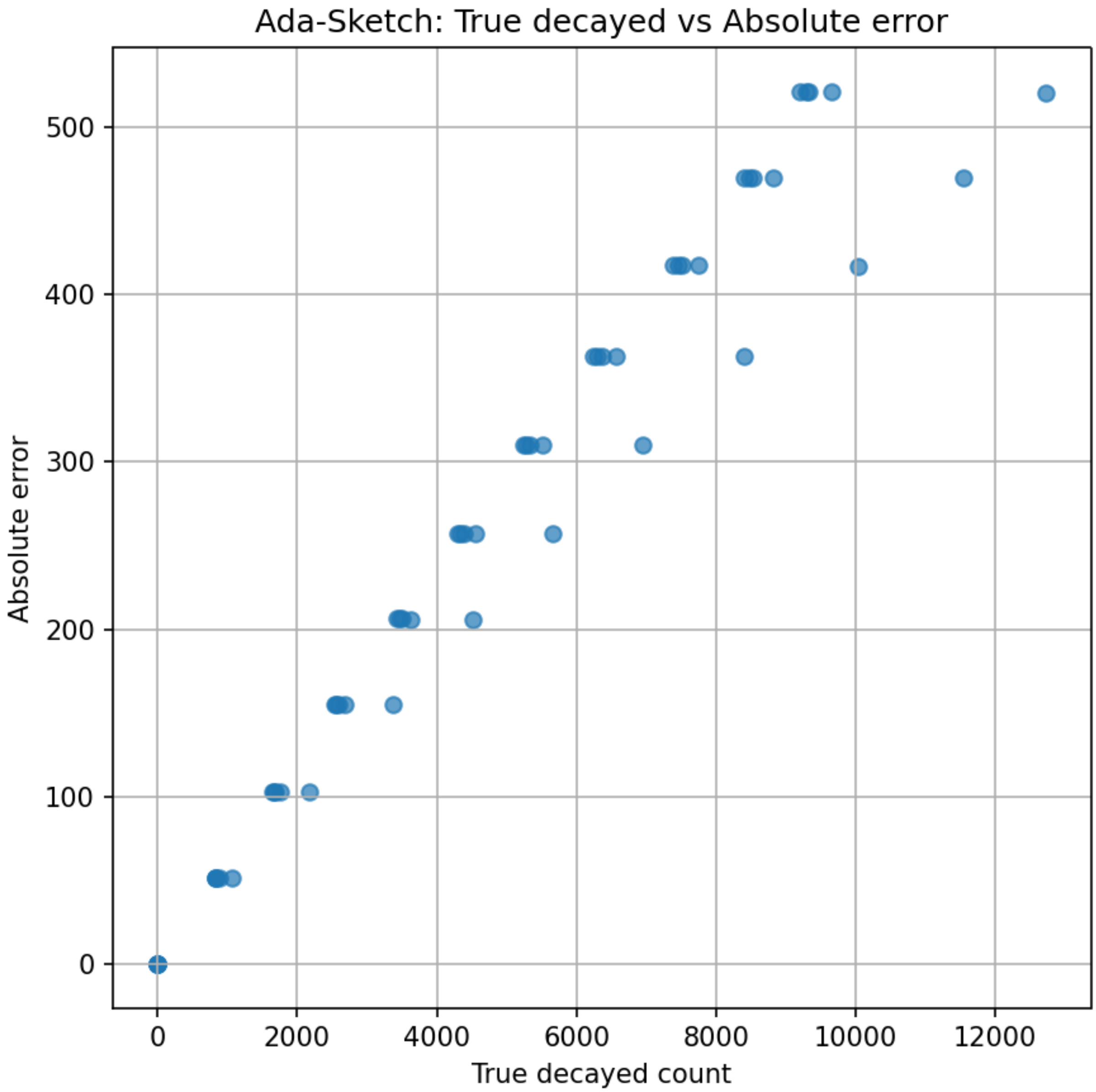

Figure 74: True counts versus absolute error for Ada Sketch at width 8192 and depth four.

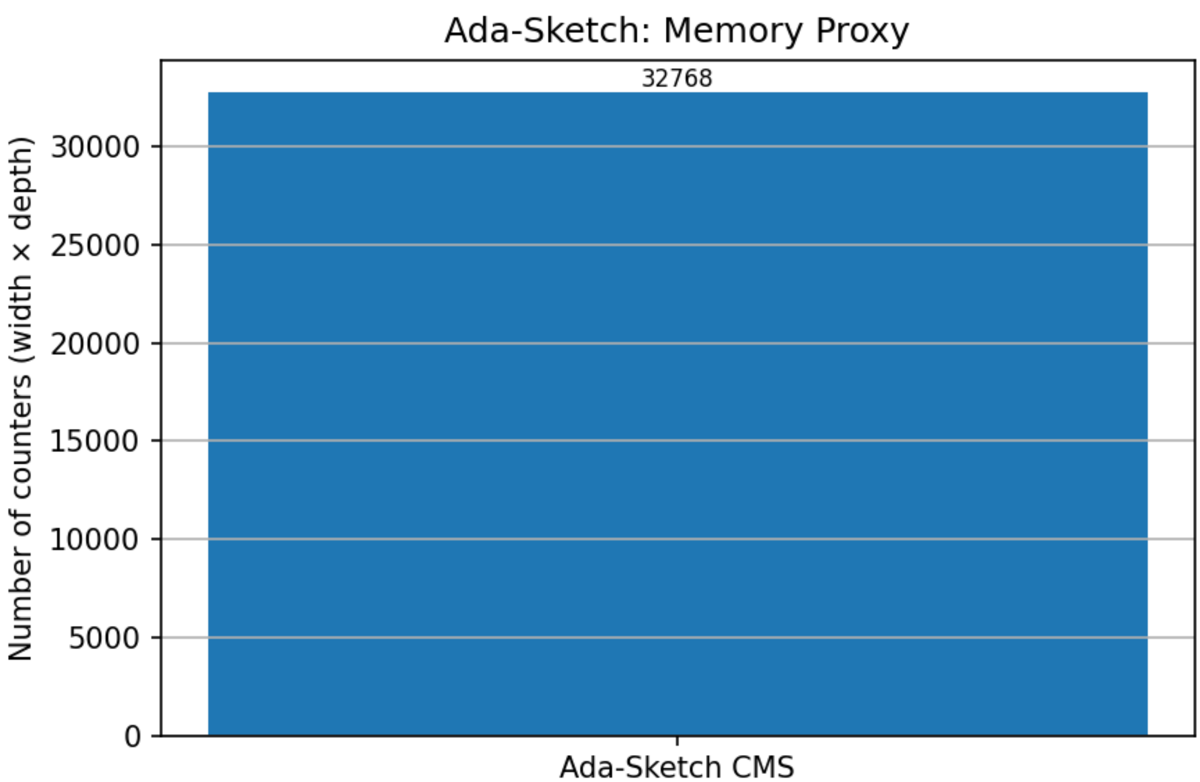

Figure 75: Memory proxy for Ada Sketch at width 8192 and depth four.

A.4.2 Width 16384 depth four

Figure 76: Ada Sketch time series estimates for width 16384 and depth four.

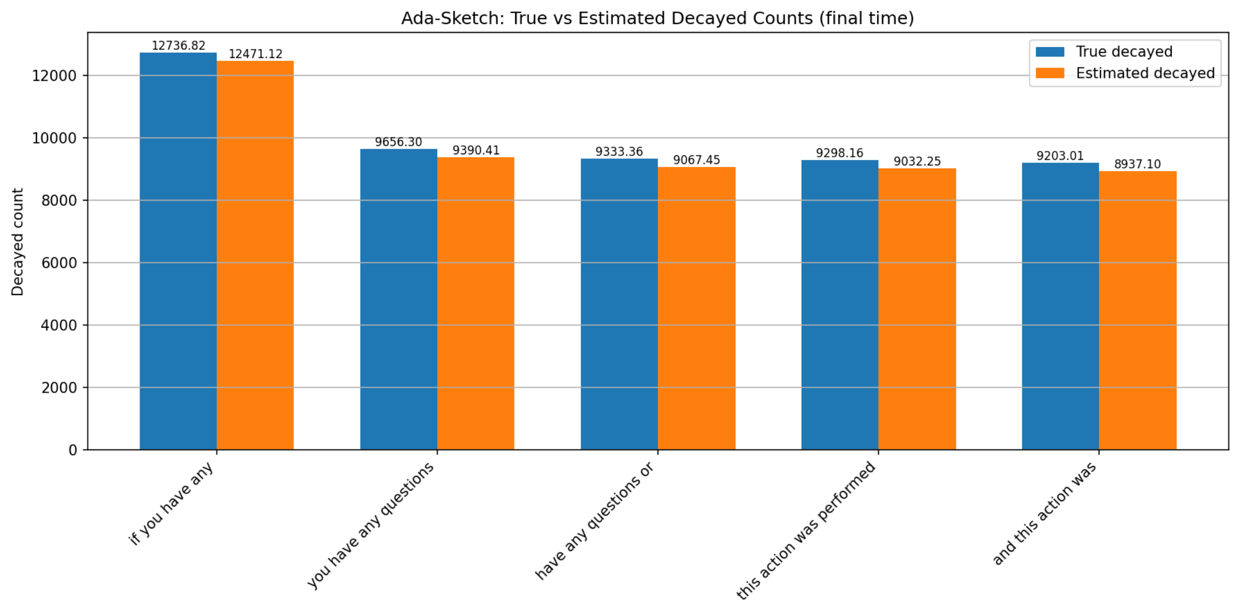

Figure 77: Ada Sketch counts versus ground truth for width 16384 and depth four.

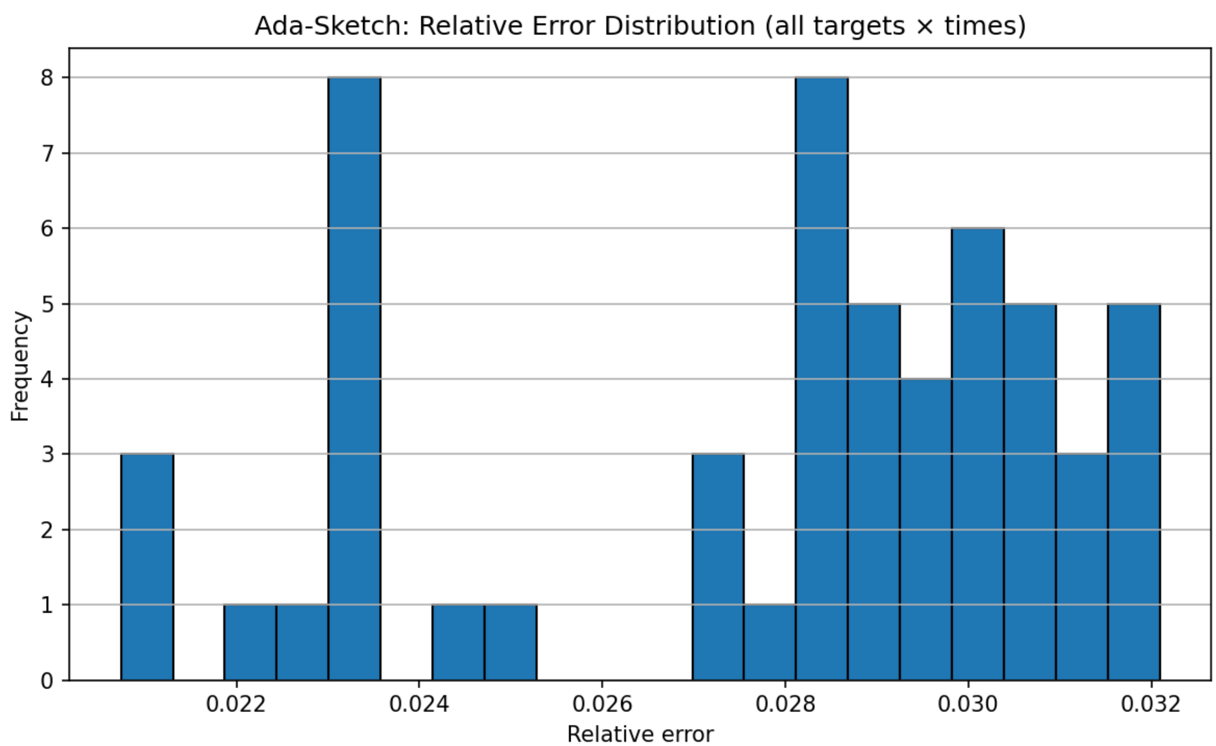

Figure 78: Histogram of relative error for Ada Sketch at width 16384 and depth four.

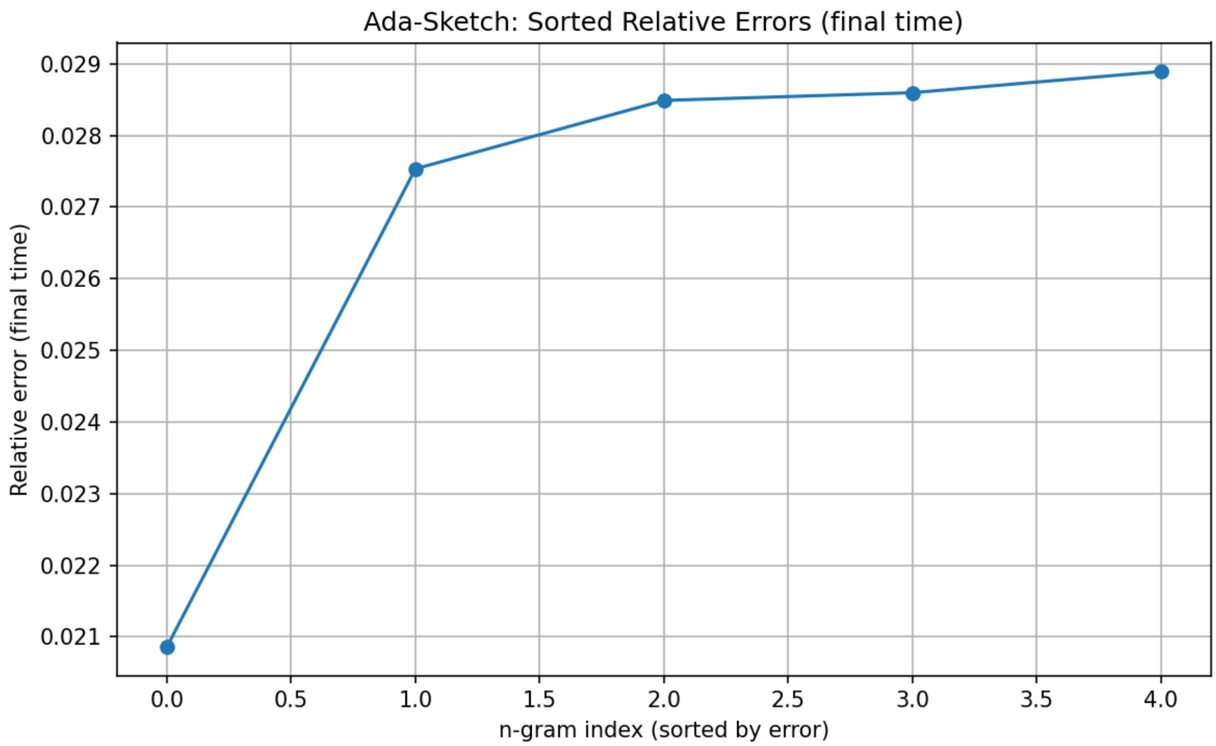

Figure 79: Sorted relative error for Ada Sketch at width 16384 and depth four.

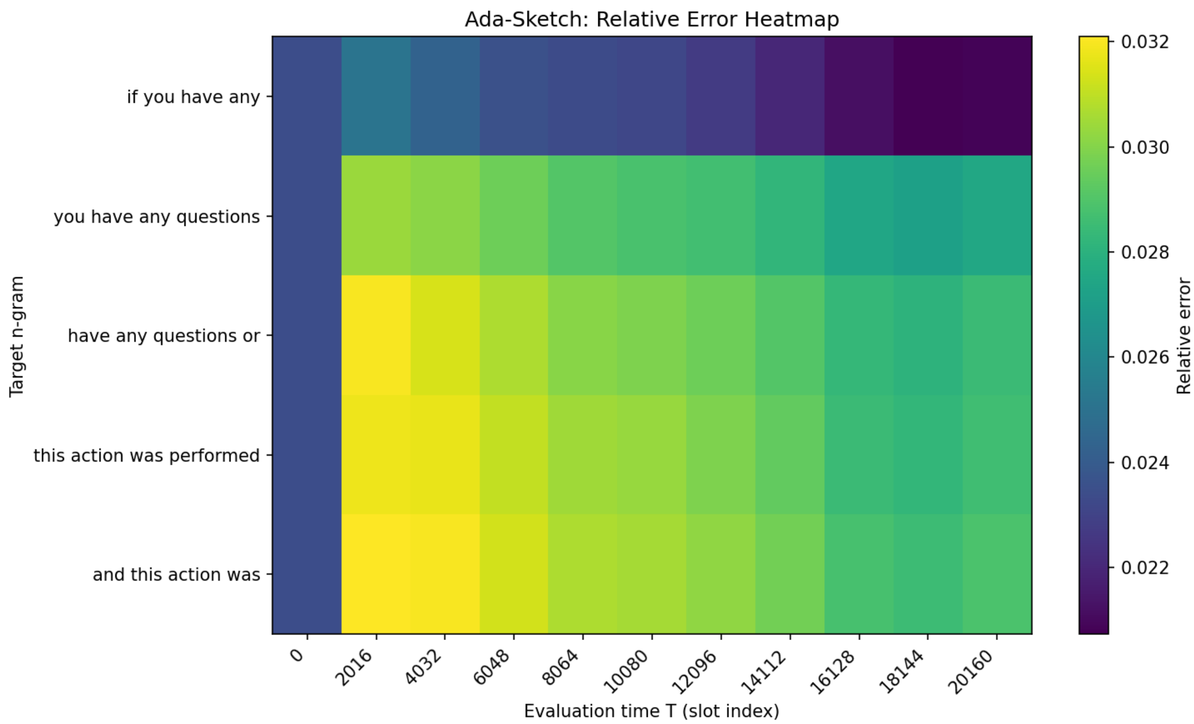

Figure 80: Heatmap of relative error for Ada Sketch at width 16384 and depth four.

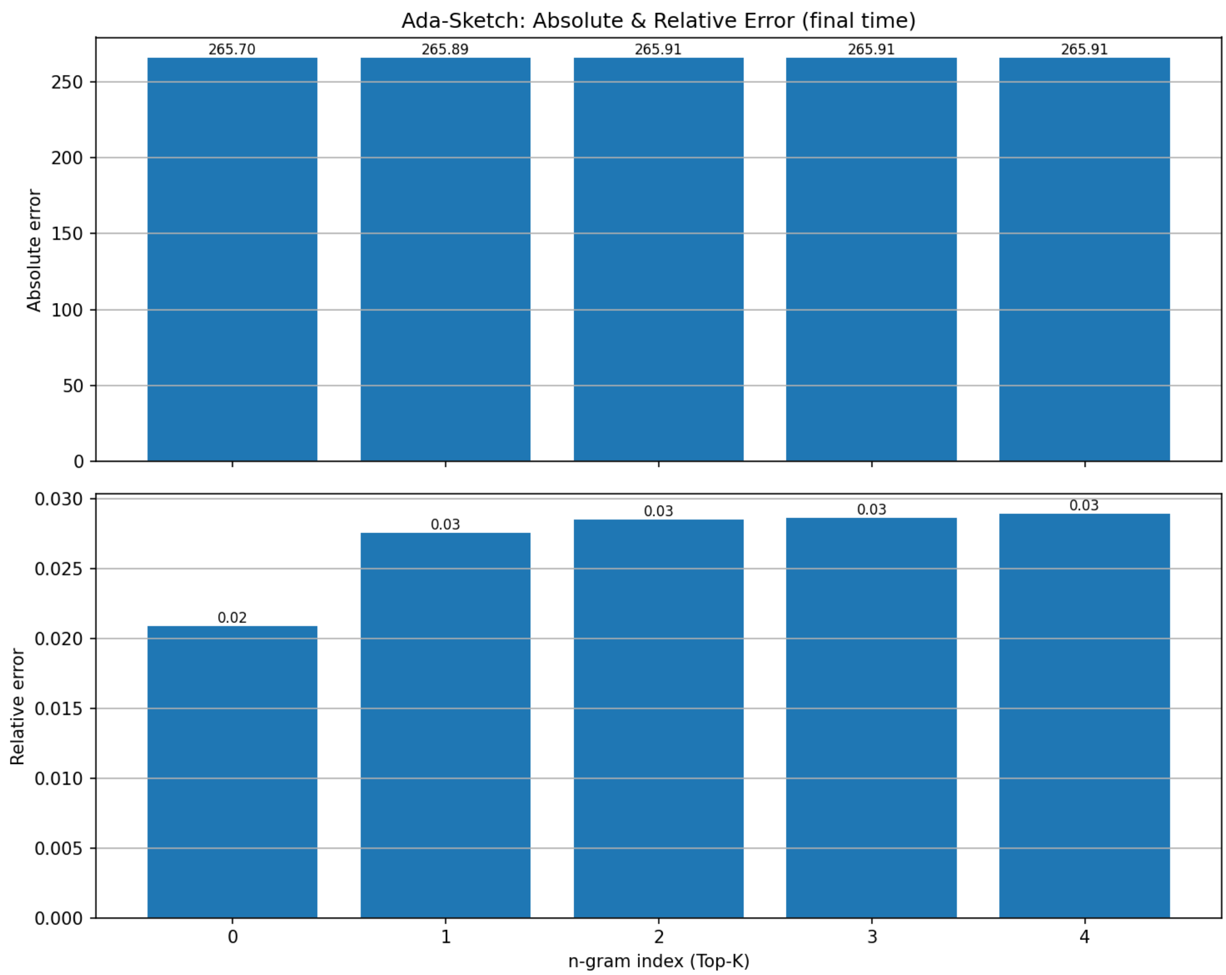

Figure 81: Absolute relative error for Ada Sketch at width 16384 and depth four.

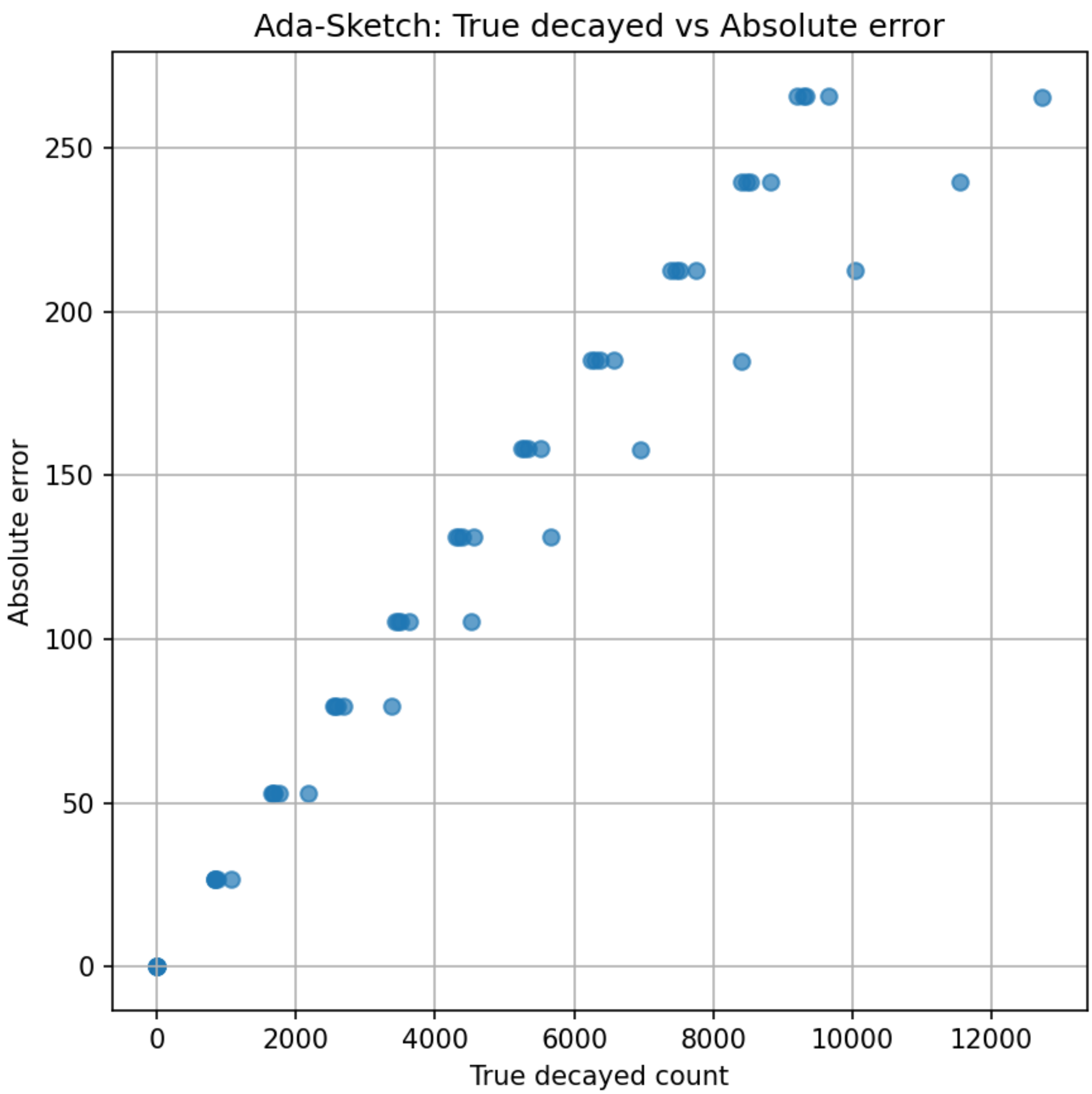

Figure 82: True counts versus absolute error for Ada Sketch at width 16384 and depth four.

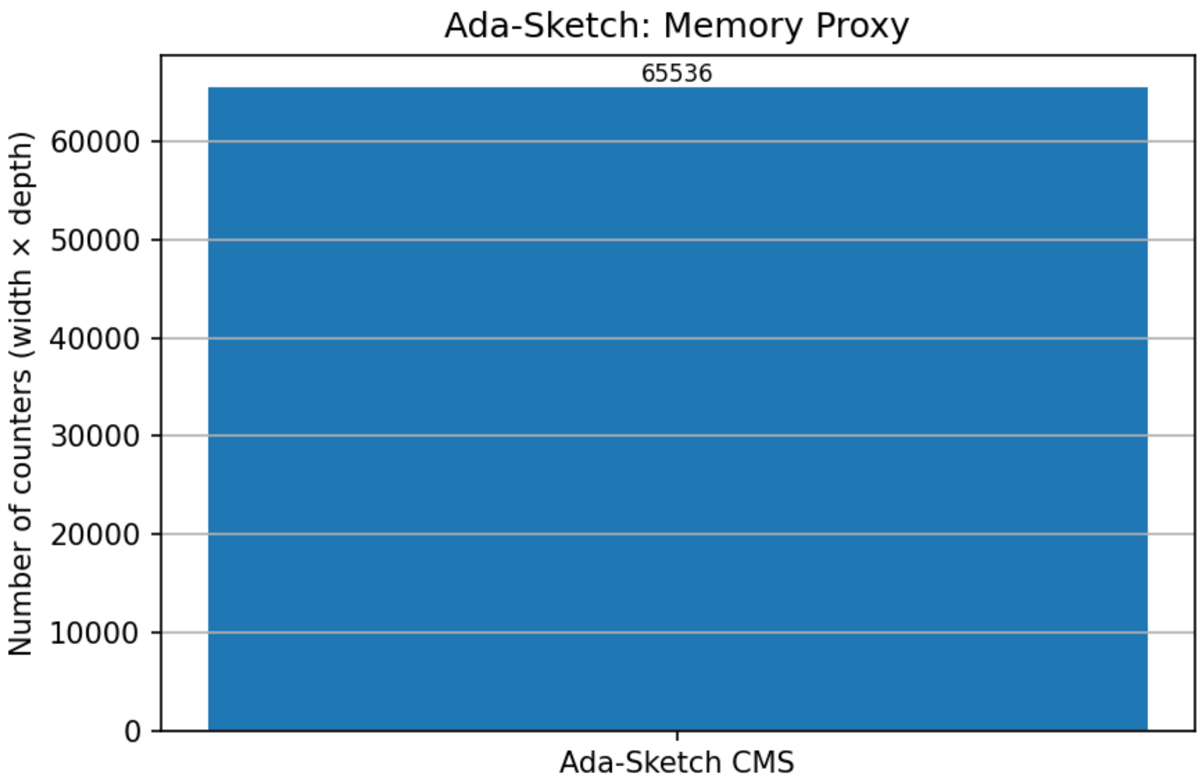

Figure 83: Memory proxy for Ada Sketch at width 16384 and depth four.

A.4.3 Width 32768 depth four

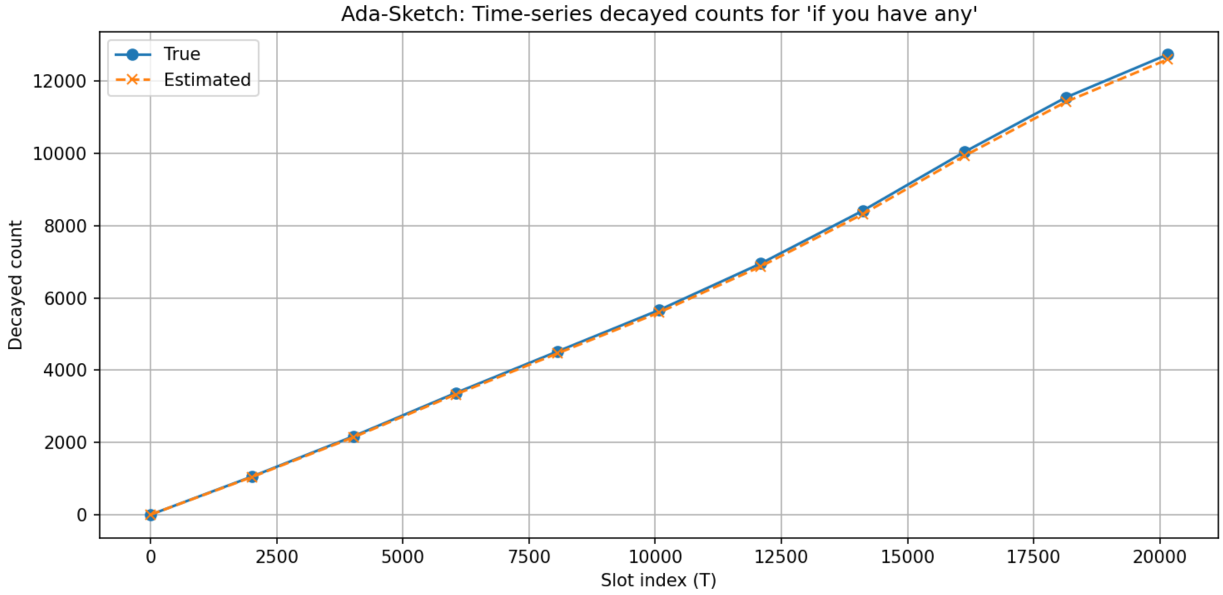

Figure 84: Ada Sketch time series estimates for width 32768 and depth four.

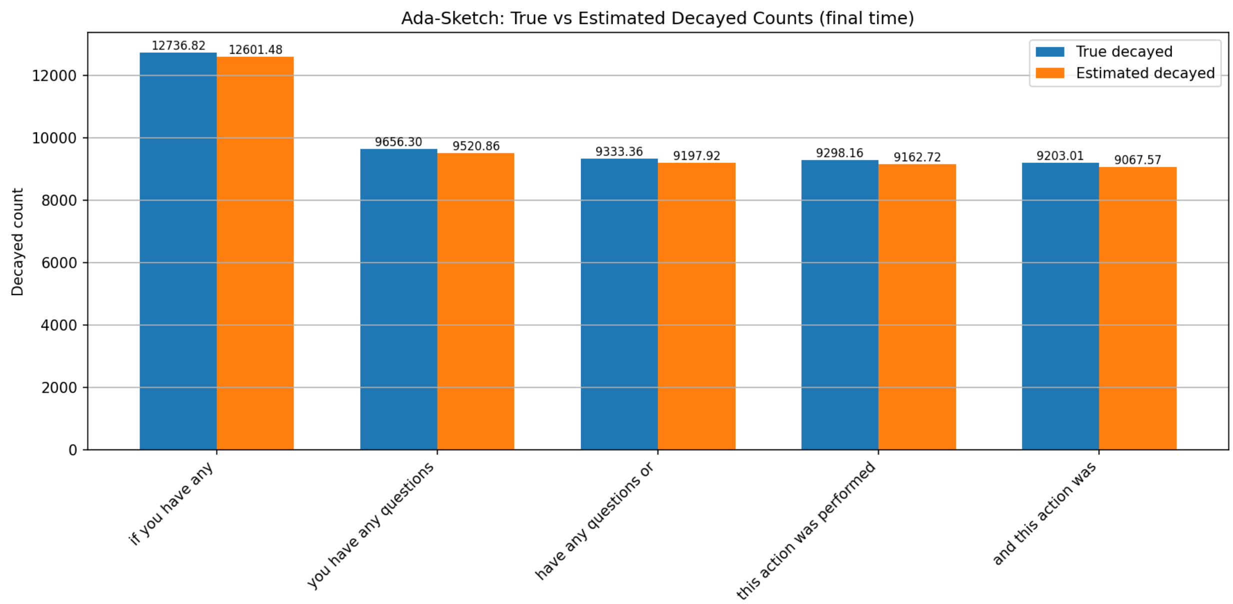

Figure 85: Ada Sketch counts versus ground truth for width 32768 and depth four.

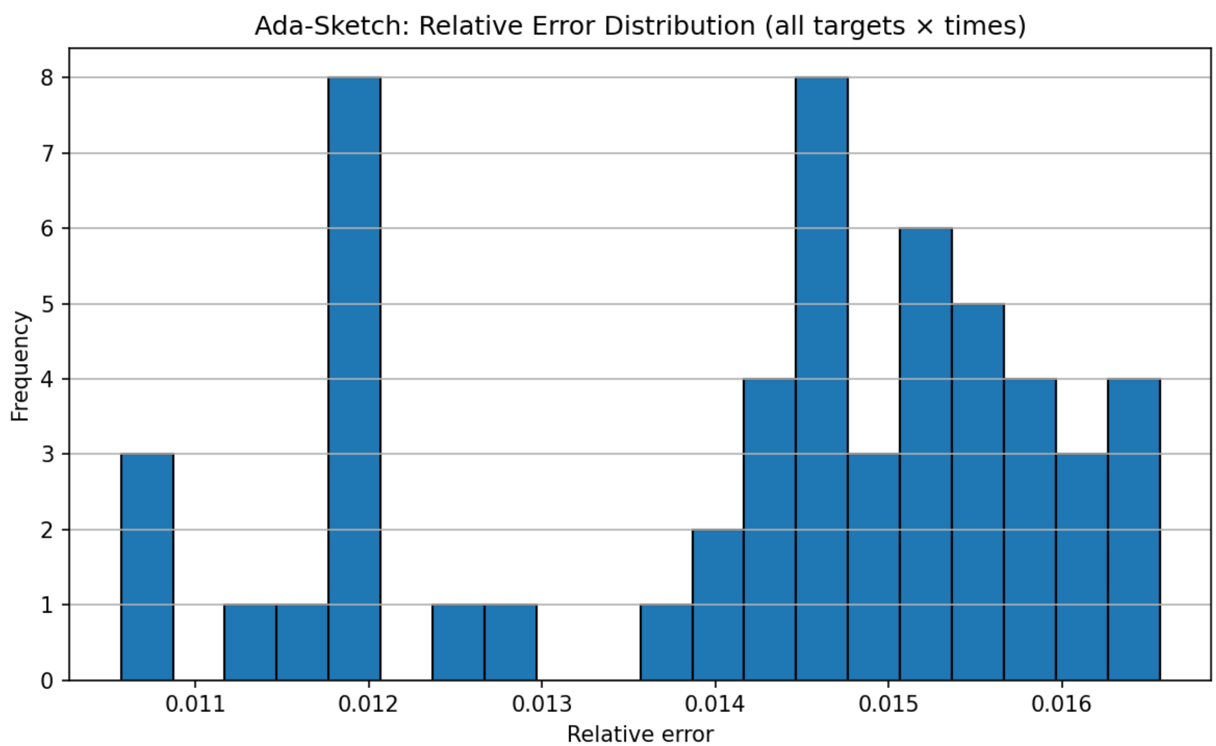

Figure 86: Histogram of relative error for Ada Sketch at width 32768 and depth four.

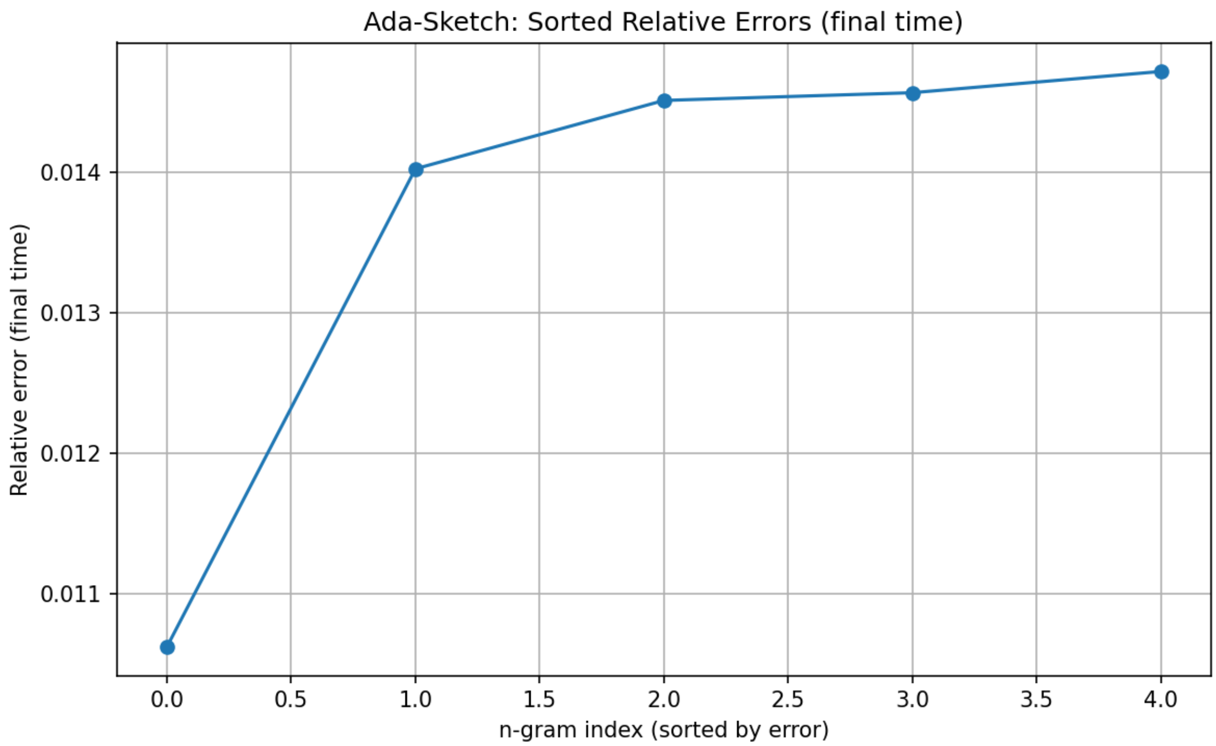

Figure 87: Sorted relative error for Ada Sketch at width 32768 and depth four.

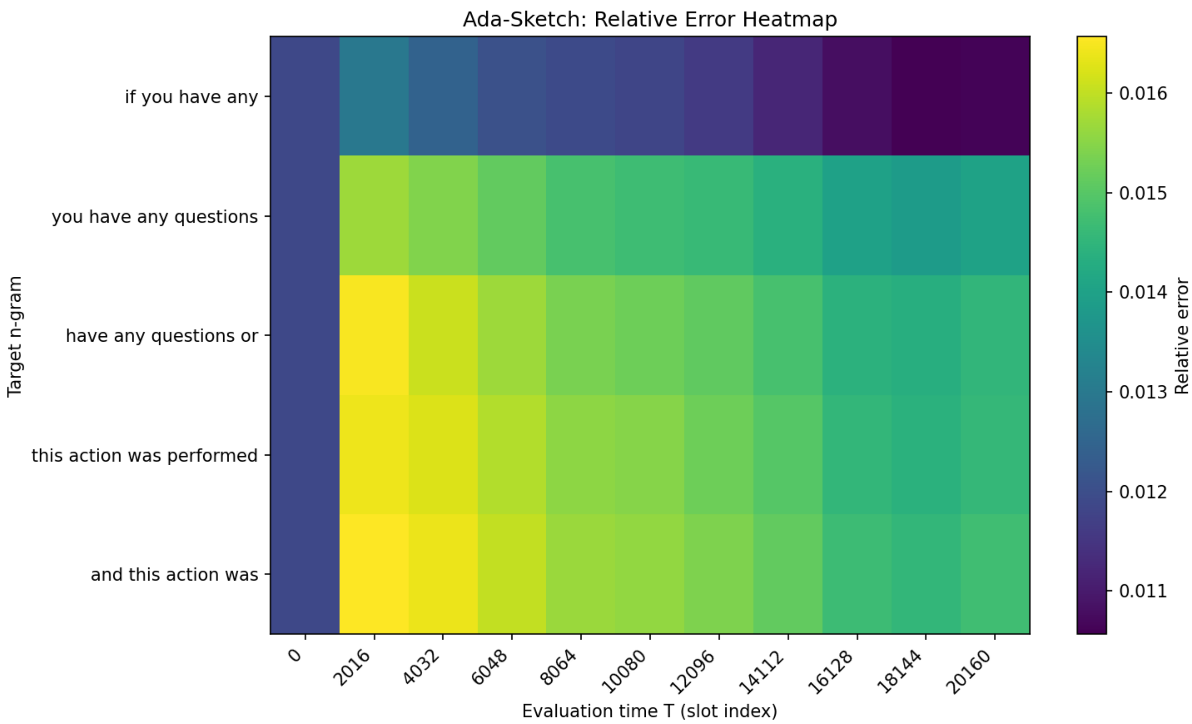

Figure 88: Heatmap of relative error for Ada Sketch at width 32768 and depth four.

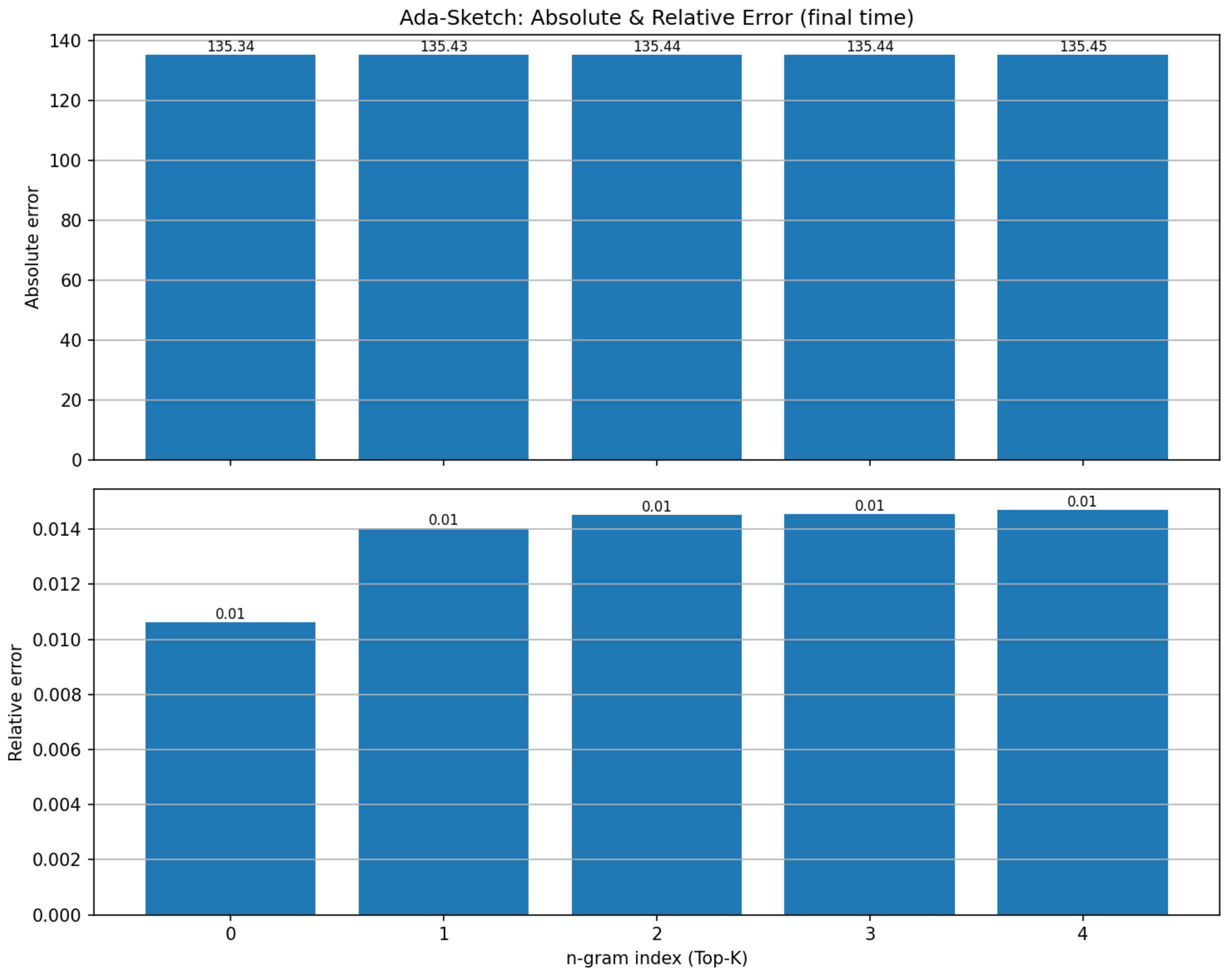

Figure 89: Absolute relative error for Ada Sketch at width 32768 and depth four.

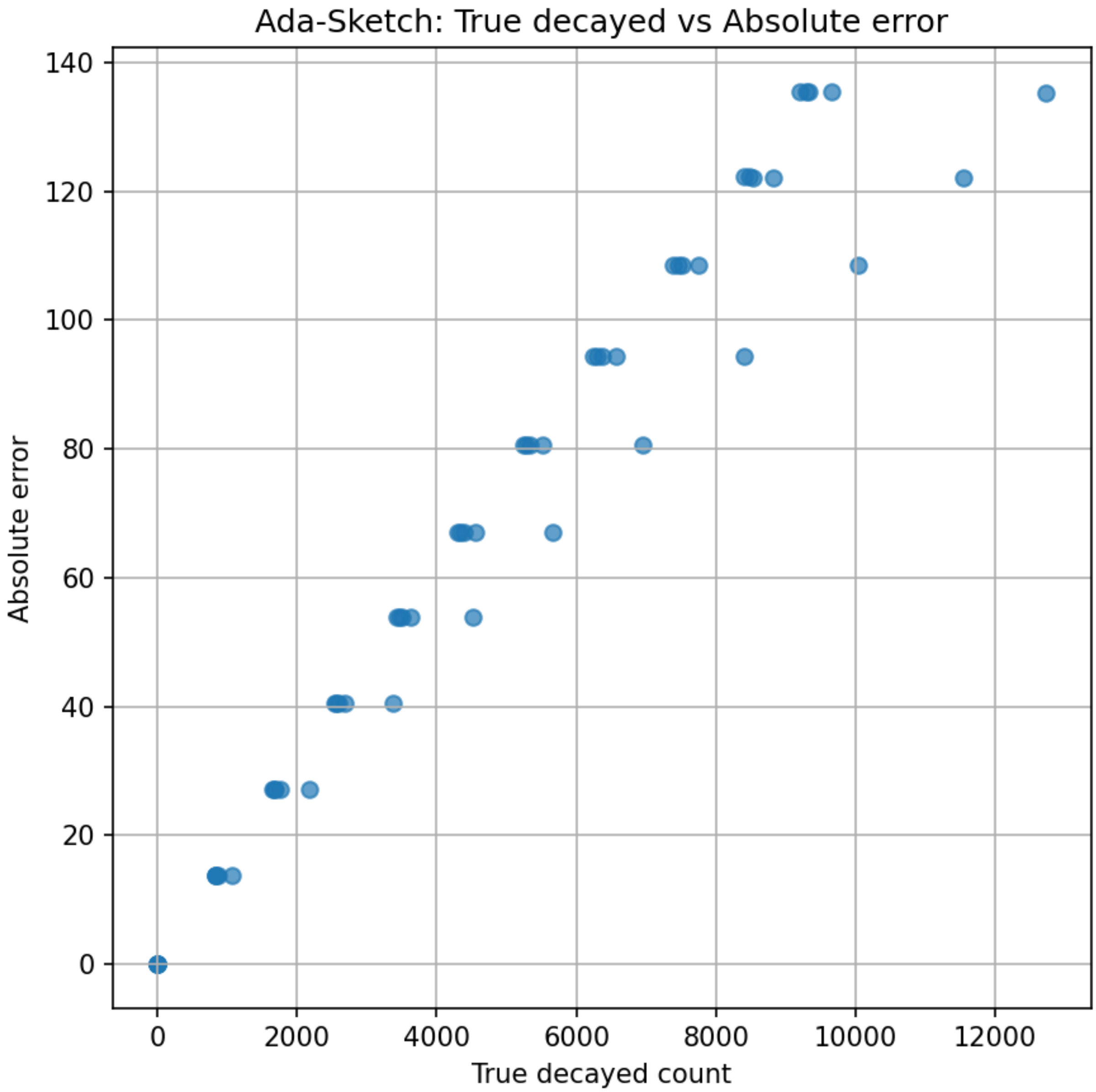

Figure 90: True counts versus absolute error for Ada Sketch at width 32768 and depth four.

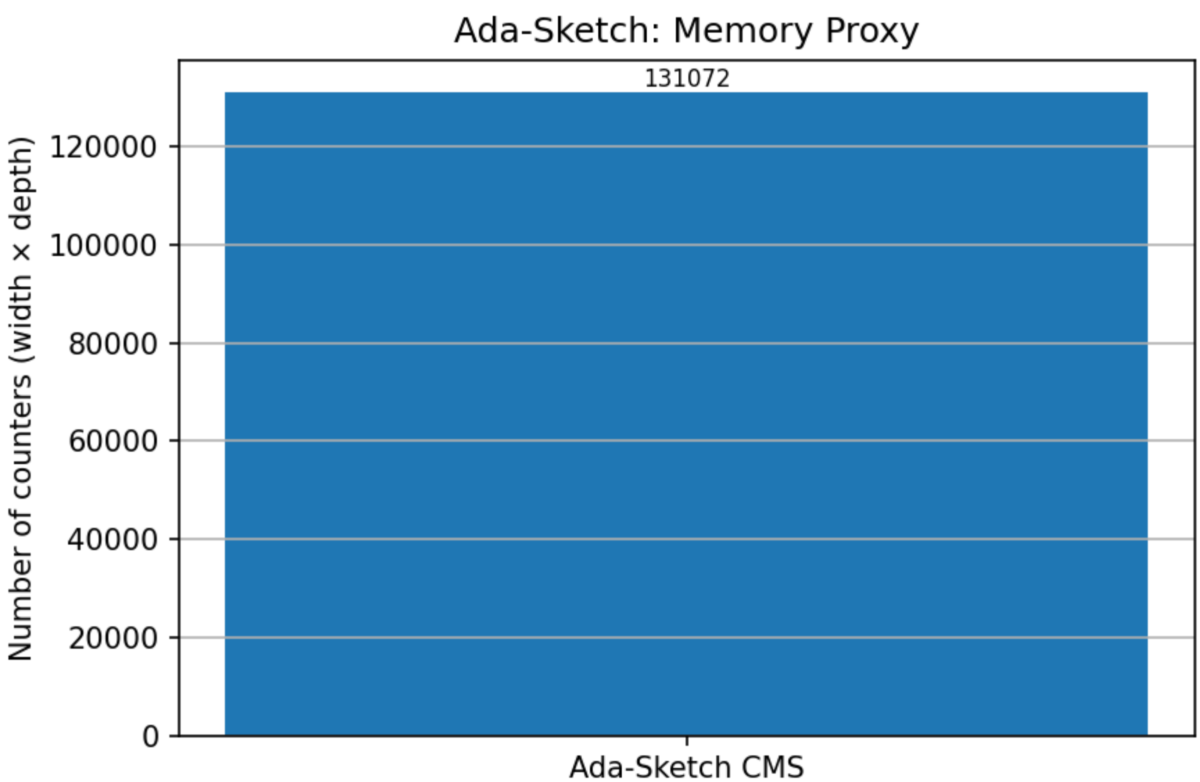

Figure 91: Memory proxy for Ada Sketch at width 32768 and depth four.

A.5 Burst Sketch results

Per-width BurstSketch plots: burst time series, burst bars, counts versus ground truth, memory proxies, error distributions, and burst score scatter views.

A.5.1 Width 8192 depth four

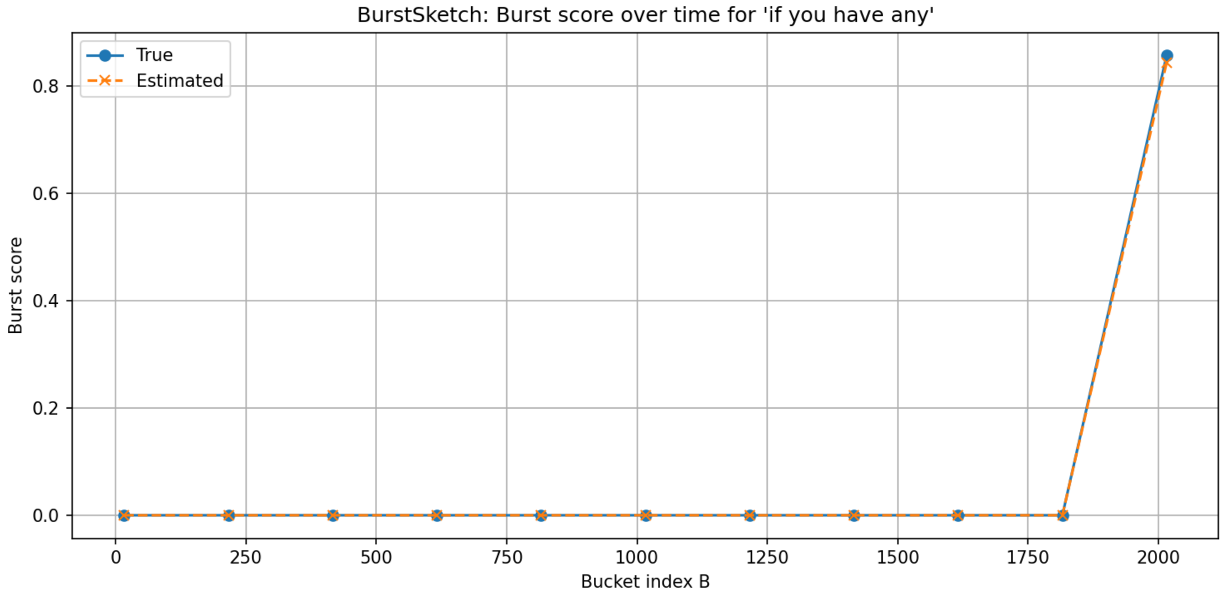

Figure 92: Burst Sketch burst score time series for width 8192 and depth four.

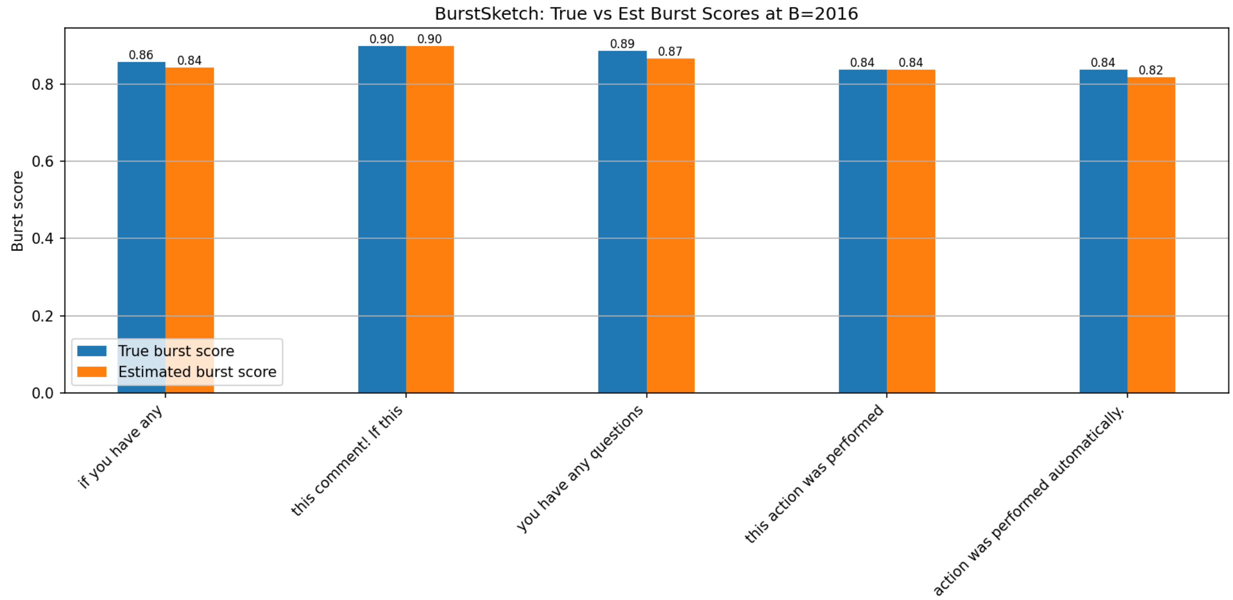

Figure 93: Burst Sketch burst score bars for width 8192 and depth four.

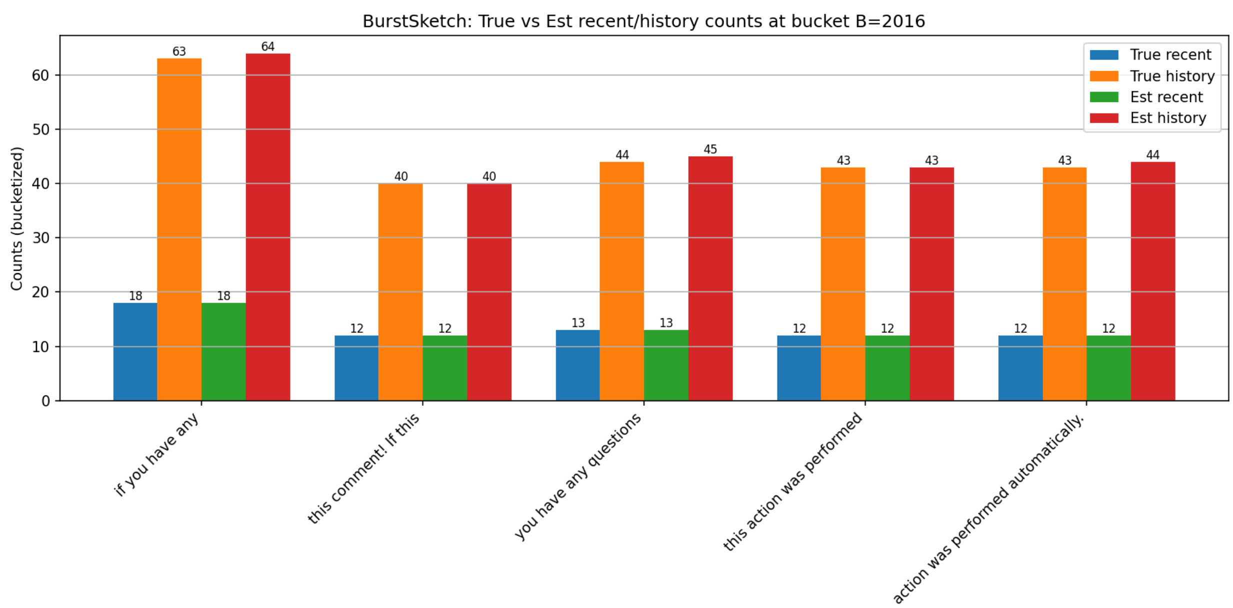

Figure 94: Burst Sketch counts versus ground truth for width 8192 and depth four.

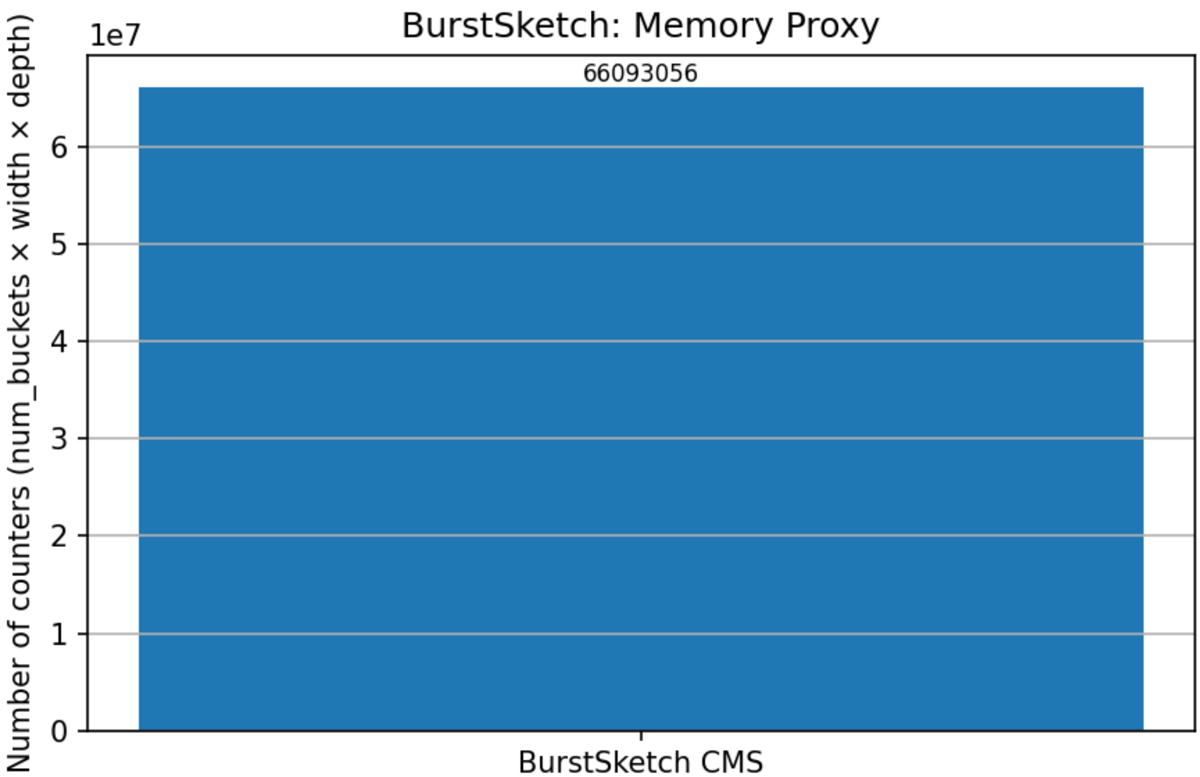

Figure 95: Memory proxy for Burst Sketch at width 8192 and depth four.

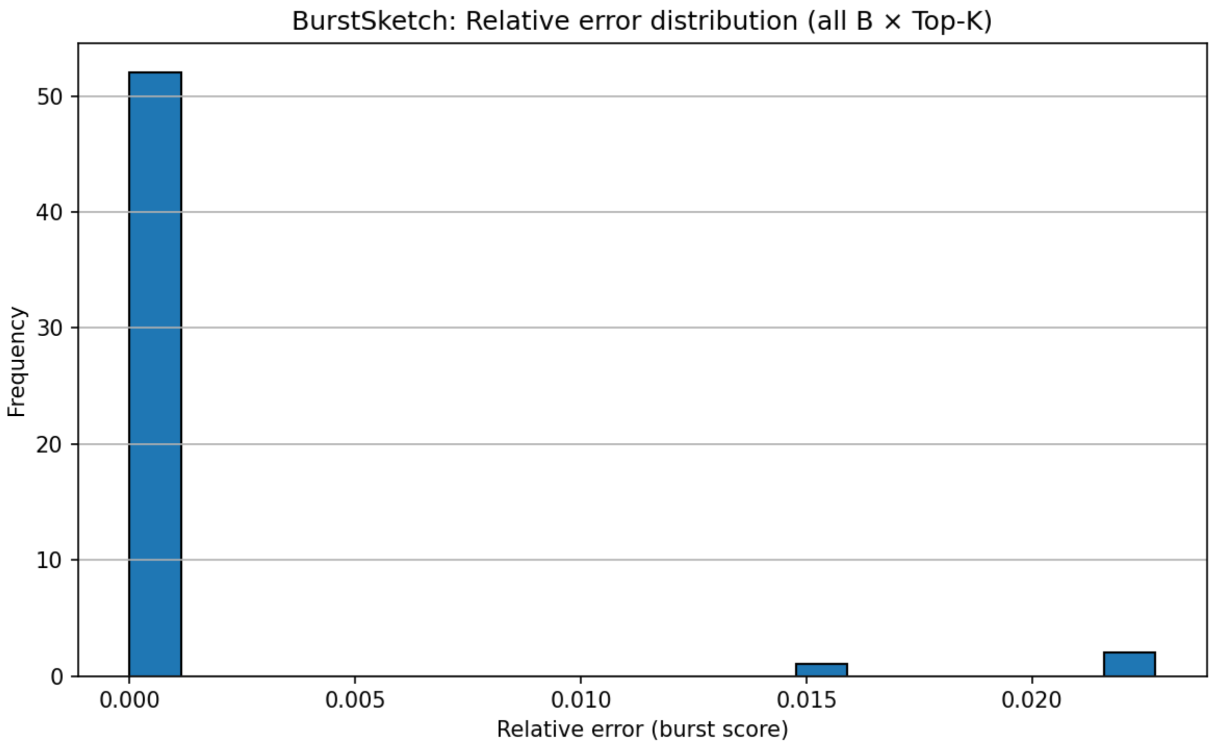

Figure 96: Histogram of relative error for Burst Sketch at width 8192 and depth four.

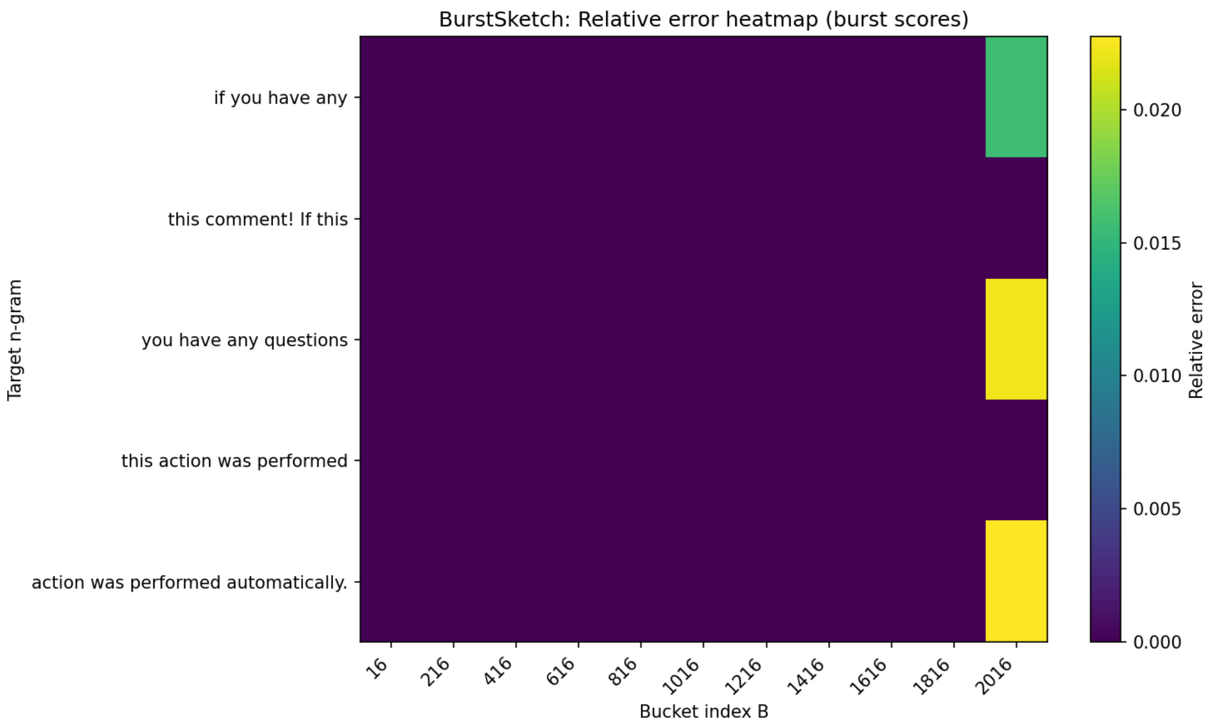

Figure 97: Heatmap of relative error for Burst Sketch at width 8192 and depth four.

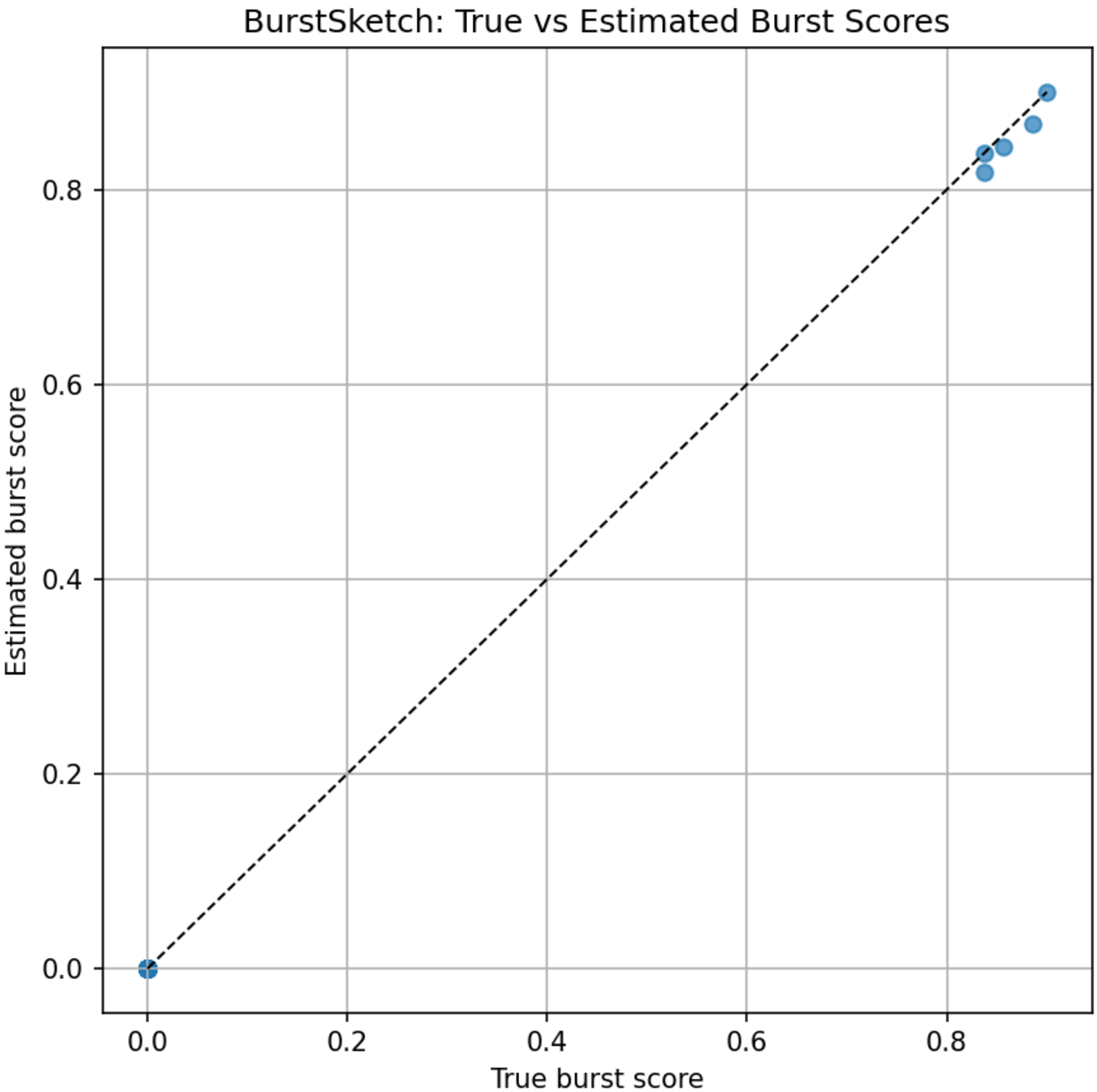

Figure 98: True burst scores versus estimated scores for Burst Sketch at width 8192 and depth four.

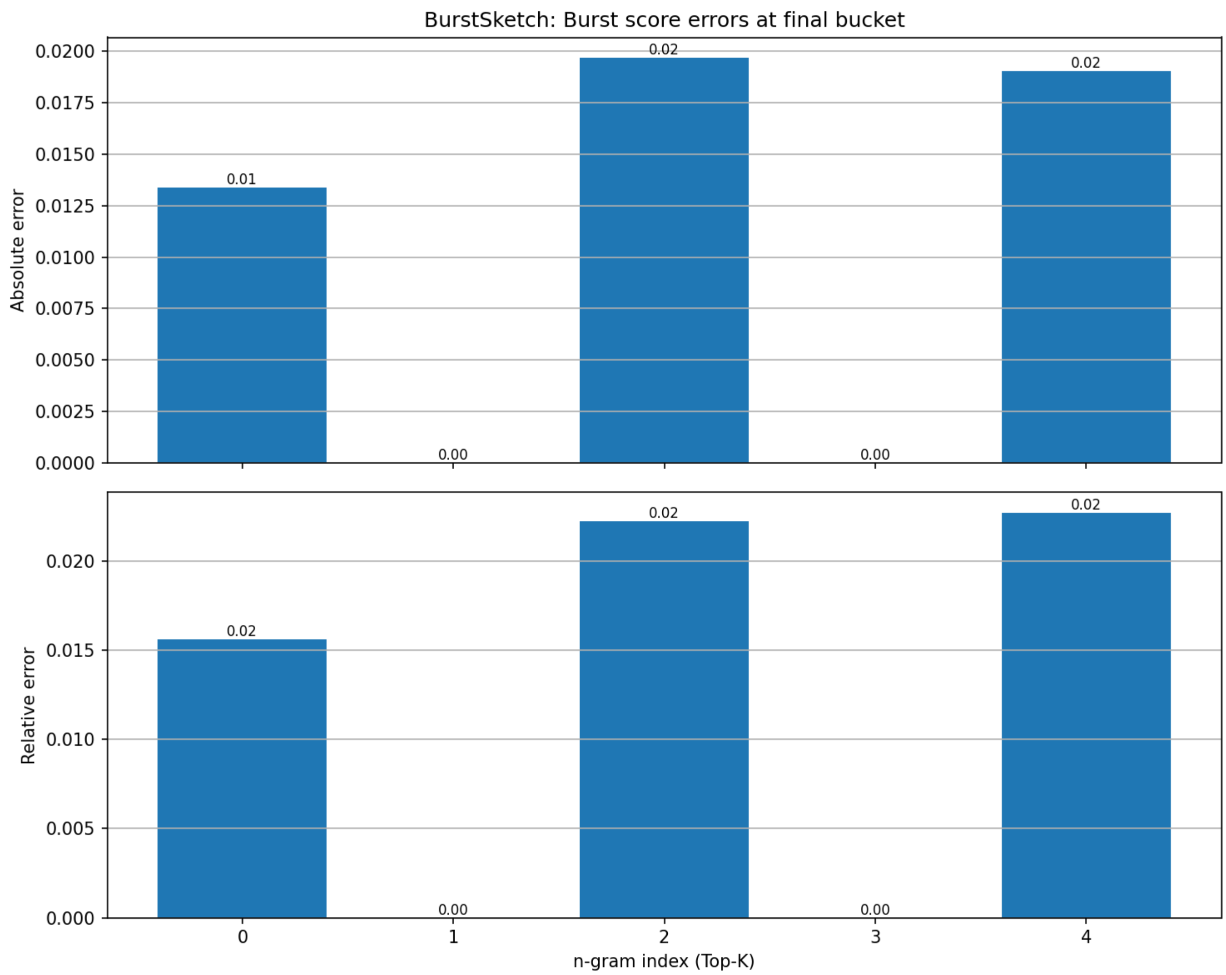

Figure 99: Absolute relative error for Burst Sketch at width 8192 and depth four.

A.5.2 Width 16384 depth four

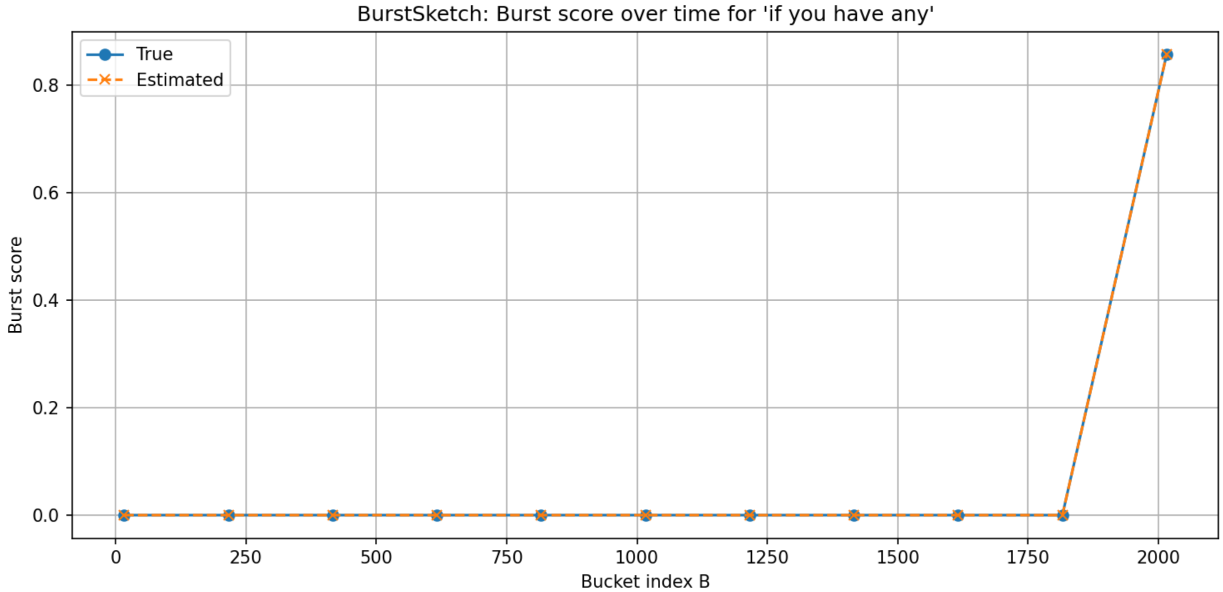

Figure 100: Burst Sketch burst score time series for width 16384 and depth four.

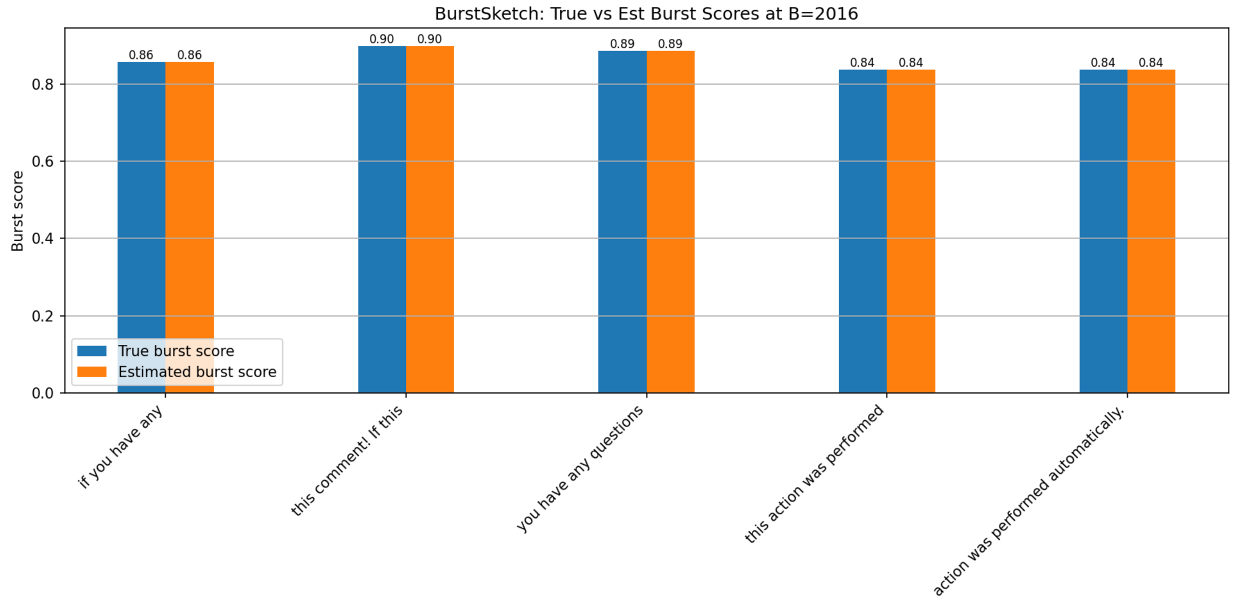

Figure 101: Burst Sketch burst score bars for width 16384 and depth four.

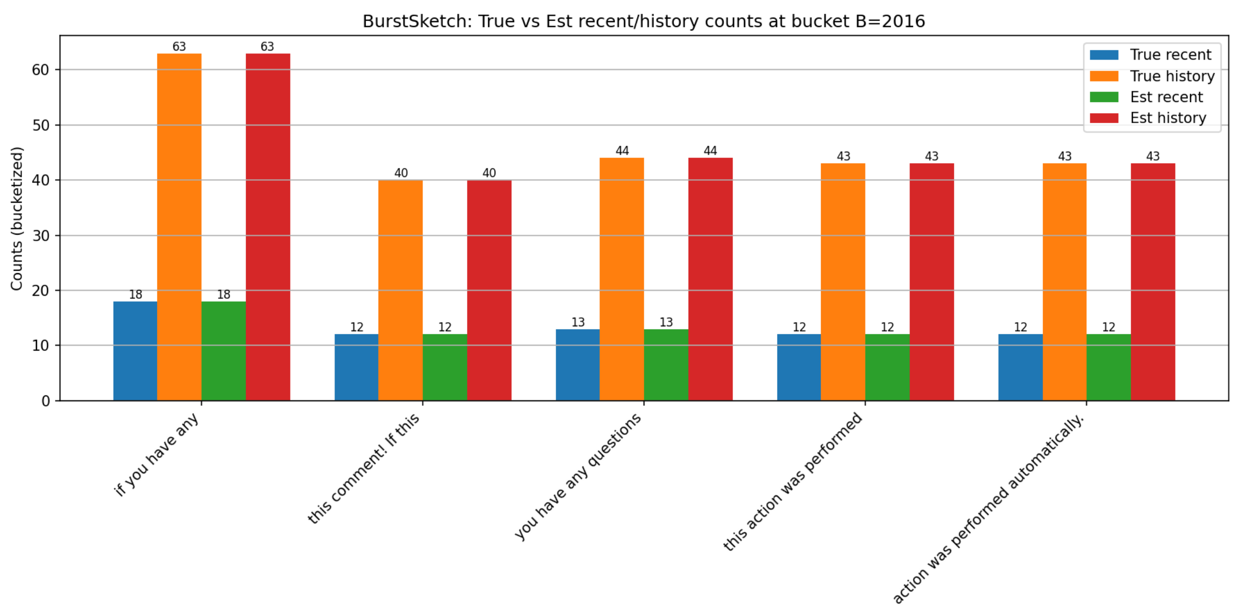

Figure 102: Burst Sketch counts versus ground truth for width 16384 and depth four.

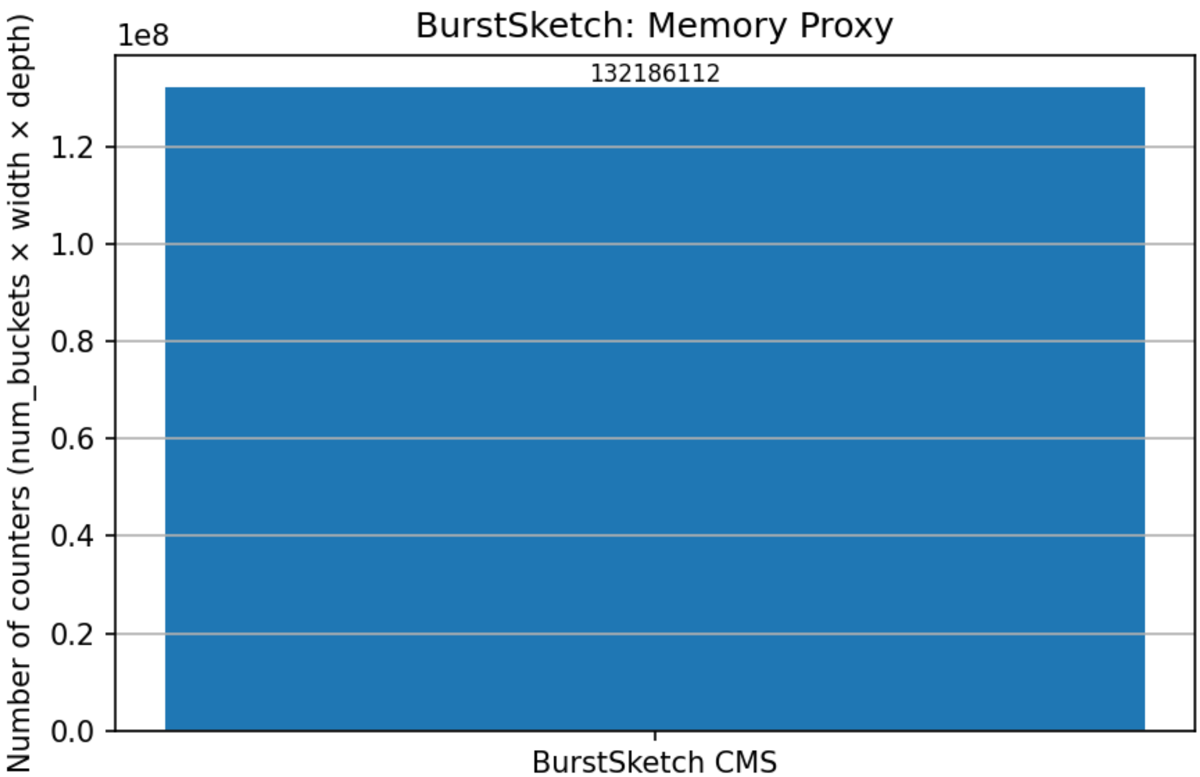

Figure 103: Memory proxy for Burst Sketch at width 16384 and depth four.

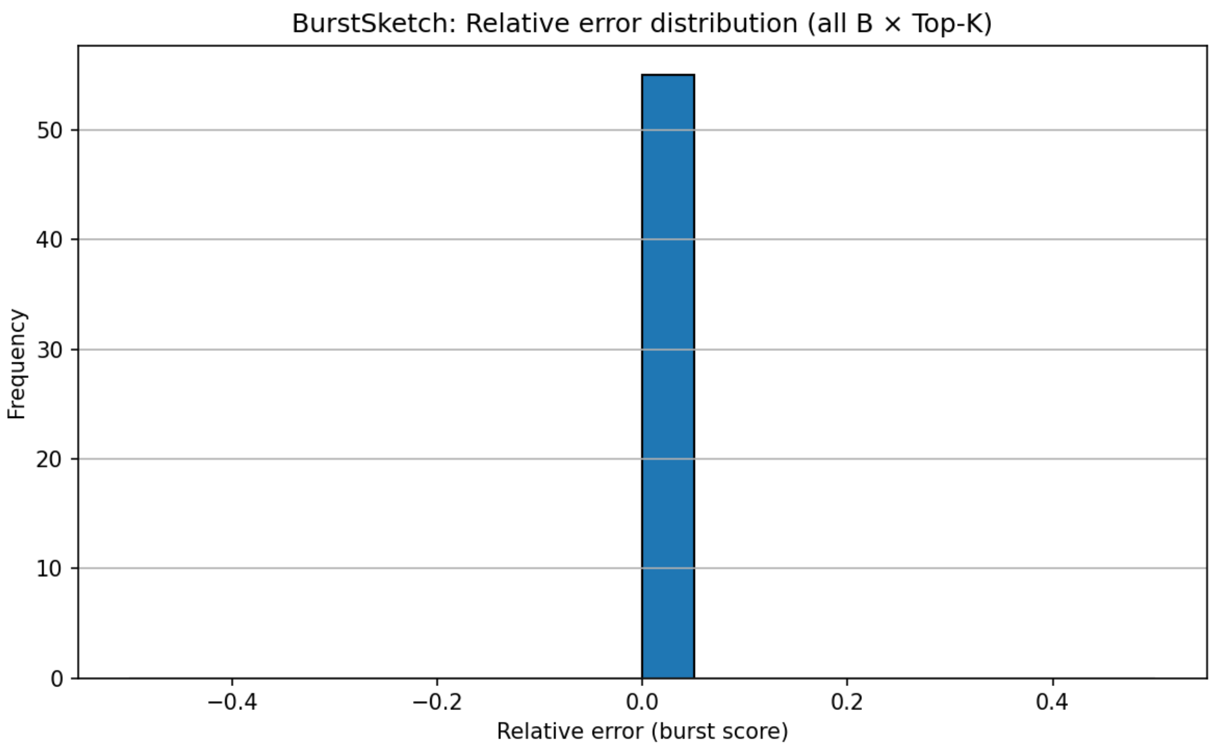

Figure 104: Histogram of relative error for Burst Sketch at width 16384 and depth four.

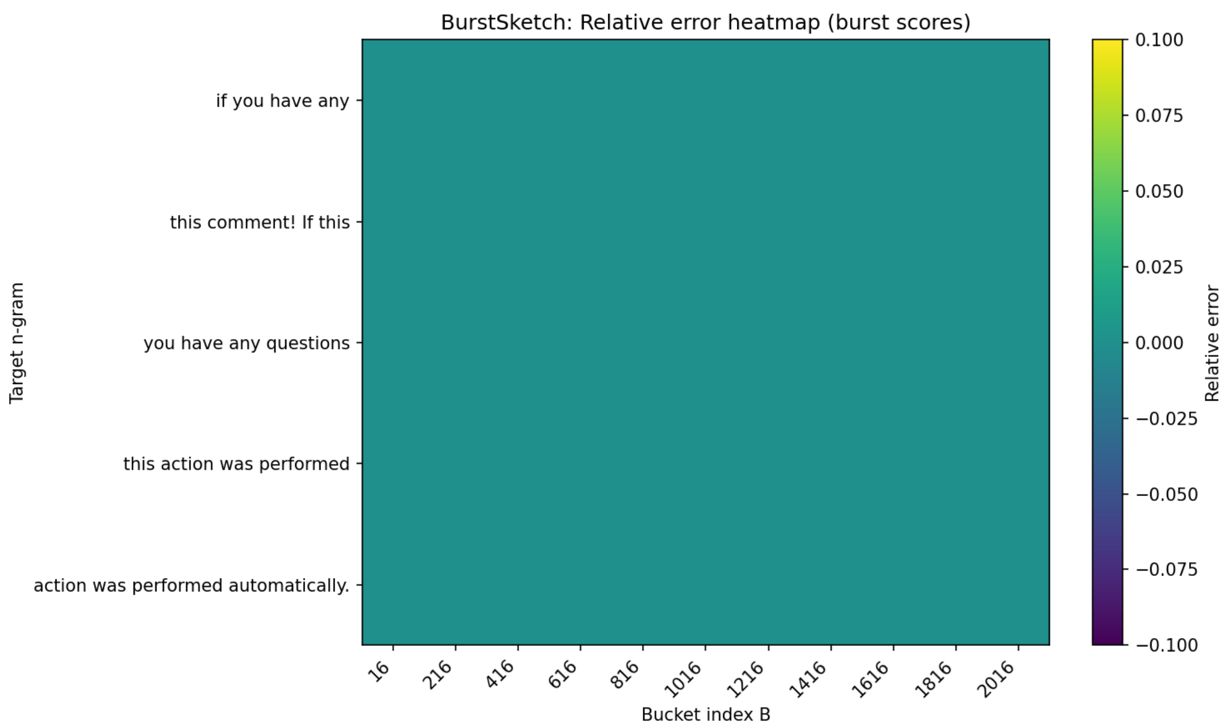

Figure 105: Heatmap of relative error for Burst Sketch at width 16384 and depth four.

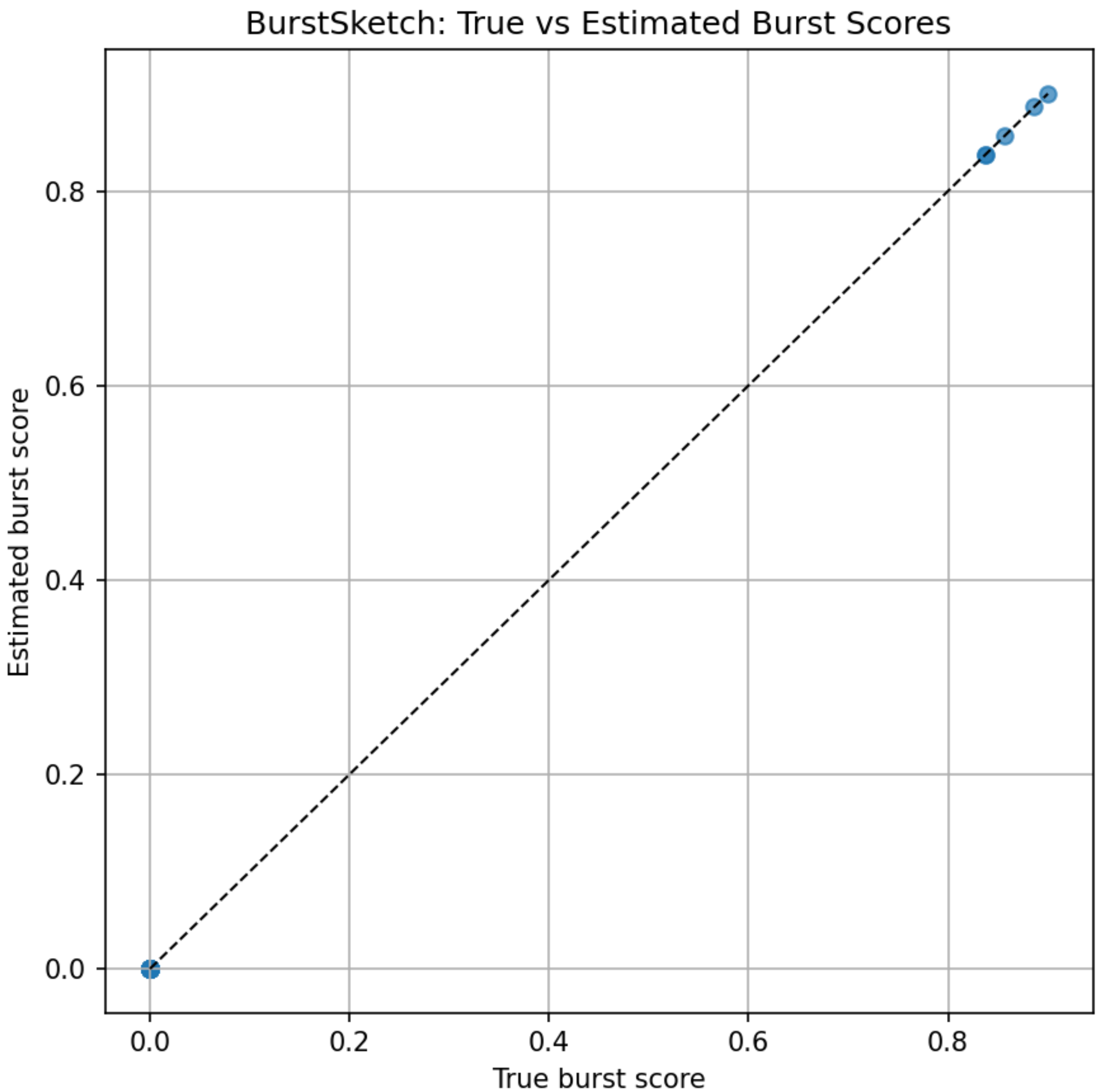

Figure 106: True burst scores versus estimated scores for Burst Sketch at width 16384 and depth four.

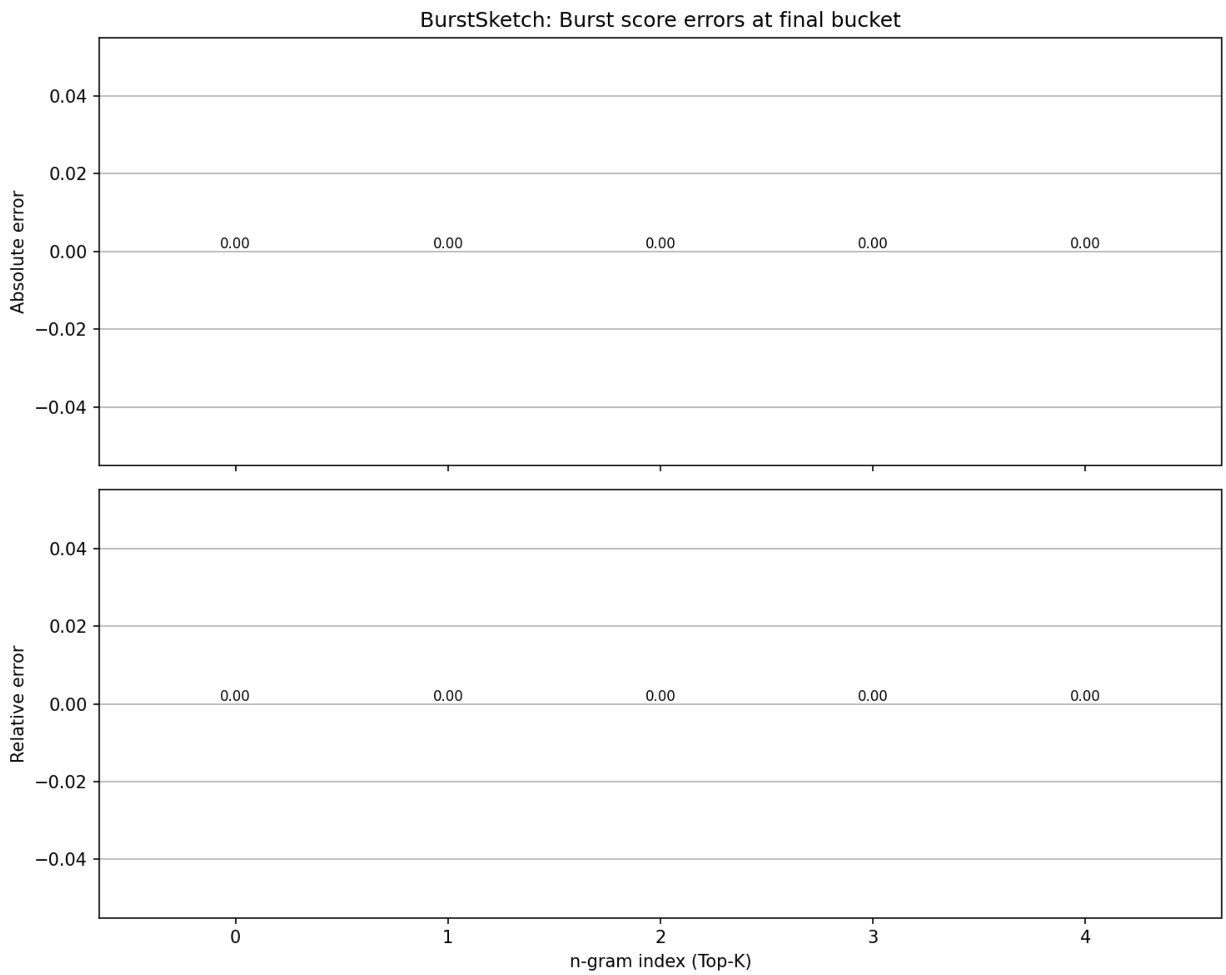

Figure 107: Absolute relative error for Burst Sketch at width 16384 and depth four.

A.5.3 Width 32768 depth four

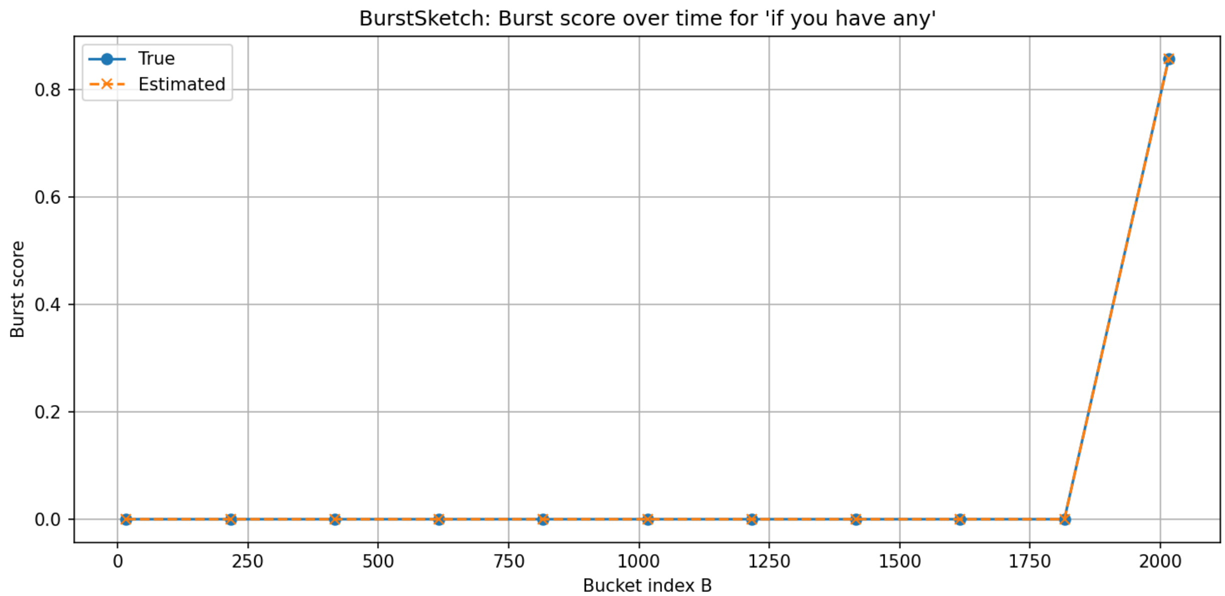

Figure 108: Burst Sketch burst score time series for width 32768 and depth four.

References

[1] Matusevych, S., Smola, A. J., and Ahmed, A. (2012). *Hokusai: Sketching Streams in Real Time*. Available at: https://www.auai.org/uai2012/papers/231.pdf

[2] Shrivastava, A., Konig, A. C., and Bilenko, M. (2016). *Time Adaptive Sketches (Ada-Sketches) for Summarizing Data Streams*. Proceedings of SIGMOD 2016. DOI: https://doi.org/10.1145/2882903.2882946

[3] Zhong, Z., Yan, S., Li, Z., Tan, D., Yang, T., and Cui, B. (2021). *BurstSketch: Finding Bursts in Data Streams*. Available at: https://yangtonghome.github.io/uploads/BurstSketch.pdf

[4] Cormode, G., Shkapenyuk, V., Srivastava, D., and Xu, B. (2009). *Forward Decay: A Practical Time Decay Model for Streaming Systems*. Available at: http://dimacs.rutgers.edu/ graham/pubs/papers/fwddecay.pdf

[5] Metwally, A., Agrawal, D., and Abbadi, A. E. (2005). *Efficient Computation of Frequent and Top-k Elements in Data Streams*. Proceedings of ICDT 2005.

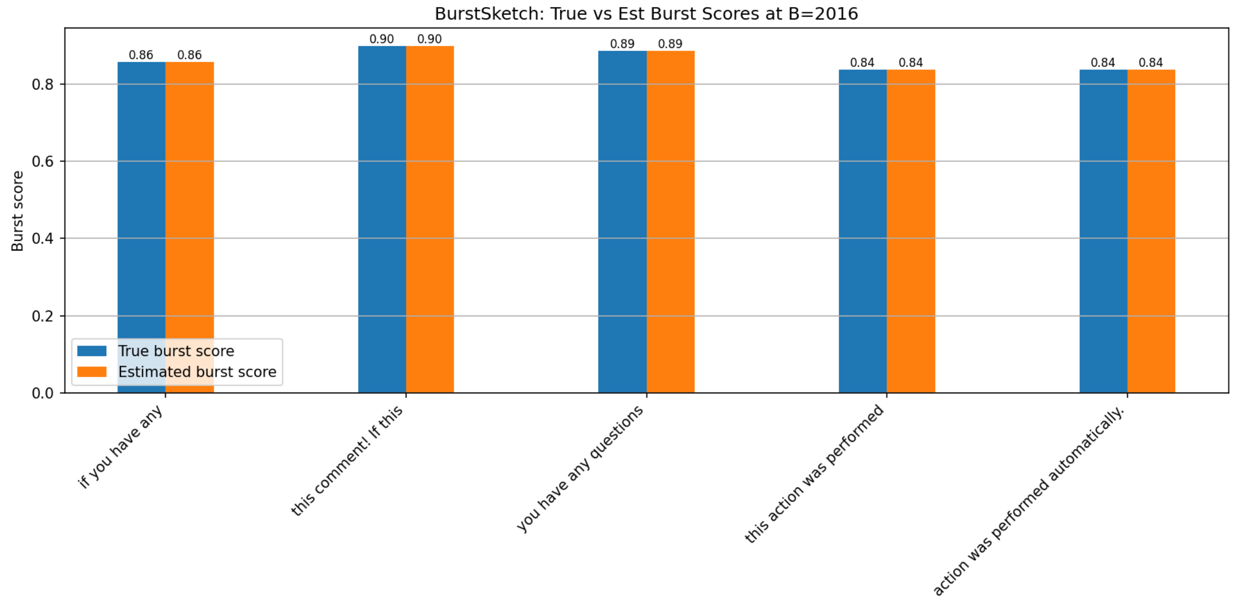

Figure 109: Burst Sketch burst score bars for width 32768 and depth four.

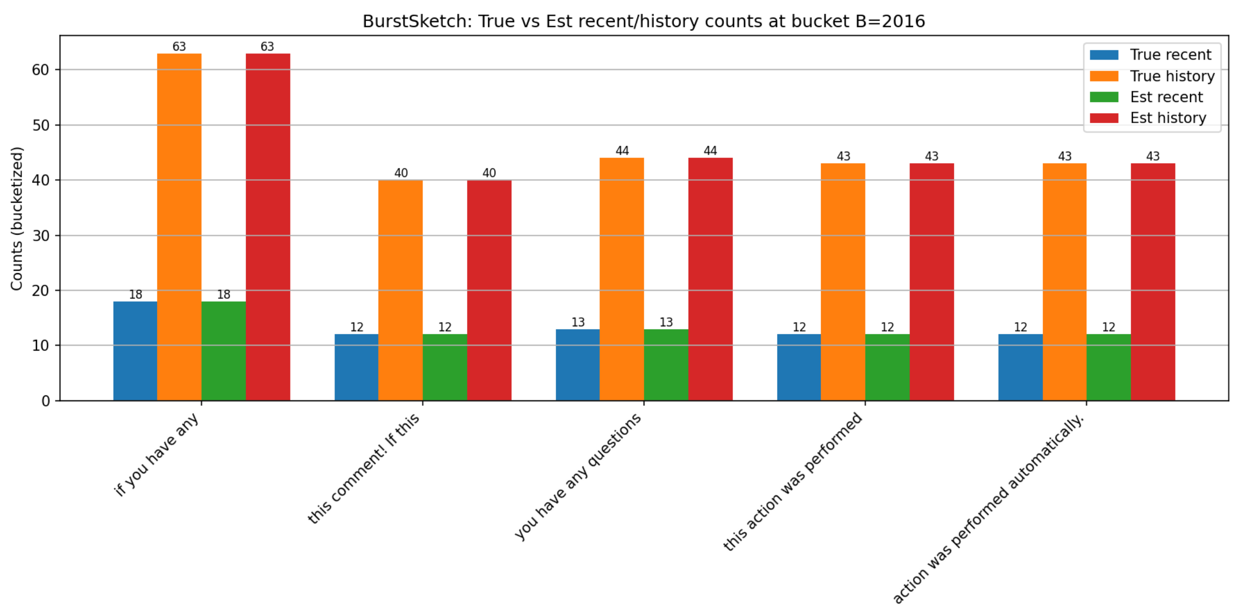

Figure 110: Burst Sketch counts versus ground truth for width 32768 and depth four.

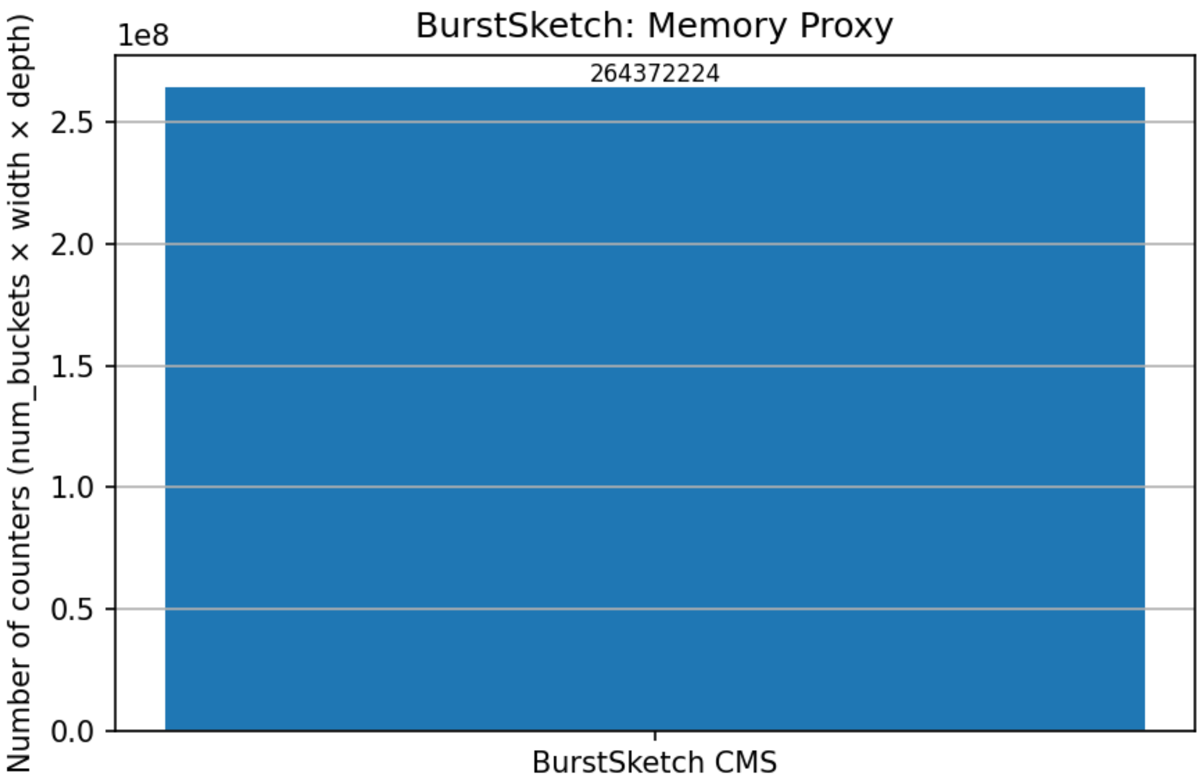

Figure 111: Memory proxy for Burst Sketch at width 32768 and depth four.

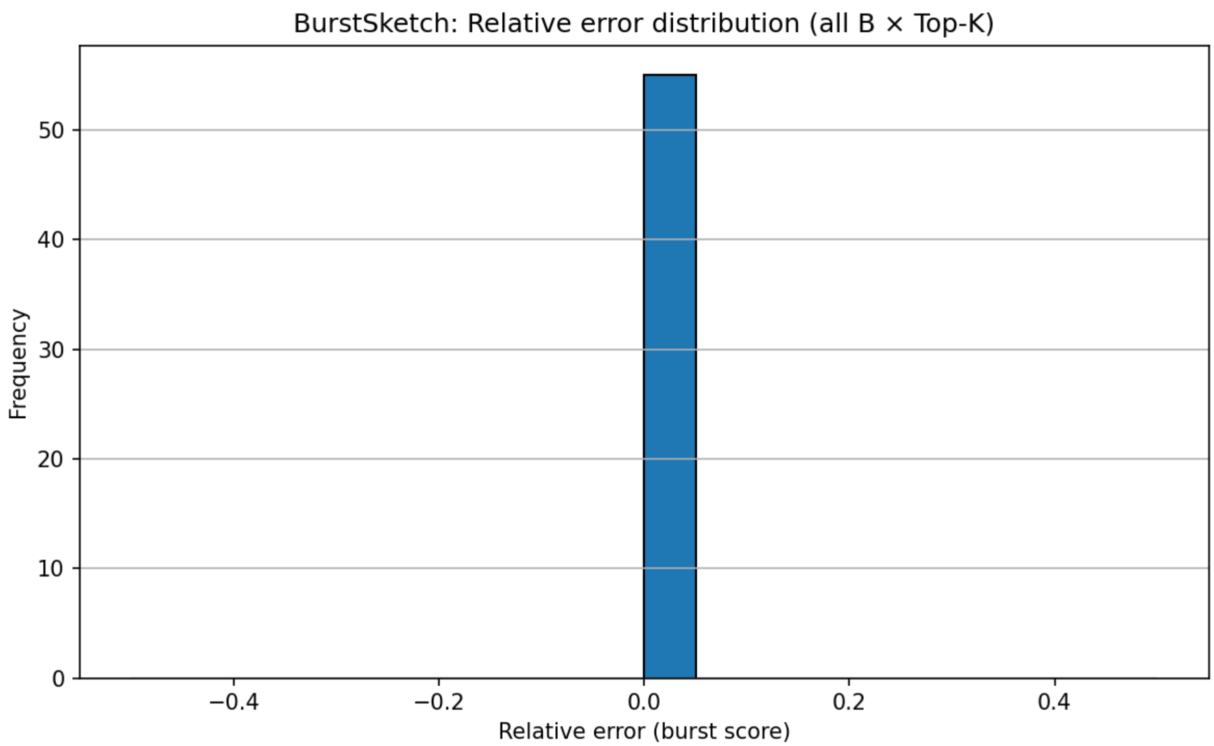

Figure 112: Histogram of relative error for Burst Sketch at width 32768 and depth four.

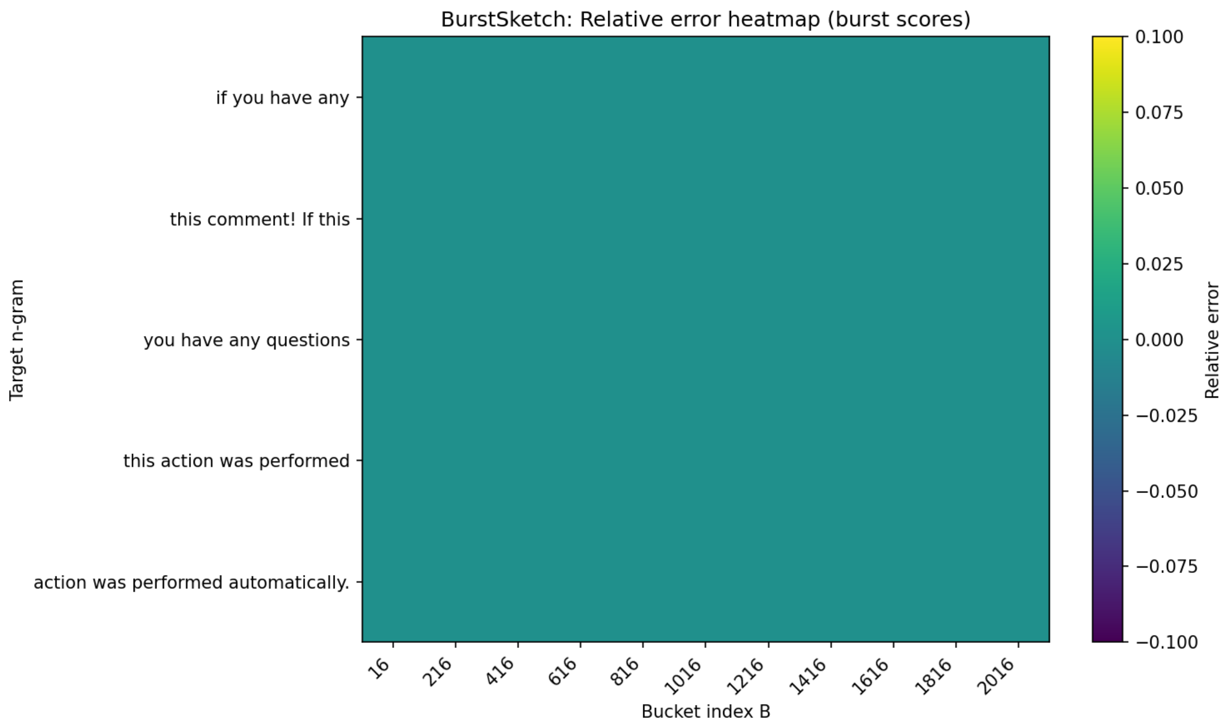

Figure 113: Heatmap of relative error for Burst Sketch at width 32768 and depth four.

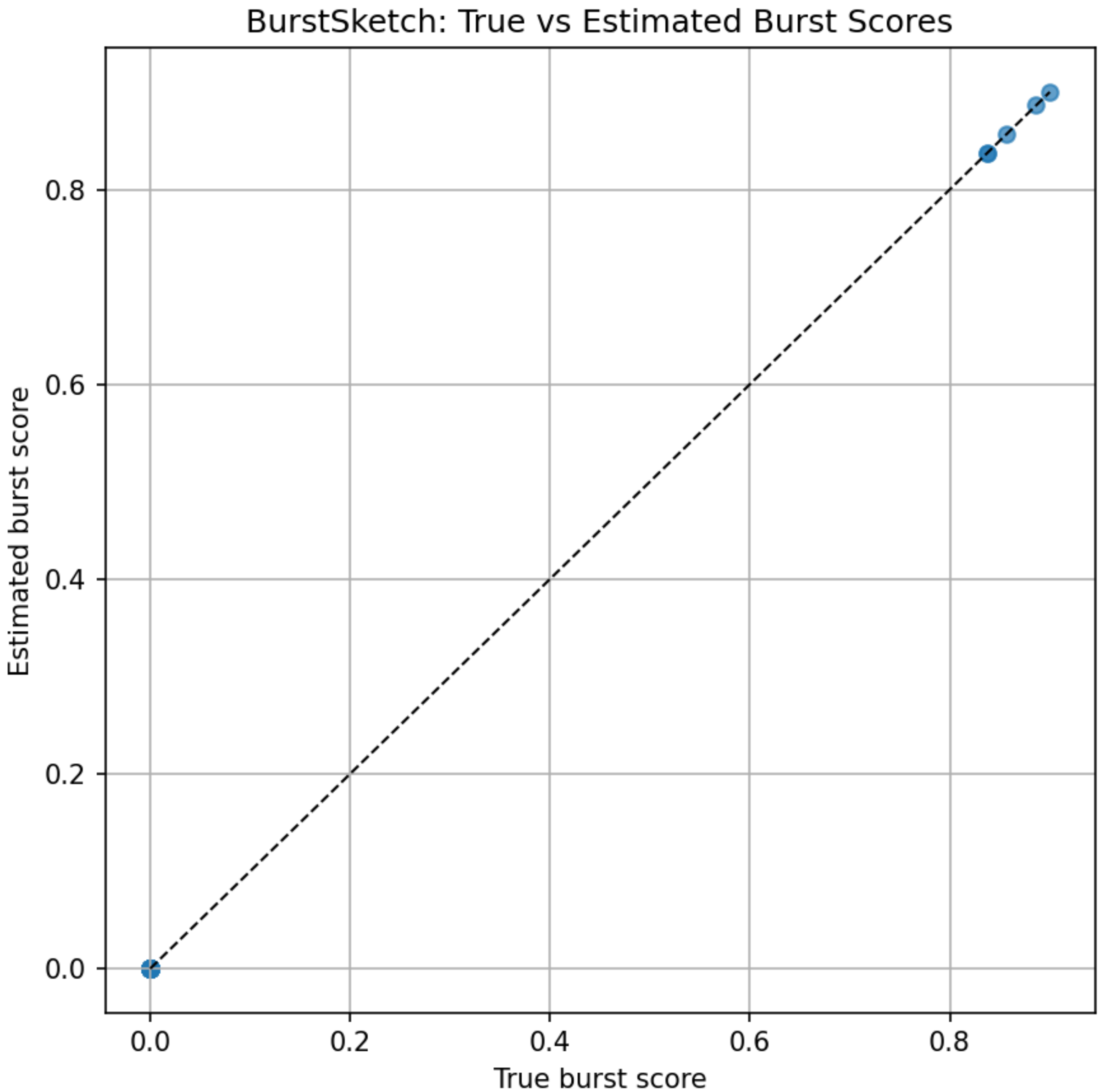

Figure 114: True burst scores versus estimated scores for Burst Sketch at width 32768 and depth four.

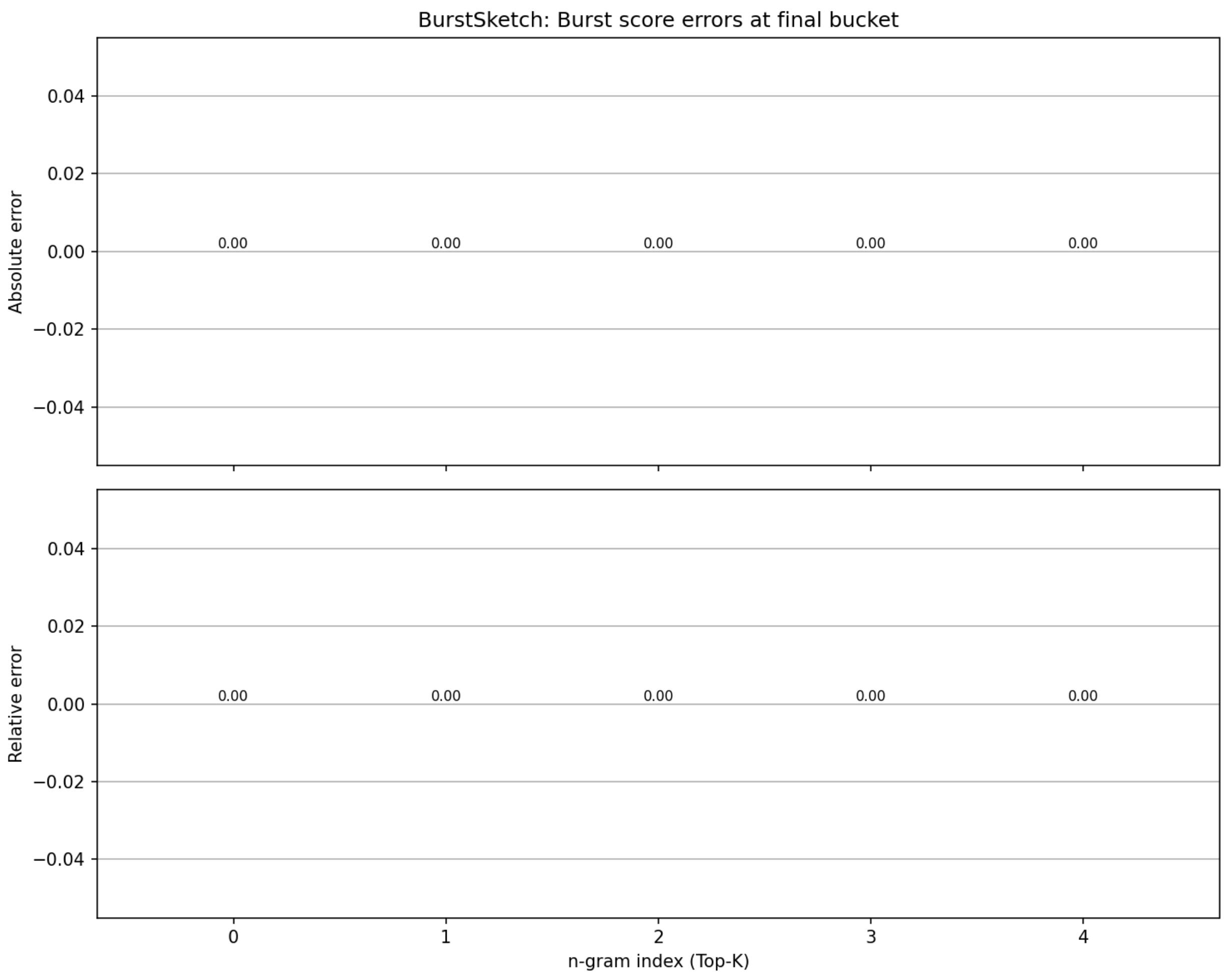

Figure 115: Absolute relative error for Burst Sketch at width 32768 and depth four.