

GEOMETRIC PERCEPTUAL IMPORTANCE ALGORITHM FOR REAL-TIME VISUALIZATION OF STREAMING TIME SERIES

Hotamov F. *, Nuraliyev F. , Gubo S. 

Abstract High-rate data streams require aggregation before screen rendering. The M4 method retains four points within each pixel column. However, important outliers, sharp bends, and level shifts can be lost. This study proposes AdaptiveM4 for improved time-series point selection. AdaptiveM4 retains the M4 rasterization structure and changes point selection. A weighted importance score is assigned to candidate points. The score combines deviation, local slope, and local curvature. Each component is normalized within the processed data batch. Outliers and trend reversals receive higher selection priority. An optional domain rule also preserves predefined critical points. The output size is bounded by the specified budget. A time-aware version supports irregularly sampled time series. Ten baseline methods are compared under equal output budgets. Four synthetic datasets and one real labeled dataset are evaluated. Geometric, perceptual, and retention metrics are used for comparison. The highest anomaly-retention F1 reaches 0.318 with tuned weights. The strongest baseline reaches an F1 score of 0.283. However, the real-data F1 decreases to 0.232 under held-out tuning. Therefore, the reported improvement is not confirmed on unseen data. AdaptiveM4 processes 100,000 points in approximately 20 milliseconds. This result indicates substantially lower computational cost than geometric simplifiers. However, whole-curve fidelity remains lower than several baseline methods. The advantage also disappears under a loose matching tolerance. Weight selection remains sensitive to the evaluation dataset. All experiments remain reproducible with fixed random seeds.

Keywords: time-series visualization; curve simplification; M4; largest triangle three buckets; perceptual fidelity; Frechet distance; structural similarity; anomaly retention; real-time streaming; computer graphics; Information Systems; Modeling and Simulation; Applied Mathematics.

AMS Mathematics Subject Classification: 68T05, 68T01, 92B20.

DOI: 10.32523/2306-6172-2026-14-3-64-84

1 Introduction

High-rate data streams are routinely displayed on dashboards. Screen resolution is often exceeded by their sampling rates. Each plot panel is limited to several thousand pixels. This width is quickly filled by fast sensors. Consequently, many samples are assigned to each pixel column. Most samples are transferred and drawn without visual benefit. Data aggregation is therefore used before visualization. The n input samples are reduced to $m \ll n$ points. The visual form of the full series is preserved by these points. This task is defined as curve simplification under perceptual and temporal constraints. The rendered line is represented by a planar time-value polyline. Its distance from the original curve is minimized. The stream arrival rate is also maintained. Several aspects of this task are addressed by established

*Corresponding Author.

methods. Rasterization is addressed directly by M4 [1]. Four samples are retained from each pixel column. These samples are the minimum, maximum, first, and last values. Therefore, the full line’s two-color rasterization is reproduced exactly. A perceptual objective is pursued through largest triangle three buckets (LTTB) [2]. The point with the largest local triangle is retained from each bucket. Geometric deviation is minimized by Douglas-Peucker (DP) [3]. Points with the smallest effective triangle areas are removed by Visvalingam-Whyatt (VW) [4]. Equal-width frames are averaged by piecewise aggregate approximation (PAA) [5]. The averages are quantized into a discrete alphabet by symbolic aggregate approximation (SAX) [6]. The series is smoothed by Automatic Smoothing for Attention Prioritization (ASAP) [7]. Noise is thereby suppressed before display.

Recent database and visualization work extends these ideas. M4 is generalized to a relational operator set by visualization-driven data aggregation (VDDA) [8] and computed over log-structured storage in Apache IoTDB [9]. Progressive queries are served by an ordered multi-level min-max representation (OM3) [10], and interaction is accelerated by a min-max cache with error bounds [11]. LTTB is accelerated by a min-max pre-selection [12], scaled to interactive charts by Plotly-Resampler [13], and improved on convex hulls by iterative largest triangle sampling (ILTS) [14]; point selection for line charts is assessed with metrics in [15].

Error-bounded simplification is also relevant. Swing and slide filters are used by online piecewise linear approximation (PLA) [16], similar segments are merged by Sim-Piece [17] and Mix-Piece [18], and autocorrelation is preserved by CAMEO [19]. A discord score for anomalies is supplied by the matrix profile [20]. Perceptual studies rank position above area and curvature [21], and graphical perception is also modeled by neural networks [22]. Time-series representations are surveyed in [23], anomaly detectors are benchmarked in [24], common benchmarks are criticized in [25], and the symbolic representation is formalized further in [26].

A common gap runs through these families: each optimizes a single geometric quantity. M4 uses column extrema, LTTB and VW use triangle area, DP uses chord deviation, and ASAP uses variance and kurtosis. None fuses statistical rarity, slope change, and curvature into one budgeted score, and none offers a first-class hook for domain rules. As a result a spike, a level shift, or the onset of a drift may be dropped whenever it fails the one criterion a given method happens to measure.

AdaptiveM4 is designed to close this gap. It keeps the M4 rasterization backbone but replaces positional selection with a weighted composite geometric score; prioritizes outliers and trend reversals through a multi-path nomination rule; adapts the level of detail to a latency budget through an adaptive bucket rule; and preserves domain-critical points through an optional semantic rule. Because the output is capped to a fixed budget, all methods can be compared at equal cost. The main contributions are as follows.

1. A scale-free composite importance score fusing statistical deviation, local slope magnitude, and triangle-area curvature, each normalized per batch to the unit range. The uniform-sampling assumption is stated, and a time-aware form is given for irregular sampling.
2. A multi-path nomination rule that prioritizes outliers, reversals, and optional domain points, retaining them whenever the nominated set fits the budget. A strict output cap keeps comparison fair.
3. An adaptive bucket rule with an explicit update equation that rescales the bucket count from the latency ratio, plus an optional semantic module.
4. A reproducible geometric-perceptual evaluation over ten baselines classified by output semantics, four synthetic patterns, and one real dataset, including a weight search, an

ablation, a leakage check on the weight selection, and multiplicity-corrected significance tests.

The terminology used throughout is fixed here, because the compared methods are not of one kind and the distinction affects how they may be measured. Selection returns a subset of the input samples unchanged, as in M4, LTTB, MinMax, DP, VW, and PLA. Aggregation replaces a group of samples by a summary value, as in PAA. Smoothing replaces each value by a filtered value over a neighbourhood, as in ASAP. Symbolic representation maps segments to a finite alphabet and reconstructs representative values, as in SAX. Approximation fits a model, such as a sequence of line segments, subject to an error bound. AdaptiveM4 is a selection method with an M4 rasterization backbone. Only selection methods preserve original timestamps exactly, which is what makes timestamp-level retention measurable for them and not for the others.

2 Materials and Methods

2.1 The AdaptiveM4 algorithm

A batch of samples is denoted by D , each sample a pair (t_i, x_i) of a timestamp and a value, with strictly increasing timestamps. Given an output budget $m \ll n$, a subset S of at most m points is produced that minimizes the distance to D while prioritizing salient points.

Equations (1)-(3) assume a **uniform sampling interval**, so that $t_{i+1} - t_i = \Delta t$ is constant. Both evaluated data sources satisfy this assumption: the building-health series is recorded at one sample per second, and the synthetic series are generated on a one-second grid. Under this assumption the index spacing is a faithful proxy for the time spacing, and the geometric quantities may be written in terms of values alone. The non-uniform generalization in Equations (2a)-(3a) below is the intended extension of the method but is not exercised by the experiments reported here.

Three scale-free geometric quantities are computed per interior sample from a mean μ and a standard deviation σ whose estimation is specified below: the standard score in Equation (1), the central slope in Equation (2), whose absolute value is the local slope magnitude, and the local curvature in Equation (3). For uniformly spaced samples this triangle area is proportional to the magnitude of the second finite difference; it is not the second derivative itself.

$$Z_i = \left| \frac{x_i - \mu}{\sigma} \right| \quad (1)$$

$$\text{slope}_i = \frac{x_{i+1} - x_{i-1}}{2} \quad (2)$$

$$A_i = \frac{|x_{i-1} - 2x_i + x_{i+1}|}{2} \quad (3)$$

For irregular sampling the two geometric terms are replaced by time-aware forms: the central derivative is taken over the true time span in Equation (2a), and the triangle area as one half of the determinant of the three (t, x) points in Equation (3a). Equations (2) and (3) are recovered from these when $t_{i+1} - t_i$ is constant.

$$\text{slope}_i^* = \frac{x_{i+1} - x_{i-1}}{t_{i+1} - t_{i-1}} \quad (2a)$$

$$A_i^* = \frac{1}{2} |(t_i - t_{i-1})(x_{i+1} - x_{i-1}) - (t_{i+1} - t_{i-1})(x_i - x_{i-1})| \quad (3a)$$

The statistics are windowed and batch-causal. A rolling window of the last $W = 100$ observed values supplies μ and σ ; it is carried across batches and updated only after the current batch has been aggregated, so the statistics used to score a batch depend only on preceding batches. On a cold start the window is empty and the batch mean and standard deviation are used instead. A degenerate spread is guarded: σ is replaced by one when it is zero or undefined.

The scope of this causality is stated precisely, since the method is not causal at the sample level. Within a batch the algorithm looks ahead: the slope and area terms of Equations (2) and (3) read the succeeding sample x_{i+1} , the normalizing maxima are taken over the whole batch, and the cold-start fallback uses that batch's own statistics. The method is therefore causal at batch granularity and non-causal within a batch. This is the appropriate model for visual aggregation, where a batch corresponds to a rendered window that is complete before it is drawn, but it is not a sample-by-sample online estimator and should not be described as one. A strictly sample-causal variant would replace the central difference by a backward difference and the batch maxima by running maxima, at a cost in curvature sensitivity that is not evaluated here.

Each quantity is divided by its maximum within the current batch. The normalized terms are denoted by $S_{\text{stat},i}$, $S_{\text{slope},i}$, and $S_{\text{area},i}$. A zero maximum is replaced by one. Thus, a constant batch is mapped to zero. An undefined value is therefore avoided. Maximum normalization is selected because only within-batch rankings affect selection. Under a fixed budget, identical selections are produced by identical rankings. Therefore, the same points are selected after any strictly monotone rescaling. However, sensitivity to one extreme value is introduced by this normalization. The other scores are compressed by that extreme value. Percentile or interquartile normalization is considered a robust alternative. Its evaluation is reserved for future work. The composite score is defined in Equation (4). The three weights are constrained to sum to one.

$$\text{Score}_i = w_{\text{stat}} S_{\text{stat},i} + w_{\text{slope}} S_{\text{slope},i} + w_{\text{area}} S_{\text{area},i} \quad (4)$$

The endpoints are pinned to the maximum score. The released weights are $w_{\text{stat}} = 0.40$, $w_{\text{slope}} = 0.10$, and $w_{\text{area}} = 0.50$, selected by a grid search (Section 3.3) whose protocol and sensitivity are examined in Section 3.5. Curvature and deviation are dominant; steepness is least important.

The complete procedure is given in Algorithm 1 with the cost of each step. Steps are executed in the order listed; k denotes the number of candidates entering the cap.

Algorithm 1. AdaptiveM4 batch aggregation. Input: samples D of length n , budget m . Output: a subset S with $|S| \leq m$.

A point is nominated by several independent paths, as defined in Equation (5), where θ is the composite threshold, τ the standard-score threshold, δ the slope-change threshold, and $S_{\text{sem},i}$ the optional semantic flag.

$$\text{nominate}_i = [\text{Score}_i \geq \theta] \vee [Z_i > \tau] \vee [\Delta\text{slope}_i > \delta] \vee [S_{\text{sem},i} = 1] \quad (5)$$

The reversal term is defined explicitly. The one-sided differences at an interior point are $s_i^- = x_i - x_{i-1}$ and $s_i^+ = x_{i+1} - x_i$, the reversal magnitude is $\Delta\text{slope}_i = |s_i^- - s_i^+|$, and the point is nominated when $s_i^- \cdot s_i^+ < 0$ and $\Delta\text{slope}_i > \delta$. The first and last samples have no two-sided neighbourhood and are excluded from this path; they are pinned separately.

#	Step	Cost
1	If $n \leq m$, return D unchanged	$O(n)$
2	Take μ, σ from the rolling window; compute Z_i , slope_i , A_i and their batch maxima	$O(n)$
3	Normalize each term; form Score_i by Equation (4); pin the endpoints to 1	$O(n)$
4	Mark nominated points by Equation (5)	$O(n)$
5	Set m_{eff} and B by Equation (6)	$O(1)$
6	Per bucket, emit first, last, min, max and the top-score point	$O(n)$
7	Merge the nominated points; de-duplicate by timestamp	$O(k)$
8	If more than m candidates remain, rank by priority and cap to m	$O(k \log k)$
9	Append the batch to the rolling window	$O(n)$

The defaults are $\theta = 0.60$, $\tau = 2.5$, and $\delta = 1.25\sigma$. The remaining budget is filled by M4 rasterization, with the highest-score point per bucket also emitted when its score exceeds a small floor of 0.02.

The status of the cap is stated precisely, because it bounds what the method can promise. Nominated points are prioritized, not unconditionally retained. Each candidate is ranked by a priority equal to its composite score plus a boost of one for any nomination path, so that every nominated point outranks every non-nominated one. The two endpoints are pinned first, then candidates are admitted in descending priority until the budget is exhausted. Retention of all nominated points is therefore guaranteed only while the nominated set together with the two pinned endpoints fits the budget, that is while $|\{i : \text{nominate}_i\} \cup \{1, n\}| \leq m$; the endpoints are counted because they are admitted before any nominated point and consume budget regardless of their own scores. When this bound is exceeded, the lowest-scoring nominated points are discarded. This is a deliberate trade: the size contract is honoured exactly, which is what makes the equal-budget comparison in Section 3 fair.

The adaptive bucket rule is specified next. A bucket multiplier ρ is maintained, initialized to $\rho_0 = 1$ and confined to $[1, 4]$. The latency ratio is $\lambda = L_{\text{obs}}/L_{\text{target}}$, where L_{obs} is the latency observed on the previous batch and $L_{\text{target}} = 200$ ms. The update is a multiplicative controller with a dead band, given in Equation (6), invoked at most once per second, so it reacts per batch only at lower batch rates.

$$\rho \leftarrow \begin{cases} \min(4, 1.2\rho) & \lambda > 1.2 \\ \max(1, 0.9\rho) & \lambda < 0.5 \text{ and } \rho > 1 \\ \rho & \text{otherwise} \end{cases} \quad (6)$$

The bucket count follows from ρ as $B = \max(1, \min(\lfloor (m_{\text{eff}}/4)/\rho \rfloor, \lfloor n/4 \rfloor))$, where $m_{\text{eff}} = \max(1, m - |\text{nominated}|)$ is the budget left after the nominated points are reserved. The lower clamp matters when nominations alone fill or exceed the budget: at least one bucket is still formed, and the surplus is resolved by the cap rather than by an empty rasterization pass. Aggregation is thus coarsened when latency is high and refined when headroom exists. Only the bucket count is rescaled; the geometric terms are unchanged. This rule follows the level-of-detail (LOD) idea, without a weighted multi-resolution term.

Points that are critical for domain reasons, such as a strain reading above a safety limit, are marked by an optional semantic rule. The default is no rule, which yields a pure geometric-perceptual simplifier.

The cost follows from the steps above. The feature pass computes Z_i , slope_i , and A_i in constant time per sample over a constant number of passes, for $O(n)$; the nomination and bucket passes are also single linear scans, for $O(n)$; and the final cap sorts the merged

candidate set at a cost of $O(k \log k)$ in the candidate count k . The total batch cost is therefore

$$O(n + k \log k) \quad (7)$$

The size of k must be bounded for this to be informative. The bucket pass emits at most five points per bucket — first, last, minimum, maximum, and the top-score representative — over at most $\lfloor m_{\text{eff}}/4 \rfloor$ buckets, contributing at most $\frac{5}{4}m$ candidates, and the nominated set contributes the remainder. Two regimes follow. In the normal regime the nominated set is itself $O(m)$, so $k = O(m)$ and the total is $O(n + m \log m)$, which reduces to $O(n)$ when m is a fixed constant; this is the operating regime of interest, since the budget is the display width and does not grow with the stream. In the adversarial regime the thresholds admit a constant fraction of the batch — a series in which almost every point is a sign reversal exceeding δ , for instance — and then $k = O(n)$ and the bound degrades to $O(n \log n)$. No such case arises in the evaluated data, where nominations remain a small multiple of the budget. The bound is thus $O(n)$ in the input with a log-linear term in the candidate set rather than a strict $O(n)$ total, and the linear scan dominates in all measured configurations, consistent with Section 3.1.

2.2 Datasets, metrics, and evaluation protocol

A real labeled dataset of building structural health monitoring (SHM) is used, providing one thousand one-second samples over four channels — tri-axial acceleration, strain, temperature, and a binary condition label — of which two hundred ninety-six samples are labeled as anomalies. The strain channel is the primary series, the label is the ground truth, and the dataset is released with the study artifacts.

The ground-truth unit is stated explicitly, because it determines which matching rule is admissible. Each label marks a single anomalous timestamp, not an interval or an event spanning several samples, so the 296 labels are 296 individually labeled samples. A selection method either keeps a labeled sample or drops it, so exact-timestamp matching is well defined for that family; it is not well defined for methods that synthesize new timestamps, a case treated below.

Four synthetic patterns are also generated. A fixed seed of 42 is used, and all series are placed on a one-second grid with $n = 5000$ points. The generators are specified in full so that the data can be reconstructed exactly. Spike: a baseline $50 + 15 \sin(2\pi i/150)$ with additive noise $\mathcal{N}(0, 1)$; twenty positions are drawn uniformly without replacement from the interior, and each is displaced by $\pm(35 + 20u)$ with $u \sim \mathcal{U}(0, 1)$ and a random sign; the displaced samples are the labels. Level Shift: a piecewise-constant mean over eight equal segments starting at 50, where each of the seven interior boundaries adds $\pm(15 + 15u)$, with noise $\mathcal{N}(0, 1.5)$; the seven boundary samples are the labels. Gradual Drift: a ramp of total magnitude 40 applied linearly across the middle third, held constant thereafter, over a baseline $50 + 3 \sin(2\pi i/120)$ with noise $\mathcal{N}(0, 1)$; every fiftieth sample inside the drift region is labeled, giving thirty-four sparse anchors. Oscillation: a baseline $50 + 8 \sin(2\pi i/300)$ with noise $\mathcal{N}(0, 0.4)$, plus a burst $16 \sin(2\pi i/50)$ over the middle third; local maxima inside the burst are located by neighbour comparison on the realized noisy series and every second one is labeled. The burst spans about thirty-three full cycles, but because maxima are detected on the noisy signal rather than on the noiseless envelope, additive noise creates further local maxima, and the labeling procedure yields fifty-one labeled extrema at seed 42. All Oscillation recalls in Table 4 are therefore multiples of $1/51$; for example the AdaptiveM4 value 0.235 is $12/51$ and the ASAP value 0.020 is $1/51$. The count is a property of the realized series and varies slightly across seeds.

Three metric families are used. All geometric metrics are computed after both curves are mapped into a normalized $[0, 1] \times [0, 1]$ space using the bounds of the original curve on both axes, so that a simplification is always measured against the ground-truth extent and the heterogeneous time and value axes become commensurable. The discrete Fréchet distance is computed by the Eiter-Mannila dynamic program [28], with the classical continuous formulation as background [27]. The Hausdorff distance is symmetric and segment-aware: each vertex is measured to the nearest point on the other curve’s polyline rather than its nearest vertex, which is the correct treatment for a sparse simplification [29]. The area difference is the integral of the absolute vertical deviation $|x(t) - \hat{x}(t)|$, evaluated by the trapezoidal rule over the original timestamps with the simplification linearly interpolated at each of them, divided by the total plot area. Lower values are better for these three.

Perceptual fidelity is measured by two raster-image similarity scores. Both curves are rendered with an identical protocol: a 256×128 grid; one value per pixel column obtained by linear interpolation of the polyline; adjacent columns joined by a vertical fill, giving a one-pixel connected line; binary intensity with no anti-aliasing and no axes, grid, or margins; and the same original-curve bounds as above. The structural similarity index measure (SSIM) is then computed on these binary rasters with dynamic range $L = 1$, stabilizers $C_1 = 0.01^2$ and $C_2 = 0.03^2$, and averaging over non-overlapping 8×8 windows [30]. The pixel error is the fraction of differing pixels between the two renderings. A higher SSIM and a lower pixel error are better. Both are properly described as raster-image similarity measures: SSIM was developed for natural-image quality assessment, and its agreement with human judgement on binary line-chart rasters has not been established here, so it is used as a reproducible proxy for rendered-shape agreement rather than a validated model of perception.

Feature retention is measured by precision, recall, and the F1 score, with matches found by exact timestamp. Recall, the fraction of labeled anomalies that survive aggregation, is the quantity of primary interest; precision is the fraction of emitted points that are labeled anomalies. The attainable range is set by the budget rather than the label rate: with $m = 100$ emitted points and 296 labeled anomalies, recall cannot exceed $100/296 \approx 0.34$ even if every emitted point is an anomaly, while precision can in principle reach one. The observed values reflect this asymmetry, and F1 is reported because it combines the two under a common budget.

Ten methods are compared, and they are not methodologically homogeneous; this is made explicit because the retention metric treats the two groups differently. Point-selection methods return a subset of the original samples with their timestamps and values unchanged: AdaptiveM4, M4, LTTB, MinMax, DP, VW, PLA, and Random. Smoothing and symbolic methods synthesize new values or timestamps: ASAP applies a moving average and then subsamples, so its values are modified, and SAX replaces each segment by a quantized symbol reconstructed at the segment midpoint, so both its values and timestamps are new. Exact-timestamp retention is meaningful only for the first group; for the second it is structurally near-zero regardless of how well the shape is preserved, and the SAX score of exactly zero in Section 3.2 is a direct consequence of midpoint timestamps rather than evidence of poor shape fidelity. Those rows are reported for completeness and marked as not directly comparable on retention; a tolerance-based metric is introduced in Section 3.2 to rank them fairly. The geometric and perceptual metrics have no such restriction and are comparable across all ten methods.

The equal-budget protocol is defined per method, since the methods have different output semantics. Every method receives the same budget m , interpreted as the maximum number of emitted points. At $m = 100$: M4 forms $\lfloor m/4 \rfloor = 25$ buckets and emits the first, last,

minimum, and maximum of each, which is the pixel-column interpretation with twenty-five columns; when m is not divisible by four the remainder is left unused rather than allotted to a partial bucket. MinMax forms $\lfloor m/2 \rfloor = 50$ buckets and emits only the minimum and maximum of each, spending the same budget on half as many points per bucket and twice as many columns. Both may emit slightly fewer than m points when a bucket is sparse and its representatives coincide. LTTB, VW, DP, PLA, and Random select m points directly. One caveat applies to the PLA baseline: the published swing- and slide-filter formulation of [16] is an online, error-bounded method whose output size is not fixed in advance, which is incompatible with an equal-budget protocol. The baseline evaluated here is therefore a budget-constrained top-down variant that recursively splits at the point of maximum vertical error until m anchors are reached. This is close in behaviour to Douglas-Peucker, which is why the two return identical values on the real series in Tables 2 and 4, and the PLA rows should be read as a second error-driven simplifier under a fixed budget rather than as an evaluation of the online swing-filter algorithm. SAX uses $w = m$ PAA segments, one symbol each. ASAP selects the largest centred moving-average window whose smoothed series stays within a Kolmogorov-Smirnov distance threshold of the original, located by binary search over windows up to $2n/m$, and then uniformly subsamples the smoothed series to m points; this criterion is a faithful reimplement of the goal of [7] but not of its exact statistic, which ranks candidate windows by roughness and kurtosis, and the substitution affects only the ASAP baseline. No method exceeds the budget: AdaptiveM4 emits exactly m , while M4 emits 91 at $n = 1000$ and 99 on the real series and DP emits 99 at $n = 10000$, in each case because of sparse or degenerate buckets. The per-run output counts are released alongside the latencies.

Latency is the median of twenty-five timed runs at a fixed seed, preceded by eight warm-up runs to reach steady-state JIT compilation. Every value is written to a comma-separated file from which the figures are regenerated. The platform is reported in Section 3.1.

3 Results

3.1 Efficiency and output budget

Table 1 lists the median latency per method, and Figure 1 plots latency against input size for all ten. AdaptiveM4 scales linearly over the measured range: from $n = 10^4$ to $n = 10^5$ the input grows tenfold and the median latency grows from 2.03 ms to 20.08 ms, a factor of 9.9, so the per-point cost is essentially flat at about 0.20 s, matching the analysis in Section 2.1. The smallest input is the one exception, at 1.42 s per point, reflecting fixed per-batch overhead and residual JIT effects rather than the algorithm’s order. One hundred thousand points are processed in about twenty milliseconds, comfortably under the two-hundred-millisecond target. That target is not a universal definition of real time; it is adopted because roughly two hundred milliseconds is the interval below which a dashboard update is perceived as an immediate response, so the claim is conditional: the implementation meets it for inputs up to one hundred thousand points on the platform described below. The constant factor of AdaptiveM4 is higher than that of M4, LTTB, and MinMax, which complete in well under a millisecond at every size, the cost being the per-point feature computation. The separation from the accurate geometric simplifiers is the substantive result: at $n = 10^5$ AdaptiveM4 is roughly ten times faster than DP, eight times faster than VW, and six times faster than ASAP. Those three remain inside the target at 10^5 points, at 194.58 ms, 155.01 ms, and 111.89 ms, but with no headroom, and their growth is superlinear — DP rises by a factor of twelve between 10^4 and 10^5 against AdaptiveM4’s factor of ten from a base twenty times smaller

— so extrapolating their trend places them outside the target shortly beyond the measured range, whereas AdaptiveM4 retains an order of magnitude of headroom. Absolute latencies are single-machine medians and vary between runs and platforms; the ordering between method families, not the individual values, is what should be read from the table.

The benchmark environment is reported for reproducibility. An Intel Core Ultra 9 185H processor is used. Performance and efficiency cores are included in its hybrid design. Twenty-two logical processors are exposed. The system is equipped with 32 GB of memory. Windows 11 (amd64) is installed. JDK 21.0.3 and the HotSpot 64-Bit Server VM are used. The maximum heap is set to 8044 MB. Default JVM flags are used. The default garbage collector is also used. No fixed initial heap size is imposed. No garbage-collection adjustment is applied. The aggregation is executed on one thread. Parallel streams are not used.

Two factors are left uncontrolled. Dynamic CPU frequency control is retained at the system default. Core assignment is controlled by the operating system. Therefore, either a performance core or efficiency core may be assigned. The spread between runs is increased by this variation. These effects are reduced by the warm-up and 25-run median protocol. However, these effects are not eliminated. Absolute latencies should therefore be interpreted with this limitation. A stable method-family order is observed across repeated tests. Complete run metadata are released with the artifacts.

Table 1: Median latency in milliseconds under an equal one-hundred-point budget.

Method	n=1000	n=10000	n=50000	n=100000
AdaptiveM4	1.42	2.03	7.21	20.08
M4	0.21	0.22	0.13	0.36
LTTB	0.26	0.29	0.37	0.70
MinMax	0.12	0.11	0.28	0.53
DP	1.35	16.12	98.44	194.58
VW	0.69	5.59	60.28	155.01
ASAP	0.84	5.88	48.50	111.89
PLA	0.31	0.76	4.10	7.07
SAX	0.25	0.16	0.49	0.59
Random	0.17	0.27	0.68	2.25

3.2 Anomaly retention and pattern coverage

Table 2 reports the retention metrics on the real dataset, and Figure 2 shows the F1 score per method. AdaptiveM4 obtains the highest F1 and the highest recall, with strong precision. SAX scores zero because it resamples timestamps to segment midpoints; as set out in Section 2.2, exact-timestamp retention is not meaningful for SAX or ASAP, and their rows are marked accordingly rather than read as shape failures. Among the point-selection methods the ordering is the substantive result. The numerical lead is confirmed, but the significance reported in Section 3.4 shows it is not significant against the nearest baselines, and the practical margin over the strongest baseline is small: 0.318 against 0.283 for MinMax, a difference of 0.035 in F1. DP and PLA give identical results here, which follows from the budget-constrained PLA variant of Section 2.2 being algorithmically close to DP, so they should not be read as two independent confirmations.

Because exact-timestamp matching cannot score the resampling and symbolic methods fairly, a tolerance-based retention metric is introduced for them: a labeled anomaly counts

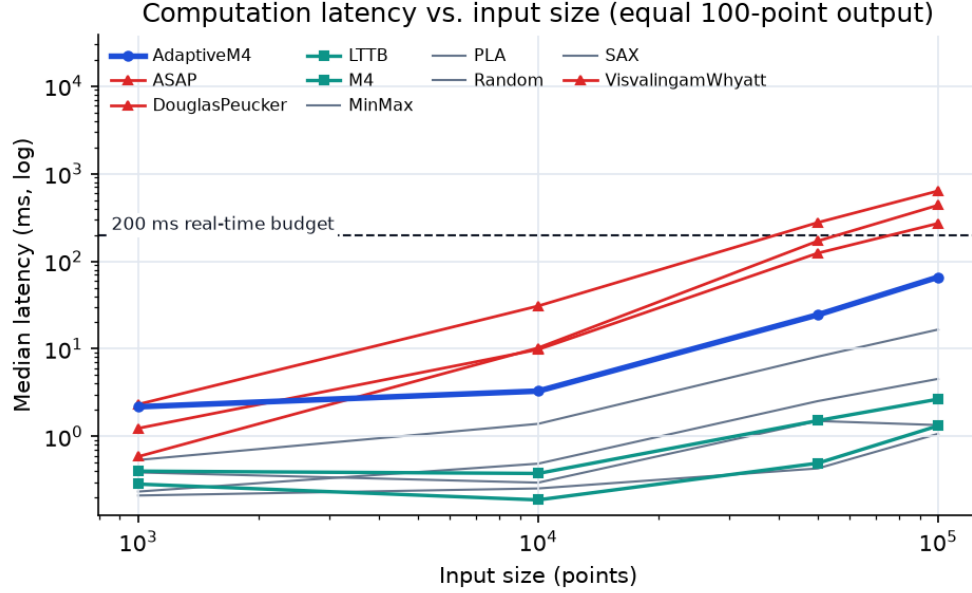


Figure 1: Median computation latency against input size under an equal one-hundred-point output. All ten compared methods are plotted and named in the legend; AdaptiveM4 is drawn in bold, the accurate geometric simplifiers in red, and the fast reference aggregators in green. Both axes are logarithmic. AdaptiveM4 scales linearly and retains an order of magnitude of headroom below the two-hundred-millisecond target, whereas the red curves grow superlinearly and approach the target at the largest measured input.

Table 2: Anomaly retention on the building-health data. The dataset holds two hundred ninety-six point-labeled anomalies. Methods marked † synthesize new values or timestamps, so exact-timestamp retention understates them; their rows are not directly comparable on this metric (Section 2.2).

Method	Class	Precision	Recall	F1
AdaptiveM4	selection	0.630	0.213	0.318
MinMax	selection	0.560	0.189	0.283
DP	selection	0.530	0.179	0.268
PLA	selection	0.530	0.179	0.268
LTTB	selection	0.510	0.172	0.258
M4	selection	0.505	0.169	0.253
VW	selection	0.420	0.142	0.212
Random	selection	0.270	0.091	0.136
ASAP †	smoothing	0.340	0.115	0.172
SAX †	symbolic	0.000	0.000	0.000

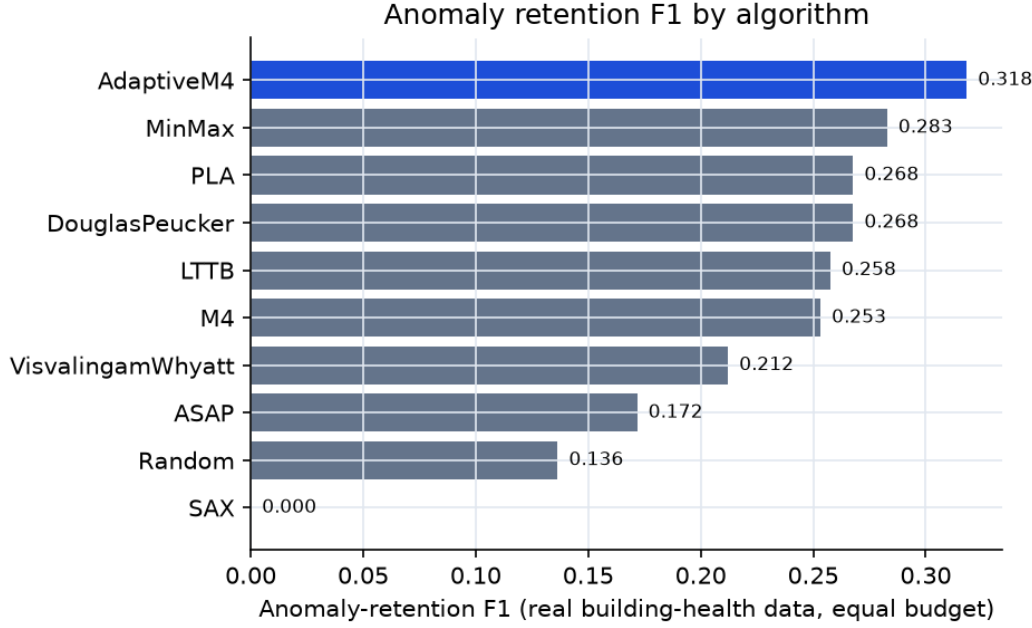


Figure 2: Anomaly-retention F1 score per method on the real building-health dataset under an equal budget. The two methods that synthesize new timestamps or values (ASAP, SAX) are shown for completeness but are not directly comparable on exact-timestamp retention.

as retained when the output places any vertex within $\pm w$ samples of its timestamp, with precision defined symmetrically. This asks whether the rendered curve carries a vertex where the anomaly occurs, which is what a reader perceives, rather than whether an original sample was returned verbatim. The series is sampled at one hertz, so w is both a sample count and a number of seconds. Table 3 reports the result.

The metric achieves what it was introduced for. SAX rises from exactly zero at $w = 0$ to 0.285 at a single sample of tolerance and to 0.990 at five, confirming that its zero in Table 2 is an artifact of midpoint reconstruction rather than a failure to preserve the anomalies; ASAP behaves likewise. Under this metric the two are not weak but among the strongest at loose tolerance, the expected behaviour of smoothing and symbolic representations, which place a representative near every region of activity while committing to no exact sample. The same table also qualifies our own method. AdaptiveM4 leads only at $w = 0$; at $w = 1$ it is level with LTTB and behind MinMax, and the gap widens with w . The interpretation is that AdaptiveM4 is optimized to return the anomalous sample itself, which is what the exact metric rewards and what matters when the selected points are inspected, queried, or used as indices into the raw stream. When the requirement is weaker, that the drawn line merely bend near the anomaly, the budget is better spent by a method that distributes vertices more evenly. Neither metric is correct in general: the exact variant suits point-selection semantics and the tolerance variant shape-level reading, and both are reported so that a reader may apply whichever matches their use. Table 4 lists the critical-point recall per pattern, with a pattern counted as detected when the recall reaches one half, and Figure 3 shows the pattern recall for three methods. AdaptiveM4 detects two of four patterns, Spike and Level Shift. No method exceeds this coverage, and DP and PLA match it exactly, so the correct statement is a tie for best rather than an outright lead. M4 detects only Spike. The level-shift gain is the clearest: all boundary points are kept by AdaptiveM4 through the reversal path, and none by

Table 3: Tolerance-based anomaly-retention F1 on the building-health data. A tolerance of $w = 0$ is exact-timestamp matching, as in Table 2. Methods marked $\dagger$ synthesize timestamps and are scored meaningfully only for $w > 0$.

Method	w=0	w=1	w=2	w=5
AdaptiveM4	0.318	0.488	0.595	0.789
MinMax	0.283	0.497	0.645	0.906
DP	0.268	0.438	0.576	0.748
PLA	0.268	0.438	0.576	0.748
LTTB	0.258	0.488	0.659	0.946
M4	0.253	0.398	0.570	0.813
VW	0.212	0.435	0.610	0.867
Random	0.136	0.353	0.530	0.756
ASAP $\dagger$	0.172	0.383	0.584	0.985
SAX $\dagger$	0.000	0.285	0.531	0.990

M4. Gradual Drift and Oscillation are recovered by no method, a limit that follows from the tight budget. See Table 4 where critical-point retention per pattern. A pattern is detected when the recall reaches one half. As in Table 2, this metric matches by exact timestamp; methods marked $\dagger$ synthesize new timestamps or values and are not directly comparable on it.

Table 4: Critical-point retention per pattern.

Method	Class	Spike	Level Shift	Drift	Oscillation	Detected
AdaptiveM4	selection	1.00	1.00	0.00	0.24	2
DP	selection	1.00	0.86	0.00	0.33	2
PLA	selection	1.00	0.86	0.00	0.33	2
VW	selection	0.00	1.00	0.03	0.14	1
M4	selection	0.95	0.00	0.00	0.10	1
MinMax	selection	0.95	0.00	0.00	0.20	1
LTTB	selection	0.85	0.29	0.03	0.12	1
Random	selection	0.05	0.00	0.00	0.00	0
ASAP $\dagger$	smoothing	0.00	0.00	0.03	0.02	0
SAX $\dagger$	symbolic	0.00	0.00	0.00	0.00	0

3.3 Fidelity trade-off, ablation, and adaptive response

Table 5 lists the quality metrics for all methods. The whole-curve fidelity of AdaptiveM4 is low: it is last on SSIM and on the area difference, and second worst on the Fréchet distance, where only MinMax is worse, since the averaging methods render smooth regions better. AdaptiveM4 is third best on the Hausdorff distance, at an observed 0.060 in normalized curve space — an empirical measurement on the evaluated data, not a derived bound, as no worst-case guarantee on point error is claimed or proved here. The pattern is consistent with the design: the budget is spent on outliers and high curvature, which keeps the largest single deviation small while reducing average fidelity. Feature fidelity is favored over uniform fidelity, and plain M4 or LTTB is preferred when uniform fidelity is the goal. See Table 5

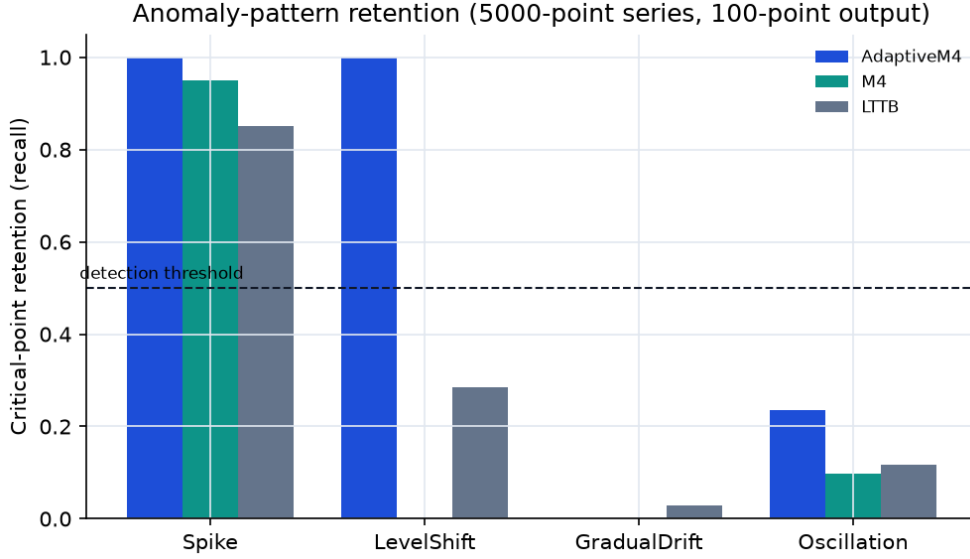


Figure 3: Critical-point retention per anomaly pattern for AdaptiveM4, M4, and LTTB, the three point-selection methods contrasted in the text.

where the real-data quality for all methods under an equal one-hundred-point budget. Lower Frechet, Hausdorff, area, and pixel values are better. A higher SSIM value is better. Unlike Tables 2 and 3, these metrics compare rendered curves rather than matched timestamps, so all ten methods are directly comparable here, including ASAP and SAX.

Table 5: Real-data quality for all methods under an equal one-hundred-point budget.

Method	Frchet	Hausdorff	Area diff	SSIM	Pixel err
AdaptiveM4	0.471	0.060	0.235	0.400	0.184
M4	0.373	0.035	0.190	0.404	0.192
LTTB	0.373	0.074	0.164	0.447	0.184
MinMax	0.555	0.016	0.180	0.406	0.195
DP	0.416	0.069	0.139	0.560	0.152
VW	0.383	0.201	0.132	0.492	0.167
PLA	0.416	0.069	0.139	0.560	0.152
ASAP	0.389	0.292	0.134	0.509	0.156
SAX	0.464	0.464	0.108	0.573	0.136
Random	0.393	0.313	0.134	0.538	0.144

The weight simplex is evaluated at one-tenth intervals. Only positive weight triples are included. Thus, thirty-six three-term combinations are evaluated. Each combination is scored by the balanced objective $F1 + \frac{1}{4} \text{SSIM} - \frac{1}{4} \text{AreaDiff}$. Scores are averaged across four datasets. Three synthetic datasets are used: Spike, Level Shift, and Oscillation. The real building-health series is also used. All objective terms are dimensionless. Each term is bounded within the unit interval. Therefore, direct comparison is permitted without rescaling. However, coefficients 1 and $\frac{1}{4}$ are selected by design. They are not obtained from optimization. Retention is assigned the primary objective. The fidelity terms are used as tie-breakers. Large rendered-shape costs are discouraged by these terms. Another trade-off point is generally selected by another

coefficient ratio. The highest-ranked combinations are listed in Table 6. The optimum is set at 0.40, 0.10, and 0.50. This combination is released as the default. Curvature and statistical deviation are assigned the largest weights.

Coefficient dependence is also tested without scalarization. The grid is evaluated through Pareto optimality. A configuration is classified as Pareto-optimal when no alternative dominates it. Dominance is defined by two conditions. An alternative must be at least equal across all three criteria. At least one criterion must also be strictly improved. Higher mean F1 and SSIM values are preferred. Lower mean area differences are preferred. Seventeen of thirty-six configurations are classified as Pareto-optimal. The released default is included among them. The highest mean F1 on the front is attained by this default. The expected trade-off is represented by the front. Under slope-dominant weights, SSIM is increased from 0.617 to 0.676. The area difference is reduced from 0.128 to 0.103. Meanwhile, mean F1 is reduced from 0.235 to 0.191. Real-data F1 is reduced from 0.318 to 0.207. The default or a nearby configuration is selected when retention is prioritized. Thus, the selected front position is determined by the coefficients. The validity of the choice is not determined by them. Dataset dependence is examined in Section 3.5; see also Table 6.

Table 6: Top weight configurations by the cross-dataset balanced objective.

w_stat	w_slope	w_area	mean F1	real F1	mean SSIM	area diff	objective
0.40	0.10	0.50	0.235	0.318	0.617	0.128	0.358
0.40	0.40	0.20	0.226	0.293	0.637	0.120	0.355
0.30	0.20	0.50	0.230	0.298	0.623	0.124	0.355
0.50	0.20	0.30	0.231	0.333	0.624	0.131	0.354

Table 7 lists a feature ablation on the real data, and Figure 4 shows it. Several facts are stated openly. The degenerate statistics-only setting lies outside the three-way fusion grid and reaches a higher F1 than the released default, as do some single-dataset settings, so the default is not the F1 maximizer on this dataset but the configuration that optimizes the cross-dataset balanced objective.

Performance is sensitive to the selected weights, and the sensitivity is not merely a matter of degree. The spread is large, from 0.485 for statistics-only down to 0.136 for slope-only, and three ablated settings fall at or below the baselines: slope-only reaches 0.136, equal to Random and well below MinMax at 0.283; no-statistics reaches 0.172; and area-only reaches 0.248. It is therefore not the case that every AdaptiveM4 configuration outperforms all baselines. What the ablation supports is narrower: the statistical deviation term is the single most important contributor to retention on this dataset, the area term is the principal contributor to rendered fidelity, and a fusion that drops either degrades to baseline level. The optional semantic module is shown as an added layer.

The adaptive bucket rule is exercised next by an injected latency trace, shown in Figure 5. The observed latency is driven to 400 ms against the 200 ms target, held there for three consecutive steps, then dropped to 30 ms. The multiplier responds as specified by Equation (6): it rises $1.000 \rightarrow 1.200 \rightarrow 1.440 \rightarrow 1.728$ while the target is exceeded, coarsening aggregation to shed load, then relaxes to 1.555 and 1.400 once headroom returns. This is a recovery, not a guarantee of continuous compliance. During the induced spike the observed latency exceeds the target by a factor of two, and the controller cannot prevent this: it reacts to the previous batch’s measurement and is rate-limited to one update per second, so it is corrective

Table 7: Feature ablation on building-health data.

Setting (stat/slope/area)	F1	SSIM	Area diff
stat only (1/0/0, degenerate)	0.485	0.445	0.254
no slope (0.30/0/0.20)	0.439	0.449	0.244
pre-tuning (0.30/0.25/0.20)	0.384	0.475	0.215
no area (0.30/0.25/0)	0.338	0.507	0.188
equal (1/1/1)	0.333	0.513	0.197
released default (0.40/0.10/0.50)	0.318	0.400	0.235
default with semantic (strain above p95)	0.308	0.441	0.216
area only (0/0/1)	0.248	0.422	0.211
slope only (0/1/0)	0.136	0.566	0.131

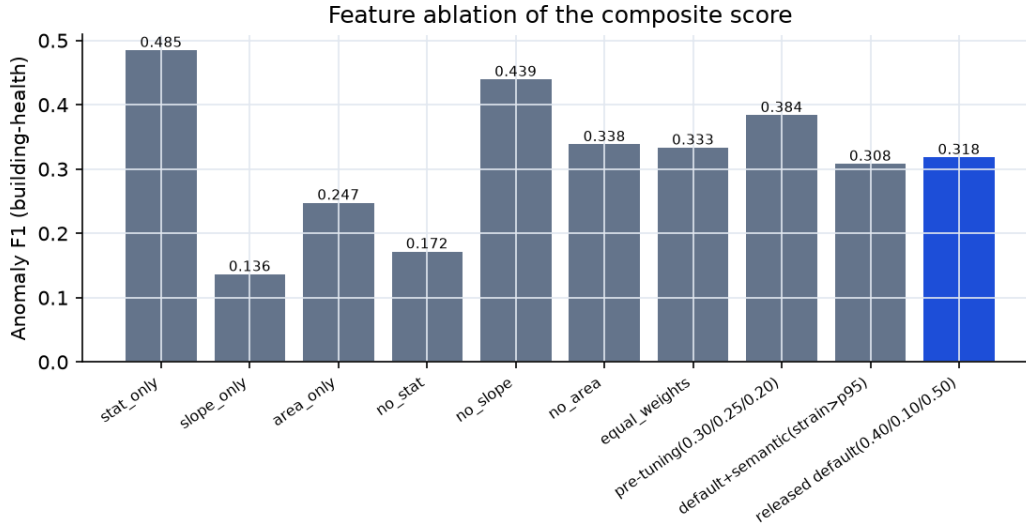


Figure 4: Feature ablation of the composite score on the building-health dataset.

rather than preventive. The output size contract is honoured at every step, with exactly one hundred points emitted throughout; see figures 4 and 5.

3.4 Statistical significance

The experimental unit is defined first, together with its limitation. One block is one (seed, pattern) pair, and the synthetic evaluation is repeated over ten seeds and four patterns, giving forty blocks. All algorithms see identical data within a block, which licenses the *pairing* across algorithms. Independence across blocks holds only partially: the ten seeds are independent, but the four patterns within a seed are drawn from the same seeded generator stream and are therefore not strictly independent, while the Wilcoxon and Friedman tests treat all forty blocks as exchangeable. The effective number of independent replicates is thus closer to ten than to forty, and the reported p-values are correspondingly anti-conservative. Two considerations bound the severity: the four patterns differ in generating process, not merely in noise draw, so their shared dependence is weak; and the conclusions below rest on effect sizes and intervals unaffected by this issue, with the significant comparisons having adjusted p-values between 10^{-6} and 10^{-4} , several orders of magnitude from the threshold. A design with one pattern

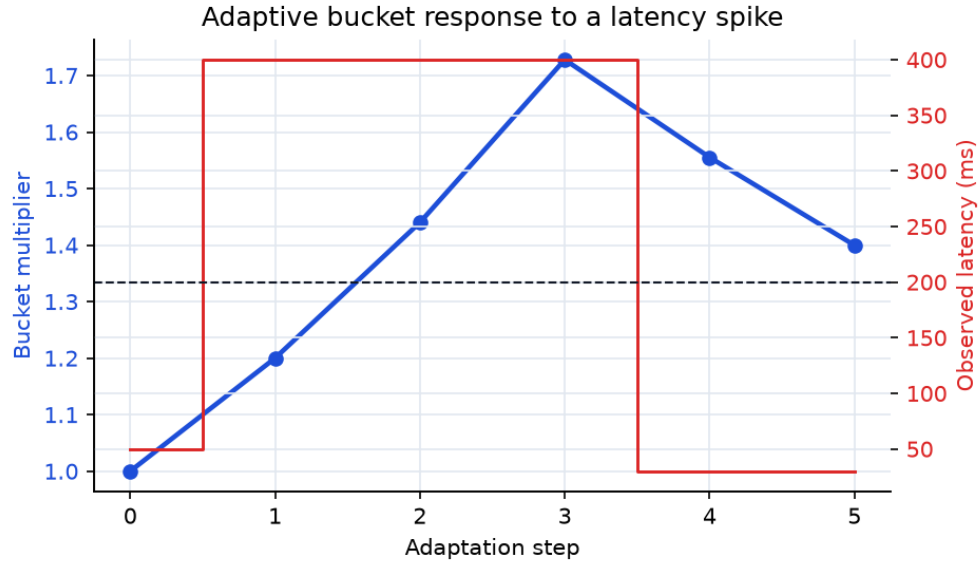


Figure 5: Adaptive bucket response to a latency spike.

per seed would remove the concern outright and is the recommended protocol for future replication. The real-data retention is tested by the McNemar exact test over the 296 labeled anomalies, and Figure 6 shows the mean F1 with standard deviation.

A Friedman omnibus test is applied first: the ten methods differ with high significance ($\chi^2 = 193.1$, $p \approx 9 \times 10^{-37}$). This should not be over-read. It includes SAX and ASAP, whose near-constant scores under exact-timestamp matching inflate the between-method spread, and it treats DP and the budget-constrained PLA variant as two methods although they are algorithmically close and nearly collinear here, so the ten columns are not ten independent treatments. The omnibus establishes only that the methods are not interchangeable; the pairwise tests carry the substantive comparisons. The same caveat applies to the Holm correction, which is conservative in the usual sense but computed over a family containing one near-duplicate pair, see Table 8. Nine pairwise comparisons are made against the same AdaptiveM4 sample in each family, so all pairwise p-values are adjusted by the Holm-Bonferroni step-down procedure. Table 8 reports the paired Wilcoxon signed-rank results on the synthetic blocks, with the statistic W , raw and adjusted p-values, the matched-pairs rank-biserial effect size r , and a percentile bootstrap 95% interval for the mean paired difference over 10,000 resamples.

AdaptiveM4 is significantly ahead of M4, LTTB, and MinMax after correction, with large effect sizes and intervals excluding zero. DP and the budget-constrained PLA variant are significantly ahead of AdaptiveM4, by 0.015 and 0.014 in mean F1; the gap is small in absolute terms but consistent, and its interval also excludes zero. The comparison against VW is not counted as a win: the median paired difference is exactly zero, so the significant p-value reflects an asymmetry in the tails rather than a shift in the centre, and the wide interval $[+0.044, +0.133]$ confirms the instability. On the real data the McNemar differences are not significant after correction against the nearest baselines — M4, LTTB, MinMax, DP, and PLA — but remain significant against ASAP, SAX, and Random. The comparison against VW is not significant after adjustment: its raw $p = 0.019$ becomes a Holm-adjusted $p = 0.112$. The practical sizes are small: the retention-rate gap over MinMax is +0.024 and over M4

Table 8: Paired Wilcoxon signed-rank tests on the forty synthetic blocks, AdaptiveM4 against each baseline. Positive differences favour AdaptiveM4. Significance is judged on the Holm-adjusted p-value.

Baseline	median diff	mean diff [95% CI]	r	W	p raw	p Holm
M4	+0.0656	+0.0561 [+0.0425, +0.0696]	+0.979	6.0	9.2e-07	8.3e-06
LTTB	+0.0644	+0.0490 [+0.0366, +0.0609]	+0.950	15.0	1.4e-06	1.1e-05
MinMax	+0.0324	+0.0378 [+0.0231, +0.0529]	+0.795	54.0	8.5e-05	3.4e-04
VW	+0.0000	+0.0868 [+0.0438, +0.1331]	+0.589	108.5	3.5e-03	4.9e-03
ASAP	+0.1121	+0.1352 [+0.0973, +0.1736]	+0.866	55.0	1.7e-06	1.1e-05
SAX	+0.1225	+0.1452 [+0.1076, +0.1833]	+1.000	0.0	1.5e-06	1.1e-05
Random	+0.1123	+0.1402 [+0.1035, +0.1775]	+1.000	0.0	1.7e-06	1.1e-05
DP	0.0078	0.0148 [0.0228, 0.0072]	0.729	55.0	7.5e-04	2.2e-03
PLA	0.0063	0.0142 [0.0222, 0.0066]	0.667	63.0	2.4e-03	4.9e-03

is +0.044, corresponding to an F1 difference over MinMax of 0.035. Table 9 lists the F1 rank across output budgets. Rank one is held on the real dataset at every budget, with two qualifications: at the tightest budget, $m = 50$, the rank is shared with MinMax, DP, and PLA at an identical F1 of 0.191, so it is a tie rather than a lead; and at the larger budgets, where the rank is held outright, the margins above are the appropriate measure of its worth, none of them statistically significant against the nearest baselines.

Table 9: Anomaly-F1 rank of AdaptiveM4 across output budgets among ten methods. “(tie)” marks a budget at which one or more baselines attain the identical F1 score, so rank one is shared rather than held outright.

Dataset	m=50	m=100	m=200	m=400
BuildingHealth (real)	1 (tie)	1	1	1
Spike (synthetic)	1	1 (tie)	1 (tie)	2
LevelShift (synthetic)	1 (tie)	1 (tie)	1 (tie)	1 (tie)

Different results are observed across datasets. Against M4, LTTB, and MinMax, a significant synthetic-data advantage is observed for AdaptiveM4. Different results are obtained against DP and budget-constrained PLA. On synthetic blocks, higher mean F1 values are achieved by both methods. Margins of 0.015 and 0.014 are observed. Significance is retained after correction. Thus, AdaptiveM4 is placed behind both methods. On real data, no significant difference is detected in either direction. Thus, parity is indicated rather than superiority. Both results are required for a complete comparison. On real data, AdaptiveM4 is ranked first at every budget. However, significance is not detected against the five nearest baselines. The ranking is based on weights tuned with the same dataset. Therefore, only competitive retention at interactive speed is claimed. The M4 pixel-column structure is also retained. This structure is not provided by DP or PLA. Comparable retention is achieved by them at much higher latency.

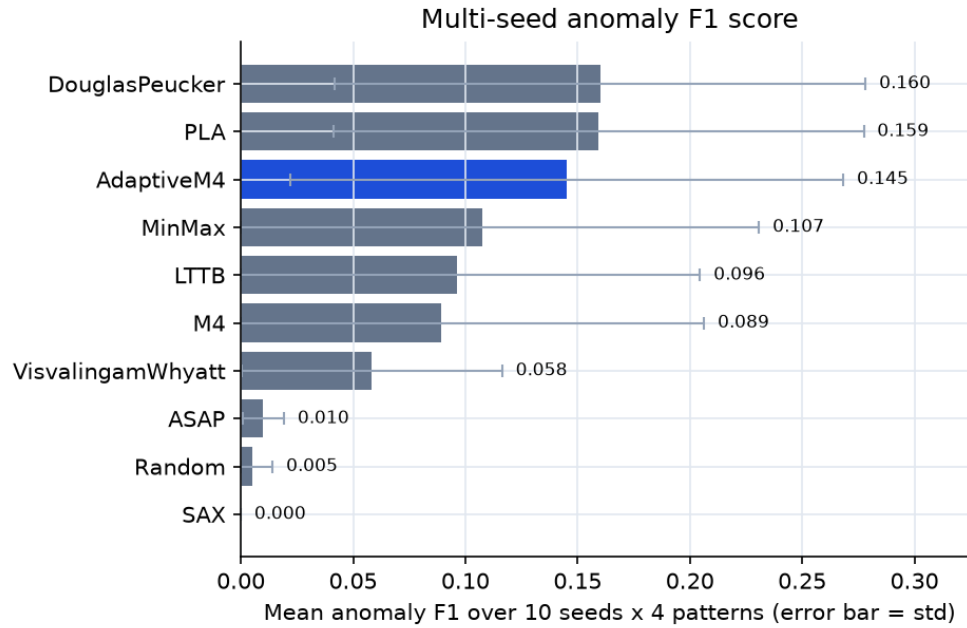


Figure 6: Multi-seed anomaly F1 score. The mean and the standard deviation over ten seeds and four patterns are shown.

3.5 Validity of the weight selection

The released weights are selected with an objective that includes real data. The same dataset is used for the headline evaluation. Thus, real data are included in parameter selection. An optimistic bias is therefore introduced into the reported figures. This bias is quantified through a held-out analysis. Synthetic and real objective components are recorded separately in the released grid. Therefore, selection is recomputed without the real dataset.

Different weights are selected when the real dataset is excluded. The 0.10/0.50/0.40 weights are selected by the synthetic-only objective. The released default is 0.40/0.10/0.50. With held-out selection, a real-data F1 of 0.232 is obtained. With the released default, a real-data F1 of 0.318 is obtained. Thus, selection optimism is estimated at +0.086 F1. The 0.035 margin over the strongest baseline is exceeded by this value. Two qualifications are required. First, weak discrimination is provided by the synthetic-only objective. A mean synthetic F1 range of only 0.004 is observed. The choice is determined mainly by fidelity tie-breakers, not retention. Second, the strongest baseline is exceeded by sixteen configurations on real data. Therefore, the advantage is not attributed entirely to selection. Thus, a real but weight-dependent retention advantage is supported. However, its size is overstated by the single released value. The 0.318 result should therefore be treated as tuned. It should not be treated as unbiased on unseen data. An independent test set is required for confirmation. Additional labeled real datasets are also required.

Conclusions

AdaptiveM4 is introduced as a feature-based geometric selection method. Time-series aggregation is supported by this method. Deviation, local slope, and curvature are combined into one score. Outliers and trend reversals are prioritized by a multi-path rule. These points are retained when permitted by the output budget. An optional semantic selection rule is also

provided. Linear computational complexity is retained with respect to input size. Strong anomaly retention is observed on the evaluated data. A maximum anomaly-retention F1 of 0.318 is obtained with tuned weights. An F1 score of 0.283 is obtained by the strongest baseline. Under held-out weight selection, F1 is reduced to 0.232. Thus, improvement is not confirmed on unseen data. A significant advantage over fast aggregators is found on synthetic data. No significant difference is found against the five nearest baselines. Whole-curve fidelity is reduced because feature selection is prioritized. Gradual drift and fine oscillations are poorly retained at small budgets. Larger budgets or multiple passes are therefore required. More computation is required by AdaptiveM4 than by M4 and LTTB. However, the tested time target is met. Equal output budgets and strict size limits are used. Several metric groups are used to reduce measurement bias. Paired nonparametric tests and Holm correction are also applied. External validity is identified as the main limitation. Only one real dataset and one computing platform are evaluated. Uniformly sampled data are used in the main experiments. Weight selection is also influenced by the evaluation data. Additional real-world datasets should be evaluated in future studies. Irregularly sampled time series should also be examined. Independent weight selection should be applied to new datasets. Gradual drift and fine oscillations should be addressed further. Online weight adaptation should also be investigated.

Acknowledgment

All numbers are produced by a fixed-seed Java harness on JDK 21.0.3, with one seed for point estimates and ten seeds for significance. Results, run metadata, significance tests, and figure scripts are released, together with the per-run output counts, the full weight-search grid, and the adaptive-controller trace, and the experiments are regenerated by the provided commands. The statistical output, including the Holm-adjusted p-values, effect sizes, bootstrap intervals, and held-out weight-selection check, is stored in the stats file. Every table and figure is generated from the released comma-separated files and reports no quantity absent from them. Parts of the software and the drafting are supported by AI, under the authors' direction. All design choices and measurements are verified by the authors, and every claim is checked against the released artifacts.

References

- [1] U. Jugel, Z. Jerzak, G. Hackenbroich, and V. Markl, "M4: A visualization-oriented time series data aggregation," *Proc. VLDB Endow.*, vol. 7, no. 10, pp. 797–808, 2014, doi: 10.14778/2732951.2732953.
- [2] S. Steinarsson, "Downsampling time series for visual representation," M.S. thesis, Fac. Ind. Eng., Mech. Eng. Comput. Sci., Univ. Iceland, Reykjavik, Iceland, 2013.
- [3] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica*, vol. 10, no. 2, pp. 112–122, 1973, doi: 10.3138/FM57-6770-U75U-7727.
- [4] M. Visvalingam and J. D. Whyatt, "Line generalisation by repeated elimination of points," *Cartogr. J.*, vol. 30, no. 1, pp. 46–51, 1993, doi: 10.1179/000870493786962263.
- [5] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality reduction for fast similarity search in large time series databases," *Knowl. Inf. Syst.*, vol. 3, no. 3, pp. 263–286, 2001, doi: 10.1007/PL00011669.
- [6] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proc. 8th ACM SIGMOD Workshop Res. Issues Data Mining Knowl. Discov. (DMKD)*, 2003, pp. 2–11, doi: 10.1145/882082.882086.

- [7] K. Rong and P. Bailis, “ASAP: Prioritizing attention via time series smoothing,” *Proc. VLDB Endow.*, vol. 10, no. 11, pp. 1358–1369, 2017, doi: 10.14778/3137628.3137645.
- [8] U. Jugel, Z. Jerzak, G. Hackenbroich, and V. Markl, “VDDA: Automatic visualization-driven data aggregation in relational databases,” *VLDB J.*, vol. 25, no. 1, pp. 53–77, 2016, doi: 10.1007/s00778-015-0396-z.
- [9] L. Rui, X. Huang, S. Song, Y. Kang, C. Wang, and J. Wang, “Time series representation for visualization in Apache IoTDB,” *Proc. ACM Manag. Data*, vol. 2, no. 1, Art. no. 35, 2024, doi: 10.1145/3639290.
- [10] Y. Wang *et al.*, “OM3: An ordered multi-level min-max representation for interactive progressive visualization of time series,” *Proc. ACM Manag. Data*, vol. 1, no. 2, Art. no. 145, 2023, doi: 10.1145/3589290.
- [11] S. Maroulis, V. Stamatopoulos, G. Papastefanatos, and M. Terrovitis, “Visualization-aware time series min-max caching with error bound guarantees,” *Proc. VLDB Endow.*, vol. 17, no. 8, pp. 2091–2103, 2024, doi: 10.14778/3659437.3659460.
- [12] Jeroen Van Der Donckt, Jonas Van Der Donckt, M. Rademaker, and S. Van Hoecke, “MinMaxLTTB: Leveraging MinMax-preselection to scale LTTB,” in *Proc. IEEE Vis. Vis. Anal. (VIS)*, 2023, pp. 21–25, doi: 10.1109/VIS54172.2023.00013.
- [13] Jonas Van Der Donckt, Jeroen Van Der Donckt, E. Deprost, and S. Van Hoecke, “Plotly-Resampler: Effective visual analytics for large time series,” in *Proc. IEEE Vis. Vis. Anal. (VIS)*, 2022, doi: 10.1109/VIS54862.2022.00013.
- [14] L. Rui, X. Huang, S. Song, C. Wang, J. Wang, and Z. Cao, “Largest triangle sampling for visualizing time series in database,” *Proc. ACM Manag. Data*, vol. 3, no. 1, Art. no. 49, 2025, doi: 10.1145/3709699.
- [15] Jeroen Van Der Donckt, Jonas Van Der Donckt, M. Rademaker, and S. Van Hoecke, “Data point selection for line chart visualization: Methodological assessment and evidence-based guidelines,” 2023, arXiv:2304.00900.
- [16] H. Elmeleegy, A. K. Elmagarmid, E. Cecchet, W. G. Aref, and W. Zwaenepoel, “Online piece-wise linear approximation of numerical streams with precision guarantees,” *Proc. VLDB Endow.*, vol. 2, no. 1, pp. 145–156, 2009, doi: 10.14778/1687627.1687645.
- [17] X. Kitsios, P. Liakos, K. Papakonstantinou, and Y. Kotidis, “Sim-Piece: Highly accurate piecewise linear approximation through similar segment merging,” *Proc. VLDB Endow.*, vol. 16, no. 8, pp. 1910–1922, 2023, doi: 10.14778/3594512.3594521.
- [18] X. Kitsios, P. Liakos, K. Papakonstantinou, and Y. Kotidis, “Flexible grouping of linear segments for highly accurate lossy compression of time series data,” *VLDB J.*, vol. 33, no. 5, pp. 1569–1589, 2024, doi: 10.1007/s00778-024-00862-z.
- [19] C. E. Muñoz-Cuza, M. Boehm, and T. B. Pedersen, “CAMEO: Autocorrelation-preserving line simplification for lossy time series compression,” in *Proc. 29th Int. Conf. Extending Database Technol. (EDBT)*, 2026, pp. 15–28, doi: 10.48786/edbt.2026.02.
- [20] C.-C. M. Yeh *et al.*, “Matrix Profile I: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets,” in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, 2016, pp. 1317–1322, doi: 10.1109/ICDM.2016.0179.
- [21] W. S. Cleveland and R. McGill, “Graphical perception: Theory, experimentation, and application to the development of graphical methods,” *J. Amer. Statist. Assoc.*, vol. 79, no. 387, pp. 531–554, 1984, doi: 10.1080/01621459.1984.10478080.
- [22] D. Haehn, J. Tompkin, and H. Pfister, “Evaluating ‘graphical perception’ with CNNs,” *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 1, pp. 641–650, 2019, doi: 10.1109/TVCG.2018.2865138.
- [23] P. Esling and C. Agon, “Time-series data mining,” *ACM Comput. Surv.*, vol. 45, no. 1, Art. no. 12, 2012, doi: 10.1145/2379776.2379788.
- [24] S. Schmidl, P. Wenig, and T. Papenbrock, “Anomaly detection in time series: A comprehensive evaluation,” *Proc. VLDB Endow.*, vol. 15, no. 9, pp. 1779–1797, 2022, doi: 10.14778/3538598.3538602.
- [25] R. Wu and E. J. Keogh, “Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 3, pp. 2421–2429, 2023, doi: 10.1109/TKDE.2021.3112126.
- [26] J. Lin, E. Keogh, L. Wei, and S. Lonardi, “Experiencing SAX: A novel symbolic representation of time series,” *Data Mining Knowl. Discov.*, vol. 15, no. 2, pp. 107–144, 2007, doi: 10.1007/s10618-007-0064-z.
- [27] H. Alt and M. Godau, “Computing the Fréchet distance between two polygonal curves,” *Int. J. Comput. Geom. Appl.*, vol. 5, no. 1–2, pp. 75–91, 1995, doi: 10.1142/S0218195995000064.
- [28] T. Eiter and H. Mannila, “Computing discrete Fréchet distance,” Christian Doppler Lab. Expert Syst., TU Vienna, Vienna, Austria, Tech. Rep. CD-TR 94/64, 1994.

- [29] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, 1993, doi: 10.1109/34.232073.
- [30] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004, doi: 10.1109/TIP.2003.819861.

Farhod Hotamov,
Tashkent University of Information Technologies
Tashkent, Uzbekistan. f.hotamov@tuit.uz

Fakhri Nuraliyev,
Tashkent University of Information Technologies,
Tashkent, Uzbekistan.

Stefan Gubo,
Faculty of Economics and Informatics,
J. Selye University, Slovakia.

Received 10.03.2026, Accepted 24.07.2026, Available online 30.09.2026.