arXiv:2301.07703v1 [cs.CG] 18 Jan 2023

利用拓扑信号处理鲁棒地检测含噪信号中的零交叉点

Sunia Tanweer* Firas A. Khasawneh† Elizabeth Munch‡

2023年1月19日

摘要

本文探讨了拓扑数据分析（TDA）中的零维持久同调在确定一维连续函数及均匀采样时间序列的零交叉点所在区间方面的一种新颖应用。我们提出了一种算法，并展示了该算法在多种采样频率下存在噪声时的鲁棒性。与用于寻找时间序列零点的先进软件方法相比，我们的方法通常收敛更快、精度更高，而且能够找出给定区间内的所有根，而不是只收敛到其中一个根。我们还介绍并比较了若干自动设置持久性阈值参数的方法，该参数会影响根区间确定的准确性。

1 引言

零交叉点是连续函数改变符号的位置。确定这些零交叉点是许多领域中的一个经典而重要的问题，例如工程、医学和物理科学。具体而言，零交叉点已用于信号频率测定[1–3]，肌肉疲劳估计 [4]，中子与伽马射线的区分 [5]，感应电动机短路故障检测 [6]，直流电动机调速 [7]，陆地车辆速度测量 [8]，弹性成像应变估计 [9]，甚至手写字符识别 [10]。

尽管寻找函数 $f(x) = 0$ 何处过零的问题意义重大，且历史悠久，但文献中提出的零点区间定位方法种类繁多，各有缺陷。这些方法可分为开放式区间定位方案——如牛顿-拉夫森法、不动点法和割线法——以及闭合式区间定位方案，如二分法和假位法 [11]。最基本的闭合式零点区间定位方法是二分法。尽管速度较慢，但只要给定初始区间 $[a, b]$，使得 $f(a)f(b) < 0$，该方法就能保证对连续函数 $f(x)$ 收敛。另一种常用的区间定位方法是假位

*tanweer1@msu.edu
†khasawn3@msu.edu
‡muncheli@msu.edu

法 [11] ，除函数斜率过平或过陡这两种棘手情况外，其收敛速度快于二分法。以这些算法为基础，人们进行了多种改进。其中一种改进见于 Suhadolnik 的 [12] 结合方法，该方法在二分法与假位法之间切换，以求解非线性方程的根。在该方法中，Suhadolnik 使用二次插值，将已知的函数两点以及通过二分法／假位法得到的估计值拟合到一条抛物线上。Alojz [13], Razbani [14], Kavvadias [15], Kodnyanko [16], Badr 等人 [17], Hussein 等人 [18], 以及 Daponte [19]等人还提出了其他根区间定位方法。Sadrpour 等人 [20], Fried [21], Li 等人 [22] 以及 Kim 等人 [23]还设计了一些更为特殊的非线性系统零点检测算法。虽然这些传统方法中有些需要能够确定函数的导数，另一些则要求对根作出合理的初始猜测，才能获得较快的收敛速度，但无论各自优缺点如何，这些算法都需要给出函数表达式，因此无法从未知函数采样得到的离散时间序列中求得根。

工程领域已成功运用多种方法，借助二极管、比较器和滤波器等硬件元件，从时间序列中捕获过零点 [24, 25]。然而，构建用于寻找零交叉点的电子电路并非总是切实可行。相比之下，Molinaro 和 Sergeyev [26] 开发了一种基于估计信号利普希茨常数的算法式零点查找方法，与 Daponte 等人 [19, 27]的方法类似。该算法虽然计算速度快，但只能估计定义域内的第一个零交叉，遗漏函数的所有其余零点。此外，该算法要求定义域左边界处的函数值为正。

我们的工作通过提出一种新方法，利用应用拓扑学中的工具——零维持久性——对时间序列的零点进行区间定位，从而克服了这些重大缺陷。该算法运行快速，准确度高于可比方法，并能够对性质良好的信号中的所有零点进行区间定位。

2 持久同调

在本文中，我们将运用持久同调的思想，不过仅限于一个极其简单的情形：即实数集 $\mathbb{R}$ 中点云的零维持久性。因此，我们将只关注这一情形，并留待读者自行探索向更高维度的推广[28–30]。

持久同调的主要思想，是对不断变化的拓扑空间结构进行编码。在我们的情形中，数据是一组点 $P \subset \mathbb{R}$。我们可以想象以每个点为中心扩张区间（$a - \varepsilon/2, a + \varepsilon/2$），其中 $a \in P$，并观察随着 ε 增大，实数轴上的覆盖情况如何变化。具体而言，我们关注这些区间合并、使连通分支数量减少时 ε 的取值。由于 $\mathbb{R}$ 的简单性，只要对 $P = \{a_1 < a_2 < \cdots < a_n\}$ 的值排序，连通分支就会在 $a_{i+1} - a_i$ 处合并。因此，我们用集合 $\mathrm{dgm}(P) = \{a_{i+1} - a_i \mid i = 1, \cdots, n-1\}$ 来表示点集连通分支的变化。对于熟悉相关领域的读者，我们要致歉：我们将一个不完全符合文献定义的对象称为持久性图。不过，我们认为这样称呼是合理的，因为它实际上是严格意义上的零维持久性图中的死亡时间集合，而且在这一情形下，所有出生时间均为 0。

以图1为例。点云示例 $P = \{a_1, \cdots, a_n\}$ 展示在顶行。我们可以计算 $\mathrm{dgm}(P)$，并用两种方式之一将其可视化。首先，由于 $\mathbb{R}$ 具有自然的

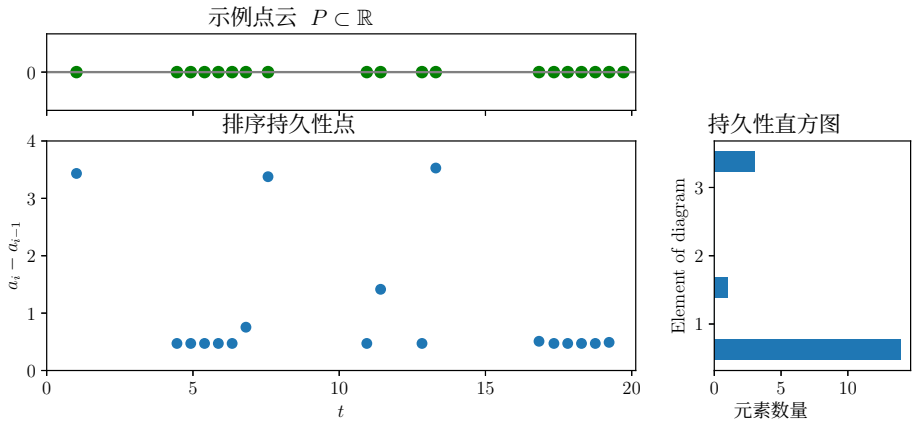

图 1：图的顶部给出了实数集 $\mathbb{R}$ 中的一个点云示例。左下方给出了持久性图中的点（$a_i - a_{i-1}$），绘制在 a_{i-1} 值对应的位置。请注意，该图中的高值点出现在点云分裂处。右侧则展示了持久性图中（死亡时间）点的直方图。

全序，我们可以根据用于构造全序的坐标对这些点进行排序，从而得到左下图中的有序点。如果不需要排序信息，也可以像右侧面板中那样，将这些信息可视化为持久性直方图。我们通常会关注高持久性点；也就是说，持久图 dgm(P) 中超过某个阈值 μ 的离群值。排序图中的高持久性点对应于点云中原始点云相邻点之间间距较大的位置。我们可以利用直方图来确定 μ，寻找较大的跃变。

在图 1 的示例中，选择 $\mu = 2$ 附近任意位置的阈值，都会将三个点标记为高持久性。点云中位置之间的三个间隔（通常对应图中的 k 个点）对应原始数据中的四个（通常为 $k+1$ 个）簇。在下一节中，我们将把持久性图这一概念用于从信号中提取的两个一维点云，以界定预期出现零交叉的位置。

3 方法

本节介绍计算方法，并给出一种基于零维持久同调的算法，用于检测含噪信号中的零交叉点。假设给定一个离散时间序列 $x(t_0)$ $x(t_1)$ $\cdots, x(t_N)$，其时间为 $\{t_0 < t_1 < \cdots < t_N\}$。我们关注实数集 $\mathbb{R}$ 中的两个点云：$x(t)$ 为正和为负时对应的时间。出于技术原因，我们将端点添加到每个集合中。将它们分别记为 $P = \{t_i \mid f(t_i) > 0\} \cup \{t_0, t_N\} = \{p_1 < \cdots < p_\ell\}$ 和 $Q = \{t_i \mid f(t_i) < 0\} \cup \{t_0, t_N\} = \{q_1 < \cdots < q_m\}$。然后，我们按照第 2 节所述计算两个持久图 dgm(P) $= \{p_i - p_{i-1} \mid i = 1, \cdots, \ell - 1\}$ 和 dgm(Q) $= \{q_i - q_{i-1} \mid i = 1, \cdots, m - 1\}$。图 2 展示了这些记号的示意图。

固定一个阈值 μ。基于该阈值，我们可以提取 dgm(P) 和 dgm(Q) 中高于 μ 的值。也就是说，令 $\text{dgm}(P)_\mu = \{a \in \text{dgm}(P) \geq \mu\}$ 并类似地定义 $\text{dgm}(Q)_\mu$。

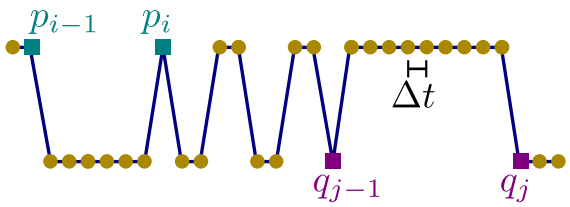

图 2：展示 p 和 q 记法的脉冲信号。 P 点位于上方一行； Q 位于下方。对于阈值 $\mu = 3\Delta t$，我们看到来自 dgm P的条目（p_{i-1}， p_i），以及来自 dgm Q的条目（q_{j-1}， q_j）。

在此设定下，我们关注移除区间后 $\mathbb{R}$ 中剩余的区域

$$\mathcal{I}_P = \{(p_{i-1}, p_i) \mid p_i - p_{i-1} \in \mathrm{dgm}(P)_\mu\}$$

and

$$\mathcal{I}_Q = \{(q_{i-1}, q_i) \mid q_i - q_{i-1} \in \mathrm{dgm}(Q)_\mu\}.$$

具体而言，将这些区间排序并交错排列后，直线的剩余部分可以写作

$$\mathbb{R} \setminus \bigcup_{I \in \mathcal{I}_P \cup \mathcal{I}_Q} I = \bigcup_j [r_{2j-1}, r_{2j}]$$

其中，每个 r_{2j-1} （分别为 r_{2j}）都是 $\mathcal{I}_P$ 或 $\mathcal{I}_Q$中某个区间的右端点（分别为左端点）。我们还可以假设这些端点已排序，因此将端点集记为 $R = \{r_1 < \cdots < r_{2i} < r_{2i+1} < \cdots < r_{2K}\}$。

$\mathcal{I}_P$ 和 $\mathcal{I}_Q$ 中的区域是 $\mathbb{R}$ 中我们非常确信函数保持同一符号的部分，因此，直线的剩余部分就是我们预期会找到零交叉点的位置。基于此，我们的算法将区间 $[r_{2i-1}, r_{2i}]$ 作为可能发生过零的位置返回。此外，如果两个 r 点分别来自 P 和 Q，那么根据介值定理，我们相当确信它们之间存在一个零交叉。

命题 3.1。 如果这些值取样自连续函数 f，那么任意端点类型不同的区间内至少存在一个过零点。

证明。 设所讨论的区间为 $[a, b]$，不失一般性，我们可以假设 $a = f(t_a)$ 是一个 P-区间的较高值，而 $b = f(t_b)$ 是一个 Q-区间的下界。根据 P 点和 Q 点的定义，这意味着 $a > 0$ 且 $b < 0$。因此，根据介值定理，存在一个 $t \in [t_a, t_b]$，使得 $f(a) = 0$。 □

当端点类型相同时，能得出的结论较少。事实上，在简单情况下，可能出现的情形是 0 个或 2 个过零点，但任何偶数个过零点都有可能。因此，我们的算法（算法 1）会返回一个可能包含零交叉点的区间列表，并为以下区间添加不确定性标记：区间两侧均由相同类型的端点界定，或区间来自边界。

找到这些区间后，可以采用多种方法来确定过零点的可能值，例如对每个区间使用二分法，或对其进行线性插值以求得根的估计值。在我们的计算中，我们使用每个返回区间两个端点的平均值来估计根的位置。完整方法总结见算法 1。

算法 1： 用于界定零交叉点的持久性算法

。数据：具有时间 $\{t_1 < \cdots < t_N\}$ 的时间序列 $x(t_1) \quad \cdots, x(t_N)$，持久性阈值 $\mu \geq 0$.结果：包含信号 $x(t)$ 潜在过零点的区间。计算 $P = \{p_i \,|\, \text{sgn}(x(p_i)) > 0\} \cup \{t_0, t_N\}$ 和 $Q = \{q_i \,|\, \text{sgn}(x(q_i)) < 0\} \cup \{t_0, t_N\}$。对列表排序，分别记为 $P = \{p_1 < \cdots < p_\ell\}$ 和 $Q = \{q_1 < \cdots < q_m\}$。在每个图中保留持久性高于 μ 的配对 $\mathcal{I}_P = \{(p_i, p_{i+1}) \mid p_{i+1} - p_i \geq \mu, i = 1, \cdots, \ell - 1\}$ 和 $\mathcal{I}_Q = \{(q_i, q_{i+1}) \mid q_{i+1} - q_i \geq \mu, i = 1, \cdots, m - 1\}$。交错排列并对区间 $R = \{r_0 < \cdots < r_{2i} < r_{2i+1} < \cdots < r_{2K-1}\}$ 的端点排序。如果端点值尚未包含在内，则将其添加到 R $R = \{t_0 = r_{-1} \leq r_0 < r_1 < \cdots < r_{2i} < r_{2i+1} < \cdots < r_{2K-1} \leq r_{2K} = t_N\}$ 返回区间

$$\{[r_{2i-1}, r_{2i}] \mid r_{2i-1} \neq r_{2i},\ i = 0, \cdots, K\}$$

如果端点并非来自不同类型的区间，则会标记不确定性标记。

3.1 示例

我们用以下多频正弦波演示该算法：

$$x(t) = \sin(t) + \sin(10t/3). \tag{1}$$

如图 3所示。我们在公式 (1) 中加入加性白噪声，并将所得信号转换为脉冲波。以下位置的 t 值位于 P 和 Q 中，显示在原始信号的上方和下方。请注意， t_0 和 t_N 处的蓝色三角形是在事后添加到 P 和 Q 中的，用于处理边界。

信号下方是该示例完整的排序持久性图。对于以虚线表示的 $\mu = 0.4$ ，所得区间为

$$\text{dgm}_\mu(P) = \{(r_2, r_3), (r_4, r_5)\}$$
$$\text{dgm}_\mu(Q) = \{(r_0, r_1), (r_6, r_7)\}$$

我们可以扩展 R，将端点 $R = \{r_{-1} = r_0 < \cdots < r_7 \leq r_8\}$ 纳入其中。算法返回的未带不确定性标记的区间，即端点并非同时位于 P 或 Q 中的区间，为 $[r_1, r_2]$ 和 $[r_5, r_6]$。其余带有不确定性标记的区间为 $[r_3, r_4]$ 和 $[r_7, r_8]$。

根据阈值 μ的选择，算法会返回不同的区间集合。如果我们选择稍高一些的阈值 $\mu = 0.6$，点 r_6 和 r_7 就不会被纳入零交叉的潜在边界，因此会改为提供区间 $[r_5, r_8]$ ，并附带不确定性标记。

3.2 选择阈值 μ

算法1利用了夹住根的点具有更高持久性的思想。通过将 dgm(P) 和 dgm(Q) 绘制在一起，可以对其进行可视化量化。该算法

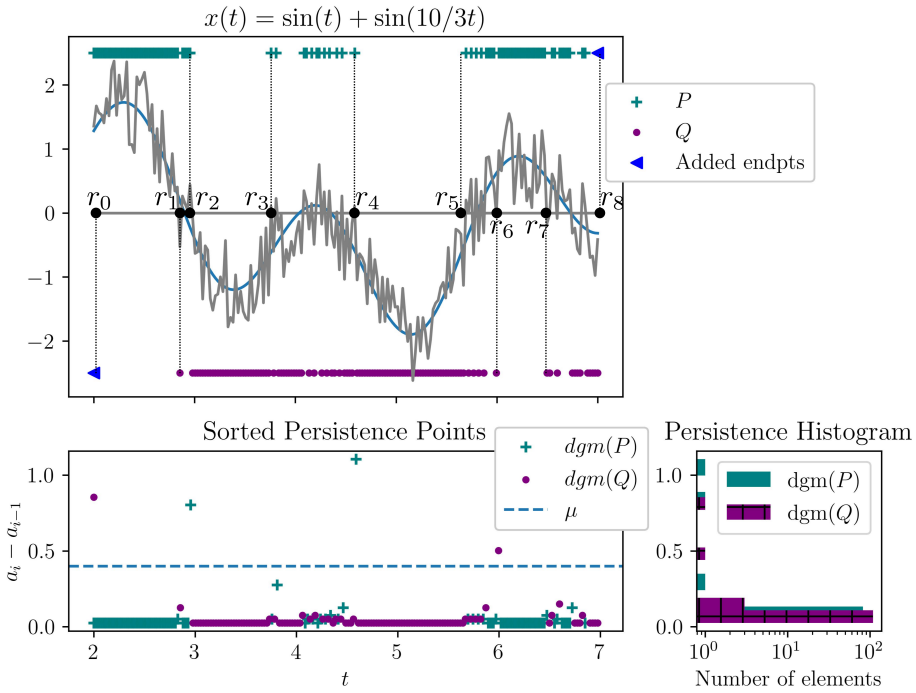

图3：使用 $x(t) = \sin(t) + \sin(10/3t)$ 的示例。算法返回的区间为 $[r_1, r_2]$ 和 $[r_5, r_6]$（没有不确定性标记）；以及 $[r_3, r_4]$ 和 $[r_7, r_8]$（带有不确定性标记）。右侧显示了持久性图。阈值选为 $\mu = 0.4$，但选择不同的 μ 会得到不同的零交叉区间。

提出使用一个数值 μ 来设定此持久性阈值，但从示例中可以看出，所得区间相当依赖于 μ 的取值。

在特别受限的设置下，有些情况下离群值的确定尤其简单。第一种情况是信号没有噪声，此时两个均匀采样点之间的时间差 $1 \cdot dt$ 是一个无误差的持久性阈值。第二种情况是已知区间内根的数量。若区间内有 n 个根，则持久性图中最高的 $n+1$ 个点代表所需的区间。

然而，当我们没有这样理想的输入数据时，就必须寻找其他方法来以数学方式确定μ 。由于在这些情况下高持久性点出现的频率较低，且差异可以量化，因此可以将它们视为数据集中的离群值。因此，可以应用统计和机器学习离群值检测方法来识别它们。这里，我们介绍两种检测高持久性点的方法，并将在下一节的实验中使用。检测离群值最简单的方法是采用统计度量。我们以表 x_9 1中的信号所计算出的图为例，展示这种方法。图 μ展示了均值$\mu+3\sigma$、3 的 z 分数（[31,32]）、1.5 倍四分位距（IQR）[31,32],、中位数 $\tilde{x}$ 以及中位数绝对偏差（MAD）[31] ——这些值都曾被用作数据集中离群值检测的截断值。从这个例子可以看出， $\mu + 3\sigma$ 是对持久性阈值的可靠估计。

最后，我们探讨机器学习算法孤立森林 [33] 在排序后的

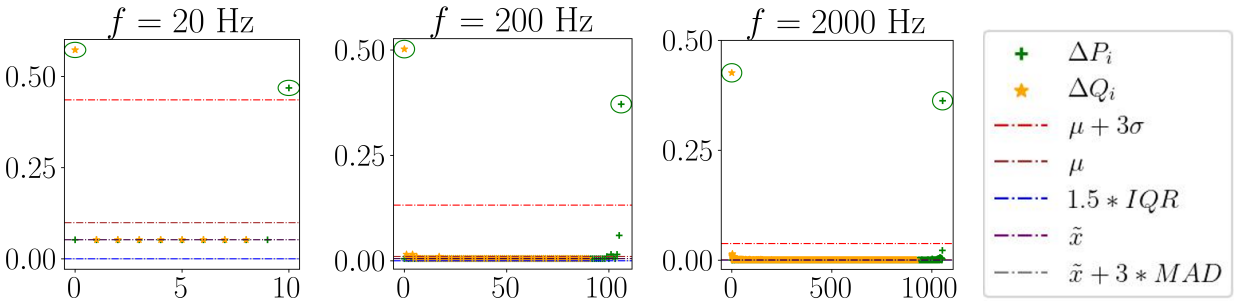

图 4： x_9 （使用信噪比 = 15 dB）的排序持久性点，并绘制了用于异常值检测的统计度量。

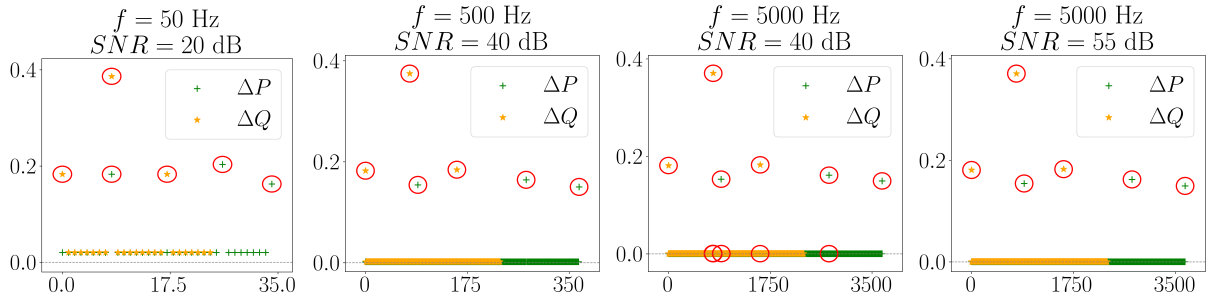

图 5：对 x_4的排序持久性点应用孤立森林。

持久点 x_4 表1。圈出的点是该方法返回的点，而真实根所需的点是从上往下的前六个点。图5 表明，孤立森林适用于所有信噪比下的低采样频率；但随着采样频率升高，需要更干净的信号（更高的信噪比）。例如，图中显示，该方法在噪声为 40 dB、频率为 5000 Hz 时失效，但在相同频率下，噪声为 55 dB 时则能找到所需的点。

在图 4中，我们注意到，随着采样频率降低， 3σ 阈值会接近一个所需的离群值；如果频率进一步降低，阈值就会超过该离群值。这表明，与低采样频率相比，该方法在高采样频率下效果更好。另一方面，如前所述，随着采样频率提高，孤立森林需要信号越来越干净。因此，在处理含噪信号时，一般而言，对于高频采样的信号，应优先采用前一种方法；对于采样率较低的信号，则应采用后一种方法。

4 结果与讨论

我们使用表 1 中的算法，在一系列采样频率和信噪比（SNR）下，对求函数的根进行了一系列实验。信噪比以分贝为单位，其定义见式（2）。

$$\mathrm{SNR} = 10 \cdot \log_{10} \frac{P_s}{P_n} \tag{2}$$

其中 P_s 和 P_n 分别表示干净信号和噪声的功率。图 6 展示了

	$f(t)$	过零点	区间
1	$x_1(t) = (1/6)t^6 - (52/25)t^5 + (39/80)t^4 + (71/10)t^3 - (79/20)t^2 - t + 1/10 + 1000$	4.052	[-1.5, 5]
2	$x_2(t) =$ 正弦$(t) +$正弦$(10/3t)$	2.9, 4.039, 4.3499, 5.79986, 6.73198, 7.2598	[2.7, 7.5]
3	$x_3(t) = \left(-16t^2 + 24t - 5\right)e^{-t} + 3$	2.064	[1.9, 3.9]
4	$x_4(t) = (-3t + 1.4)$ 正弦$(18t) + 0.1$	0.181, 0.3349, 0.7059, 0.868, 1.0504	[0, 1.2]
5	$x_5(t) =$ 正弦$(t) +$正弦$(2/3t)$	3.77, 7.54, 9.425	[3.1, 11]
6	$x_6(t) = -t \cdot \sin(t) + 0.5$	0.741, 2.973, 6.362	[0, 8]
7	$x_7(t) = -(2\cos(t) + \cos(2t))$	-1.196, 1.196, 5.087	[-1.57, 6.28]
8	$x_8(t) =$ 正弦$^3(t) +$余弦$^3(t)$	2.356, 5.5	[0, 6.28]
9	$x_9(t) = -t^3 + \left(t^2 - 1\right)^6$	0.525	[0.001, 0.99]
10	$x_{10}(t) = -e^{-t}\sin(2\pi t) + 0.5$	0.092, 0.371	[0, 4]
11	$x_{11}(t) = (t^2 - 5t + 6)/(t^2 + 1)$	2, 3	[-5, 3]
12	$x_{12}(t) = \begin{cases} (t-2)^2 & \text{若 } t \le 3 \\ 2\ln(t-2) + 1 & \text{否则} \end{cases}$	2	[0, 6]
13	$x_{13}(t) = -t +\sin(3t) + 1$	1.035	[0, 6.5]
14	$x_{14}(t) = -(t -$ 正弦 $t) * e^{-t^2} + 0.01$	0.4159	[-2, 2]

表1：用于评估所提出方法鲁棒性的示例 [26]。

算法返回的区间以及表 x_2 和 x_5 所示无噪声函数的真实根（表 1 ），采样频率较低，为 25 赫兹。这些图表明，该算法能够捕捉给定区间内的多个根。然而，函数 x_{11} 和 x_{12} 揭示了该方法固有的一些缺陷，即使在没有噪声的情况下也是如此。首先，如果根位于边界上，该方法就很难找到它，例如 x_{11}中的例子。另一个问题是，对于无噪声数据，如果零点处的二阶导数为正，该零点可能完全无法被检测到。在 x_{12}中，如果零点本身没有被实际采样，就不会检测到零点。有趣的是，在这种情况下，如果系统中存在噪声，我们反而能更好地检测到过零点。其余示例见附录 B。

图7 展示了在信号中添加不同信噪比的高斯噪声后，对表1中函数 x_8 和 x_{14} 在采样频率为 1000 Hz 时，算法返回的区间所产生的影响。图中显示，无论信噪比如何，算法都能有效地为所有根确定区间。然而，当信噪比低到足以导致信号中出现人工交叉点时，算法可能会返回更多的根（例如， x_{14} 在信噪比为 15 dB 时），也可能返回更少的根（例如， x_2 在信噪比为 15 dB 时，如图8所示）。其余情况见附录C。

为了超越个别示例，我们开展了一系列实验，以全面比较本文提出的算法与 Molinaro 和 Sergeyev 开发的先进软件零点检测算法 [26]。该方法为时间序列构造支撑函数，并利用局部 Lipschitz常数的估计值，在区间内近似求出第一个零交叉。作者已证明，对于离散信号中的这一问题，他们的算法明显优于简单的网格法。附录 A 详细介绍了他们的

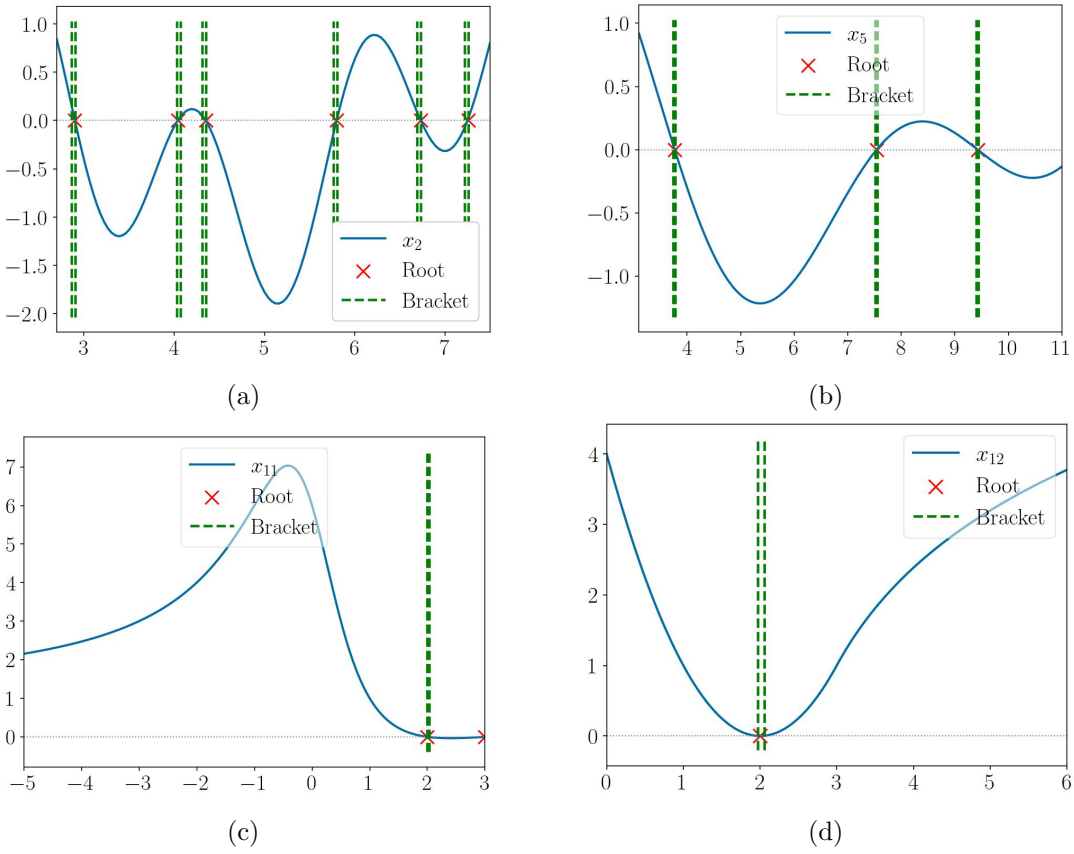

图6：采样频率为 25 赫兹时，(a) x_2、(b) x_5、(c) x_{11} 和 (d) x_{12} 的真实根与返回的区间

算法。由于该算法只返回区间内的第一个零交叉，因此我们的比较仅考察第一个根，并将收敛判据 ε 设为均匀采样序列中两个相邻时间值之差。所采用的收敛判据为

$$\varepsilon = \frac{t_b - t_a}{n-1} \tag{3}$$

其中，$n = f(t_b - t_a)$ 是以 f 的采样频率从区间 $[t_a, t_b]$ 中采样得到的离散值数量。

图 9 展示了对所有考察的函数，使用两种算法得到的相对误差对比，以及信噪比为 45 dB、30 dB 和 15 dB 的信号在 500 赫兹频率下采样时的收敛所需时间。条形图显示，总体而言，Molinaro 和 Sergeyev 的算法产生的相对误差显著高于所提算法。此外，显然，相对误差总体上会随着信噪比的提高而降低。

图10展示了在一系列信噪比和采样频率范围内，使用两种算法计算函数 x_2时的相对误差和收敛所需时间热力图。图中表明，所提算法产生的最大相对误差为 $\sim 3.5\%$，而 Molinaro算法的最大相对误差为 $\sim 6\%$；同时，Molinaro算法所需的最长时间是

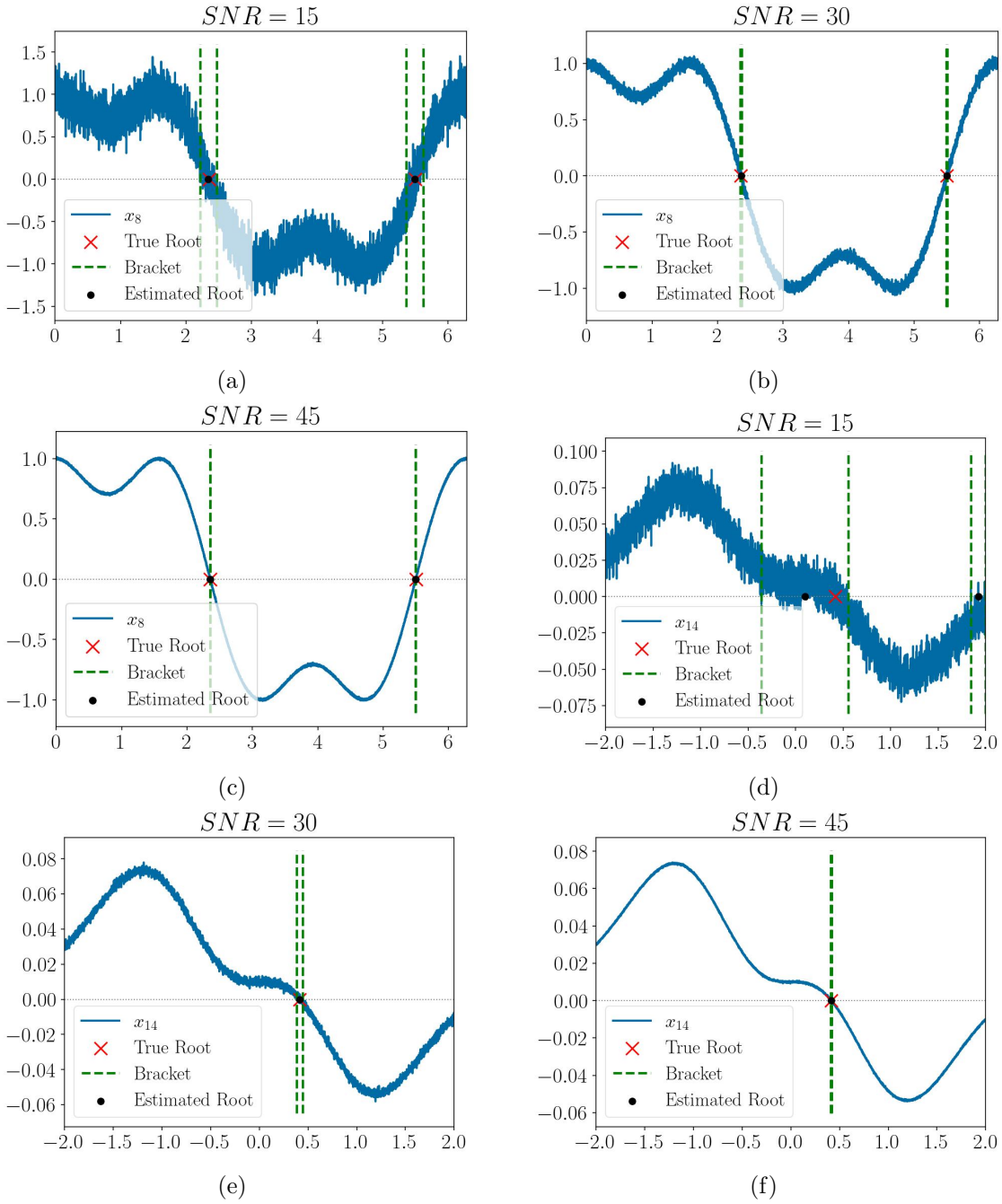

图7：在低、中、高信噪比下，以 1000 Hz 的采样频率对 x_8和 x_{14} 进行处理后得到的真实根与估计根

本例中所提算法所需时间的两倍。这两个方面进一步印证了所提算法效率更高这一结论。其余案例见附录D。

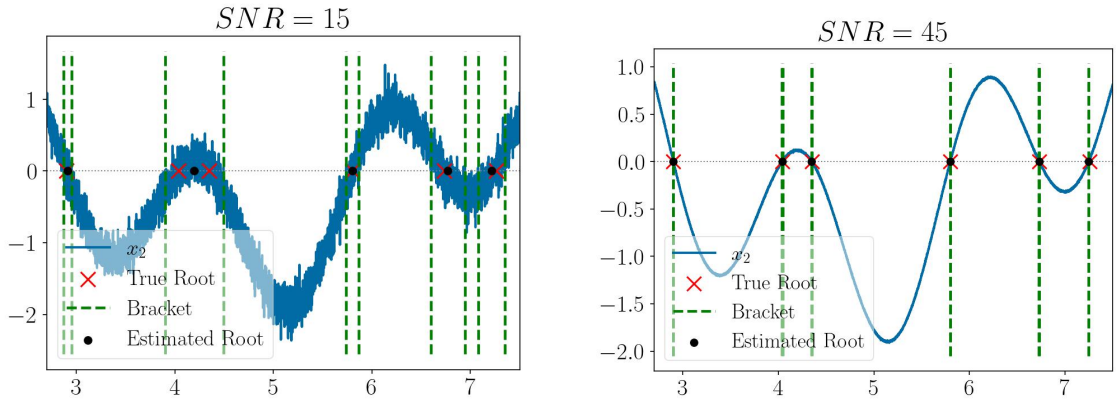

图 8：采样频率为 1000 Hz 时， x_2 的真实根与估计根。部分 (a) 显示，算法因噪声过高而漏掉了一个根。

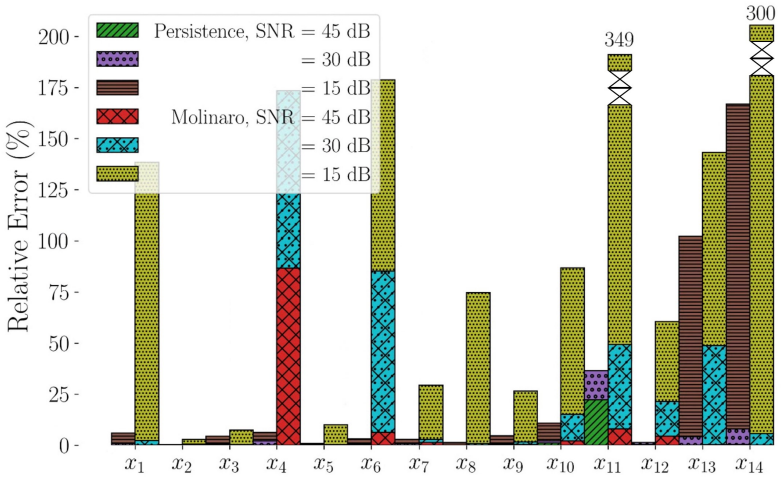

图 9：两种算法在 $f = 500$ Hz 下的低、中、高信噪比对应的相对误差

5 结论

求信号的根这一数学问题意义重大，在科学和工程领域有许多应用。然而，文献中缺乏能够确定函数所有根的技术，尤其是针对离散时间序列的技术。一些算法只有在已知函数表达式时才能求根，而另一些算法则只会收敛到给定区间内的单个根，忽略可能存在的其他根。

为克服这些局限性，有人提出了一种用于识别离散时间信号过零点的新算法。该算法基于从信号中获得的二值化脉冲的零维持久性。脉冲中与正确零点区间相对应的点具有一个显著特征：持久性较高。由于这些点持久性高而频率低，因此可以将其视为离群值，并利用广受欢迎的 z 分数和孤立森林方法高效地检测出来。一般而言，z 分数技术

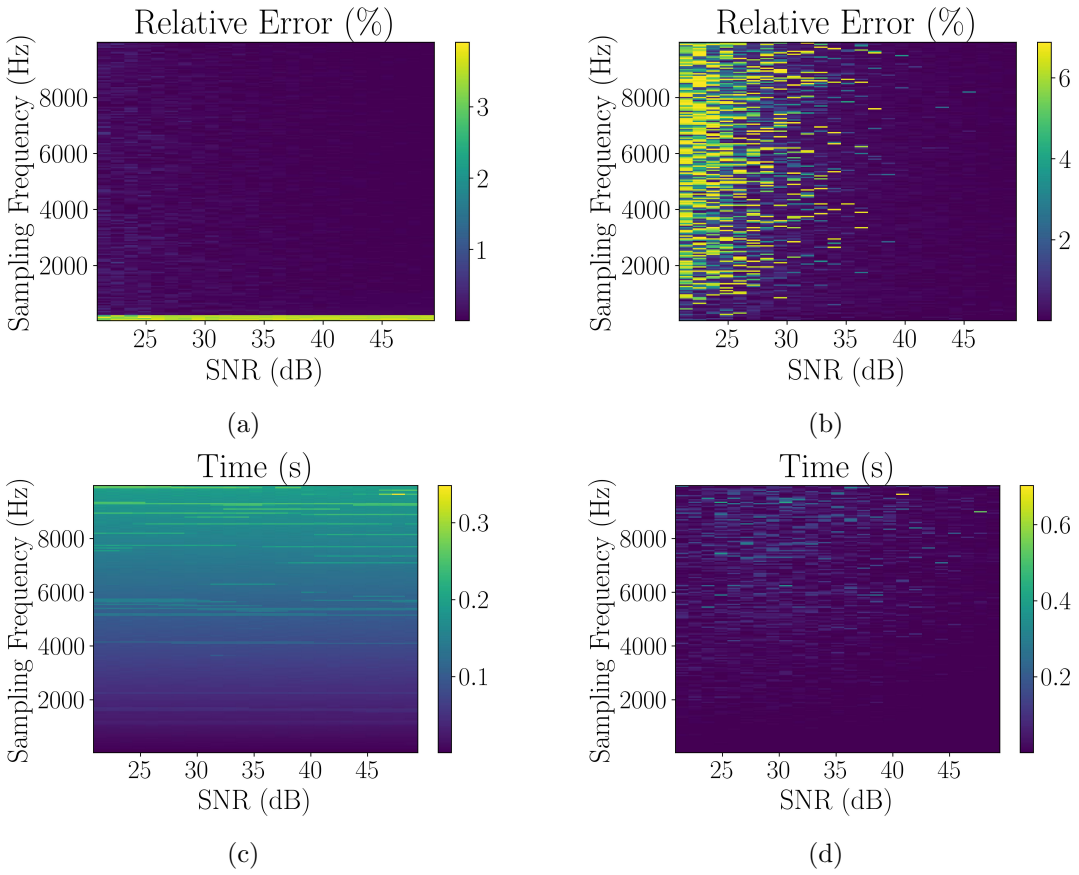

图10：0维持久性（a、c）和 Molinaro算法（b、d）的相对误差与收敛所需时间，针对 x_2

更适用于中高采样频率，而孤立森林则更适用于低采样频率或小型数据集。

此外，结果表明，该算法在多种情况下都具有良好的抗噪声能力，但在噪声产生人工交叉点，或无噪声数据中的根恰好位于x轴或区间边界上时除外。不过，显然这些局限性是大多数零点区间算法所固有的。

最后，我们将该算法与 Molinaro 和 Sergeyev 提出的一种用于检测离散信号中过零点的现有软件技术进行了比较 [26]。比较结果表明，所提算法通常速度更快、精度更高，并且能够返回给定区间内的所有过零点。与 Molinaro算法相比，所提算法在某些情况下可能需要更长时间才能收敛；但由于作为对比的算法仅返回区间内的第一个过零点，而我们的算法会返回所有过零点，因此无法对这两种算法进行公平的计时比较。总之，所提过零算法是一种可靠且强大的工具，在科学和工程领域具有众多潜在应用。

6 致谢

本材料基于空军科学研究办公室资助的工作，资助编号为FA9550-22-1-0007。

参考文献

[1] M. B. Duric 和 Z. R. Durisic。利用傅里叶和过零技术测量存在谐波时电力网络中的频率。*IEEE*，2005。[2] Milenko B. Djuric 和 Zeljko R. Djurisic。利用傅里叶和过零技术测量畸变信号的频率。《电力系统研究》，2008。[3] Vladimir Friedman。白噪声中单个正弦波频率估计的过零算法。《*IEEE*信号处理汇刊》，42:1565–1569，1994。[4] Tadashi Masuda、Hisao Miyano 和 Tsugutake Sadoyama。利用梯度阈值过零法测量肌纤维传导速度。《*IEEE*生物医学工程汇刊》，BME-29:673–678，1982。[5] E. Bayat、N. Divani-Vais、M. M. Firoozabadi 和 N. Ghal-Eh。采用过零法和 NE213、UGLTT闪烁体进行中子-伽马甄别的比较研究。《辐射物理与化学》，2012。[6] Abhishek Ukil、Shuo Chen 和 Andrea Andenna。利用电机电流过零时刻检测三相感应电动机的定子短路故障。《电力系统研究》，2011。[7] R.M. Pindoriya、A.K. Mishra、B.S. Rajpurohit 和 R. Kumar。采用反电动势过零技术分析无传感器无刷直流电机的位置与速度控制。载于*2016*年*IEEE*第一届电力电子、智能控制与能源系统国际会议。IEEE，2016年7月。[8] P. Misans 和 M. Terauds。基于连续波多普勒雷达、利用过零法和最小二乘法测量陆地车辆速度的算法。载于*2012*年第*13*届波罗的海电子学双年会。IEEE，2012年10月。[9] S. Srinivasan 和 J. Ophir。用于弹性成像的过零应变估计器。《医学与生物学中的超声》，29(2):227–238，2003。[10] G. Raju。利用小波系数的零交叉识别无约束手写马拉雅拉姆字符。载于*2006*年先进计算与通信国际会议。IEEE，2006年12月。[11] H. M. Antia。《科学家与工程师的数值方法》。Birkh¨auser，2002。[12] Alojz Suhadolnik。求解非线性方程的组合区间夹逼法。《应用数学快报》，2012。

[13] Alojz Suhadolnik。求解非线性方程的超线性括区间法。《应用数学与计算》，2013。[14] M. A. Razbani。具有自适应网格细化的全局根括区间法。《应用数学与计算》，2015。[15] D. J. KAVVADIAS、F. S. MAKRI 和 M. N. VRAHATIS。高效计算函数的多个根。《科学计算期刊》，2005。[16] Vladimir Kodnyanko。用于非线性方程数值求解的改进抛物线括区间法。《应用数学与计算》，400，2021年7月。[17] Elsayed Badr、Hala Attiya 和 Abdallah El Ghamry。结合开方法与括区间法优点的全新根求解混合算法。《亚历山大工程期刊》，61:11579–11588，2022年12月。[18] Bahadin Muhammad Hussein、Bawar Mohammed Faraj、Shnyar Karim Rahman、Deni Adnan Mohammed、Muhammad Hussein、Azad Salam、Khadija Rzgar Mohammed、Bawar Mohammed、Berivan Azad Salam 和 Mohammed Faraj。基于 Ridders 方法的非线性方程数值解改进括区间法。《矩阵科学数学》，6:30–33，2022。[19] P. Daponte、D. Grimaldi、A. Molinaro 和 Ya.D. Sergeyev。求导数满足利普希茨条件的时间信号零交叉点的算法。《测量》，16(1):37–49，1995。[20] Amir Sadrpour、Luis G. Crespo 和 Sean P. Kenny。利用伯恩斯坦展开分析非线性系统。载于*AIAA*制导、导航与控制（*GNC*）会议。美国航空航天学会，2013年8月。[21] Isaac Fried。高阶迭代括区间法。《国际工程数值方法期刊》，94:708–714，2013年5月。[22] Zigang Li、Jun Jiang、Ling Hong 和 J. Q. Sun。用于多自由度非线性系统全局零点求解的子空间扩展技术。《应用数学与力学（英文版）》，41:769–784，2020年5月。[23] Sunyoung Kim、Masakazu Kojima 和 Kim Chuan Toh。用于简单锥优化问题的牛顿括区间法。《优化方法与软件》，第1–18页，2020。[24] R.W. Wall。检测零交叉的简单方法。载于*IECON'03. IEEE*工业电子学会第*29*届年会（*IEEE*目录号*03CH37468*），第3卷，第2477–2481页。IEEE，2003。[25] K. R. Sreenivasan、A. Prabhu 和 R. Narasimha。湍流信号中的零交叉点。《流体力学期刊》，137:251–272，1983年12月。[26] A. Molinaro 和 Ya.D. Sergeyev。数字化测量信号的高效零交叉检测算法。《测量》，30(3):187–196，2001年10月。[27] P. Daponte、D. Grimaldi、A. Molinaro 和 Ya. D. Sergeyev。快速检测测量信号集中的第一个过零点。《测量》，1996。

[28] Steve Y. Oudot。《持久性理论：从箭表示到数据分析（数学调查与专著）》。美国数学学会，2017年。[29] Tamal K Dey 和 Yusu Wang。《用于数据分析的计算拓扑学》。剑桥大学出版社，2021年。[30] Elizabeth Munch。拓扑数据分析用户指南。《学习分析期刊》，4(2)，2017年。[31] Sehar Saleem、Maria Aslam 和 Mah Rukh Shaukat。单变量离群值检测方法的综述与实证比较。《巴基斯坦统计学杂志》，37(4):447–462，2021年。[32] Maher M El-Masri、Fabrice I Mowbray、Susan M Fox-Wasylyshyn。单变量离群值：护士研究者的概念综述。《加拿大护理研究杂志》，51，2018年。[33] F. Pedregosa、G. Varoquaux、A. Gramfort、V. Michel、B. Thirion、O. Grisel、M. Blondel、P. Prettenhofer、R. Weiss、V. Dubourg、J. Vanderplas、A. Passos、D. Cournapeau、M. Brucher、M. Perrot 和 E. Duchesnay。Scikit-learn：Python 中的机器学习。《机器学习研究杂志》，12:2825–2830，2011年。

基于全局优化的过零点算法

算法 2： Molinaro 和 Sergeyev 提出的第一个过零点算法 [26]

数据： 一个时间序列 $f(t_1) \quad \cdots, \quad f(t_N)$，其时间为 $\{t_1 < \cdots < t_N\}$，满足$t^{(1)} = f(t_1)$ $t^{(2)} = f(t_N)$，且 $t^{(n+1)}$按以下方式求得，其中 n 为迭代次数，$t \in T_\delta$ 适用于 δ 网格。

结果： 信号 $f(t)$ 的第一个零交叉点或全局最小值。

将试探点 $t^{(j)}$，$1 \le j \le n$ 按照 $t_1 < t_2 < \cdots < t_k = t^{(n)} \le t_N$ 排序。对每个区间 $[t_{i-1}, t_i]$、$[t_i, t_{i+1}]$、$2 \le i \le N-1$，将局部 Lipschitz 常数估计为 m_i。

$$m_i = r \cdot max\{\lambda_i', \lambda_i'', \varepsilon\}$$

其中 $r > 1$ 是可靠性参数，ε 是一个较小的数，使得 $0 < \varepsilon \le 10^{-3}$，并且$\lambda'$，$\lambda''$ 按如下方式求得:

$$\lambda' = \begin{cases} max\{S_{k-1}, S_k\} & \text{if } k = 3 \text{ or } k = n \\ max\{S_{k-1}, S_k, S_{k+1}\} & \text{if otherwise} \end{cases}$$

$$S_i = |f(t_i) - f(t_{i-1})|/(t_i - t_{i-1}), \quad 2 \le i \le k$$

$$\lambda_i'' = \lambda_{max}^{(n+1)}(t_i - t_{i-1})/\Delta t_{max}^{(n+1)}, \quad 2 \le i \le k$$

$$\lambda_{max}^{(n+1)} = max\{S_i : 2 \le i \le k\}$$

$$\Delta t_{max}^{(n+1)} = max\{t_i - t_{i-1} : 2 \le i \le k\}$$

计算 R_i 和 τ^{th} 区间以计算 $t^{(n+1)}$，如下所示:

$$R_i = 0.5 \cdot \{f(t_i) + f(t_{i-1}) - m_i \cdot (t_i - t_{i-1})\}$$

$$\hat{t}_i = 0.5 \cdot \{t_i + t_{i-1} - (f(t_i) - f(t_{i-1}))/m_i\}$$

如果任一 $R_i \le 0$，则存在过零点。使用第一个 $R_i \le 0$，令 $\tau = i$ 和

$$t_\tau^* = t_{\tau-1} + f(t_{\tau-1})/m_\tau \qquad t^{(n+1)} = argmin\{|t_\tau - t| : t \in T_\delta$$

如果所有 $R_i > 0$，则求全局最小值。

$$\tau = min\{j : j = argmin\{R_i : 2 \le i \le k\}\}$$

$$t^{(n+1)} = argmin\{|\hat{t_\tau} - t| : t \in T_\delta\}$$

验证是否 $|t^{(n+1)} - t_{\tau-1}| \le \sigma$，其中 σ 是所需容差。如果不是，则在 $f(t^{(n+1)} \le 0$ 时，使用 $t_N = t^{(n+1)}$ 重复该过程；否则使用 $t_1 < \cdots < t^{(n+1)} < \cdots < t_N$。

返回 区间 $[t_1, t_N]$ 内的第一个过零点（如果存在）或全局最小值，容差为 σ。

B 过零区间

图 11 和 12 展示了该算法准确圈定任意时间序列中所有零交叉点的能力。

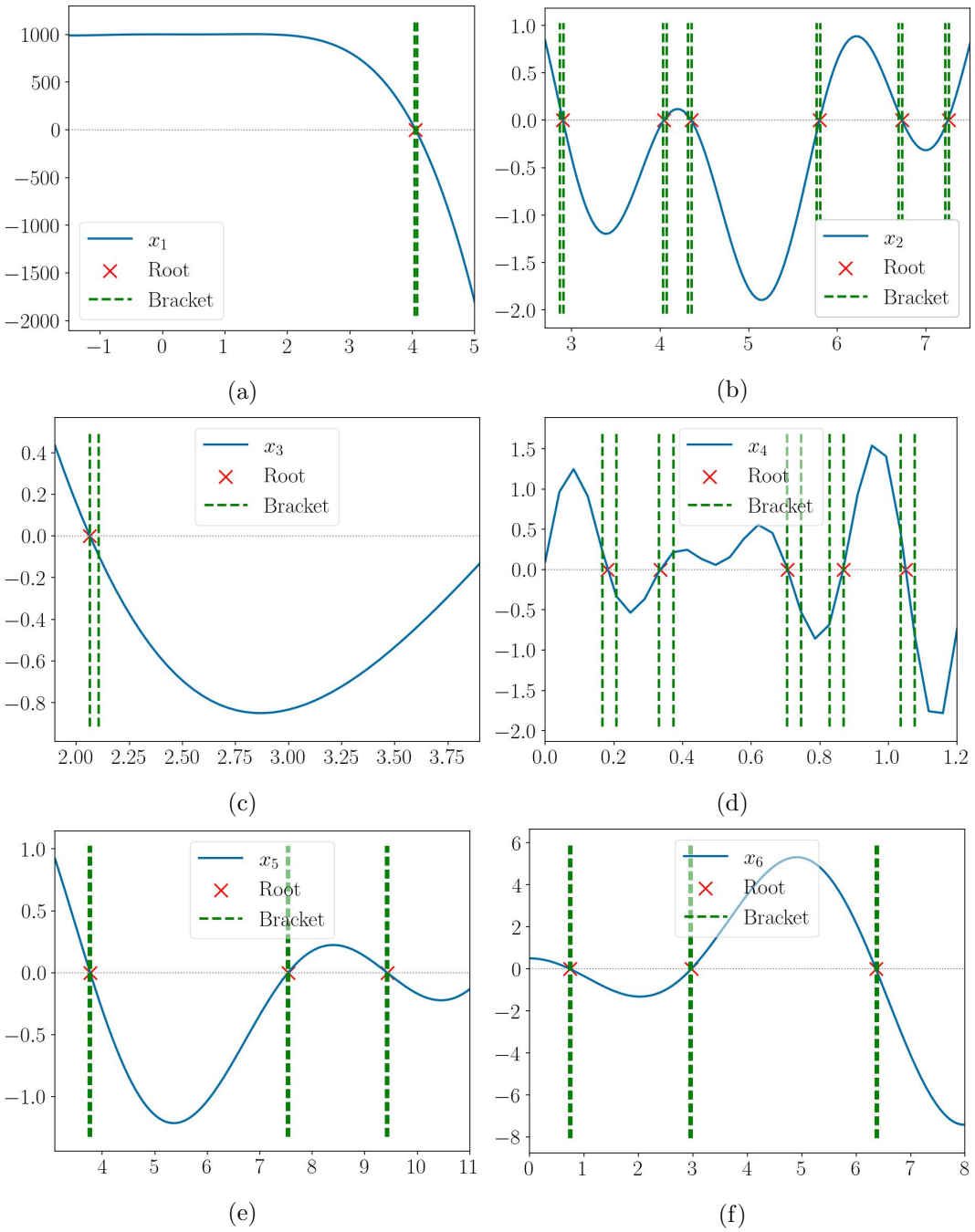

图 11：在较低的 25 赫兹采样频率下，算法针对函数 x_1 至 x_6 返回的真实过零点及其区间。

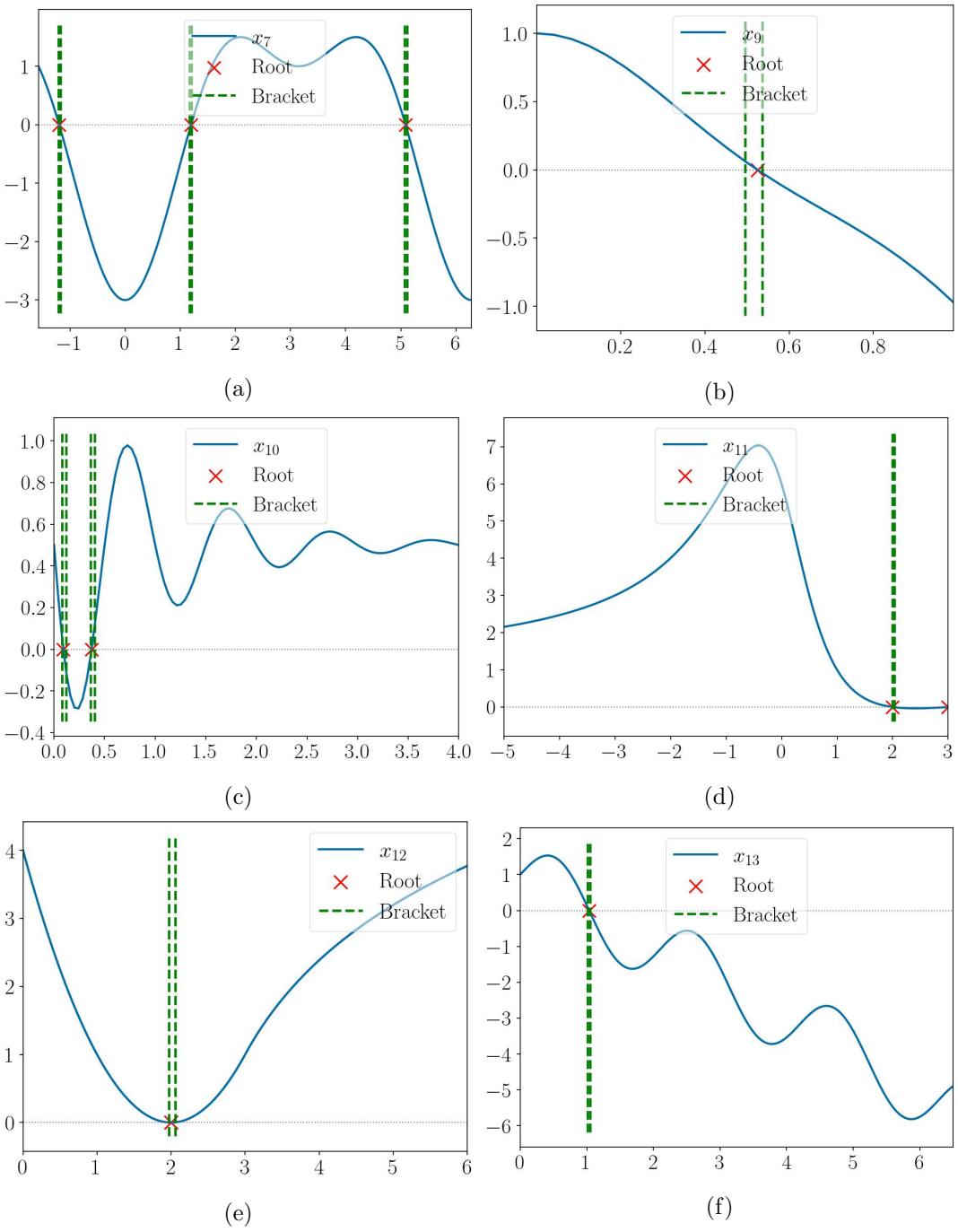

图 12：在 25 赫兹的低采样频率下，算法为函数 x_7 和 x_9 至 x_{13} 返回的真实过零点和区间。

C 不同信噪比下的根括区间

图 13 至 18展示了该算法对噪声的鲁棒性：即使在低信噪比下，也能以足够的精度估计根。

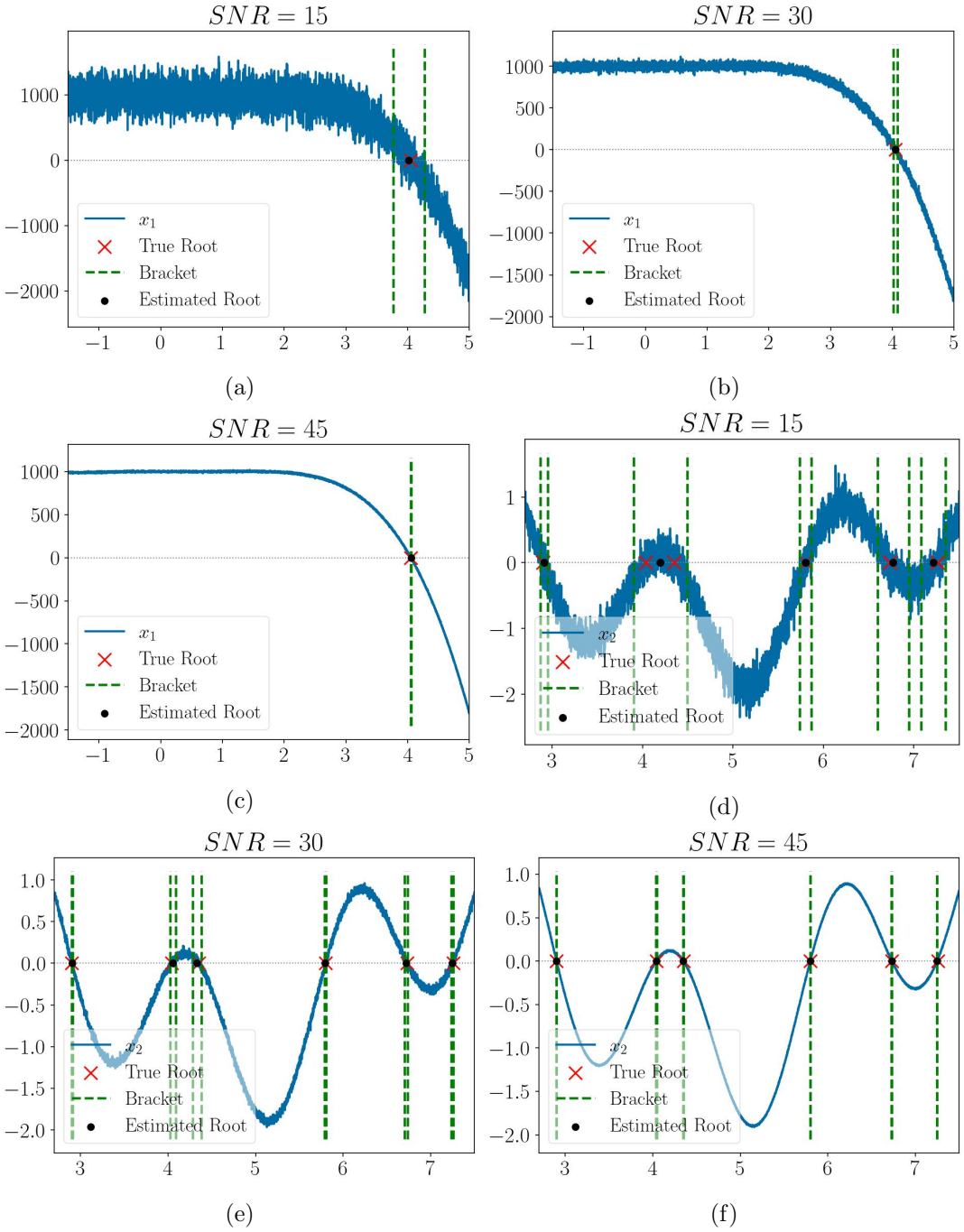

图 13：函数 x_1 和 x_2 在采样频率为 1000 Hz、低、中、高信噪比下返回的真实根、区间和估计根

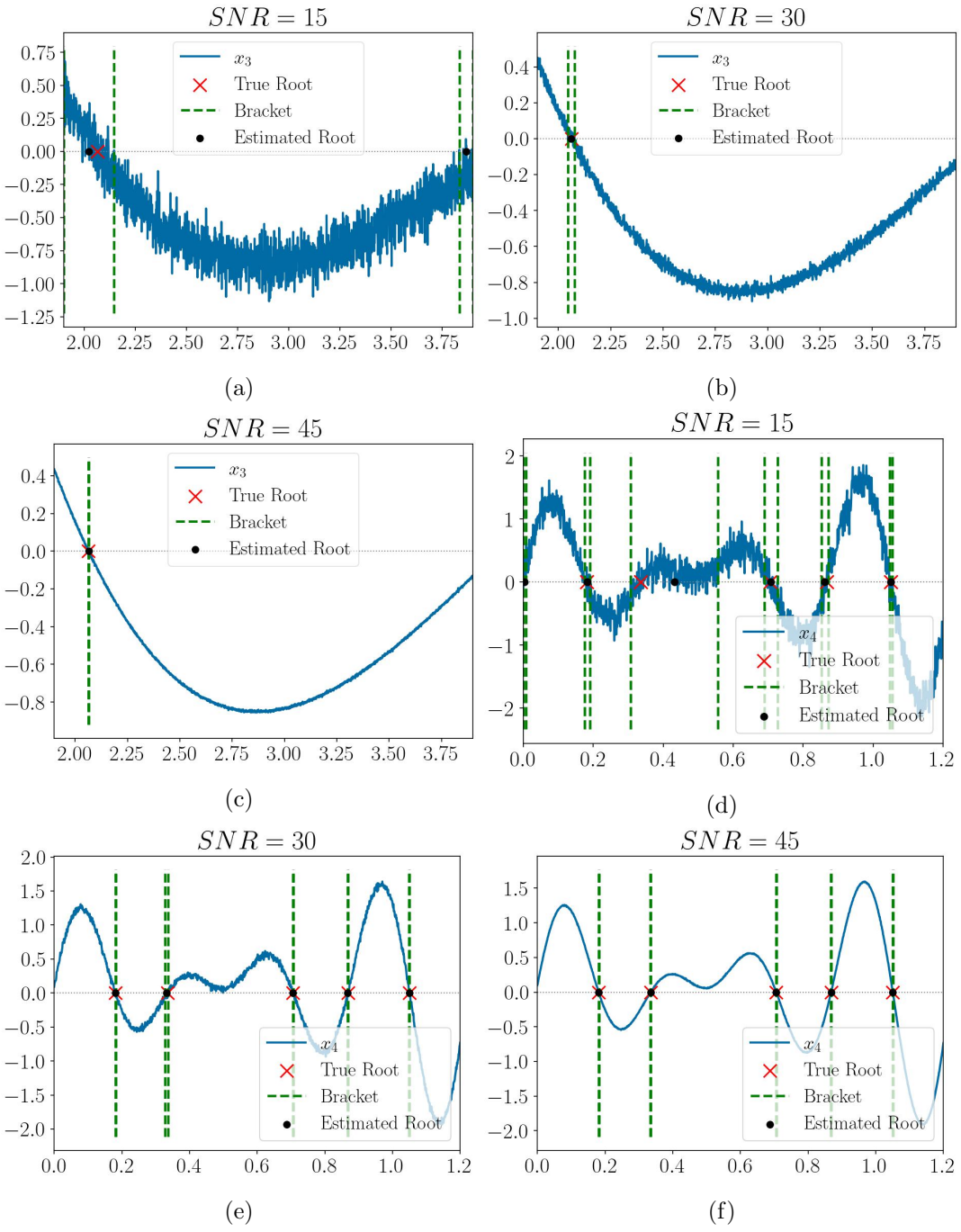

图14：函数 x_3 和 x_4 在低、中、高信噪比下以 1000 Hz 采样频率得到的真实根、区间和估计根

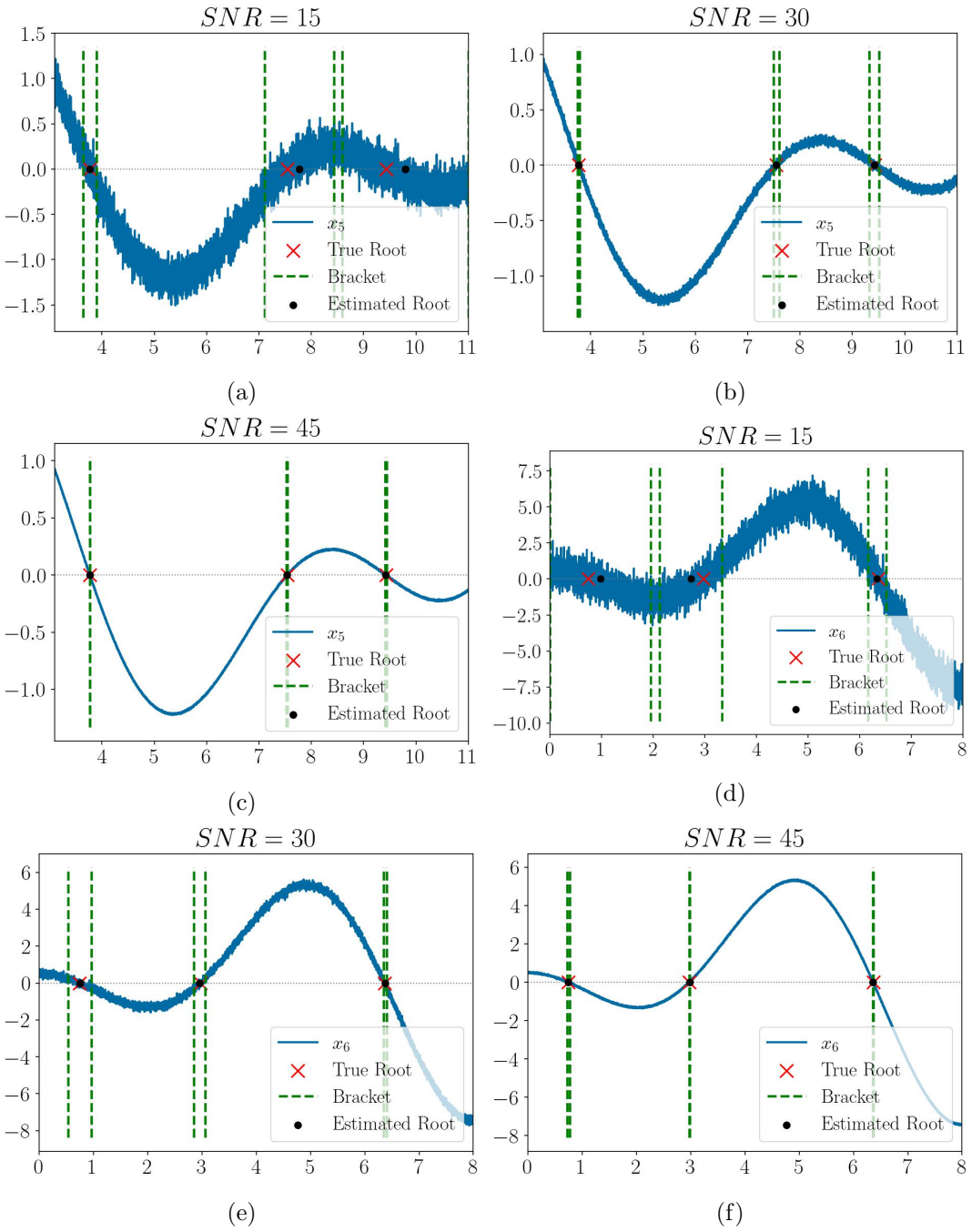

图15：函数 x_5 和 x_6 在低、中、高信噪比下，以 1000 Hz 采样频率返回的真实根、区间和估计根

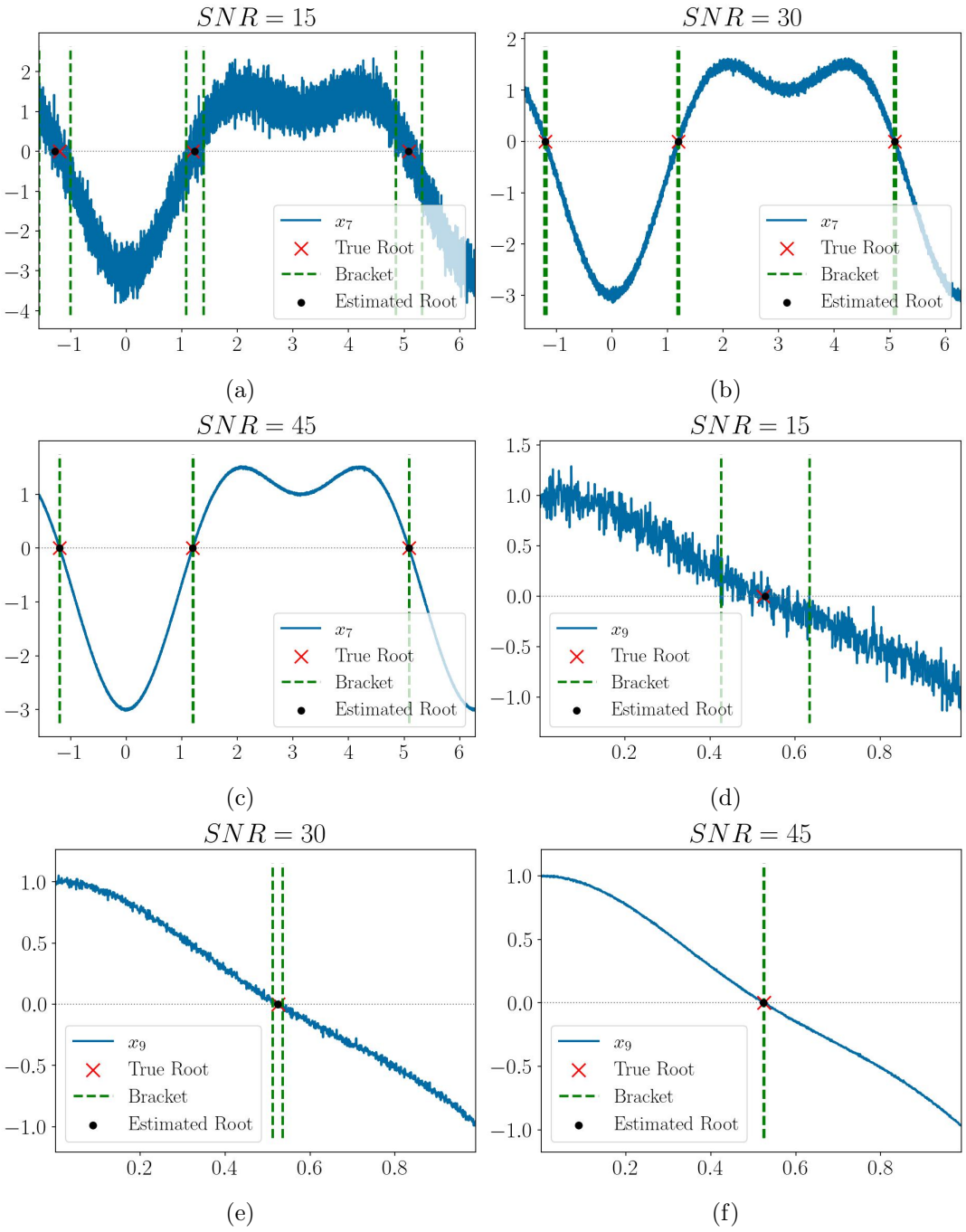

图16：函数 x_7 和 x_9在低、中、高信噪比下，以 1000 Hz 的采样频率得到的真实根、区间和估计根

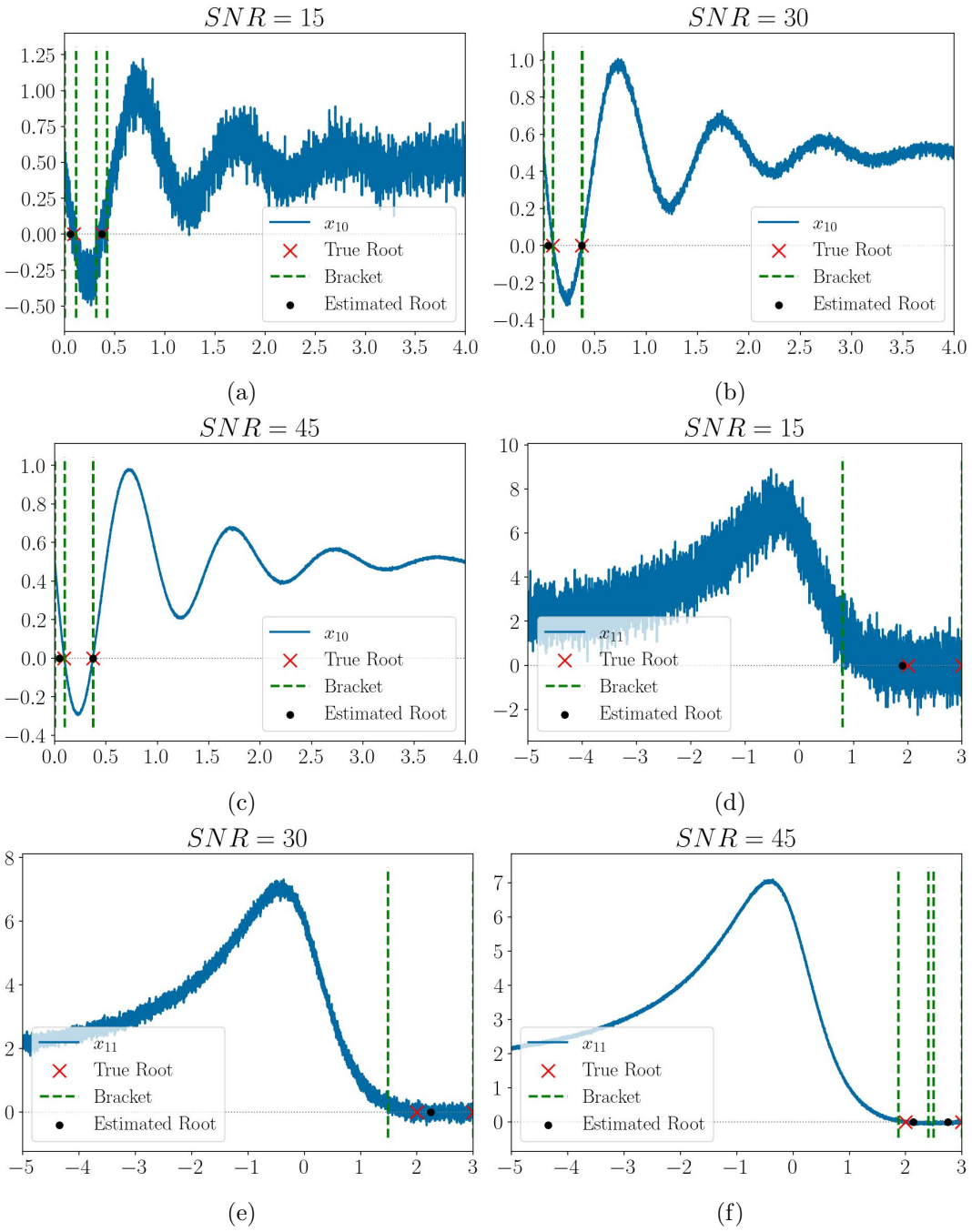

图17：函数 x_{10} 和 x_{11} 在低、中、高信噪比下以 1000 Hz 采样频率得到的真实根、区间和估计根

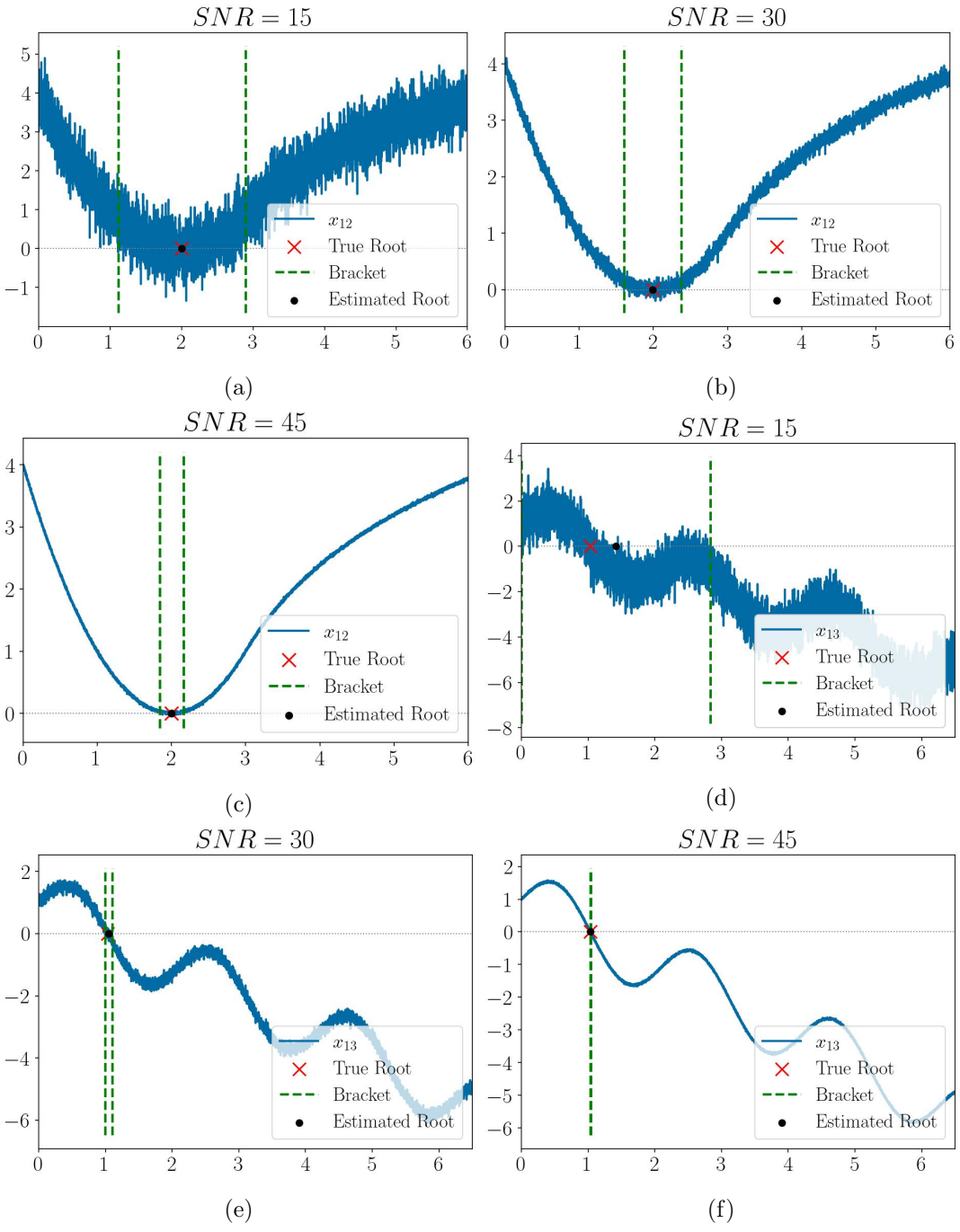

图18：函数 x_{12} 和 x_{13} 在低、中、高信噪比下，以 1000 Hz 采样频率得到的真实根、区间和估计根

D 相对误差与收敛时间

图 19显示，0维持久性算法的最大相对误差为 6%，而全局优化算法产生的最大误差超过 100%。同样，持久性算法的最长收敛时间为 0。15 秒，而全局优化算法则需要超过 200 秒。图 20 至 31 中所有函数的热力图也呈现出同样的差异，证明了 0维持久性算法的效率更高。

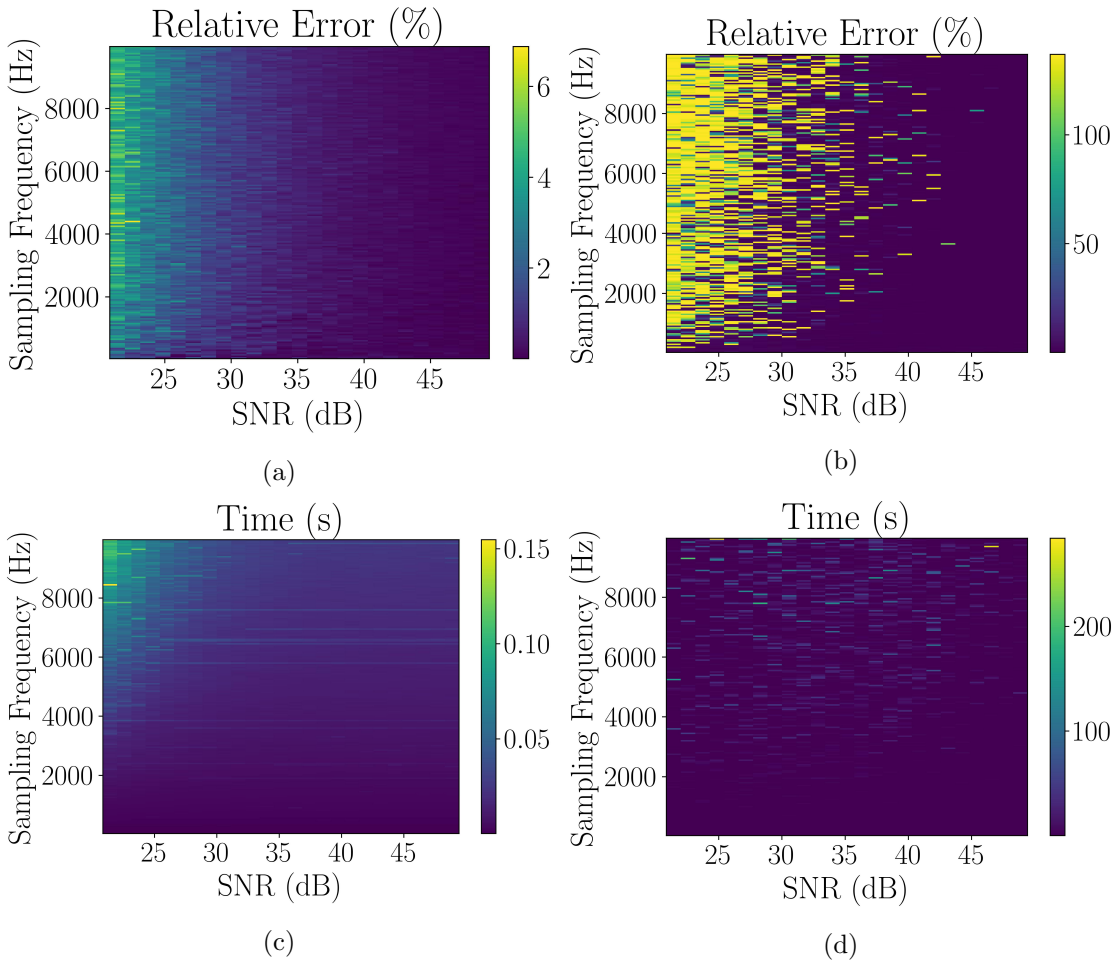

图 19：0维持久性算法（a、c）与 Molinaro算法（b、d）的相对误差和收敛所需时间，适用于 x_1

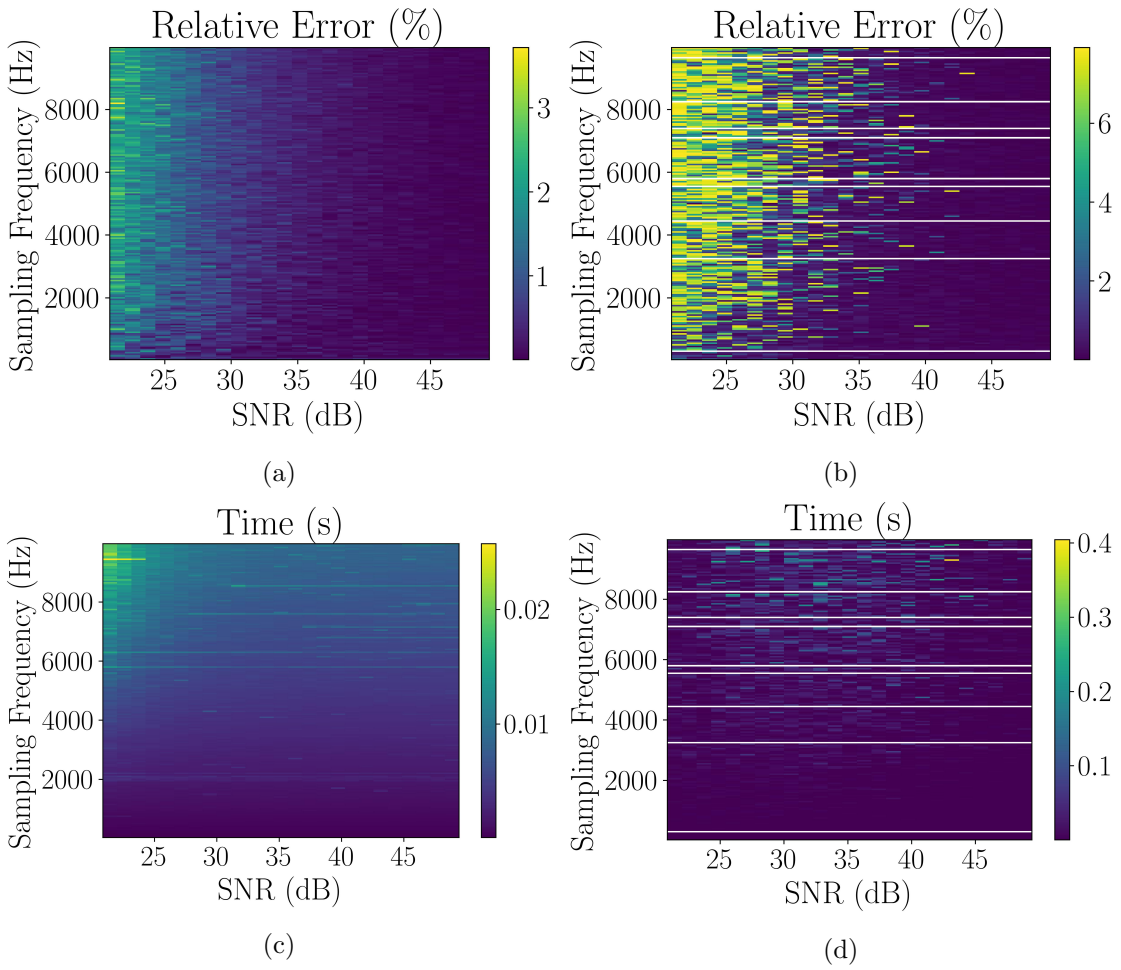

图20：0维持久性（a、c）和 Molinaro算法（b、d）的相对误差与收敛所需时间，适用于 x_3

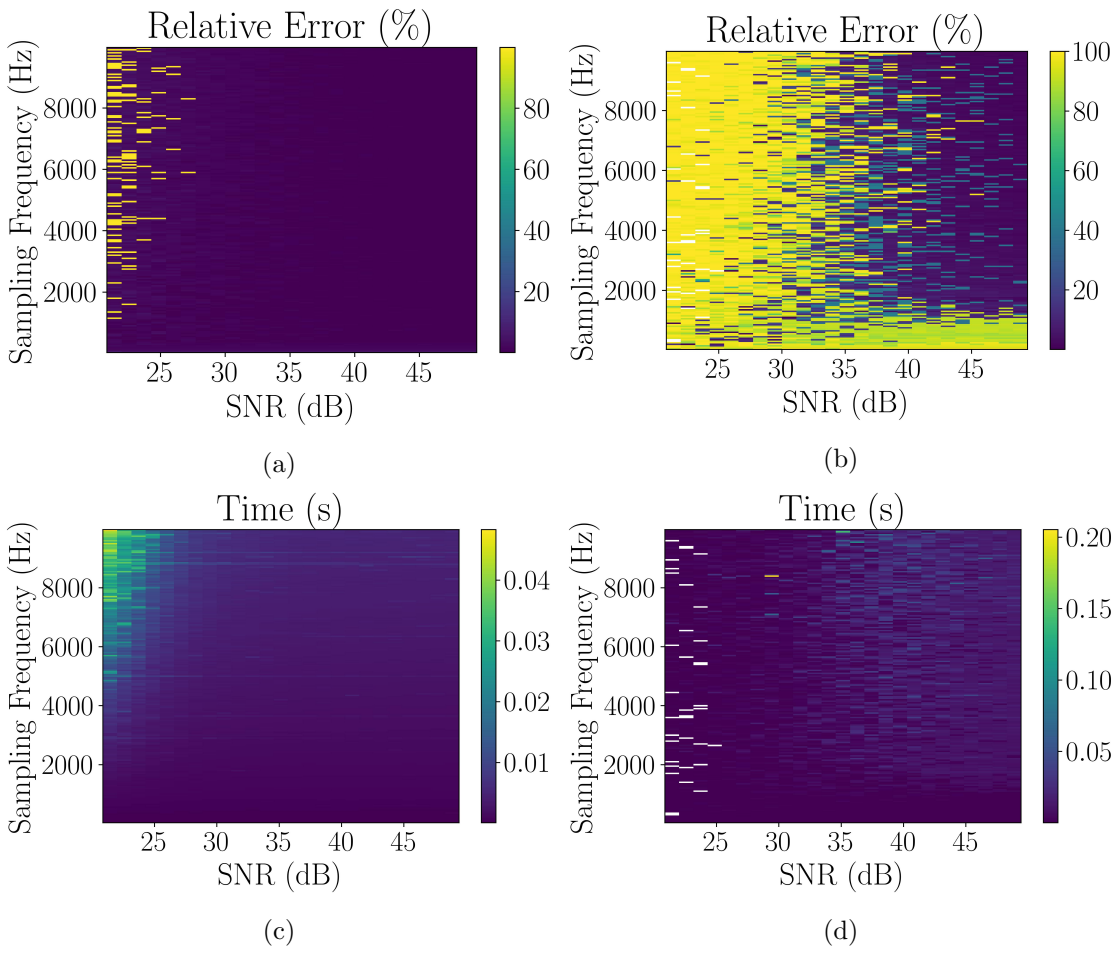

图21：0维持久性（a、c）和 Molinaro算法（b、d）的相对误差与收敛所需时间，针对 x_4

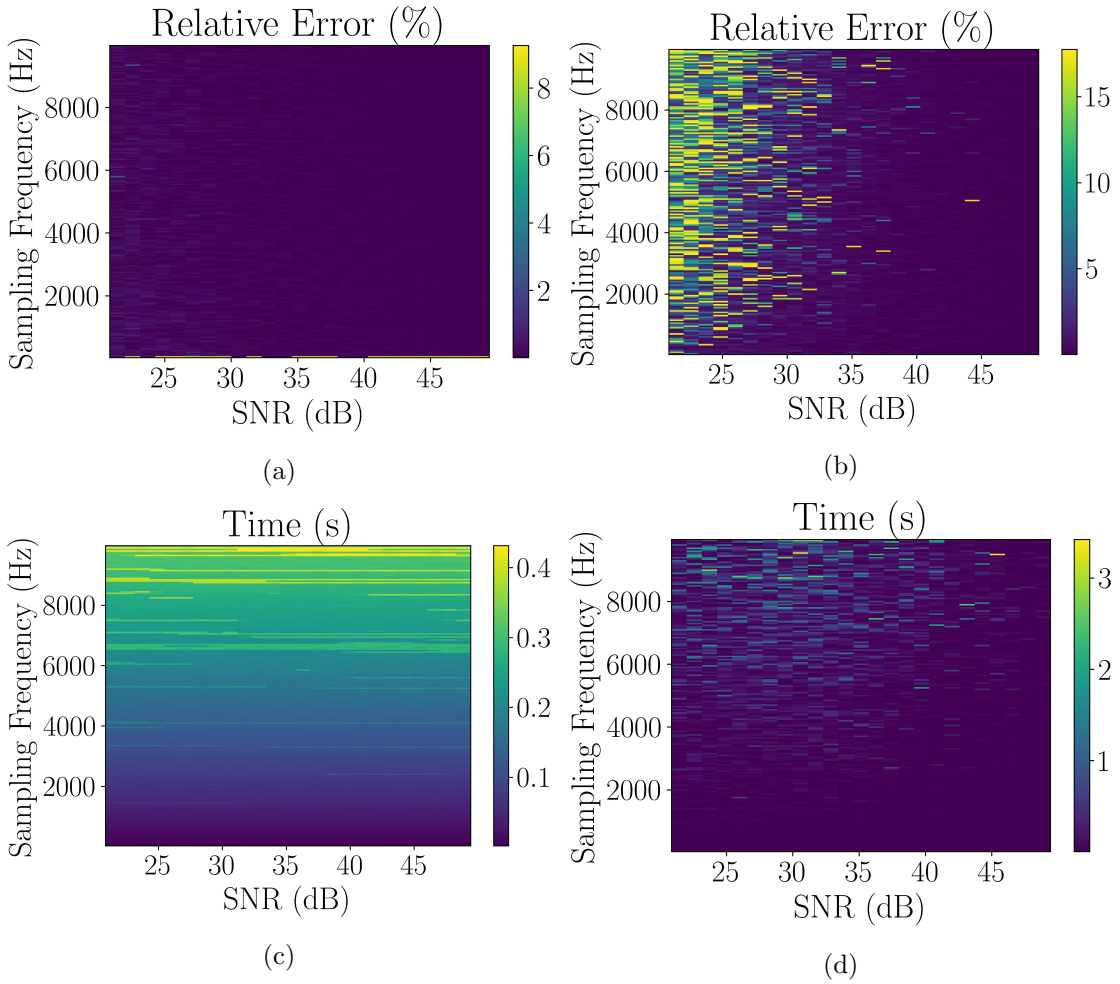

图22： x_5的0维持久性（a、c）和Molinaro算法（b、d）的相对误差与收敛所需时间

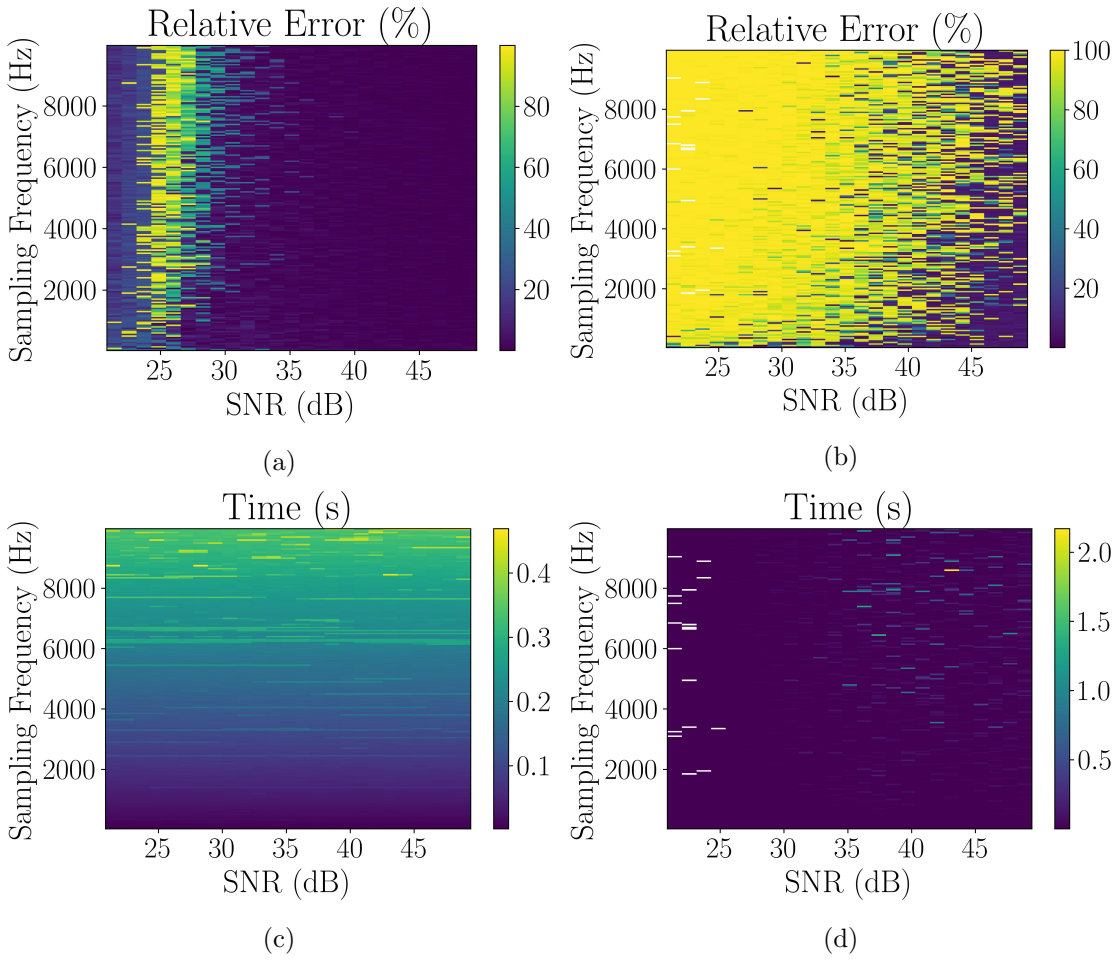

图23： x_6的0维持久性（a、c）和Molinaro算法（b、d）的相对误差及收敛所需时间

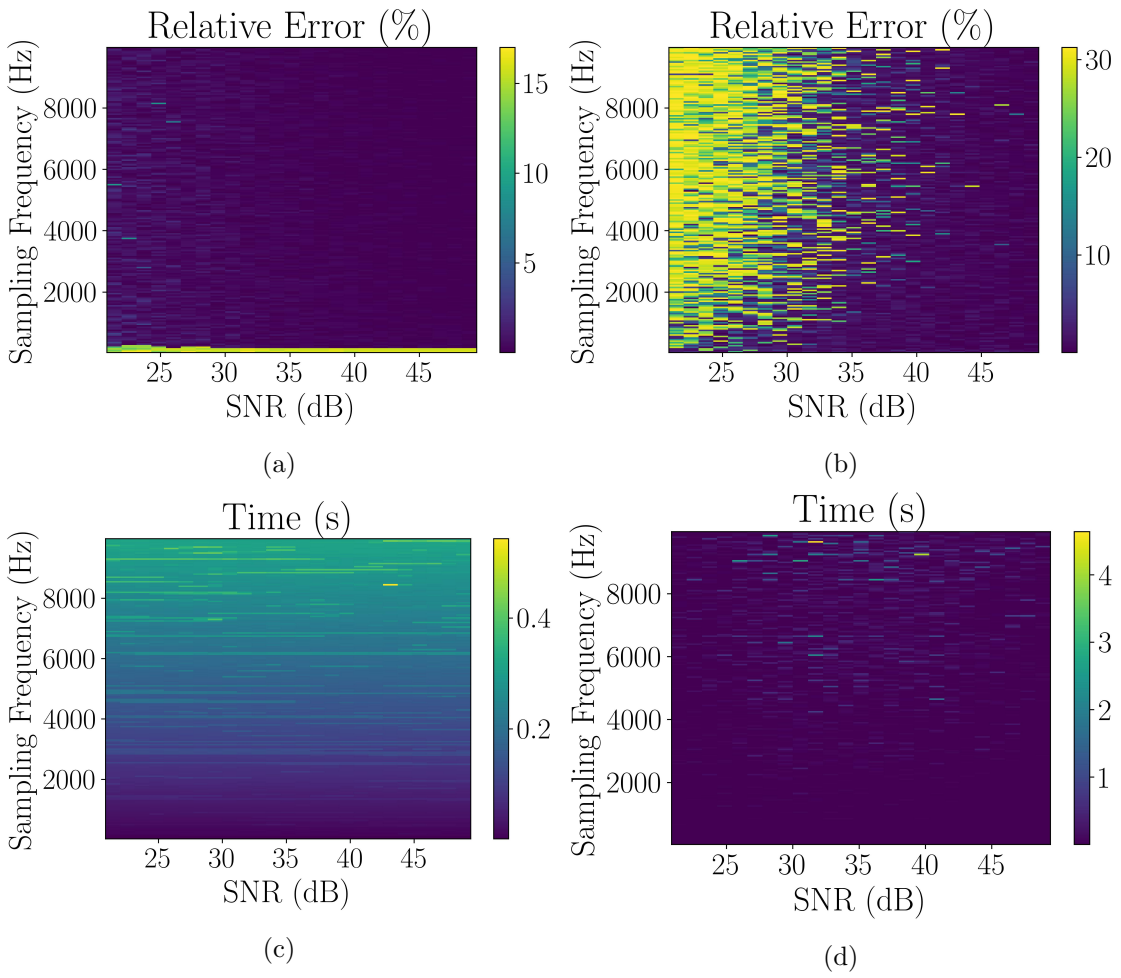

图24：0维持久性（a、c）和 Molinaro算法（b、d）的相对误差与收敛所需时间，针对 x_7

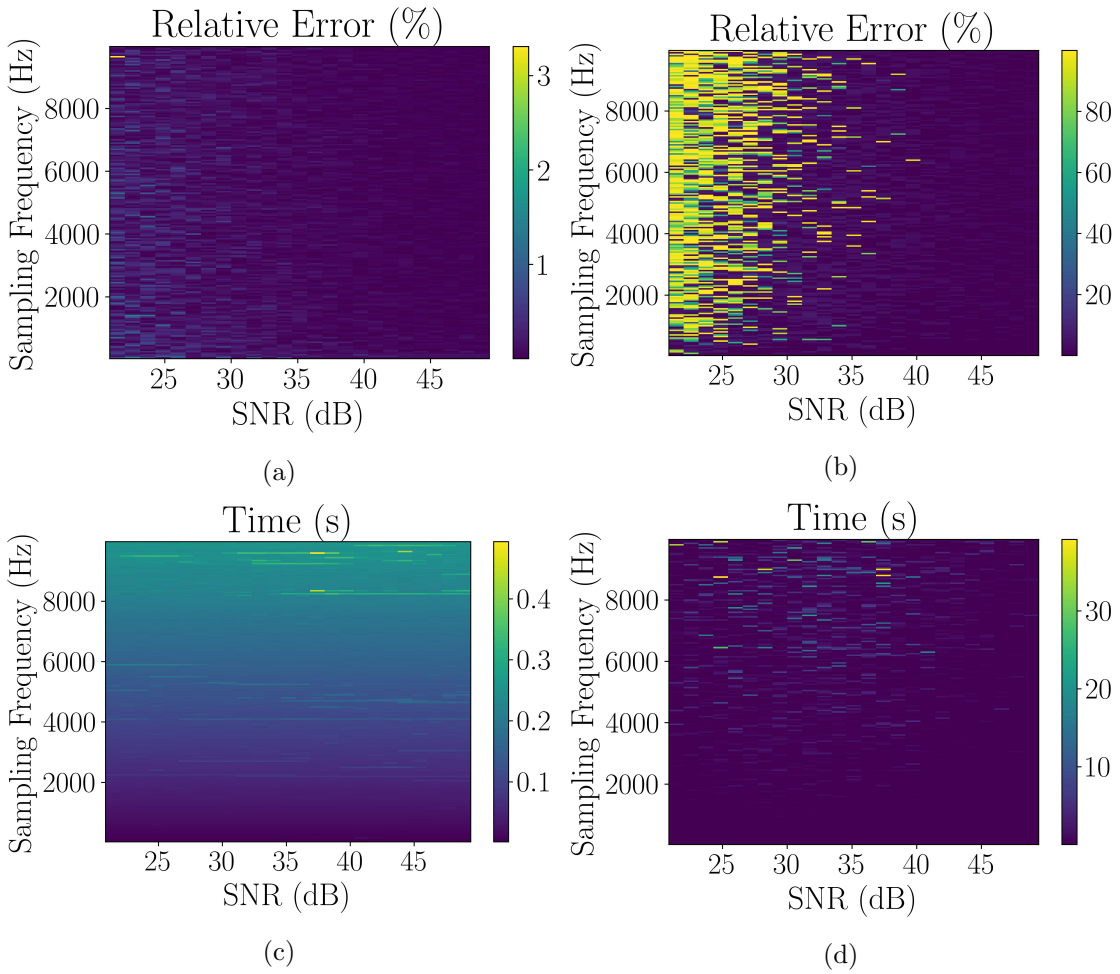

图25：0维持久性（a、c）和 Molinaro算法（b、d）的相对误差与收敛所需时间，针对 x_8

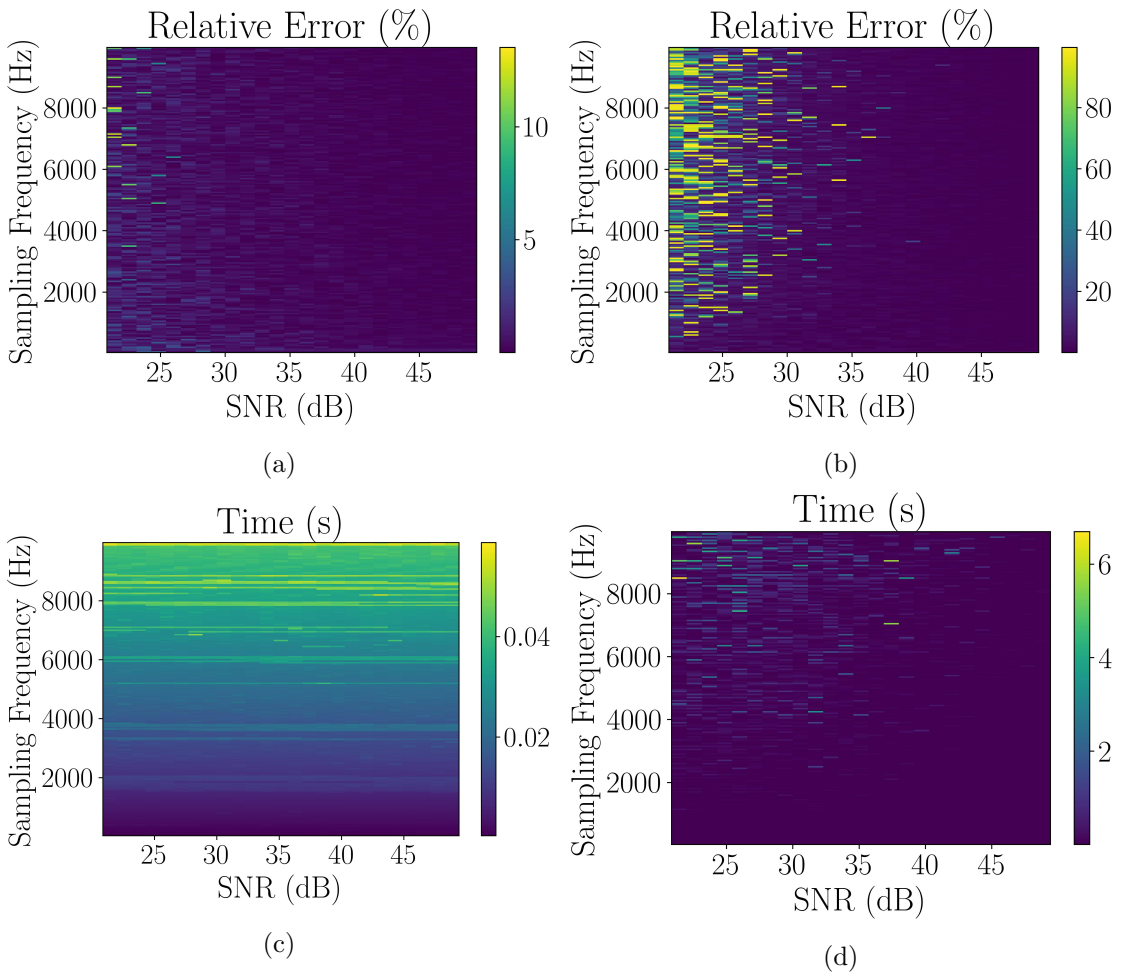

图26： x_9的0维持久性（a、c）和Molinaro算法（b、d）的相对误差及收敛所需时间

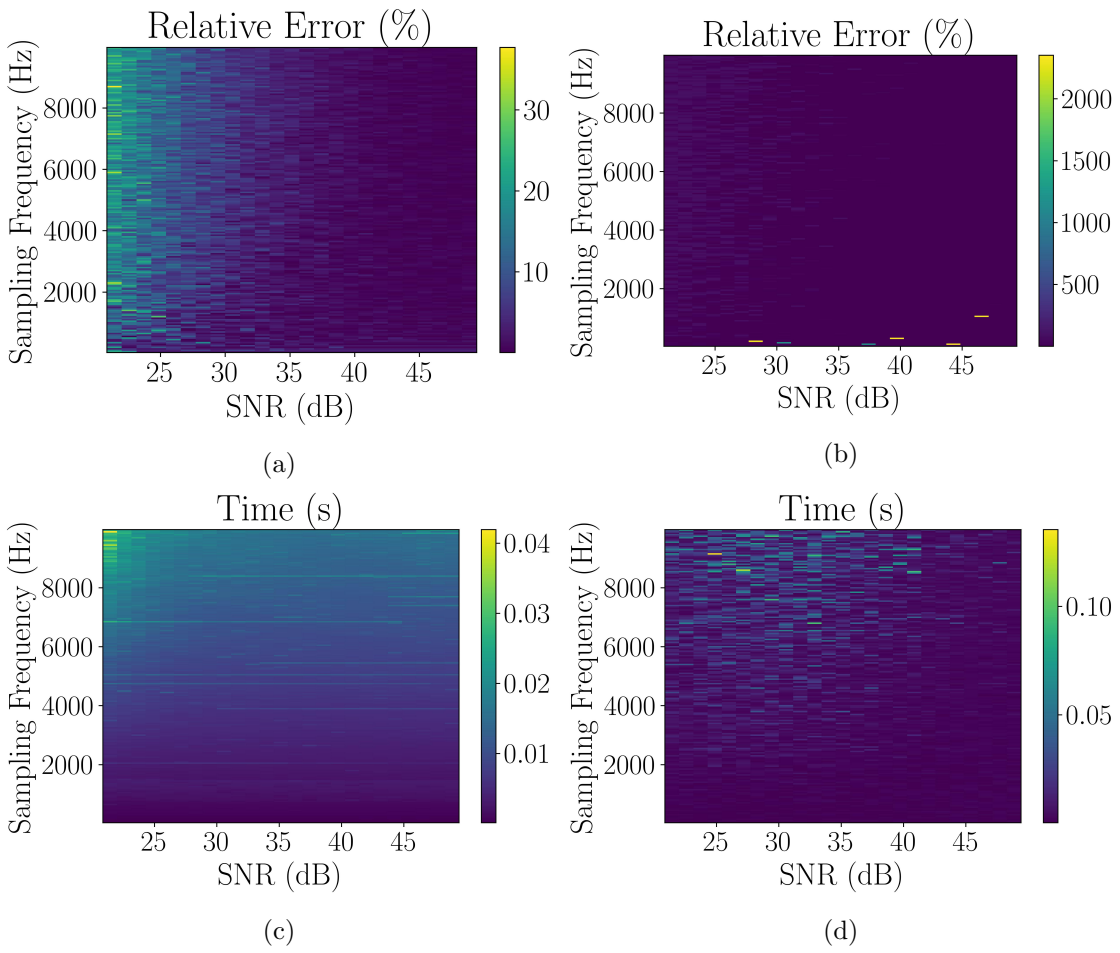

图27：0维持久性（a、c）和 Molinaro算法（b、d）的相对误差及收敛所需时间，针对 x_{10}

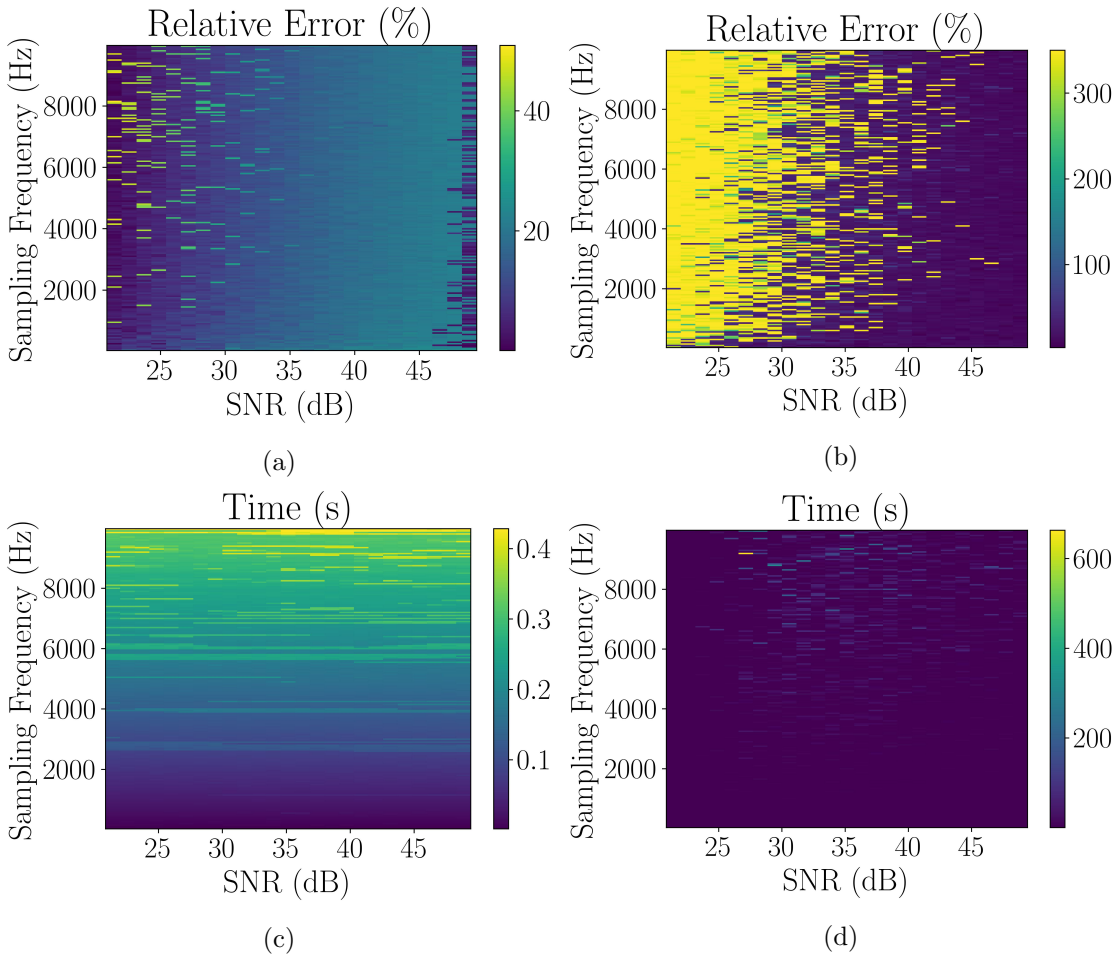

图28： x_{11}的0维持久性（a、c）和Molinaro算法（b、d）的相对误差与收敛所需时间

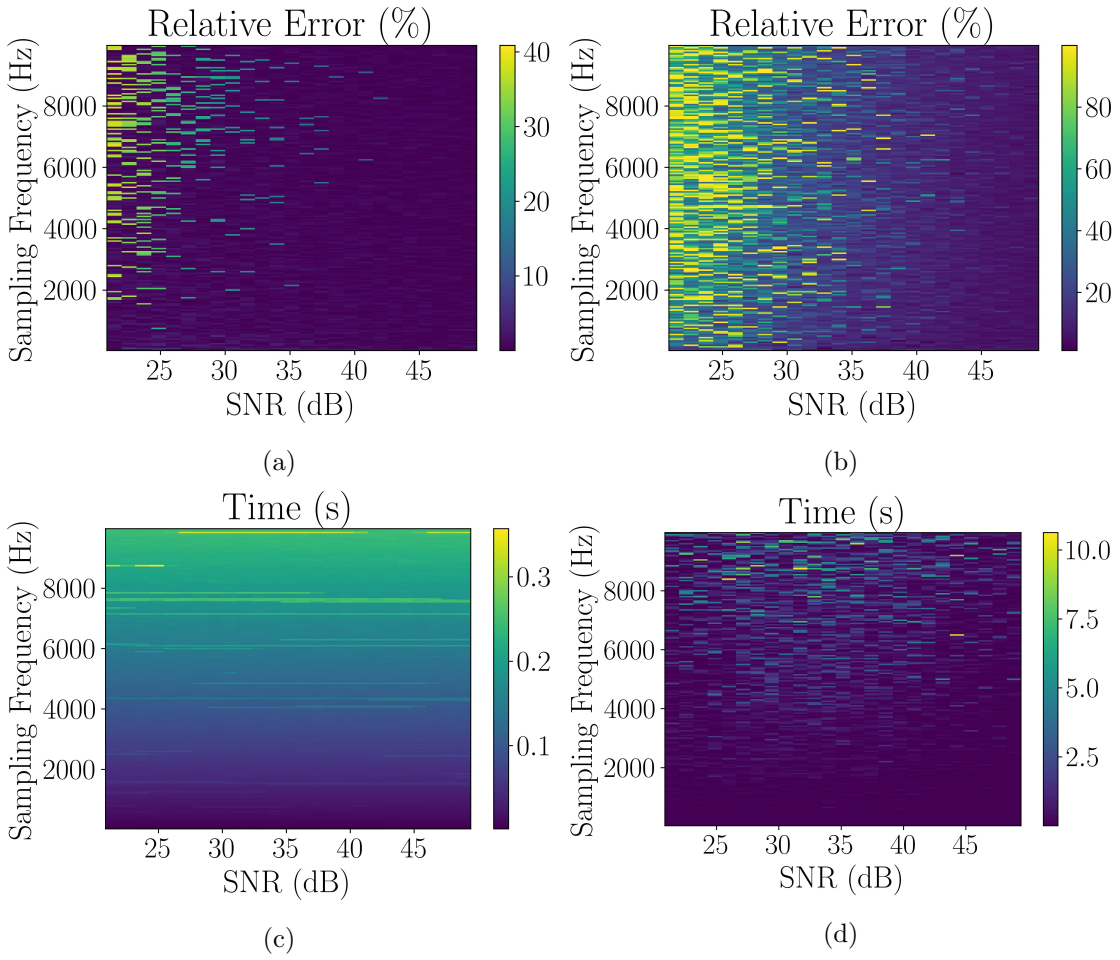

图29： x_{12}的0维持久性（a、c）和Molinaro算法（b、d）的相对误差与收敛所需时间

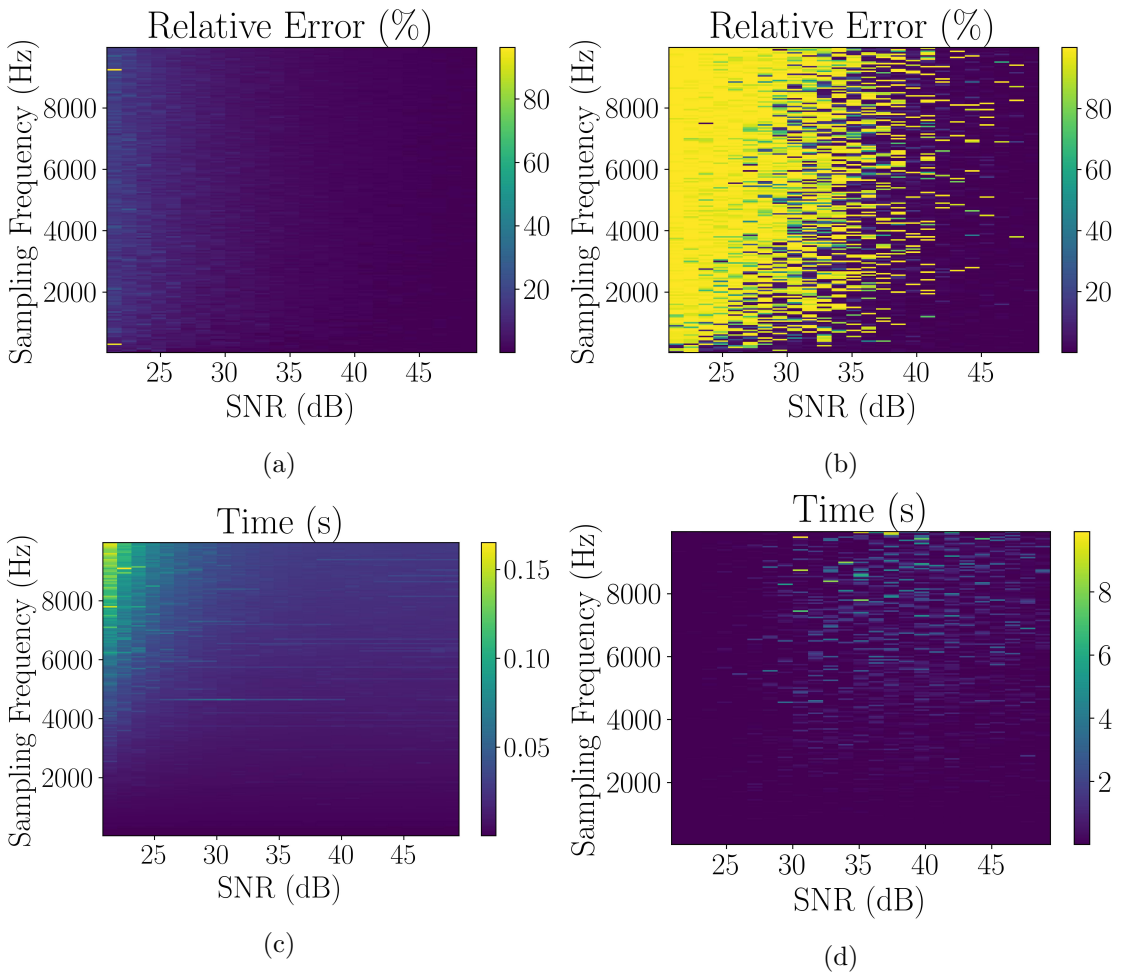

图30：0维持久性（a、c）和 Molinaro算法（b、d）对 x_{13}的相对误差和收敛所需时间

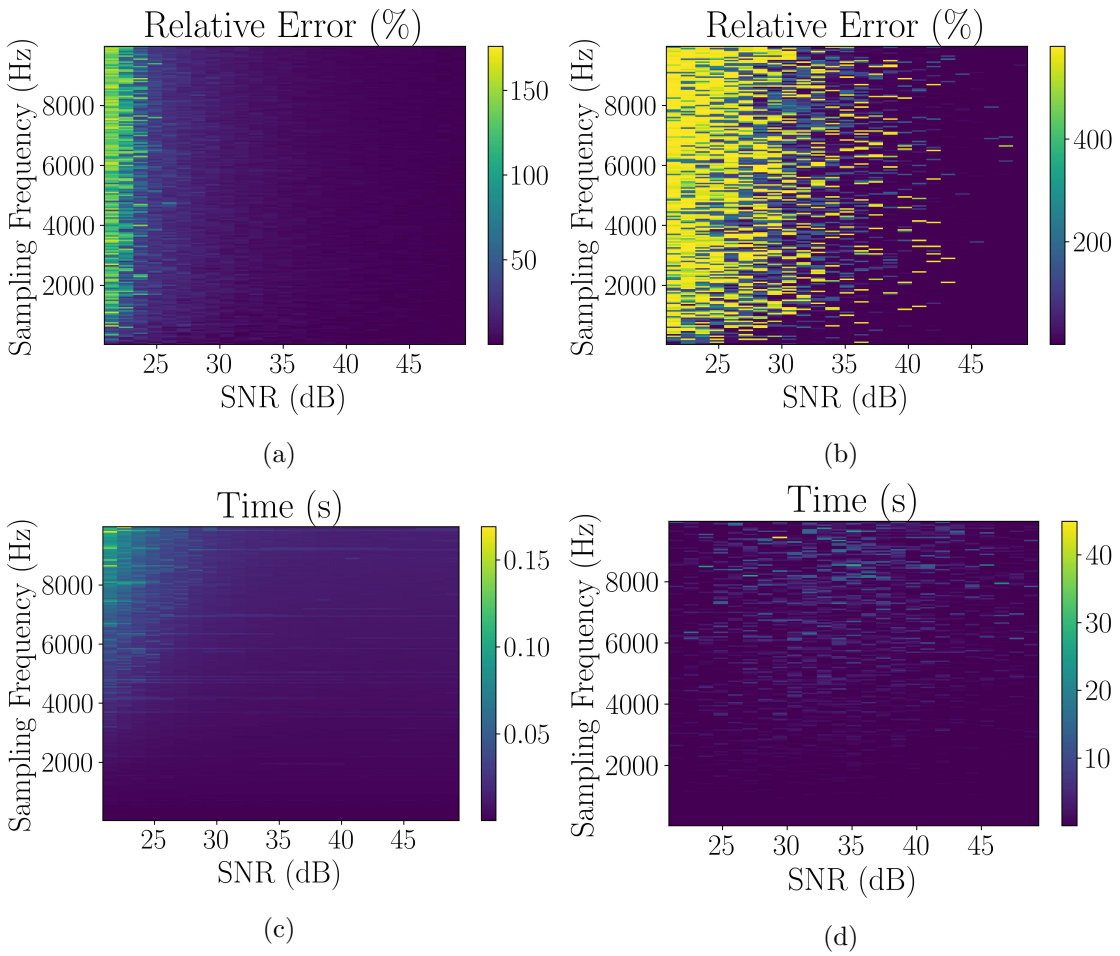

图31：0维持久性（a、c）和 Molinaro算法（b、d）的相对误差和收敛所需时间，针对 x_{14}