Polynomial-Time Solvability of the Multi-Dimensional Slope Change Problem

Apoorva Narula, Santanu S. Dey
H. Milton Stewart School of Industrial and Systems Engineering (ISyE),
Georgia Institute of Technology, USA
anarula34@gatech.edu, santanu.dey@isye.gatech.edu

Abstract

Slope change detection seeks to identify changes in the trends of time-ordered signals. We study the problem of fitting a multidimensional time series by continuous piecewise-linear functions with a common, continuously varying breakpoint and unrestricted slopes. We consider both the dense setting, in which every dimension may change slope, and the sparse setting, in which at most k dimensions may change slope. Although the resulting optimization problems are nonconvex mixed integer nonlinear programs and may have irrational optimal solutions even for rational input data, we prove that all six variants, corresponding to the ℓ_1, ℓ_2, and ℓ_∞ losses in the dense and sparse settings, are exactly solvable in polynomial time. In particular, optimal solutions admit polynomial-size algebraic representations and can be computed with polynomial bit complexity. For the least-squares loss, we obtain especially efficient algorithms requiring $O(JT)$ operations in the dense setting and $O(J^2T \log J)$ operations in the sparse setting, where T is the number of time points and J is the number of dimensions. We implement these least-squares algorithms and evaluate them on simulated data and on three real-world multidimensional time series, namely, regional surface temperatures (ERA5), hand-tracking data from a mixed-reality headset (HoloAssist), and turbofan engine sensor data (NASA C-MAPSS). The dense algorithm solves instances with up to $J = 10^5$ and $T = 100$, that is, 10^7 observations, in less than 0.1 seconds on a laptop, while the sparse algorithm solves instances with about 10^3 dimensions and up to $T = 240$ in under a second. Compared with a mixed-integer nonlinear programming (MINLP) formulation solved by Gurobi on the same instances, our algorithms are several orders of magnitude faster (approximately 8.9×10^3 to 3.4×10^6 times) on those instances where Gurobi terminates within its default optimality tolerance of 0.01% optimality gap. On larger instances, Gurobi frequently reaches a 10,000-second time limit with substantial optimality gaps.

Keywords: High-dimensional slope change detection; Mixed integer nonlinear program; Irrational solutions; Polynomial-time solvability.

1 Introduction

Slope change detection seeks the location in a time-ordered sequence, called a *breakpoint*, at which the rate of change of an underlying signal changes. Unlike mean-shift detection, which seeks abrupt changes in the signal level, slope change detection considers changes in trend while preserving continuity. Slope-change models have been used in a variety of applications, including surface-temperature trends in climatology [2, 48, 7, 57], rainfall trends [20], changes in health outcomes measured repeatedly over time [17], and sensor-based monitoring [19]. In many applications, multiple signals are observed at each observation time, and a common event, such as a regime shift

or a system fault, may change their trends simultaneously. This motivates fitting a slope change that is shared across the dimensions of a multidimensional time series.

1.1 Problem formulation

Let $T \geq 4$ be an integer, let $x_1 < x_2 < \cdots < x_T$ be rational time coordinates, and let $y_i = (y_i^1, \ldots, y_i^J) \in \mathbb{Q}^J$ be the observation at time x_i, so that y_i^j is the observation of dimension $j \in [J]$ at time x_i. For a breakpoint $z \in \mathbb{R}$, every continuous function that is affine on $(-\infty, z]$ and on $[z, \infty)$ can be written uniquely as

$$x \mapsto \alpha + \beta x + \delta(x - z)_+,$$

where $(t)_+ := \max\{t, 0\}$. We call z the breakpoint of this function. This function has slope β and intercept α before the breakpoint, and slope $\beta + \delta$ and intercept $\alpha - \delta z$ after it. Thus δ is the change in slope, and continuity at z is built into the representation. We fit one such function to each dimension, with a breakpoint common to all dimensions. Given $z \in \mathbb{R}$ and $\alpha, \beta, \delta \in \mathbb{R}^J$, the residual of dimension j at time x_i is

$$r_i^j = y_i^j - \alpha_j - \beta_j x_i - \delta_j (x_i - z)_+, \qquad i \in [T],\ j \in [J], \tag{1}$$

and $r \in \mathbb{R}^{TJ}$ denotes the vector of these TJ residuals. For $p \in \{1, 2, \infty\}$, define the loss

$$L_p(r) := \begin{cases} \sum_{i,j} |r_i^j|, & p = 1, \\ \sum_{i,j} (r_i^j)^2, & p = 2, \\ \max_{i,j} |r_i^j|, & p = \infty. \end{cases}$$

Given data $(x_i, y_i)_{i \in [T]}$ with $x_i \in \mathbb{Q}$ and $y_i \in \mathbb{Q}^J$, and $k \in \{0, 1, \ldots, J\}$, the *slope change problem* is

$$\mathrm{OPT}_{p,k} := \min_{z,\alpha,\beta,\delta,r} \big\{ L_p(r) :\ r \text{ satisfies (1)},\ \|\delta\|_0 \leq k,\ z \in [x_2, x_{T-1}] \big\}. \tag{2}$$

The constraint $\|\delta\|_0 \leq k$ allows at most k dimensions to change slope, and each remaining dimension is fitted by a single line. We refer to the case $k = J$ as the *dense* problem and to the case $k < J$ as the *sparse* problem. The constraint $z \in [x_2, x_{T-1}]$ ensures that at least two observations lie on each side of the breakpoint, since a change in slope cannot be inferred from a segment containing a single observation.

Three features make this problem nontrivial. First, the breakpoint z is a continuous decision variable, and the terms $\delta_j(x_i - z)_+$ introduce nonconvexity. Second, the breakpoint is shared across all dimensions, coupling their fitting problems. Third, the sparse problem introduces the additional combinatorial problem of selecting which dimensions may change slope. Although the problem can be formulated as a mixed-integer nonlinear program [47], we show that its special structure permits polynomial-time algorithms.

1.2 Related literature

Single-dimension time series. Most methods for a single time series allow several breakpoints but restrict the breakpoints to the observation times or a prespecified candidate set. Without continuity, dynamic programming finds exact solutions [38, 5, 39]. With continuity, [25] use functional dynamic programming with an ℓ_0 penalty on the number of slope changes, and [24] allow unevenly spaced observations and more flexible candidate sets. The convex alternative, ℓ_1 trend filtering

[42, 55], penalizes the second differences of the fitted values, so its breakpoints arise implicitly at observation times. Fewer methods treat breakpoint locations as continuous. For one breakpoint under least squares, the setting closest to ours, [37] obtains the exact fit by examining each interval between consecutive observations, with statistical properties studied in [36, 26] and a grid-search alternative proposed by [44]. For several breakpoints, iterative linearization [46] guarantees only local optimality, while mixed-integer programming provides exact formulations [56, 49, 30, 47].

Multiple-dimension time series with shared breakpoints. Here, the breakpoints are common to all dimensions. [40] estimates a break in the intercept, slope, or both of a linear trend, common to all series in a large panel, by minimizing the total sum of squared residuals over all break dates, and [41] estimates the break date jointly with latent common factors in the errors. For shared changes in the mean, the group fused lasso [11] gives a convex formulation. [19] detects, online, a common change point after which a subset of the dimensions moves from a constant mean to a linear drift, with sparsity modeled through the probability that each dimension is affected. Sparse changes in the mean of high-dimensional time series are studied by [58] and [23]. These works differ from our setting in their restrictions on breakpoint locations, assumptions on the form of the slope change, or statistical detection objectives, and none establishes polynomial-time global optimization for the formulation (2). [47] extends their mixed-integer formulations to shared breakpoints. These formulations are exact, but they are not known to be solvable in polynomial time.

Parametric linear programming. [15] optimizes the value of $\min\{c^\top x : (A_0 + \lambda A_1)x \le b,\ x \ge 0\}$ over a scalar λ in an interval, extending sensitivity analysis [28]. They prove that both minimizing and maximizing this value are NP-hard, and identify easier cases where A_1 has entries of one sign or a single nonzero row or column. For $p \in \{1, \infty\}$ and fixed i^*, our dense problem is the minimization version with $\lambda = z$. However, because of the absolute values, A_1 always has entries of both signs, and for $J \ge 2$ it has J nonzero columns, so none of these cases applies. We nevertheless solve the dense and sparse problems exactly in polynomial time, even though the optimal z may be irrational and the sparse problem is not an LP for fixed i^*.

Complexity and connection to ReLU networks. Closely related function recovery problems are NP-hard [60], and [25] notes that it is unclear whether detecting multiple slope changes is NP-hard. We study the setting with a single breakpoint that is shared across dimensions and located continuously. This setting is also the global training of a ReLU network whose input is time, with a single hidden unit $(x-z)_+$ shared by J outputs with affine skip connections, where the breakpoint z is restricted to $[x_2, x_{T-1}]$. Training ReLU networks is NP-hard for other architectures [22, 14, 29], and optimal weights can be irrational [1, 9]. These results do not imply hardness for ours. Exact polynomial-time training is known for one hidden layer with a single output and fixed input dimension [4], but this does not cover multiple outputs.

2 Main complexity results

Throughout, $\langle \mathcal{D} \rangle$ denotes the binary encoding length of the data $\mathcal{D} := (x_i, y_i)_{i \in [T]}$. The existence and representation of rational solutions are important issues in mixed-integer nonlinear programming. For example, mixed-integer quadratic programming admits polynomial-size certificates of feasibility [21], whereas systems of cubic polynomial inequalities with rational data may have only irrational feasible solutions, and deciding the existence of rational solutions is NP-hard [10]. Problem (2) exhibits both phenomena, that is, optimal breakpoints can be irrational, yet we show that

optimal solutions admit polynomial-size algebraic representations and can be computed exactly in polynomial time. To see that optimal breakpoints can be irrational, let $T = 6$, $J = 2$, $x_i = i - 1$ for $i \in [6]$, $y^1 = (-2, 2, -2, -2, 5, 0)$, and $y^2 = (-9, 4, 8, -6, -4, 0)$. Then

$$\mathrm{OPT}_{2,2} = \frac{74531 - \sqrt{477755737}}{420} \approx 125.4128,$$

attained at $z^\star = (-58563+5\sqrt{477755737})/49162 \approx 1.0318$. Since 477755737 is not a perfect square, both $z^\star$ and $\mathrm{OPT}_{2,2}$ are irrational. This irrationality is caused by sharing the breakpoint across dimensions. For $J = 1$ and $p = 2$ a rational optimal breakpoint always exists (see Section 3). We therefore specify what an algorithm returns.

Definition 1. *An algorithm* solves (2) *if it returns an optimal breakpoint $z^\star$, represented as a real algebraic number by a univariate polynomial with rational coefficients together with an interval with rational endpoints that contains $z^\star$ and no other root of the polynomial, and returns $\alpha^\star, \beta^\star, \delta^\star$ as explicit rational functions of $z^\star$ with rational coefficients.*

As we shall see, for $p \in \{2, \infty\}$, our algorithms return an optimal breakpoint $z^\star$ that is algebraic of degree at most two over $\mathbb{Q}$. For $p = 1$, its degree is at most $\max\{2, 2J - 2\}$ in the dense problem and at most $\max\{2, 2k - 2\}$ in the sparse problem. In all cases, the optimal fitting coefficients can be chosen in $\mathbb{Q}(z^\star)$, the smallest subfield of $\mathbb{R}$ containing $\mathbb{Q}$ and $z^\star$, which consists of the values at $z^\star$ of rational functions with rational coefficients.

The encoding length of an algebraic number represented as in Definition 1 is the total binary encoding length of the coefficients of the polynomial and the endpoints of the interval. All rational and algebraic numbers arising in our algorithms admit representations of encoding length polynomial in T, J, and $\langle \mathcal{D} \rangle$. Standard algorithms for real algebraic numbers perform exact arithmetic, root isolation, sign evaluation, and comparison on these representations in polynomial time [6, 51]. Consequently, the optimal breakpoint, objective value, and fitting coefficients can also be approximated to within 2^{-t} in time polynomial in T, J, $\langle \mathcal{D} \rangle$, and t [16, 51].

2.1 Poly-time complexity

Theorem 1. *For every $p \in \{1, 2, \infty\}$ and $k \in \{0, 1, \ldots, J\}$, problem (2) can be solved, in the sense of Definition 1, in time polynomial in T, J, and $\langle \mathcal{D} \rangle$. More precisely, the algorithms perform the arithmetic operations counted in Table 1. All numbers involved are rational or algebraic, with encoding length polynomial in T, J, and $\langle \mathcal{D} \rangle$. For $p \in \{2, \infty\}$, these operations include, and the table counts among them, computations of roots of quadratic polynomials and exact comparisons of quadratic algebraic numbers. For $p = 1$, the algorithm additionally performs root isolations and algebraic-number comparisons, counted in Table 1. All these algebraic operations have polynomial bit complexity.*

When $k = 0$, each dimension is fitted by a single line, and the problem is solved directly. For $p = 2$, the dense algorithm is linear in the number JT of observations, which is optimal up to a constant factor since every observation must be read, and the sparse algorithm is linear in T and nearly quadratic in J. In our experiments, a dense instance with $J = 10^5$ and $T = 100$, that is, 10^7 observations, is solved in less than 0.1 seconds on a laptop, and sparse instances with about 10^3 dimensions are solved in under a second. Section 5 describes the experiments, software, and hardware. We have not tried to optimize the running-time for $p \in \{1, \infty\}$, whose purpose is to establish polynomial-time solvability.

Table 1: Number of operations of the algorithms. Quadratic root computations and comparisons of quadratic algebraic numbers count as single operations. For $p = 1$, the table also lists the root isolations of the polynomials of degree at most $2J - 2$ or $2k - 2$, which are needed only if $J \geq 2$ or $k \geq 2$, and the comparisons of the resulting algebraic numbers. All these operations have polynomial bit complexity (Theorem 1).

	Dense ($k = J$)	**Sparse** ($k < J$)
$p = 2$	$O(JT)$	$O(J^2T \log J)$
$p = \infty$	$O(J^3T^9 \log(JT))$	$O(JT + J \log J + k^3T^9 \log(kT))$
$p = 1$	$O(J^3T^9 + JT^9 \log(JT))$ arithmetic operations; $O(JT^9)$ root isolations, degree $\leq 2J - 2$ ($J \geq 2$); $O(J^2T^9)$ algebraic comparisons	$O((J + k^2)J^3T^9 + JT^9 \log(JT))$ arithmetic operations; $O(J^3T^9)$ root isolations, degree $\leq 2k - 2$ ($k \geq 2$); $O(kJ^3T^9)$ algebraic comparisons

2.2 Outline of the proof of Theorem 1

The three losses, each in the dense and the sparse setting, give six variants of the problem. All six variants are solved by the same six-step strategy described below. The details of each step depend on the variant. The main idea is that, for a fixed breakpoint z, the problem splits into J small fitting problems, one for each dimension. We study how the optimal error of each of these problems varies with z, combine these errors over the dimensions, and reduce the search for z to finitely many candidates.

Step 1: reduction to intervals. The nonlinear terms $(x_i - z)_+$ depend on which observations lie to the right of z. Once we restrict z to an interval between consecutive observations, this set is fixed and each residual term, r_i^j, becomes affine in z. For $i^* \in \{2, \dots, T-2\}$, let $\mathrm{OPT}_{p,k}^{i^*}$ denote the optimal value of (2) with $z \in [x_{i^*}, x_{i^*+1}]$. Since these intervals cover $[x_2, x_{T-1}]$, we have that $\mathrm{OPT}_{p,k} = \min_{i^* \in \{2,\dots,T-2\}} \mathrm{OPT}_{p,k}^{i^*}$. For $z \in [x_{i^*}, x_{i^*+1}]$, including the endpoints, the residuals (1) become

$$r_i^j = y_i^j - \alpha_j - \beta_j x_i - \delta_j h_i(z), \qquad h_i(z) := \begin{cases} 0, & i \leq i^*, \\ x_i - z, & i > i^*. \end{cases}$$

Since $2 \leq i^* \leq T - 2$, each side of the breakpoint contains at least two observations with distinct time coordinates. We show this implies the three columns $(1)_{i\in[T]}$, $(x_i)_{i\in[T]}$, and $(h_i(z))_{i\in[T]}$ of the fitting problem, corresponding to α_j, β_j, and δ_j, are linearly independent for every z, so none of them becomes redundant. We fix i^* for the rest of the outline.

Step 2: decoupling for fixed z. For fixed z, the dimensions interact only through the choice of which of them may change slope. Each dimension is otherwise fitted separately using the three variables $(\alpha_j, \beta_j, \delta_j)$. For $j \in [J]$ and $z \in [x_{i^*}, x_{i^*+1}]$, let $f_j(z)$ be the minimum error when dimension j may change slope at z. This error is $\sum_i |r_i^j|$, $\sum_i (r_i^j)^2$, or $\max_i |r_i^j|$ for $p = 1, 2, \infty$, respectively. Let U_j be the corresponding minimum with $\delta_j = 0$. Thus U_j is the error of the best single line, independent of z, and $f_j(z) \leq U_j$. Then

$$\mathrm{OPT}_{p,k}^{i^*} = \min_{z \in [x_{i^*}, x_{i^*+1}]} \; \min_{S \subseteq [J],\, |S| \leq k} \Big(\bigoplus_{j \in S} f_j(z) \Big) \oplus \Big(\bigoplus_{j \notin S} U_j \Big),$$

where $\oplus$ denotes addition for $p \in \{1, 2\}$ and maximum for $p = \infty$, and an empty aggregation equals zero. Thus the problem reduces to understanding and combining the functions $f_j(\cdot)$.

Step 3: structure of each $f_j(\cdot)$. We establish that $f_j(\cdot)$ is continuous. As we cannot evaluate $f_j(z)$ for every z, we show that each $f_j(\cdot)$ is piecewise rational on $[x_{i^*}, x_{i^*+1}]$, with polynomially many pieces. Each piece has numerator and denominator of degree at most two, and its endpoints are roots of polynomials of degree at most two with rational coefficients.

For $p = 2$, the least-squares projection gives a single piece, $f_j(z) = U_j - \nu_j(z)^2/D(z)$, where $\nu_j(z)$ is affine in z and $D(z)$ is a positive quadratic in z. Crucially, $D(\cdot)$ is common to all dimensions.

For $p \in \{1, \infty\}$, $f_j(z)$ is the optimal value of a linear program. We show that its feasible region is pointed. Since its objective is bounded below by zero, the linear program has an optimal vertex. Since z appears only, and affinely, in the column of δ_j, the Cramer rule shows that every basic solution has entries $(a + bz)/(c + dz)$ with $a, b, c, d \in \mathbb{Q}$. We call such functions linear-fractional. Such a function can be shown to be monotone on every interval that avoids its pole, and two distinct ones cross at most twice. We bound the number of vertices and determine where each is feasible.

- For $p = \infty$, the linear program minimizes η over the four variables $(\alpha_j, \beta_j, \delta_j, \eta)$, subject to the $2T$ constraints $-\eta \le r_i^j \le \eta$ for $i \in [T]$. A vertex is determined by four active constraints, so there are $O(T^4)$ candidate vertices. For each candidate, η is linear-fractional in z, and the candidate is feasible on a union of $O(T)$ intervals. Hence $f_j(\cdot)$ is the minimum of $O(T^5)$ linear-fractional functions, each restricted to one interval.

- For $p = 1$, we prove that an optimal fit can be chosen to pass through three observations, not all on the same side of the breakpoint. Thus, there are $O(T^3)$ choices. Each determines $(\alpha_j, \beta_j, \delta_j)$ uniquely as linear-fractional functions of z. Every such fit is feasible, so, unlike for $p = \infty$, no restriction on z is needed. Its loss $\sum_i |r_i^j|$ is piecewise linear-fractional, with $O(T)$ pieces separated by residual sign changes. Thus $f_j(\cdot)$ is the minimum of $O(T^4)$ linear-fractional functions, each restricted to one interval.

Step 4: aggregation in the dense problem. Here we minimize $\bigoplus_{j\in[J]} f_j(z)$. Each f_j is piecewise rational on $[x_{i^*}, x_{i^*+1}]$, and together their breakpoints partition this interval into cells. On each cell, every f_j is given by a single rational expression whose denominator does not vanish on the cell. We minimize the resulting univariate function on each cell and take the best value over all cells.

- If $\oplus$ is addition, the objective on each cell is a single rational function. We show that its minimum occurs at a cell endpoint or a root of the numerator of its derivative. This numerator has degree at most two for $p = 2$ and at most $2J - 2$ for $p = 1$.

- If $\oplus$ is the maximum, the objective on each cell is the upper envelope of J linear-fractional functions. We show each is monotone on the cell, so a minimum occurs at a cell endpoint or a crossing of two functions.

Step 5: support selection in the sparse problem. Allowing dimension j to change slope gives the *gain* $u_j(z) := U_j - f_j(z) \ge 0$. Our task is to choose at most k dimensions. The optimal choice depends on the aggregation.

First, suppose $\oplus$ is addition. For a set S, the total error is $\sum_{j\in[J]} U_j - \sum_{j\in S} u_j(z)$. For fixed z, an optimal S consists of the k dimensions with the largest gains. The ranking of the gains, however, varies with z, and the set of the k largest gains can change only at crossings of two gains, where their relative order may change. Therefore, between consecutive crossings, S is fixed and we solve the dense problem restricted to S. For $p = 1$, the gains are linear-fractional on each cell of Step 4, so two distinct gains cross at most twice, giving $O(J^2)$ crossings per cell. For $p = 2$, the common

denominator in $u_j(z) = \nu_j(z)^2/D(z)$ means that the ranking is determined by $|\nu_j(z)|$, which gives the subcells on which the ranking is constant.

Next, suppose $\oplus$ is maximum. For $S \subseteq [J]$ with $|S| = k$, the objective is

$$\Phi_S(z) := \max\{\max_{j\in S} f_j(z), \max_{j\notin S} U_j\}.$$

A dimension outside S keeps its single-line error U_j, so the dimensions with the largest U_j should be the ones allowed to change slope. We verify this as follows. Let $S^\star$ be the set of the k dimensions with the largest single-line errors U_j, with arbitrary tie-breaking. This set is independent of z. We claim that $\Phi_{S^\star}(z) \le \Phi_S(z)$ for every z and every S of size k. If $S \neq S^\star$, choose $j \in S^\star \setminus S$ and $j' \in S \setminus S^\star$. Then $U_j \ge U_{j'}$. Swapping j and j' removes U_j and $f_{j'}(z)$ from the maximum and adds $f_j(z) \le U_j$ and $U_{j'} \le U_j$. Thus the maximum cannot increase. Repeated swaps yield $S^\star$, proving the claim. Setting $M := \max_{j\notin S^\star} U_j$, the problem becomes $\min_{z\in[x_{i^*},x_{i^*+1}]} \max\Big\{ \max_{j\in S^\star} f_j(z),\ M\Big\}$, which is the dense problem on $S^\star$, followed by a maximum with the constant M.

Step 6: candidates and exact comparison. Steps 4 and 5 reduce the search to finitely many candidates, namely cell endpoints, stationary points, and crossings. These are algebraic numbers of low degree (see Definition 1), so we evaluate the objective at each and compare the values exactly. Given an optimal breakpoint $z^\star$, solving the fitting problems of Step 2 at $z^\star$ recovers $(\alpha^\star, \beta^\star, \delta^\star)$.

Complexity. Each $f_j(\cdot)$ has one piece for $p = 2$ and $O(T^8)$ pieces for $p \in \{1, \infty\}$. Merging the endpoints of all pieces, and refining by pairwise crossings where needed, gives polynomially many cells. The bounds in Table 1 multiply the number of cells by the cost per cell and by the $T - 3$ intervals of Step 1.

Detailed proofs are presented in Section 3 for the dense case and Section 4 for the sparse case.

3 Polynomial-Time Solvability: Dense Setting

This section and the next carry out Steps 3 to 6 of Section 2.2, with the notation introduced there. Throughout, we fix $i^* \in \{2, \dots, T-2\}$ and write $\mathcal{I} := [x_{i^*}, x_{i^*+1}]$. By Step 1, the algorithm is run once for each of the $T - 3$ values of i^*, and the best of the resulting solutions is returned. Hence, all operation counts below are per interval, and they are multiplied by T in Table 1. For $i \in [T]$ and $z \in \mathbb{R}$, let $a_i(z) := (1, x_i, h_i(z))^\top$, and for $\theta_j = (\alpha_j, \beta_j, \delta_j)^\top$ let $r_i^j(\theta_j, z) := y_i^j - a_i(z)^\top\theta_j$ be the residual of Step 1. For a linear-fractional of the form $(a + bz)/(c + dz)$ with $(c, d) \neq (0, 0)$, its derivative is $(bc - ad)/(c + dz)^2$, whose sign is that of the constant $bc - ad$ wherever $c + dz \neq 0$. Hence it is strictly increasing, strictly decreasing, or constant on every interval on which $c+dz \neq 0$.

3.1 Preliminaries

Lemma 1. *Fix $j \in [J]$ and $p \in \{1, 2, \infty\}$.*

(a) For every $z \in \mathbb{R}$, the vectors $\mathbf{1}$, x, and $h(z) := (h_i(z))_{i\in[T]}$ are linearly independent. Equivalently, the vectors $a_i(z)$, $i \in [T]$, span $\mathbb{R}^3$.

(b) For every $z \in \mathcal{I}$, the minimum defining $f_j(z)$ is attained, and f_j is continuous on $\mathcal{I}$.

Proof. (a) Let $\lambda_1\mathbf{1} + \lambda_2 x + \lambda_3 h(z) = 0$. Since $i^* \ge 2$ and $h_1(z) = h_2(z) = 0$, we have $\lambda_1 + \lambda_2 x_i = 0$ for $i \in \{1, 2\}$, and $x_1 \neq x_2$ gives $\lambda_1 = \lambda_2 = 0$. Since $i^* \le T - 2$, we have $h_i(z) = x_i - z$ for $i \in \{T - 1, T\}$, and these cannot vanish, so $h(z) \neq 0$ and $\lambda_3 = 0$.

(b) Let $\ell_j(\theta, z)$ denote the loss of dimension j, that is, $\sum_i |r_i^j|$, $\sum_i (r_i^j)^2$, or $\max_i |r_i^j|$. We remind the reader that $U_j = \min_{\alpha_j,\beta_j} L_p\big((y_i^j - \alpha_j - \beta_j x_i)_{i\in[T]}\big)$ is the error of the best single line, so $f_j(z) \le U_j$ for every z. Let $\rho := U_j + 1$ for $p \in \{1, \infty\}$ and $\rho := \sqrt{U_j + 1}$ for $p = 2$. Let

$$\Theta := \big\{\theta \in \mathbb{R}^3 :\ \ell_j(\theta, z) \le U_j + 1 \text{ for some } z \in \mathcal{I}\big\},$$

which does not depend on z. Let $\theta \in \Theta$, and let $z \in \mathcal{I}$ satisfy $\ell_j(\theta, z) \le U_j + 1$. Then $|r_i^j(\theta, z)| \le \rho$ for every i. Using $h_1(z) = h_2(z) = 0$, $h_i(z) = x_i - z$ for $i \in \{T-1, T\}$, and

$$r_1^j - r_2^j = y_1^j - y_2^j - \beta(x_1 - x_2), \qquad r_T^j - r_{T-1}^j = y_T^j - y_{T-1}^j - (\beta + \delta)(x_T - x_{T-1}),$$

we obtain $|\beta| \le (|y_1^j - y_2^j| + 2\rho)/(x_2 - x_1)$ and $|\beta + \delta| \le (|y_T^j - y_{T-1}^j| + 2\rho)/(x_T - x_{T-1})$. Since $\alpha = y_1^j - \beta x_1 - r_1^j$, we have $|\alpha| \le |y_1^j| + |\beta||x_1| + \rho$. Hence Θ is bounded, and its closure $\bar{\Theta}$ is compact.

Fix $z \in \mathcal{I}$. We show that the minimum defining $f_j(z)$ is attained. By definition, $f_j(z) = \inf_{\theta\in\mathbb{R}^3} \ell_j(\theta, z)$, and $f_j(z) \le U_j < U_j + 1$. If $\theta \notin \Theta$, then, by the definition of Θ, $\ell_j(\theta, z') > U_j + 1$ for every $z' \in \mathcal{I}$, and in particular $\ell_j(\theta, z) > U_j + 1 > f_j(z)$. Hence the points outside Θ do not affect the infimum, and $f_j(z) = \inf_{\theta\in\Theta} \ell_j(\theta, z)$. Moreover, $\Theta \subseteq \bar{\Theta} \subseteq \mathbb{R}^3$ gives

$$f_j(z) = \inf_{\theta\in\mathbb{R}^3} \ell_j(\theta, z) \le \inf_{\theta\in\bar{\Theta}} \ell_j(\theta, z) \le \inf_{\theta\in\Theta} \ell_j(\theta, z) = f_j(z),$$

so $f_j(z) = \inf_{\theta\in\bar{\Theta}} \ell_j(\theta, z)$.

Each residual $r_i^j(\theta, z) = y_i^j - \alpha - \beta x_i - \delta h_i(z)$ is continuous in (θ, z). Since ℓ_j is a sum of absolute values ($p = 1$), a sum of squares ($p = 2$), or a maximum of absolute values ($p = \infty$) of these residuals, ℓ_j is continuous on $\mathbb{R}^3 \times \mathcal{I}$. In particular, $\ell_j(\cdot, z)$ is continuous, and since $\bar{\Theta}$ is compact and nonempty, the infimum is attained, that is, $f_j(z) = \min_{\theta\in\bar{\Theta}} \ell_j(\theta, z)$. As ℓ_j is continuous on the compact set $\bar{\Theta} \times \mathcal{I}$, $f_j(\cdot)$ is continuous on $\mathcal{I}$, by Berge's maximum theorem [8] (Chapter VI, Section 3). □

For $p \in \{1, \infty\}$, f_j is the pointwise minimum of finitely many linear-fractional functions, each available on a part of $\mathcal{I}$. The next lemma turns such a minimum into a piecewise linear-fractional function. It is also used to aggregate the dimensions for $p = \infty$.

Lemma 2. *Let $\mathcal{C} = [l, u]$ with $l < u$, where l and u are rational or roots of quadratic polynomials with rational coefficients. For $k \in [m]$, let $\varphi_k = (a_k + b_k z)/(c_k + d_k z)$ have rational coefficients. Let $V_k \subseteq \mathcal{C}$ be the set of points at which φ_k takes part in the minimum below. Assume the following.*

(i) Each V_k is a union of at most r intervals. Each interval may be open, closed, half-open, or a single point.

(ii) Each endpoint of these intervals is rational, equal to l or u, or a root of a quadratic polynomial with rational coefficients.

(iii) For every k, $c_k + d_k z \neq 0$ for all $z \in V_k$.

(iv) $V_1 \cup \cdots \cup V_m = \mathcal{C}$, and $\varphi(z) := \min_{k\in[m]:\, z\in V_k} \varphi_k(z)$ is continuous on $\mathcal{C}$.

Then there exist points $l = \zeta_0 < \zeta_1 < \cdots < \zeta_P = u$ with $P \le 2mr + m^2$, and linear-fractional functions $\psi_1, \ldots, \psi_P$ with rational coefficients, with the following properties. For every $\ell \in [P]$, the denominator of ψ_ℓ has no zero in $[\zeta_{\ell-1}, \zeta_\ell]$, and $\varphi = \psi_\ell$ on $[\zeta_{\ell-1}, \zeta_\ell]$. Each ζ_ℓ is l, u, an endpoint of an interval of some V_k, or a root of a nonzero polynomial of degree at most two with

rational coefficients. Given the φ_k and the endpoints of the intervals of the V_k, the points ζ_ℓ and the functions ψ_ℓ are computed with $O((m^2+mr)\log(m+r))$ operations. The same holds with the minimum replaced by the maximum.

Proof. For $k \neq k'$, let $\pi_{kk'} := (a_k + b_k z)(c_{k'} + d_{k'} z) - (a_{k'} + b_{k'} z)(c_k + d_k z)$, a polynomial of degree at most two. Wherever both functions are defined, we have

$$\varphi_k(z) - \varphi_{k'}(z) = \frac{\pi_{kk'}(z)}{(c_k + d_k z)(c_{k'} + d_{k'} z)}.$$

Let E be the set of points of (l, u) that are endpoints of the intervals forming the sets V_k, or roots of a polynomial $\pi_{kk'} \not\equiv 0$. Then $|E| \leq 2mr + m(m-1)$. Let $\zeta_0 < \cdots < \zeta_P$ be the points of $E \cup \{l, u\}$, where points that coincide are listed once, which is detected by exact comparisons as in Remark 1, so that $P \leq |E| + 1 \leq 2mr + m^2$, and consider an open cell $C_\ell := (\zeta_{\ell-1}, \zeta_\ell)$. No endpoint of an interval of V_k lies in C_ℓ, so either $C_\ell \subseteq V_k$ or $C_\ell \cap V_k = \emptyset$. Let $A_\ell := \{k : C_\ell \subseteq V_k\}$. Since $V_1 \cup \cdots \cup V_m = \mathcal{C}$, the set A_ℓ is nonempty.

Let $k, k' \in A_\ell$. Then $C_\ell \subseteq V_k \cap V_{k'}$, so both denominators have no zero in C_ℓ by the assumption on V_k and $V_{k'}$. If $\pi_{kk'} \equiv 0$, then $\varphi_k = \varphi_{k'}$ on C_ℓ. Otherwise, $\pi_{kk'}$ has no zero in C_ℓ, since all its roots in (l, u) belong to E. Hence $\varphi_k - \varphi_{k'}$ is continuous and nonzero on C_ℓ, and by the intermediate value theorem it has a constant nonzero sign on C_ℓ. Order A_ℓ by $k \prec k'$ if $\varphi_k < \varphi_{k'}$ on C_ℓ, or if $\varphi_k = \varphi_{k'}$ on C_ℓ and $k < k'$. This order does not depend on $z \in C_\ell$, so its first element k_ℓ satisfies $\varphi = \varphi_{k_\ell}$ on C_ℓ.

We now extend the formula $\varphi = \varphi_{k_\ell}$ from C_ℓ to its closure $[\zeta_{\ell-1}, \zeta_\ell]$. If $b_{k_\ell} c_{k_\ell} = a_{k_\ell} d_{k_\ell}$, then (a_{k_ℓ}, b_{k_ℓ}) is a multiple of (c_{k_ℓ}, d_{k_ℓ}), so φ_{k_ℓ} is a rational constant on its domain, and we let ψ_ℓ be this constant. Otherwise, φ_{k_ℓ} is nonconstant, and we let $\psi_\ell := \varphi_{k_\ell}$. The denominator $c_{k_\ell} + d_{k_\ell} z$ of ψ_ℓ has no zero in C_ℓ, since $C_\ell \subseteq V_{k_\ell}$. We show that it has no zero at $\zeta_{\ell-1}$ or ζ_ℓ either. Since φ is continuous on the compact interval $[l, u]$, it is bounded. As $\psi_\ell = \varphi$ on C_ℓ, the function ψ_ℓ is bounded on C_ℓ. A nonconstant linear-fractional function, however, is unbounded near a zero of its denominator. Hence the denominator of ψ_ℓ does not vanish at $\zeta_{\ell-1}$ or ζ_ℓ. Thus, in both cases, ψ_ℓ and φ are continuous on $[\zeta_{\ell-1}, \zeta_\ell]$ and agree on C_ℓ, so they agree on $[\zeta_{\ell-1}, \zeta_\ell]$.

It remains to compute k_ℓ for all ℓ, and for this we compare neighboring cells. For $\ell \in [P-1]$, call an index k *involved at* ζ_ℓ if ζ_ℓ is an endpoint of an interval of V_k or a root of $\pi_{kk'} \not\equiv 0$ for some k', and let E_{ζ_ℓ} be the set of these indices. Indices that are not involved at ζ_ℓ are unaffected when passing from C_ℓ to $C_{\ell+1}$. Indeed, let $k \notin E_{\zeta_\ell}$. Since ζ_ℓ is not an endpoint of an interval of V_k, either ζ_ℓ is an interior point of an interval of V_k, and then $k \in A_\ell \cap A_{\ell+1}$ and $\zeta_\ell \in V_k$, or $k \notin A_\ell \cup A_{\ell+1}$. If, moreover, $k' \notin E_{\zeta_\ell}$ and $k, k' \in A_\ell$, then both denominators are nonzero at $\zeta_\ell \in V_k \cap V_{k'}$, and either $\pi_{kk'} \equiv 0$ or $\pi_{kk'}(\zeta_\ell) \neq 0$. Hence $\varphi_k - \varphi_{k'}$ is identically zero or has the same sign on C_ℓ and $C_{\ell+1}$, so the relative order of k and k' is the same on both cells.

This suggests the following sweep. Sort E, which takes $O(|E| \log |E|)$ operations, and visit the cells from left to right, keeping the indices of the current set A_ℓ in a balanced binary search tree ordered by $\prec$. When the sweep passes ζ_ℓ, delete from the tree the indices of E_{ζ_ℓ} that it contains, and insert those indices of E_{ζ_ℓ} that belong to $A_{\ell+1}$, each at its position on $C_{\ell+1}$. By the observation above, the tree then contains $A_{\ell+1}$ ordered by $\prec$ on $C_{\ell+1}$, and its first element is $k_{\ell+1}$. Each comparison of two indices k, k' during an insertion is decided by the index if $\pi_{kk'} \equiv 0$, and otherwise by the signs on $C_{\ell+1}$ of $\pi_{kk'}$ and of the two denominators. Each of these is a nonzero polynomial g of degree at most two with no zero in $C_{\ell+1}$. By the exact Taylor expansion $g(\zeta_\ell + t) = g(\zeta_\ell) + g'(\zeta_\ell)t + \frac{1}{2}g''(\zeta_\ell)t^2$, its sign on $C_{\ell+1}$ is the sign of the first nonzero number among $g(\zeta_\ell)$, $g'(\zeta_\ell)$, and $g''(\zeta_\ell)$. Each index k is involved at no more than the $2r$ endpoints of the intervals of V_k and the $2(m-1)$ roots of the polynomials $\pi_{kk'}$. Hence $\sum_\ell |E_{\zeta_\ell}| \leq 2mr + 2m(m-1)$,

and the sweep, including the construction of the initial tree for C_1, takes $O((m^2+mr)\log(m+r))$ operations (see [31] for the runtime guarantees of balanced binary search tree operations). For the maximum, the same argument applies with the order of the values reversed. □

Remark 1. *Two numbers $a+b\sqrt{\Delta}$ and $a'+b'\sqrt{\Delta'}$ with $a,b,a',b',\Delta,\Delta'\in\mathbb{Q}$ and $\Delta,\Delta'\geq 0$ can be compared exactly, by sign tests on rational numbers after squaring at most twice, in time polynomial in the encoding lengths of a,b,a',b',Δ,Δ'.*

3.2 The ℓ_2 loss

Let $A := [\mathbf{1}\ x]\in\mathbb{Q}^{T\times 2}$ and $G := A^\top A$, which is invertible since the x_i are distinct. Let $M := I_T - AG^{-1}A^\top$ be the orthogonal projector onto the orthogonal complement of the column space of A. Write $h(z) = v - zw$, where $v, w\in\mathbb{Q}^T$ are given by

$$v_i := \begin{cases} 0, & i\leq i^*,\\ x_i, & i>i^*,\end{cases} \qquad w_i := \begin{cases} 0, & i\leq i^*,\\ 1, & i>i^*.\end{cases}$$

Lemma 3 establishes the structure of f_j, which completes Step 3 for $p=2$.

Lemma 3. *For every $j\in[J]$ and $z\in\mathbb{R}$,*

$$f_j(z) = U_j - \frac{\nu_j(z)^2}{D(z)}, \qquad U_j = (y^j)^\top M y^j,$$

where

$$\nu_j(z) := (y^j)^\top M h(z) = s_{j,0} + s_{j,1}z, \qquad D(z) := h(z)^\top M h(z) = d_0 + d_1 z + d_2 z^2,$$

with $s_{j,0} := (y^j)^\top M v$, $s_{j,1} := -(y^j)^\top M w$, $d_0 := v^\top M v$, $d_1 := -2v^\top M w$, and $d_2 := w^\top M w$. Moreover, $D(z)>0$ for every $z\in\mathbb{R}$, and the minimum defining $f_j(z)$ is attained at

$$\delta_j = \frac{\nu_j(z)}{D(z)}, \qquad (\alpha_j,\beta_j)^\top = G^{-1}A^\top\big(y^j - \delta_j h(z)\big).$$

Proof. For fixed δ_j, the minimum of $\|y^j - \delta_j h(z) - A(\alpha_j,\beta_j)^\top\|_2^2$ over (α_j,β_j) is attained at $G^{-1}A^\top(y^j-\delta_j h(z))$, with value $\|M(y^j-\delta_j h(z))\|_2^2$. Since M is symmetric and $M^2=M$, this value equals $U_j - 2\delta_j\nu_j(z) + \delta_j^2 D(z)$, where $U_j = (y^j)^\top M y^j$ is the value for $\delta_j = 0$ (that is, the error of the best single line). By Lemma 1(a), $h(z)$ is not in the column space of A, so $Mh(z)\neq 0$ and $D(z) = \|Mh(z)\|_2^2 > 0$. Minimizing the quadratic $U_j - 2\delta_j\nu_j(z) + \delta_j^2 D(z)$ in δ_j gives the claim. □

Lemma 4. *Let D be as in Lemma 3, a polynomial of degree at most two with $D>0$ on $\mathbb{R}$. Let N be a polynomial of degree at most two, let $c\in\mathbb{R}$, and let $F := c - N/D$. Then $R := N'D - ND'$ has degree at most two, and $F' = -R/D^2$ on $\mathbb{R}$. Consequently, for every interval $[l,u]$, the minimum of F over $[l,u]$ is attained either at l, or at u, or, if $R\not\equiv 0$, at a root of R in (l,u).*

Proof. If $N = n_0 + n_1 z + n_2 z^2$, the coefficient of z^3 is $2n_2 d_2$ in both $N'D$ and ND', so $\deg R\leq 2$. Since $D>0$, F is differentiable on $\mathbb{R}$ with $F' = -R/D^2$. The continuous function F attains its minimum over $[l,u]$. At an interior minimizer, $F'=0$, so $R=0$. If $R\equiv 0$, then F is constant, and l is a minimizer. □

Aggregation and candidates (Step 4). By Step 2, $\mathrm{OPT}_{2,J}^{i^*} = \min_{z\in\mathcal{I}} F(z)$, where, by Lemma 3,

$$F(z) := \sum_{j\in[J]} f_j(z) = C - \frac{N(z)}{D(z)}, \qquad C := \sum_{j\in[J]} U_j, \qquad N(z) := \sum_{j\in[J]} \nu_j(z)^2.$$

The polynomial $N = n_0 + n_1 z + n_2 z^2$, with $n_0 = \sum_j s_{j,0}^2$, $n_1 = 2\sum_j s_{j,0}s_{j,1}$, and $n_2 = \sum_j s_{j,1}^2$, has degree at most two irrespective of J. By Lemma 4, an optimal breakpoint on $\mathcal{I}$ lies in the set consisting of x_{i^*}, x_{i^*+1}, and at most two roots in (x_{i^*}, x_{i^*+1}) of the quadratic $R = r_0 + r_1 z + r_2 z^2$ with rational coefficients. Hence every candidate lies in $\mathbb{Q}(\sqrt{\Delta})$ with $\Delta := r_1^2 - 4r_0r_2 \in \mathbb{Q}$, and so does its value $C - N(z)/D(z)$. By Remark 1, the values at the at most four candidates are compared exactly in polynomial time, which yields an optimal breakpoint on $\mathcal{I}$.

Remark 2. *For $J = 1$, this procedure is closely related to the classical approach of [37], which also examines each interval between consecutive observations. For $J \geq 2$, the key point is that D depends only on the time coordinates, so N has degree at most two for every J, which is what makes the bound $O(JT)$ possible. Sharing the breakpoint does change the arithmetic, however. For $J = 1$, we have $R = \nu_1 \cdot (2\nu_1' D - \nu_1 D')$, and $2\nu_1' D - \nu_1 D' = (2s_{1,1}d_0 - s_{1,0}d_1) + (s_{1,1}d_1 - 2s_{1,0}d_2)z$ is affine. Hence every candidate is rational, and a rational optimal breakpoint exists, whereas for $J = 2$ the optimal breakpoint can be irrational (Section 2).*

Recovering the solution. Given an optimal breakpoint $z^\star$, Lemma 3 gives $\delta_j^\star = \nu_j(z^\star)/D(z^\star)$ and $(\alpha_j^\star, \beta_j^\star)^\top = G^{-1}A^\top(y^j - \delta_j^\star h(z^\star))$, which lie in $\mathbb{Q}(z^\star)$. This takes $O(JT)$ operations.

Operation count. Hence the coefficients d_0, d_1, d_2 of D, the coefficients $s_{j,0}, s_{j,1}$ of ν_j, and U_j are determined by G, the vectors $A^\top y^j = (\sum_i y_i^j, \sum_i x_i y_i^j)$, the numbers $\sum_i (y_i^j)^2$, and the suffix sums

$$\sum_{i>i^*} x_i, \qquad \sum_{i>i^*} x_i^2, \qquad \sum_{i>i^*} y_i^j, \qquad \sum_{i>i^*} x_i y_i^j,$$

together with $T - i^*$, since $A^\top v = (\sum_{i>i^*} x_i, \sum_{i>i^*} x_i^2)$, $A^\top w = (T - i^*, \sum_{i>i^*} x_i)$, $v^\top v = \sum_{i>i^*} x_i^2$, $v^\top w = \sum_{i>i^*} x_i$, $w^\top w = T - i^*$, $(y^j)^\top v = \sum_{i>i^*} x_i y_i^j$, and $(y^j)^\top w = \sum_{i>i^*} y_i^j$. The quantities that do not depend on i^* are computed once with $O(JT)$ operations. The suffix sums are updated from one value of i^* to the next with $O(J)$ operations, after which D, the $s_{j,0}, s_{j,1}$, N, and R are formed with $O(J)$ operations. Each interval then requires one quadratic root computation and a constant number of comparisons. Over all intervals, and with the recovery step, this gives $O(JT)$ operations.

3.3 The ℓ_∞ loss

Fix j and we drop the superscript j for simplicity. For $z \in \mathcal{I}$, $f(z)$ is the optimal value of the linear program

$$\min_{\theta\in\mathbb{R}^3,\, \eta\in\mathbb{R}} \Big\{\eta :\ \sigma\big(y_i - a_i(z)^\top\theta\big) \leq \eta \ \text{ for all } (i,\sigma) \in [T]\times\{-1,1\}\Big\}. \tag{3}$$

Call a set $B \subseteq [T]\times\{-1,1\}$ with $|B| = 4$ a *basis*, and let $K_B(z) \in \mathbb{R}^{4\times 4}$ be the matrix with rows $(a_i(z)^\top, \sigma)$ for $(i,\sigma) \in B$. The constraint (i,σ) holds with equality if and only if $a_i(z)^\top\theta + \sigma\eta = y_i$.

Lemma 5. *(a) For every basis B, $\det K_B(z)$ is an affine function of z. If $\det K_B \not\equiv 0$, let $(\theta_B(z), \eta_B(z))$ be the solution of $a_i(z)^\top\theta + \sigma\eta = y_i$, $(i,\sigma) \in B$. Then η_B is linear-fractional with rational coefficients and denominator $\det K_B$, and, for every (i,σ), the slack $\eta_B(z) - \sigma r_i(\theta_B(z), z)$ is an affine function with rational coefficients divided by $\det K_B(z)$.*

(b) For each basis B with $\det K_B \not\equiv 0$, let

$$V_B := \{t \in \mathcal{I} : \ \det K_B(t) \neq 0 \text{ and all } 2T \text{ slacks of } (\theta_B(t), \eta_B(t)) \text{ are nonnegative}\},$$

the set of breakpoints at which B defines a feasible vertex of (3). Each V_B is a union of at most $4T+5$ intervals with rational endpoints. Moreover, for every fixed $z \in \mathcal{I}$,

$$f(z) = \min\{\eta_B(z) : \ B \text{ a basis with } z \in V_B\}.$$

Proof. (a) Only the third column of $K_B(z)$ depends on z, and its entries $h_i(z)$ are affine in z. Expanding along this column shows that $\det K_B(z)$ is affine in z. By Cramer's rule, each entry of $(\theta_B(z), \eta_B(z))$ is a determinant divided by $\det K_B(z)$. For δ_B, the third column is replaced by the right-hand side, so the numerator is a rational constant κ. For α_B, β_B, and η_B, the numerator still contains the third column, so it is affine. Hence η_B is linear-fractional. Moreover, $\delta_B(z)h_i(z) = \kappa h_i(z)/\det K_B(z)$, so every residual $r_i(\theta_B(z), z) = y_i - \alpha_B(z) - \beta_B(z)x_i - \delta_B(z)h_i(z)$, and therefore every slack, is an affine function divided by $\det K_B(z)$.

(b) By Lemma 1(a), the vectors $a_i(z)$ span $\mathbb{R}^3$. Since the rows $(a_i(z)^\top, 1)$ and $(-a_i(z)^\top, 1)$ of the constraints of (3) add up to $(0,0,0,2)$, the constraint matrix has rank four. Hence the feasible region contains no line, that is it is pointed. As (3) is feasible and objective bounded below by 0, it has an optimal vertex [53]. A vertex is the unique solution of four linearly independent active constraints. Thus it equals $(\theta_B(z), \eta_B(z))$ for a basis B with $\det K_B(z) \neq 0$, and then $z \in V_B$. Conversely, for $z \in V_B$, the point $(\theta_B(z), \eta_B(z))$ is feasible, so $\eta_B(z) \geq f(z)$.

By (a), $\det K_B$ is an affine function with rational coefficients, and it is not identically zero, so it has at most one zero, which is rational. Also by (a), each of the $2T$ slacks has the form $n(t)/\det K_B(t)$, where n is an affine function with rational coefficients. If $n \not\equiv 0$, then n has at most one zero, which is rational. If $n \equiv 0$, the slack is zero wherever it is defined and never changes sign. Hence the points at which $\det K_B$ or some slack numerator vanishes are at most $2T+1$ rational points. They split $\mathcal{I}$ into at most $2T+2$ open intervals, on each of which all signs are constant, and at most $2T+3$ remaining points, namely these zeros and the two endpoints of $\mathcal{I}$. Hence V_B is a union of at most $(2T+2)+(2T+3) = 4T+5$ intervals, some of which may be single points. □

Structure of f_j (Step 3). We discard the bases with $\det K_B \equiv 0$. We apply Lemma 2 on $[l,u] = \mathcal{I}$, whose endpoints are rational, to the functions η_B on the sets V_B, with $m \leq \binom{2T}{4} = O(T^4)$ and $r \leq 4T+5$. Its hypotheses hold for the following reasons. By Lemma 5(a), each η_B is linear-fractional with rational coefficients and denominator $\det K_B$, which is nonzero on V_B by the definition of V_B. By Lemma 5(b), each V_B is a union of at most $4T+5$ intervals with rational endpoints, the sets V_B cover $\mathcal{I}$, and $f_j(z) = \min_{B:\, z \in V_B} \eta_B(z)$. By Lemma 1(b), f_j is continuous on $\mathcal{I}$. Hence f_j is piecewise linear-fractional with $O(T^8)$ pieces. On each piece, f_j is a linear-fractional function with rational coefficients whose denominator has no zero on the closed piece. The endpoints of the pieces are rational or roots of quadratic polynomials with rational coefficients. Forming the bases and the sets V_B takes $O(T^5 \log T)$ operations, and Lemma 2 takes $O(T^8 \log T)$ operations.

Aggregation and candidates (Step 4). By Step 2, $\mathrm{OPT}^{i^*}_{\infty,J} = \min_{z \in \mathcal{I}} F(z)$ with $F := \max_{j \in [J]} f_j$. We merge the endpoints of the pieces of all f_j, which partitions $\mathcal{I}$ into $O(JT^8)$ closed cells. On a cell $\mathcal{C}$, each f_j is a linear-fractional function ψ_j with rational coefficients whose denominator has no zero on $\mathcal{C}$, and F is continuous. Lemma 2, applied to the maximum with $m = J$, $r = 1$,

and $V_j = \mathcal{C}$, splits $\mathcal{C}$ into $O(J^2)$ closed subcells, on each of which F equals one linear-fractional function whose denominator has no zero on the subcell.

On a subcell $[\alpha, \beta]$, we have $F(z) = \frac{az+b}{cz+d}$ with $cz + d \neq 0$ for all $z \in [\alpha, \beta]$. Since $cz + d$ is continuous and has no zero on $[\alpha, \beta]$, it has constant sign on $[\alpha, \beta]$. Hence, for all $z, w \in [\alpha, \beta]$,

$$F(z) - F(w) = \frac{(az+b)(cw+d) - (aw+b)(cz+d)}{(cz+d)(cw+d)} = \frac{(ad-bc)(z-w)}{(cz+d)(cw+d)},$$

$$(cz+d)(cw+d) > 0.$$

Thus F is nondecreasing on $[\alpha, \beta]$ if $ad - bc \geq 0$ and nonincreasing on $[\alpha, \beta]$ if $ad - bc \leq 0$. In either case,

$$\min_{z \in [\alpha,\beta]} F(z) = \min\{F(\alpha), F(\beta)\}.$$

Hence an optimal breakpoint is among the endpoints of the subcells, and it remains to select an endpoint with the smallest value of F. Each endpoint lies in $\mathbb{Q}(\sqrt{\Delta})$ for some $\Delta \in \mathbb{Q}$, and so does the value of F at it, since F is linear-fractional with rational coefficients on the subcell. By Remark 1, these values are compared exactly, so an optimal breakpoint is found exactly in polynomial time.

Recovering the solution. Given an optimal breakpoint $z^\star$ and $j \in [J]$, Lemma 5(b) gives an optimal vertex $(\theta_B(z^\star), \eta_B(z^\star))$, where B minimizes $\eta_B(z^\star)$ over the $O(T^4)$ bases with $z^\star \in V_B$. By Cramer's rule, its entries lie in $\mathbb{Q}(z^\star)$. This takes $O(JT^5)$ operations.

Operation count. For each interval, computing the pieces of all f_j takes $O(JT^8 \log T)$ operations, and sorting their endpoints takes $O(JT^8 \log(JT))$ operations. Each of the $O(JT^8)$ cells takes $O(J^2 \log J)$ operations, and each of the resulting $O(J^3T^8)$ candidates is evaluated with $O(1)$ operations. This gives $O(J^3T^8 \log(JT))$ operations per interval, and $O(J^3T^9 \log(JT))$ operations in total.

3.4 The ℓ_1 loss

Fix $j \in [J]$. For convenience, we drop the superscript j. For $z \in \mathcal{I}$, $f(z)$ is the optimal value of the linear program

$$\min_{\theta \in \mathbb{R}^3,\, e \in \mathbb{R}^T} \Big\{ \sum_{i \in [T]} e_i : \ \sigma\big(y_i - a_i(z)^\top \theta\big) \leq e_i \ \text{ for all } (i, \sigma) \in [T] \times \{-1, 1\} \Big\}. \tag{4}$$

For $Q \subseteq [T]$ with $|Q| = 3$, let $A_Q(z) \in \mathbb{R}^{3\times 3}$ have rows $a_i(z)^\top$, $i \in Q$. If $\det A_Q(z) \neq 0$, let $\theta_Q(z) := A_Q(z)^{-1}(y_i)_{i \in Q}$ be the unique fit with zero residual at the observations in Q, and let $L_Q(z) := \sum_{i \in [T]} |r_i(\theta_Q(z), z)|$ be its loss.

Lemma 6. *(a) If $\det A_Q(z) \neq 0$, then Q contains an index $i \leq i^*$ and an index $i > i^*$.*

(b) For $z \in \mathcal{I}$, $f(z) = \min\{L_Q(z) : |Q| = 3,\ \det A_Q(z) \neq 0\}$.

(c) If $\det A_Q(z) \not\equiv 0$, then $\det A_Q(z)$ is affine in z, and $r_i(\theta_Q(z), z) = n_{Q,i}(z)/\det A_Q(z)$ for affine functions $n_{Q,i}$ with rational coefficients. Consequently, $\det A_Q$ has at most one zero, and each of the at most $T - 3$ functions $n_{Q,i} \not\equiv 0$, $i \notin Q$, has at most one zero. These at most $T - 2$ points split $\mathcal{I}$ into at most $T - 1$ closed intervals. On each of them, with the zero of $\det A_Q$ removed, $L_Q = \big(\sum_i \sigma_i n_{Q,i}\big)/(\sigma_0 \det A_Q)$ for fixed signs $\sigma_0, \sigma_i \in \{-1, 1\}$, which is linear-fractional.

Proof. (a) If $Q \subseteq [i^*]$, the third column of $A_Q(z)$ is zero. If $Q \cap [i^*] = \emptyset$, the third column equals the second column minus z times the first. In both cases, $A_Q(z)$ is singular.

(b) We show that $f(z) = \min\{L_Q(z) : \ Q \subseteq [T],\ |Q| = 3,\ \det A_Q(z) \neq 0\}$.

Step 1: $f(z) \leq L_Q(z)$ for every such Q. The point $(\theta_Q(z), (|r_i(\theta_Q(z), z)|)_i)$ is feasible for (4), and its objective value is $L_Q(z)$. Hence $f(z) \leq L_Q(z)$.

Step 2: $f(z) = L_Q(z)$ for some such Q. We first show that (4) has an optimal vertex. Every feasible point satisfies $e_i \geq |r_i(\theta, z)| \geq 0$ for all i. Let $(\theta, e) + t(\theta', e')$, $t \in \mathbb{R}$, be a line in the feasible region. Since $e + te' \geq 0$ for all $t \in \mathbb{R}$, we have $e' = 0$. Then $|y_i - a_i(z)^\top \theta - t\, a_i(z)^\top \theta'| \leq e_i$ for all $t \in \mathbb{R}$ and all i, so $a_i(z)^\top \theta' = 0$ for all i. By Lemma 1(a), the vectors $a_i(z)$, $i \in [T]$, span $\mathbb{R}^3$, so $\theta' = 0$. Hence the feasible region contains no line. As (4) is feasible and its objective is bounded below by 0, it has an optimal vertex $(\bar\theta, \bar e)$ [53].

We next count the active constraints at $(\bar\theta, \bar e)$. By optimality, $\bar e_i = |r_i(\bar\theta, z)|$ for every i, since otherwise $\bar e_i$ could be decreased. Let $\bar Q := \{i : r_i(\bar\theta, z) = 0\}$. For $i \notin \bar Q$, exactly one of the two constraints of i is active. For $i \in \bar Q$, both are active, and their rows $(a_i(z)^\top, \mathbf{1}_i^\top)$ and $(-a_i(z)^\top, \mathbf{1}_i^\top)$ span the same space as $(a_i(z)^\top, 0)$ and $(0, \mathbf{1}_i^\top)$, where $\mathbf{1}_i$ is the ith unit vector in $\mathbb{R}^T$. Hence the rank of the active constraints is at most

$$\operatorname{rank}\{a_i(z) : i \in \bar Q\} + |\bar Q| + (T - |\bar Q|) = \operatorname{rank}\{a_i(z) : i \in \bar Q\} + T.$$

At a vertex of a polyhedron in $\mathbb{R}^{T+3}$, this rank equals $T + 3$. Therefore the vectors $a_i(z)$, $i \in \bar Q$, span $\mathbb{R}^3$, and some $Q \subseteq \bar Q$ with $|Q| = 3$ has $\det A_Q(z) \neq 0$. Since $a_i(z)^\top \bar\theta = y_i$ for $i \in Q$, we get $\bar\theta = \theta_Q(z)$, and therefore $L_Q(z) = \sum_i |r_i(\bar\theta, z)| = \sum_i \bar e_i = f(z)$.

Steps 1 and 2 together prove the claim.

(c) As in the proof of Lemma 5(a), $\det A_Q(z)$ is affine, and by Cramer's rule, δ_Q is a rational constant divided by $\det A_Q(z)$, while α_Q and β_Q are affine functions divided by $\det A_Q(z)$. Hence each residual is an affine function $n_{Q,i}$ divided by $\det A_Q(z)$, with $n_{Q,i} \equiv 0$ for $i \in Q$. Neither $\det A_Q$ nor any $n_{Q,i} \not\equiv 0$ vanishes in the interior of one of the closed intervals. At an endpoint, the $n_{Q,i}$ that vanish contribute zero with either sign. This gives the formula. □

Structure of f_j (Step 3). By Lemma 6(b) and (c), f_j is the minimum of the $m \leq \binom{T}{3}(T - 1) = O(T^4)$ linear-fractional functions of Lemma 6(c), each available on a single interval. By Lemmas 1(b) and 2 with $r = 1$, f_j is piecewise linear-fractional with $O(T^8)$ pieces. On each piece, f_j is a linear-fractional function with rational coefficients whose denominator has no zero on the closed piece. The endpoints of the pieces are rational or roots of quadratic polynomials with rational coefficients. Forming the functions takes $O(T^4 \log T)$ operations, and Lemma 2 takes $O(T^8 \log T)$ operations.

Aggregation and candidates (Step 4). By Step 2, $\mathrm{OPT}_{1,J}^{i^*} = \min_{z \in \mathcal{I}} F(z)$ with $F := \sum_{j \in [J]} f_j$. Merging the endpoints of the pieces of all f_j partitions $\mathcal{I}$ into $O(JT^8)$ closed cells. On a cell $\mathcal{C}$, $f_j = (a_j + b_j z)/(c_j + d_j z)$ with rational coefficients and $c_j + d_j z \neq 0$ on $\mathcal{C}$. Hence $F = P/Q$ on $\mathcal{C}$, where

$$Q(z) := \prod_{j \in [J]} (c_j + d_j z), \qquad P(z) := \sum_{j \in [J]} (a_j + b_j z) \prod_{h \in [J] \setminus \{j\}} (c_h + d_h z),$$

both of degree at most J, and $Q \neq 0$ on $\mathcal{C}$. Then $F' = H/Q^2$ with $H := P'Q - PQ'$. If $P = \sum_{t \leq J} p_t z^t$ and $Q = \sum_{t \leq J} q_t z^t$, the coefficient of z^{2J-1} in H is $J p_J q_J - J p_J q_J = 0$, so $\deg H \leq 2J - 2$. As in the proof of Lemma 4, the minimum of F over $\mathcal{C}$ is attained at an endpoint of $\mathcal{C}$ or, if $H \not\equiv 0$, at a real root of H in the interior of $\mathcal{C}$. We isolate these roots and compare the

values of F at all candidates exactly. Every candidate is rational, a root of a quadratic polynomial with rational coefficients, or a root of H, which has rational coefficients and degree at most $2J-2$. Hence every candidate and every value of F at a candidate is a real algebraic number of degree at most $2J-2$ whose representation has encoding length polynomial in T, J, and $\langle\mathcal{D}\rangle$, and such numbers are compared exactly in polynomial time, even when their degrees differ [6, 51].

Recovering the solution. Given an optimal breakpoint $z^\star$ and $j \in [J]$, Lemma 6(b) gives an optimal fit $\theta_Q(z^\star)$, where Q minimizes $L_Q(z^\star)$ over the triples with $\det A_Q(z^\star) \neq 0$. By Cramer's rule, its entries lie in $\mathbb{Q}(z^\star)$. We find such a Q by enumerating the $O(T^3)$ triples and evaluating $L_Q(z^\star)$ for each with $O(T)$ operations, which takes $O(JT^4)$ operations in $\mathbb{Q}(z^\star)$ over all j.

Operation count. For each interval, computing the pieces of all f_j takes $O(JT^8 \log T)$ operations, and sorting their endpoints takes $O(JT^8 \log(JT))$ operations. On each of the $O(JT^8)$ cells, P, Q, and H are formed with $O(J^2)$ operations, the real roots of one polynomial of degree at most $2J-2$ are isolated, and at most $2J$ candidates are compared. This gives $O(J^3T^9 + JT^9 \log(JT))$ arithmetic operations, $O(JT^9)$ root isolations, and $O(J^2T^9)$ algebraic comparisons in total.

4 Polynomial-Time Solvability: Sparse Setting

We now carry out Step 5 of the proof outline, which handles the sparsity constraint. Let $1 \le k < J$, since the case $k = 0$ is solved directly. For $p \in \{1, 2\}$, let $u_j := U_j - f_j \ge 0$ be the gains of Step 5. Since the gains are nonnegative, the constraint $|S| \le k$ may be replaced by $|S| = k$. By Step 2,

$$\mathrm{OPT}_{p,k}^{i^*} = \min_{z \in \mathcal{I}} \Phi(z), \qquad \Phi(z) := \sum_{j \in [J]} U_j - \max_{S \subseteq [J],\, |S| = k} \sum_{j \in S} u_j(z).$$

Lemma 7. *Let $p \in \{1, 2\}$, let $x_{i^*} = t_0 < t_1 < \cdots < t_n = x_{i^*+1}$, and for each $q \in [n]$ let $S_q \subseteq [J]$ with $|S_q| = k$ attain the maximum in the definition of $\Phi(z)$ for every $z \in (t_{q-1}, t_q)$. Then*

$$\mathrm{OPT}_{p,k}^{i^*} = \min_{q \in [n]} \min_{z \in [t_{q-1}, t_q]} \Big(\sum_{j \notin S_q} U_j + \sum_{j \in S_q} f_j(z) \Big).$$

Proof. Let $\Phi_q(z) := \sum_{j \notin S_q} U_j + \sum_{j \in S_q} f_j(z) = \sum_{j \in [J]} U_j - \sum_{j \in S_q} u_j(z)$. Then $\Phi_q \ge \Phi$ on $\mathcal{I}$, with equality on (t_{q-1}, t_q). Both functions are continuous by Lemma 1(b), so they are equal on $[t_{q-1}, t_q]$. □

4.1 ℓ_2 loss with sparsity constraint

By Lemma 3, $u_j(z) = U_j - f_j(z) = \nu_j(z)^2 / D(z)$, where $\nu_j(z) = (y^j)^\top M h(z)$ is affine in z and $D(z) = h(z)^\top M h(z)$ is a quadratic in z that is positive on $\mathbb{R}$ and the same for all j. For each z, rank the dimensions by $j \prec j'$ if $\nu_j(z)^2 > \nu_{j'}(z)^2$, or if $\nu_j(z)^2 = \nu_{j'}(z)^2$ and $j < j'$. For every z, the first k dimensions in this ranking attain the maximum in the definition of $\Phi(z)$. The ranking can change only at points where $\nu_j(z)^2 = \nu_{j'}(z)^2$ for some $j \neq j'$, so we partition $\mathcal{I}$ by these points. Let E be the set of roots in (x_{i^*}, x_{i^*+1}) of the affine functions $\nu_j(z) - \nu_{j'}(z)$ and $\nu_j(z) + \nu_{j'}(z)$, for the pairs $j \neq j'$ with $\nu_j(z)^2 - \nu_{j'}(z)^2 \not\equiv 0$. These roots are rational, and $|E| \le J(J-1)$. On each open cell between consecutive points of $E \cup \{x_{i^*}, x_{i^*+1}\}$, the function $\nu_j(z)^2 - \nu_{j'}(z)^2 = (\nu_j(z) - \nu_{j'}(z))(\nu_j(z) + \nu_{j'}(z))$ is either identically zero or has a constant nonzero

sign. Hence the ranking, and therefore the set S of its first k dimensions, is constant on the cell. By Lemma 7, on the closed cell we minimize

$$\sum_{j\in[J]} U_j - \frac{N_S(z)}{D(z)}, \qquad N_S(z) := \sum_{j\in S} \nu_j(z)^2,$$

where $N_S(z)$ has degree at most two irrespective of k and J. Lemma 4 gives at most four candidates per cell. Each lies in $\mathbb{Q}$ or in a real quadratic field over $\mathbb{Q}$ and, by Remark 1, the candidates are compared exactly in polynomial time. Given an optimal breakpoint $z^\star$ and the set S of its cell, we set $\delta_j^\star = 0$ and $(\alpha_j^\star, \beta_j^\star)^\top = G^{-1}A^\top y^j$ for $j \notin S$, and we recover $(\alpha_j^\star, \beta_j^\star, \delta_j^\star)$ for $j \in S$ as in Section 3.2.

Sweep. The cells are visited from left to right. On the first cell, the ranking is obtained by sorting, with comparisons at a rational point of the cell. For $t \in E$, let E_t be the set of dimensions j such that t is a root of $\nu_j - \nu_{j'}$ or of $\nu_j + \nu_{j'}$ for some j' with $\nu_j^2 - \nu_{j'}^2 \not\equiv 0$. If $j \notin E_t$, then, for every j', the sign of $\nu_j^2 - \nu_{j'}^2$ is the same on the two cells adjacent to t, so j has the same rank on both cells. Hence the ranks held by the dimensions in E_t are also the same on both cells. The set S of the cell to the right of t is therefore obtained from the set S of the cell to the left of t by replacing $S \cap E_t$ with the first $|S \cap E_t|$ dimensions of E_t in the ranking on the right cell. This takes $O(|E_t| \log |E_t|)$ operations. The coefficients of N_S are updated with $O(|E_t|)$ operations. Since $\sum_{t\in E} |E_t| \le 2J(J-1)$, the sweep takes $O(J^2 \log J)$ operations.

Operation count. For each interval, the coefficients $s_{j,0}, s_{j,1}$ and d_0, d_1, d_2 are obtained with $O(J)$ operations from the suffix sums of Section 3.2. Forming and sorting E takes $O(J^2 \log J)$ operations, the sweep takes $O(J^2 \log J)$ operations, and each of the $O(J^2)$ cells requires a constant number of operations, quadratic root computations, and comparisons. Over all intervals, and with the recovery step, this gives $O(J^2 T \log J)$ operations.

4.2 ℓ_∞ loss with sparsity constraint

By Step 5, where $S^\star$ consists of the k dimensions with the largest U_j, any minimizer of $g := \max_{j\in S^\star} f_j$ over $\mathcal{I}$ also minimizes $\max\{g, M\}$, since $t \mapsto \max\{t, M\}$ is nondecreasing. Hence we run the algorithm of Section 3.3 on the dimensions in $S^\star$, which takes $O(k^3 T^9 \log(kT))$ operations. For $j \notin S^\star$, we use an optimal single line with $\delta_j^\star = 0$, obtained as a rational optimal vertex of a linear program with three variables and $2T$ constraints in $O(T)$ operations [45]. This also gives U_j, so computing all U_j and sorting them takes $O(JT + J \log J)$ operations.

4.3 ℓ_1 loss with sparsity constraint

Each U_j is the optimal value of a linear program, so it can be computed in polynomial time by any polynomial-time linear programming algorithm. To obtain an operation count that does not depend on the encoding length of the data, we instead compute it as follows. The proof of Lemma 6(b), with $a_i(z)$ replaced by $(1, x_i)^\top$, shows that some optimal single line passes through two observations. Hence U_j is the smallest loss among the $\binom{T}{2}$ lines through two observations. Computing all U_j, together with optimal lines, takes $O(JT^3)$ operations.

By Section 3.4, each f_j, and hence each gain $u_j = U_j - f_j$, is piecewise linear-fractional with $O(T^8)$ pieces. Merging the endpoints of the pieces of all f_j partitions $\mathcal{I}$ into $O(JT^8)$ closed cells. On a cell $\mathcal{C}$, each $u_j = (a_j + b_j z)/(c_j + d_j z)$ has rational coefficients, and $c_j + d_j z \neq 0$ on $\mathcal{C}$. For

each pair $j \neq j'$, let $\pi_{jj'}$ be the polynomial of degree at most two defined in the proof of Lemma 2. We refine $\mathcal{C}$ by the at most $J(J-1)$ roots in its interior of the polynomials $\pi_{jj'} \not\equiv 0$. As in the proof of Lemma 2, on each open subcell the order of the values u_j, with ties broken by index, is constant. Hence the set S of the k largest gains is constant on the subcell. We find S with $O(J)$ comparisons by linear-time selection [12]. To compare u_j and $u_{j'}$ on the subcell, let ζ be its left endpoint, and note that

$$u_j(z) - u_{j'}(z) = \frac{\pi_{jj'}(z)}{(c_j + d_j z)(c_{j'} + d_{j'} z)}.$$

If $\pi_{jj'} \equiv 0$, the tie is broken by index. Otherwise, $\pi_{jj'}$ has no zero on the open subcell, and the denominator has no zero on the closed cell $\mathcal{C}$. Hence the sign of $u_j - u_{j'}$ is constant on the open subcell, and it equals the sign of $(c_j + d_j\zeta)(c_{j'} + d_{j'}\zeta)$ times the sign of $\pi_{jj'}(z)$ for $z > \zeta$ close to ζ. Since $\pi_{jj'}(\zeta + \varepsilon) = \pi_{jj'}(\zeta) + \pi'_{jj'}(\zeta)\varepsilon + \frac{1}{2}\pi''_{jj'}(\zeta)\varepsilon^2$, the latter is the sign of the first nonzero number among $\pi_{jj'}(\zeta)$, $\pi'_{jj'}(\zeta)$, and $\pi''_{jj'}(\zeta)$. As ζ lies in $\mathbb{Q}$ or in a real quadratic field over $\mathbb{Q}$, these signs are determined exactly with $O(1)$ operations by Remark 1.

If $\pi_{jj'} \equiv 0$, the tie is broken by index. By Lemma 7, on the closed subcell we minimize $\sum_{j \notin S} U_j + \sum_{j \in S} f_j(z)$. This is the problem of Section 3.4 with J replaced by k. Its minimum is attained at an endpoint of the subcell or at a root of a polynomial H_S of degree at most $2k-2$, formed with $O(k^2)$ operations. For $k = 1$, H_S is constant, and only the endpoints are candidates. Given an optimal breakpoint $z^\star$ and the set S of its subcell, the fits of the dimensions in S are recovered as in Section 3.4, and an optimal line is used for every $j \notin S$.

Operation count. For each interval, computing the pieces of all f_j and sorting their endpoints takes $O(JT^8 \log(JT))$ operations. Each of the $O(JT^8)$ cells requires $O(J^2 \log J)$ operations to form and sort the crossings, and each of its $O(J^2)$ subcells requires $O(J + k^2)$ operations, one root isolation of a polynomial of degree at most $2k-2$, and $O(k)$ algebraic comparisons. Together with the computation of the U_j, this gives $O((J + k^2)J^3T^9 + JT^9 \log(JT))$ arithmetic operations, $O(J^3T^9)$ root isolations, and $O(kJ^3T^9)$ algebraic comparisons in total.

5 Computational Results

In this section, we evaluate the practical performance of our algorithms for the least-squares loss ($p = 2$). We solve the dense and sparse problems on simulated time series and on three public datasets, and we compare our algorithms against Gurobi applied to a mixed-integer nonlinear programming (MINLP) formulation.

5.1 Scope of the experiments

Why only the ℓ_2 loss. We restrict the experiments to $p = 2$. Least squares is the standard loss for fitting a continuous piecewise-linear trend, and it corresponds to maximum likelihood estimation under Gaussian noise [37, 40, 25]. It is also the case in which our algorithms are practical. By Table 1, the dense algorithm is linear in the size of the input, and the sparse algorithm performs $O(J^2T \log J)$ operations. Although the algorithms for $p \in \{1, \infty\}$ run in polynomial time, their high-degree dependence on T makes them impractical for moderate-sized instances. This is not merely an artifact of the worst-case analysis: for $p = \infty$, the algorithm explicitly enumerates all $\binom{2T}{4}$ bases of (3) for each dimension and interval, resulting in more than 6.4×10^7 bases per dimension when $T = 100$.

Why floating-point arithmetic. Theorem 1 guarantees exact computation, but our implementation uses double-precision arithmetic, so its solutions are approximately optimal and not certified in the sense of Definition 1. Exact arithmetic is possible, but for large J its cost would be dominated by large-integer operations, since the coefficients of N aggregate all J dimensions.

5.2 Datasets

Table 2 summarizes the instances. Each dataset gives one dense instance and two sparse instances with $k = \lfloor 0.20J \rfloor$ and $k = \lfloor 0.10J \rfloor$. We consider sparsity levels of 20% and 10% to evaluate the algorithms under different degrees of sparsity, where changes are restricted to a subset of dimensions, motivated by similar sparsity levels considered in the computational experiments of [33, 3]. The subsets of the three public datasets used in our experiments, together with visualizations, are available in the repositories listed below.

Simulated data. We generate correlated stock-price trajectories with market, sector, and stock-specific fluctuations, as described in Appendix A. Each dimension independently undergoes a change in its expected daily log-return with a fixed probability, at a randomly selected time near the middle of the observation horizon. We use $T = 100$ and $x_i = i - 1$.

ERA5 temperature data. ERA5 is a global atmospheric reanalysis produced by the European Centre for Medium-Range Weather Forecasts, which combines historical observations with numerical weather models [35]. Motivated by studies of changing warming rates [18, 48, 27], we use the 2 m air temperature on a 201×201 latitude-longitude grid with 0.25° spacing, covering 10°N to 60°N and 125°W to 75°W, including land and ocean. Each grid point is one dimension, so $J = 40{,}401$. We use annual values for 2000 to 2025, so $T = 26$ and $x_i = 1999 + i$, where each value is the average over the year of the daily temperatures at 17:00 UTC. The data are available at https://github.com/ApoorvaGOR/ERA5-Regional-Temperature.

HoloAssist hand-tracking data. The HoloAssist dataset [59] contains hand movements tracked by the Microsoft HoloLens mixed-reality headset. We use an 8.013-second sequence sampled at about 30 frames per second, so $T = 240$, and the time coordinates are the timestamps in seconds. Each hand has 26 tracked joints with 21 measurements per joint, namely 16 entries of a pose transformation matrix, 3 spatial coordinates, and 2 tracking-status indicators. This gives 546 dimensions per hand and $J = 1{,}092$ for both hands. The data are available at https://github.com/ApoorvaGOR/HoloAssist-Hand-Sample.

NASA C-MAPSS turbofan data. The Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset [52] contains multivariate sensor measurements from simulated aircraft engines undergoing progressive degradation, a setting that motivates multidimensional slope change detection [19]. We use training engine 1 of the FD001 subset, with all 21 sensor channels over its complete trajectory of 192 cycles, so $J = 21$ and $T = 192$. Operational settings and remaining useful life (RUL) labels are not fitted. With $R_0 = 191$ the initial RUL and R_i the RUL at observation i, we set $x_i = R_0 + 1 - R_i$, which gives the operating cycles $1, \ldots, 192$. The data are available at https://github.com/ApoorvaGOR/C-MAPSS-Turbofan-Data.

Table 2: Instances considered in the computational experiments, including the dense setting and the sparse setting with sparsity levels of 20% and 10%.

Dataset	**Dense:** J	**Sparse:** $(J;k)$**,** 20%	**Sparse:** $(J;k)$**,** 10%	T	**Time coordinates**
Simulated	$10, 20, 10^2, 10^3, 10^4, 10^5$	$(10;2), (20;4), (10^2;20)$ $(10^3;200), (10^4;2000)$ $(10^5;20000)$	$(10;1), (20;2), (10^2;10)$ $(10^3;100), (10^4;1000)$ $(10^5;10000)$	100	$0, 1, \ldots, 99$
ERA5	40,401	(40,401; 8,080)	(40,401; 4,040)	26	Years 2000 to 2025
HoloAssist	1,092	(1,092; 218)	(1,092; 109)	240	Timestamps (s)
NASA C-MAPSS	21	(21; 4)	(21; 2)	192	Cycles $1, \ldots, 192$

5.3 Implementation and benchmark

Implementation and computing environment. The algorithms are implemented in Python, and their computationally intensive routines are compiled with the just-in-time (JIT) compiler Numba [43]. The experiments were run on a MacBook Pro (November 2023) with an Apple M3 chip and 16 GB of unified memory, running macOS Golden Gate 27.0.1, and the MINLP was solved with Gurobi Optimizer 13.0.2 [32]. Both our algorithms and the Gurobi MINLP benchmark were run using a single CPU thread. For our algorithms, we restricted Numba and the underlying numerical libraries to one thread, while Gurobi was configured to use one thread. The source code and computational results are publicly available at the following repository:
https://github.com/ApoorvaGOR/High-Dimensional-Piecewise-Linear-Fitting.

Runtime measurement. Numba compiles each numerical routine the first time it is called. Before the experiments, we therefore run each algorithm once on a small instance, with $J = 2$ and $T = 8$ for the dense algorithm and with $J = 4$, $T = 8$, and $k = 2$ for the sparse algorithm. The experiments for the two sparsity levels (20% and 10%) are conducted in separate executions, each with its own warmup. The warmup is performed once per execution, and the compiled code is then reused for all instances within that execution. The warmup takes 3.613 s for the dense algorithm and 5.090 s and 5.120 s for the sparse algorithm in the 20% and 10% sparsity experiments, respectively. The reported runtimes of our algorithms exclude the warmup, data acquisition or generation, time-coordinate conversion, and output storage. All runtimes are from single runs. Runtimes are measured using Python's `time.perf_counter()`, a high-resolution monotonic wall-clock timer, which uses macOS's `mach_absolute_time()` with a reported resolution of approximately 41.67 ns on our machine.

MINLP benchmark. As a benchmark, we use the MINLP formulation of [47] for a single shared breakpoint, stated with its big-M constants in Appendix B. It has T binary variables that assign the observations to the two segments, J further binary variables in the sparse problem, and J bilinear continuity constraints. Each instance is solved by Gurobi with a time limit of $10,000$ s. The reported MINLP runtime is the solver time, and it excludes the construction of the parameter bounds and of the model. Gurobi may exceed the time limit slightly while terminating. We report the best lower bound (LB), the objective value of the best solution found (UB), and the relative optimality gap. Our algorithms are run without a time limit.

5.4 Results

Table 3 reports the results for the dense and the sparse problem. The column OPT_2 gives the optimal value, that is, the sum of squared errors, computed by our algorithms.

Table 3: Computational comparison of our algorithms and the MINLP benchmark for (a) the dense problem of Section 3.2, (b) the sparse problem of Section 4 with $k = \lfloor 0.20J \rfloor$, and (c) the sparse problem with $k = \lfloor 0.10J \rfloor$.

				Our algorithm		MINLP benchmark			
Dataset	J	T	k	OPT_2	Runtime (s)	LB	UB	Gap (%)	Runtime (s)
(a) Dense problem									
Simulated	10	100	–	**25,830.48**	$\mathbf{2.600 \times 10^{-4}}$	25,827.92	**25,830.48**	0.01	1.726×10^{1}
	20	100	–	**78,409.82**	$\mathbf{2.020 \times 10^{-4}}$	78,403.28	**78,409.82**	0.01	5.374×10^{1}
	100	100	–	**417,949.81**	$\mathbf{2.090 \times 10^{-4}}$	417,908.01	**417,949.81**	0.01	7.097×10^{2}
	1,000	100	–	**3,691,434.08**	$\mathbf{4.790 \times 10^{-4}}$	3,683,252.43	**3,691,434.08**	0.22	1.000×10^{4} *
	10,000	100	–	**36,169,676.73**	$\mathbf{4.769 \times 10^{-3}}$	17,066,148.10	40,817,159.15	58.19	1.000×10^{4} *
	100,000	100	–	**358,712,206.42**	$\mathbf{7.961 \times 10^{-2}}$	0.04	607,422,150.29	100.00	1.002×10^{4} *
ERA5	40,401	26	–	**501,263.44**	$\mathbf{4.681 \times 10^{-3}}$	8.55×10^{-6}	528,198.01	100.00	1.000×10^{4} *
HoloAssist	1,092	240	–	**6,176.35**	$\mathbf{3.274 \times 10^{-3}}$	5,878.88	6,176.43	4.82	1.000×10^{4} *
NASA C-MAPSS	21	192	–	**11,686.49**	$\mathbf{2.960 \times 10^{-4}}$	11,685.34	**11,686.49**	0.01	7.520×10^{0}
(b) Sparse problem ($k = \lfloor 0.20J \rfloor$)									
Simulated	10	100	2	**31,428.50**	$\mathbf{1.950 \times 10^{-4}}$	31,425.62	**31,428.50**	0.01	4.090×10^{0}
	20	100	4	**91,112.37**	$\mathbf{4.270 \times 10^{-4}}$	91,103.76	**91,112.37**	0.01	9.139×10^{1}
	100	100	20	**500,700.13**	$\mathbf{6.138 \times 10^{-3}}$	369,423.81	503,853.64	26.68	1.000×10^{4} *
	1,000	100	200	**4,381,778.86**	$\mathbf{5.797 \times 10^{-1}}$	2,077,935.04	4,399,461.26	52.77	1.000×10^{4} *
	10,000	100	2,000	**43,815,769.93**	$\mathbf{6.768 \times 10^{1}}$	1.09×10^{-5}	53,566,082.59	100.00	1.000×10^{4} *
	100,000	100	20,000	–	–†	0.04	617,148,417.09	100.00	1.002×10^{4} *
ERA5	40,401	26	8,080	**513,464.42**	$\mathbf{8.881 \times 10^{3}}$	8.55×10^{-6}	537,144.62	100.00	1.000×10^{4} *
HoloAssist	1,092	240	218	**6,485.05**	$\mathbf{7.153 \times 10^{-1}}$	1.60×10^{-8}	7,921.85	100.00	1.000×10^{4} *
NASA C-MAPSS	21	192	4	**11,766.44**	$\mathbf{6.300 \times 10^{-4}}$	11,765.34	**11,766.44**	0.01	1.216×10^{1}
(c) Sparse problem ($k = \lfloor 0.10J \rfloor$)									
Simulated	10	100	1	**37,049.63**	$\mathbf{1.710 \times 10^{-4}}$	37,049.63	37,051.92	0.01	1.638×10^{0}
	20	100	2	**105,367.14**	$\mathbf{3.960 \times 10^{-4}}$	105,362.30	**105,367.14**	0.00	1.064×10^{1}
	100	100	10	**558,320.79**	$\mathbf{6.381 \times 10^{-3}}$	386,192.66	559,523.91	30.98	1.000×10^{4} *
	1,000	100	100	**4,888,999.40**	$\mathbf{6.119 \times 10^{-1}}$	1,063,321.38	5,225,165.60	79.65	1.000×10^{4} *
	10,000	100	1,000	**49,101,280.35**	$\mathbf{6.590 \times 10^{1}}$	1.09×10^{-5}	62,069,667.92	100.00	1.000×10^{4} *
	100,000	100	10,000	–	– †	0.04	617,148,417.09	100.00	1.002×10^{4} *
ERA5	40,401	26	4,040	**521,415.53**	$\mathbf{7.704 \times 10^{3}}$	8.55×10^{-6}	537,144.62	100.00	1.000×10^{4} *
HoloAssist	1,092	240	109	**6,807.89**	$\mathbf{7.194 \times 10^{-1}}$	1.60×10^{-8}	8,179.81	100.00	1.000×10^{4} *
NASA C-MAPSS	21	192	2	**12,756.23**	$\mathbf{1.181 \times 10^{-3}}$	12,755.38	**12,756.23**	0.01	1.048×10^{1}

Notes: The one-time JIT compilation times of approximately 3.613 s for the dense algorithm, 5.090 s for the sparse algorithm with 20% sparsity, and 5.120 s for the sparse algorithm with 10% sparsity are excluded from the reported runtimes. LB, UB, and the gap refer to the bounded MINLP, with the gap representing Gurobi's relative gap. In each row, bold runtimes indicate the shorter runtime, and bold objective values indicate the lowest displayed objective value. *The MINLP reached the prescribed time limit of 10,000 s. –†Our algorithm did not terminate within the prescribed time limit of 10,000 s; consequently, its runtime and objective value are not reported.

Dense problem. The dense algorithm solves every instance in less than 0.1 s, including the simulated instance with $J = 10^5$ and 10^7 observations, which takes less than 0.080 s. This speed matters in online detection [19], where the problem is solved again after each new observation, for example to evaluate the generalized likelihood ratio statistic [54], which under Gaussian noise is proportional to $\sum_{j\in[J]} U_j - \mathrm{OPT}_{2,J}$. On HoloAssist, one solve takes 3.27 ms, well below the interval of about 33 ms between consecutive frames at 30 frames per second.

Sparse problem. For both sparsity levels (20% and 10%), the sparse algorithm solves all instances with $J \leq 1092$ in less than 1 s. For the simulated instance with $J = 10^4$, the runtimes are 67.68 s and 65.90 s for 20% and 10% sparsity, respectively. The ERA5 instance, with $J = 40{,}401$, takes approximately 2.5 hours and 2.1 hours, respectively. For the simulated instance with $J = 10^5$, the algorithm does not finish within 10,000 s at either sparsity level. These results reflect the quadratic dependence on J in the bound $O(J^2 T \log J)$, since each interval can contain up to $J(J-1)$ crossings of two gains, approximately 1.6×10^9 for ERA5. Hence, the sparse algorithm

remains practical for instances with up to about 10^4 dimensions at both sparsity levels.

Comparison with the MINLP. Gurobi solves the MINLP to its default relative gap of 0.01% only on the NASA C-MAPSS instances and on the simulated instances with $J \leq 100$ in the dense problem and $J \leq 20$ in the sparse problem, at both 20% and 10% sparsity levels. On these instances, our algorithms obtain objective values no worse than Gurobi's, and are faster by factors ranging from approximately 8.9×10^3 to 3.4×10^6. On all other instances, Gurobi reaches the time limit of 10,000 s with gaps between 0.22% and 100%. Its best solution matches ours only on the dense simulated instance with $J = 1000$ and is otherwise worse, for example by 5.4% on the dense ERA5 instance and by 69% on the dense simulated instance with $J = 10^5$. For the sparse simulated instance with $J = 10^5$, neither method finishes within 10,000 s at either sparsity level.

6 Conclusion

We studied the problem of fitting continuous piecewise-linear functions with a single breakpoint shared by all dimensions of a multidimensional time series, where the breakpoint varies continuously. We showed that this problem is solvable in polynomial time under the ℓ_1, ℓ_2, and ℓ_∞ losses, both in the dense setting and in the sparse setting, in which at most k dimensions may change slope. Since optimal breakpoints can be irrational, we specified what it means to solve the problem exactly. Our algorithms return an optimal breakpoint as a real algebraic number, of degree at most two for $p \in \{2, \infty\}$ and of degree bounded in terms of J or k for $p = 1$, and they return the fitting coefficients as rational functions of it. All six variants are handled by the same strategy, which involves reduction to intervals between consecutive observations, decoupling of the dimensions for a fixed breakpoint, a description of each dimension-wise optimal error as a function of the breakpoint, and aggregation of these functions over the dimensions.

For the least-squares loss, the algorithms are also practical. On simulated data and on climate, hand-tracking, and turbofan data, the dense algorithm solved every instance, with up to 10^7 observations, in less than 0.1 s, and the sparse algorithm solved instances with about a thousand dimensions in less than a second. In particular, the dense algorithm is fast enough to solve the problem again each time a new observation arrives, which makes exact least-squares fitting usable for online slope change detection. A state-of-the-art MINLP solver, applied to an exact mixed-integer formulation, solved only the smallest instances within $10,000$ s and otherwise ended with large optimality gaps.

Several questions remain open. The sparse algorithm took approximately 2.5 hours and 2.1 hours on the ERA5 instance with 40,401 dimensions at 20% and 10% sparsity, respectively, primarily due to the large number of pairwise crossings of the gains that must be processed. An algorithm that tracks only the crossings that change the set of the k largest gains could improve this substantially. For $p \in \{1, \infty\}$, our operation counts are polynomial but of high degree in T, and it would be valuable to find algorithms whose dependence on T is close to that of the least-squares case. Finally, the complexity of fitting several shared breakpoints remains open. This problem is equivalent to globally training a ReLU network with time as its input, several hidden units, and J outputs that share those units. It is natural to ask whether the problem stays polynomial-time solvable when the number of breakpoints is fixed, and whether it becomes NP-hard when that number is part of the input.

Acknowledgements

The authors acknowledge the use of large language models to assist in writing the code for the computational experiments in Section 5. All code was reviewed and tested by the authors, who take full responsibility for its correctness and for the reported results.

References

[1] Mikkel Abrahamsen, Linda Kleist, and Tillmann Miltzow. Training neural networks is $\exists\mathbb{R}$-complete. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

[2] Fatemeh Aghaei A., Ewan T. Phillips, and Holger Kantz. Change point detection in ERA5 ground temperature time series. *Climatic Change*, 178(11):193, 2025.

[3] Andreas Anastasiou and Angelos Papanastasiou. Generalized multiple change-point detection in the structure of multivariate, possibly high-dimensional, data sequences. *Statistics and Computing*, 33(5):94, 2023.

[4] Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. In *International Conference on Learning Representations*, 2018.

[5] Jushan Bai and Pierre Perron. Computation and analysis of multiple structural change models. *Journal of Applied Econometrics*, 18(1):1–22, 2003.

[6] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Real Algebraic Geometry*, volume 10 of *Algorithms and Computation in Mathematics*. Springer, Berlin, second edition, 2006.

[7] Claudie Beaulieu, Colin Gallagher, Rebecca Killick, Robert Lund, and Xueheng Shi. A recent surge in global warming is not detectable yet. *Communications Earth & Environment*, 5(1):576, 2024.

[8] Claude Berge. *Topological Spaces*. Oliver and Boyd, Edinburgh and London, 1963.

[9] Daniel Bertschinger, Christoph Hertrich, Paul Jungeblut, Tillmann Miltzow, and Simon Weber. Training fully connected neural networks is $\exists\mathbb{R}$-complete. In *Advances in Neural Information Processing Systems*, volume 36, pages 36222–36237, 2023.

[10] Daniel Bienstock, Alberto Del Pia, and Robert Hildebrand. Complexity, exactness, and rationality in polynomial optimization. *Mathematical Programming*, 197(2):661–692, 2023.

[11] Kevin Bleakley and Jean-Philippe Vert. The group fused lasso for multiple change-point detection, 2011. https://arxiv.org/abs/1106.4199.

[12] Manuel Blum, Robert W. Floyd, Vaughan Pratt, Ronald L. Rivest, and Robert E. Tarjan. Time bounds for selection. *Journal of Computer and System Sciences*, 7(4):448–461, 1973.

[13] Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327, 1986.

[14] Digvijay Boob, Santanu S. Dey, and Guanghui Lan. Complexity of training ReLU neural network. *Discrete Optimization*, 44:100620, 2022.

[15] Laurie Boveroux, Margarida Carvalho, Andrea Lodi, and Quentin Louveaux. On the complexity of linear programs with parametric constraint matrices, 2026. https://arxiv.org/abs/2610.20422.

[16] Richard P. Brent and Paul Zimmermann. *Modern Computer Arithmetic*, volume 18 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 2010.

[17] Samuel L Brilleman, Laura D Howe, Rory Wolfe, and Kate Tilling. Bayesian piecewise linear mixed models with a random change point: an application to BMI rebound in childhood. *Epidemiology (Cambridge, Mass.)*, 28(6):827, 2017.

[18] Niamh Cahill, Stefan Rahmstorf, and Andrew C Parnell. Change points of global temperature. *Environmental Research Letters*, 10(8):084002, 2015.

[19] Yang Cao, Yao Xie, and Nagi Gebraeel. Multi-sensor slope change detection. *Annals of Operations Research*, 263(1):163–189, 2018.

[20] Antonio de la Casa and Olga Nasello. Breakpoints in annual rainfall trends in Córdoba, Argentina. *Atmospheric Research*, 95(4):419–427, 2010.

[21] Alberto Del Pia, Santanu S. Dey, and Marco Molinaro. Mixed-integer quadratic programming is in NP. *Mathematical Programming*, 162(1–2):225–240, 2017.

[22] Santanu S. Dey, Guanyi Wang, and Yao Xie. Approximation algorithms for training one-node ReLU neural networks. *IEEE Transactions on Signal Processing*, 68, 2020.

[23] Farida Enikeeva and Zaid Harchaoui. High-dimensional change-point detection under sparse alternatives. *The Annals of Statistics*, 47(4):2051–2079, 2019.

[24] Paul Fearnhead and Daniel Grose. CPOP: Detecting changes in piecewise-linear signals. *Journal of Statistical Software*, 109:1–30, 2024.

[25] Paul Fearnhead, Robert Maidstone, and Adam Letchford. Detecting changes in slope with an ℓ_0 penalty. *Journal of Computational and Graphical Statistics*, 28(2):265–275, 2019.

[26] Paul I. Feder. On asymptotic distribution theory in segmented regression problems–identified case. *The Annals of Statistics*, 3(1):49–83, 1975.

[27] Grant Foster and Stefan Rahmstorf. Global warming has accelerated significantly. *Geophysical Research Letters*, 53(5):e2025GL118804, 2026.

[28] Robert M. Freund. Postoptimal analysis of a linear program under simultaneous changes in matrix coefficients. *Mathematical Programming Study*, 24:1–13, 1985.

[29] Vincent Froese and Christoph Hertrich. Training neural networks is NP-hard in fixed dimension. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[30] Noam Goldberg, Steffen Rebennack, Youngdae Kim, Vitaliy Krasko, and Sven Leyffer. MINLP formulations for continuous piecewise linear function fitting. *Computational Optimization and Applications*, 79(1):223–233, 2021.

[31] Leo J Guibas and Robert Sedgewick. A dichromatic framework for balanced trees. In *19th Annual Symposium on Foundations of Computer Science (sfcs 1978)*, pages 8–21. IEEE, 1978.

[32] Gurobi Optimization, LLC. Gurobi optimizer reference manual, 2026. Version 13.0.

[33] Georg Hahn, Paul Fearnhead, and Idris A Eckley. Bayesproject: Fast computation of a projection direction for multivariate changepoint detection. *Statistics and Computing*, 30(6):1691–1705, 2020.

[34] James D Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica: Journal of the econometric society*, pages 357–384, 1989.

[35] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, et al. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020.

[36] David V. Hinkley. Inference about the intersection in two-phase regression. *Biometrika*, 56(3):495–504, 1969.

[37] Derek J. Hudson. Fitting segmented curves whose join points have to be estimated. *Journal of the American Statistical Association*, 61(316):1097–1129, 1966.

[38] Brad Jackson, Jeffrey D. Scargle, David Barnes, Sundararajan Arabhi, Alina Alt, Peter Gioumousis, Elyus Gwin, Paungkaew Sangtrakulcharoen, Linda Tan, and Tun Tao Tsai. An algorithm for optimal partitioning of data on an interval. *IEEE Signal Processing Letters*, 12(2):105–108, 2005.

[39] Rebecca Killick, Paul Fearnhead, and Idris A. Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.

[40] Dukpa Kim. Estimating a common deterministic time trend break in large panels with cross sectional dependence. *Journal of Econometrics*, 164(2):310–330, 2011.

[41] Dukpa Kim. Common breaks in time trends for large panel data with a factor structure. *The Econometrics Journal*, 17(3):301–337, 2014.

[42] Seung-Jean Kim, Kwangmoo Koh, Stephen Boyd, and Dimitry Gorinevsky. ℓ_1 trend filtering. *SIAM Review*, 51(2):339–360, 2009.

[43] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. Numba: A LLVM-based Python JIT compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pages 1–6, 2015.

[44] P. M. Lerman. Fitting segmented regression models by grid search. *Journal of the Royal Statistical Society Series C (Applied Statistics)*, 29(1):77–84, 1980.

[45] Nimrod Megiddo. Linear programming in linear time when the dimension is fixed. *Journal of the ACM*, 31(1):114–127, 1984.

[46] Vito M. R. Muggeo. Estimating regression models with unknown break-points. *Statistics in Medicine*, 22(19):3055–3071, 2003.

[47] Apoorva Narula, Santanu S. Dey, and Yao Xie. Change-point detection via piecewise linear fitting using MIP, 2026. https://arxiv.org/abs/2602.11947.

[48] Stefan Rahmstorf, Grant Foster, and Niamh Cahill. Global temperature evolution: recent trends and some pitfalls. *Environmental Research Letters*, 12(5):054001, 2017.

[49] Steffen Rebennack and Vitaliy Krasko. Piecewise linear function fitting via mixed-integer linear programming. *INFORMS Journal on Computing*, 32(2):507–530, 2020.

[50] Stephen A Ross. The arbitrage theory of capital asset pricing. *Journal of Economic Theory*, 13(3):341–360, 1976.

[51] Michael Sagraloff and Kurt Mehlhorn. Computing real roots of real polynomials. *Journal of Symbolic Computation*, 73:46–86, 2016.

[52] Abhinav Saxena, Kai Goebel, Don Simon, and Neil Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 International Conference on Prognostics and Health Management*, pages 1–9. IEEE, 2008.

[53] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[54] David Siegmund and E. S. Venkatraman. Using the generalized likelihood ratio statistic for sequential detection of a change-point. *The Annals of Statistics*, 23(1):255–271, 1995.

[55] Ryan J. Tibshirani. Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42(1):285–323, 2014.

[56] Alejandro Toriello and Juan Pablo Vielma. Fitting piecewise linear continuous functions. *European Journal of Operational Research*, 219(1):86–95, 2012.

[57] Umberto Triacca and Antonello Pasini. Recent structural breaks in global temperature series: Evidence from a changepoint analysis. *Climate*, 14(6):129, 2026.

[58] Tengyao Wang and Richard J Samworth. High dimensional change point estimation via sparse projection. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 80(1):57–83, 2018.

[59] Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, et al. HoloAssist: An egocentric human interaction dataset for interactive AI assistants in the real world. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20213–20224. IEEE, 2023.

[60] Andreas Weinmann and Martin Storath. Iterative Potts and Blake–Zisserman minimization for the recovery of functions with discontinuities from indirect measurements. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2176):20140638, 2015.

Appendix

A Synthetic Data Generation

Our synthetic data generation procedure is motivated by multifactor models of asset returns [50], models of time-varying volatility [13], and models allowing changes in the underlying return-generating process [34]. We generate J correlated stock-price trajectories, each observed over $T = 100$ consecutive business days, with observation times $x_i = i - 1$ for $i \in [T]$.

Generating stock returns. Each stock is assigned independently to one of 20 sectors and an initial price drawn uniformly from $[20, 250]$. Its daily log-return is generated as

$$\widetilde{R}_{j,t} = \mu_{j,t} + \gamma_j^M R_t^M + \gamma_j^G R_{g_j,t}^G + \sigma_j v_{j,t} \epsilon_{j,t}, \tag{A.1}$$

where R_t^M represents market-wide fluctuations, $R_{g_j,t}^G$ represents fluctuations shared by stocks in sector g_j, and $\epsilon_{j,t}$ represents independent stock-specific Gaussian noise. The coefficients γ_j^M, γ_j^G and σ_j control the contribution of these components. The market and stock-specific volatility factors vary over time based on preceding shocks, allowing periods of higher and lower variability. Thus, stocks are correlated through shared market and sector fluctuations while retaining stock-specific variability.

Introducing changes. Each stock independently undergoes a change in its baseline drift (the average daily log-return) with probability 0.70. For each stock, a potential change time is sampled from a normal distribution centered at 52% of the observation period, with a standard deviation of 10% of its length. The sampled time is rounded to an integer and restricted to lie away from the endpoints. The daily drift is then given by

$$\mu_{j,t} = \mu_j + \kappa_j \mathbb{I}\{t > \tau_j\}, \tag{A.2}$$

where τ_j is the change time and $\kappa_j \sim \mathcal{N}(0, 0.0015^2)$ for stocks undergoing a change, and $\kappa_j = 0$ otherwise. Thus, different stocks may experience changes at different times and of different magnitudes.

Constructing prices. The generated daily log-returns are capped to $[-0.18, 0.18]$ to limit extreme daily fluctuations. Prices are then computed recursively as

$$P_{j,t+1} = P_{j,t} \exp(R_{j,t}), \qquad R_{j,t} = \text{clip}(\widetilde{R}_{j,t}, -0.18, 0.18), \tag{A.3}$$

where $\text{clip}(a, l, u) := \min\{\max\{a, l\}, u\}$. The resulting observations are $y_i^j = P_{j,i}$. The dense and sparse experiments use the same generation procedure, with J and k specified in Table 2. The complete implementation, including the parameter distributions and volatility updates, is available at: https://github.com/ApoorvaGOR/High-Dimensional-Piecewise-Linear-Fitting/blob/main/data_simulator.py

.

Relation to the optimization model. These data are not generated from the continuous piecewise-linear model (2). Changes occur in the drift of log-returns rather than directly in price slopes, and their locations may differ across stocks. This allows us to evaluate our algorithms on noisy, correlated time series that do not necessarily satisfy the assumed shared-breakpoint model.

B A Mixed-Integer Nonlinear Formulation and Its Big-M Constants

We state a mixed-integer nonlinear programming (MINLP) formulation of (2) for $p = 2$ and describe its parameter bounds and big-M constants. The construction follows [47].

B.1 Formulation

For each dimension $j \in [J]$ and segment $\ell \in \{1, 2\}$, let m_ℓ^j and c_ℓ^j be the slope and the intercept of the affine function on segment ℓ, where segment 1 lies before the breakpoint and segment 2 after it. In the notation of (1), $m_1^j = \beta_j$, $c_1^j = \alpha_j$, $m_2^j = \beta_j + \delta_j$, and $c_2^j = \alpha_j - \delta_j z$. Let $\widehat{y}_i^j$ be the fitted value of dimension j at time x_i, and let the binary variable X_i equal 1 if observation i is assigned to segment 1 and 0 if it is assigned to segment 2. The dense formulation is

$$\min \quad \sum_{i\in[T]} \sum_{j\in[J]} \left(y_i^j - \widehat{y}_i^j\right)^2 \tag{B.1}$$

$$\text{s.t.} \quad -M_{1,ij}(1 - X_i) \le \widehat{y}_i^j - m_1^j x_i - c_1^j \le M_{1,ij}(1 - X_i), \qquad i \in [T],\ j \in [J], \tag{B.2}$$

$$-M_{1,ij} X_i \le \widehat{y}_i^j - m_2^j x_i - c_2^j \le M_{1,ij} X_i, \qquad i \in [T],\ j \in [J], \tag{B.3}$$

$$m_1^j z + c_1^j = m_2^j z + c_2^j, \qquad j \in [J], \tag{B.4}$$

$$x_i \le z + M_{2,i}(1 - X_i), \qquad i \in [T], \tag{B.5}$$

$$x_i \ge z - M_{3,i} X_i, \qquad i \in [T], \tag{B.6}$$

$$X_i \ge X_{i+1}, \qquad i \in [T-1], \tag{B.7}$$

$$(m, c, z) \in \Theta, \qquad X \in \{0, 1\}^T, \tag{B.8}$$

where the set Θ and the constants $M_{1,ij}$, $M_{2,i}$, and $M_{3,i}$ are defined below. The constraints (B.2) and (B.3) set $\widehat{y}_i^j$ to the value of the affine function of the segment to which observation i is assigned. The constraints (B.5)–(B.7) ensure that the observations of segment 1 are exactly those up to the breakpoint, with either assignment allowed for an observation at the breakpoint. The constraints (B.4) impose continuity at the breakpoint. They are bilinear, and they are the source of nonconvexity. For a fixed breakpoint z, every feasible point corresponds to a continuous piecewise-linear fit with breakpoint z in each dimension, and the objective (B.1) equals $L_2(r)$.

In the sparse problem, we add a binary variable ξ_j for each $j \in [J]$, where $\xi_j = 1$ indicates that dimension j may change slope, and the constraints of Section B.5.

B.2 Parameter bounds

For each dimension j, the lower and upper slope bounds are the smallest and largest slopes between two observations,

$$L_m^j := \min_{\substack{i,i'\in[T]\\ i\neq i'}} \frac{y_i^j - y_{i'}^j}{x_i - x_{i'}}, \tag{B.9}$$

$$U_m^j := \max_{\substack{i,i'\in[T]\\ i\neq i'}} \frac{y_i^j - y_{i'}^j}{x_i - x_{i'}}. \tag{B.10}$$

Given these slope bounds, the intercept bounds are

$$L_c^j := \min_{i\in[T]} \min\left\{y_i^j - L_m^j x_i,\ y_i^j - U_m^j x_i\right\}, \tag{B.11}$$

$$U_c^j := \max_{i\in[T]} \max\left\{y_i^j - L_m^j x_i,\ y_i^j - U_m^j x_i\right\}. \tag{B.12}$$

For every $j \in [J]$ and $\ell \in \{1,2\}$, we impose

$$L_m^j \le m_\ell^j \le U_m^j, \qquad L_c^j \le c_\ell^j \le U_c^j. \tag{B.13}$$

The breakpoint is restricted to $x_2 \le z \le x_{T-1}$, as in (2), so that at least two observations lie on each side of the breakpoint. Hence

$$\Theta := \Big\{(m,c,z):\ L_m^j \le m_\ell^j \le U_m^j,\ L_c^j \le c_\ell^j \le U_c^j \text{ for all } j\in[J],\ \ell\in\{1,2\},\ x_2 \le z \le x_{T-1}\Big\}. \tag{B.14}$$

The bounds defining Θ are data-driven, and they are not guaranteed to contain an optimal solution of (2), whose slopes and intercepts are unrestricted. Hence the formulation (B.1)–(B.8) is a restriction of (2), and its optimal value is at least $\mathrm{OPT}_{2,k}$. The two values are equal if some optimal solution of (2) lies in Θ. This can be checked using the solution returned by the algorithms of Sections 3 and 4.

B.3 Big-M constants for the fitted values

For $i \in [T]$ and $j \in [J]$, let

$$\underline{f}_{ij} := \min\left\{L_m^j x_i,\ U_m^j x_i\right\} + L_c^j, \tag{B.15}$$

$$\overline{f}_{ij} := \max\left\{L_m^j x_i,\ U_m^j x_i\right\} + U_c^j. \tag{B.16}$$

For every $(m,c,z) \in \Theta$ and $\ell \in \{1,2\}$, we have $m_\ell^j x_i \in \big[\min\{L_m^j x_i, U_m^j x_i\},\ \max\{L_m^j x_i, U_m^j x_i\}\big]$, and therefore

$$\underline{f}_{ij} \le m_\ell^j x_i + c_\ell^j \le \overline{f}_{ij}. \tag{B.17}$$

Let

$$M_{1,ij} := \overline{f}_{ij} - \underline{f}_{ij}. \tag{B.18}$$

At every feasible point, $\widehat{y}_i^j$ equals $m_\ell^j x_i + c_\ell^j$ for the segment ℓ to which observation i is assigned, so $\underline{f}_{ij} \le \widehat{y}_i^j \le \overline{f}_{ij}$ by (B.17). Hence, for the other segment ℓ', (B.17) gives

$$\left|m_{\ell'}^j x_i + c_{\ell'}^j - \widehat{y}_i^j\right| \le M_{1,ij}. \tag{B.19}$$

Thus the assignment constraint of the segment to which observation i is not assigned is satisfied automatically, and $M_{1,ij}$ is a valid big-M constant in (B.2) and (B.3).

B.4 Big-M constants for the breakpoint

Since $x_2 \le z \le x_{T-1}$, valid constants for (B.5) and (B.6) are

$$M_{2,i} := \max\{0, x_i - x_2\}, \qquad i \in [T], \tag{B.20}$$

$$M_{3,i} := \max\{0, x_{T-1} - x_i\}, \qquad i \in [T]. \tag{B.21}$$

If $X_i = 1$, then (B.5) enforces $x_i \le z$, and (B.6) reads $x_i \ge z - M_{3,i}$, which holds since $z \le x_{T-1}$. If $X_i = 0$, then (B.6) enforces $x_i \ge z$, and (B.5) reads $x_i \le z + M_{2,i}$, which holds since $z \ge x_2$. The monotonicity constraints (B.7) ensure that the observations of each segment form a contiguous block.

B.5 Constraints and Big-M constants for the sparse problem

The bounds (B.13) imply

$$|m_2^j - m_1^j| \le U_m^j - L_m^j, \qquad |c_2^j - c_1^j| \le U_c^j - L_c^j. \tag{B.22}$$

Hence valid big-M constants are

$$M_m^j := U_m^j - L_m^j, \qquad M_c^j := U_c^j - L_c^j, \tag{B.23}$$

and the sparse constraints are

$$-M_m^j \xi_j \le m_2^j - m_1^j \le M_m^j \xi_j, \qquad j \in [J], \tag{B.24}$$

$$-M_c^j \xi_j \le c_2^j - c_1^j \le M_c^j \xi_j, \qquad j \in [J], \tag{B.25}$$

$$\sum_{j \in [J]} \xi_j \le k, \tag{B.26}$$

$$\xi_j \in \{0, 1\}, \qquad j \in [J]. \tag{B.27}$$

If $\xi_j = 0$, these constraints enforce $m_1^j = m_2^j$ and $c_1^j = c_2^j$, so dimension j is fitted by a single line, that is, $\delta_j = 0$. If $\xi_j = 1$, they impose no restriction beyond (B.13).