arXiv:2512.18238v1 [cs.DB] 20 Dec 2025

无需猜测的同步：不完整多变量时间序列对齐

Ding Jia[1], JingyuZhu[1], Yu Sun[1], Aoqian Zhang[2], Shaoxu Song[3], Haiwei Zhang[1],和 Xiaojie Yuan[1]

[1] 中国天津，南开大学
{2310047,2120240709}@mail.nankai.edu.cn, sunyu@nankai.edu.cn, {zhhaiwei,yuanxj}@nankai.edu.cn[2] 中国北京，北京理工大学 aoqian.zhang@bit.edu.cn[3] 中国北京，清华大学 sxsong@tsinghua.edu.cn

摘要。 多变量时间序列对齐对于确保不同变量之间的连贯分析至关重要，但缺失值和时间戳不一致使这项任务极具挑战性。

现有方法通常依赖预先插补，这可能引入错误并导致对齐效果欠佳。为解决这些局限，我们提出一种面向不完整多变量时间序列的基于约束的对齐框架，无需插补即可确保时间和结构一致性。我们还设计了高效的近似算法，以平衡准确性与可扩展性。在多个真实世界数据集上的实验表明，与现有方法相比，我们的方法在不同缺失率下均能取得更优的对齐质量。我们的贡献包括：（1）正式定义不完整多重时间数据对齐问题；（2）提出三种兼顾准确性与效率的近似算法；（3）在多种真实世界数据集上验证我们的方法，结果表明，该方法在对齐准确率和对齐元组数量方面始终优于现有方法。

关键词： 不完整时间序列 · 对齐 · 多元自回归模型

1 引言

多元时间序列是从传感器监测到金融预测等应用的基础。然而，现实中的多元时间序列通常是分别采集的（例如，燃料消耗 [15] 和电压预测 [1]），由此带来两个关键的数据质量问题：时间戳不一致和缺失数据。

由于传输或网络延迟，同时生成的不同变量对应的数值可能会以错位的时间戳记录 [10]。这种现象通常称为时间戳不一致，在时间敏感型系统中十分常见 [3,9]。

此外，在不同领域收集多变量时间序列数据也面临诸多挑战。因此，数据集通常存在大量缺失数据，这可能严重损害后续分析的准确性和可靠性。例如，广泛使用的医学时间序列数据集 PhysioNet2012 报告了较高的平均缺失率，使得稳健分析变得困难 [11,12,13]。

大多数现有的对齐方法，包括动态时间规整的变体（例如，DTW [7], CTW [16]）和基于相似度的方法（例如，SAMC [3]），都假设序列是完整的。当应用于不完整数据时，现有方法通常会采用线性插值或基于模型的填充等插补技术。然而，将插补和对齐作为两个独立步骤会带来两个关键局限。首先，插补误差会传播到对齐阶段，从而累积不准确性。其次，即使插补模型利用了多种依赖关系，通常也会孤立地估计每个缺失值，而不考虑对齐所施加的结构约束。因此，插补值可能在局部上合理，但在全局上不一致，导致对齐元组违反原始数据中固有的时间依赖关系或统计依赖关系。这促使我们采用一种联合 formulation，在元组选择过程中强制执行对齐约束，而不是依赖独立插补。

为应对这些挑战，我们提出了一种无需预先插补、面向不完整多变量时间序列的基于约束的对齐方法，同时确保对齐元组不仅在时间上连贯，而且与底层时间序列动态特性保持一致。

我们的贡献如下：

(1) 我们定义了包含三项主要约束的对齐任务，以确保结果一致且合理：时间约束确保对齐后的值在时间上彼此接近；位置约束通过限制索引间隔来处理缺失时间戳；模型一致性约束则确保对齐元组在统计上有效。我们的目标是最大化对齐元组的总权重，在完整性和位置保真度之间取得平衡。(2) 我们设计了可调节权衡的高效算法：一种复杂度为指数级的组合方法、一种基于加权 k-集合打包且复杂度为多项式级的近似算法 [2]，以及两种基于贪心策略和期望策略、复杂度近线性的实用算法，可灵活权衡对齐质量与计算效率。

(3) 我们为所有提出的算法提供了近似比的理论保证。(4) 在四个真实世界数据集上进行的实验表明，当缺失率为 10%~40% 时，我们的方法始终优于六种现有基线方法，并且在数据规模和缺失模式变化时仍保持稳健。代码位于 GitHub 代码库[4]中。

[4] https://anonymous.4open.science/r/Incomplete-Multivariate-Time-Series-Alignment-8EC0

2 问题陈述

本节首先介绍支配我们对齐过程的核心约束和权重定义。这些约束确保最终得到的对齐元组不仅在时间和结构上保持一致，而且与底层时间序列模型兼容。通过使用权重函数量化元组质量，我们为后续算法步骤中选择高质量对齐结果奠定了基础。

2.1 约束

依赖精确时间戳匹配的传统方法会导致可用数据非常有限。因此，为了解决这一局限，我们引入时间约束，允许对记录在较短时间窗口内的变量值进行对齐。这种放宽增加了可能的元组数量，同时仍保留时间连贯性。遵循 SAMC [3]，我们使用时间戳差异约束来限制所有对齐元组中的时间戳。

假设我们有 m 个待对齐的时间序列，分别由时间戳 $T \in \mathbb{R}^{n\times m}$ 和对应的值 $V \in \mathbb{R}^{n\times m}$ 表示，其中每一行都包含一个时间戳–值对 (T_i, V_i)，来自第 i 个时间序列。对齐结果记为 R_m。

对齐结果 R_m 中的每个元组 r 记为 $\{(T_1, V_1), (T_2, V_2)...(T_m, V_m)\}$。$T$ 和 V 分别是每个元组的时间戳和值。

定义 1。 每个对齐元组 $r \in R_m$ 的时间戳相似度根据其中任意两个非缺失时间戳之间的差值定义。

$$\forall i, j \in [1, m], \Theta(r) = max(|T_i - T_j|). \tag{1}$$

预期每个 $r \in R_m$ 会在较短时间内对齐，因此它们的时间戳之间的距离应不大于阈值 θ。

定义 2。 如果对齐元组 r 满足时间约束，则称其满足时间约束：$\Theta(r) \leq \theta$。

为处理时间序列中缺失的时间戳，我们提出位置约束，以限制每个 $r \in R_m$中的行索引距离。

定义 3。 每个对齐元组 $r \in R_m$ 的位置相似度根据(T, V)的行索引之差定义。

$$\forall i, j \in [1, m], \Phi(r) = max(|I[(T_i, V_i)] - I[(T_j, V_j)]|). \tag{2}$$

$I[(T, V)]$ 表示原始数据中时间戳–值对（T_i, V_i）的行索引。预期每个 $r \in R_m$ 在较短距离内对齐，因此它们的行索引之差应不大于阈值 β。

定义4. 我们称对齐元组 r 满足位置约束，当且仅当$\Phi(r) \leq \beta$。

为了保证对齐结果中各项之间的一致性，我们使用时间序列模型预测其值，并通过损失评估其一致性。

首先，我们有 $A_{i,j} = |M_{i,j} - V_{i,j}| \times mask_{i,j}, i \in [1,n], j \in [1,m]$, $A_{i,j}$ 是第 j 个时间序列中第 i 个值的损失。$M_{i,j}$ 是时间序列模型对 $V_{i,j}$ 的预测值。$mask_{i,j}$ 表示 $V_{i,j}$ 的缺失情况，缺失时为 0，否则为 1。然后，我们有 $\mu_j = F_j \times (V_j^{max} - V_j^{min})$ $j \in [1,m]$, V_j 是第 j 个时间序列，F_j 是第 j 个时间序列中非缺失值的总数。并且，我们有 $L_j = \sum_{i=1}^{|R_m|} \frac{A_{i,j}}{\mu_j}, j \in [1,m]$, L_j 是第 j 个时间序列中非缺失值的归一化平均绝对误差。

定义 5. 对于每个对齐结果 R_m 和训练好的时间序列模型 M，其一致性定义如下：

$$\Delta(R_m, M) = \frac{1}{m} \sum_{j=1}^{m} L_j. \tag{3}$$

$\Delta(R_m, M)$ 越小，对齐结果内部的一致性就越高。

定义 6. 如果对齐结果满足模型约束，则称其满足模型约束：$\Delta(R, M) \leq \delta$。

2.2 权重定义

为了尽可能增大对齐结果中的非缺失对齐对数量 $\{(T_i, V_i), (T_j, V_j)\}$, $i, j \in [1,m]$，我们为每个元组赋予权重。

定义 7. 对于每个 $r \in R_m$，我们将其权重定义为 $W(r)$，即其中配对 $p(r)$ 与距离 $d(r)$ 之和。

$$\begin{cases} p(r) = \dfrac{\lambda \times (\lambda - 1)}{2}, \\ d(r) = \displaystyle\sum_{i=1}^{m} \sum_{j=i+1}^{m} \left| I[(T_i, V_i)] - I[(T_j, V_j)] \right|, i, j \in [1,m], \\ W(r) = \dfrac{k1 \times p(r) + b}{k2 \times d(r) + c}. \end{cases} \tag{4}$$

λ是 r中非缺失值的总数。

其中，k_1 和 k_2 是控制数值完整性和位置紧凑性重要程度的系数，而 b 和 c 是为确保数值稳定性并防止除零而引入的偏置项。这些参数预先设定，可灵活平衡有效配对的数量与每个元组内的位置连贯性之间的权衡，使权重函数能够适应不同的数据特征。

2.3 问题陈述

为正式解决不完整多变量时间序列对齐问题，我们定义了一组约束，每个对齐元组都必须满足这些约束。与严重依赖插补（这往往会引入噪声和不一致性）的现有方法不同，我们的方案将结构约束直接纳入对齐过程。这使元组选择更加可靠，同时确保时间连贯性、位置连续性和模型级一致性。

问题 *1*（基于约束的不完整时间序列对齐）。每个对齐元组 $r \in R_m$ 应满足：(1) 每个 $(T,\ V)$ 只能对齐一次，即，$\forall r_i,\ r_j \in R_m,\ k \in [1, m],$ $r_i[k] \neq r_j[k],$ $r[k]$ 是元组 r 中第 k 个 $(T,\ V)$。(2)$\forall r \in R_m,\ \Theta(r) \leq \theta,$ (3) $\forall r \in R_m,\ \Phi(r) \leq \beta,$ (4) $\Delta(R_m,\ M) \leq \delta,$ (5) 权重之和最大：

$$\max_{R_m} \sum_{r \in R_m} W(r). \tag{5}$$

$\sum_{r \in R_m} W(r)$ 表示 R_m 中元组的总权重。如果元组 r_i 和 r_j 没有重复对齐的 $(T,\ V)$，则记为 $r_i \nsim r_j$。否则，我们称 r_i和 r_j 彼此冲突，记为 $r_i \asymp r_j$。

这种基于约束的表述不仅避免了直接插补的弊端，还确保对齐结果在结构和语义上都与底层时间序列模型保持一致。

3 对齐算法

在我们的对齐过程中，受 SAMC [3],启发，我们通过选择满足预定义约束的候选元组来执行对齐，组成时间序列，并检查每个时间序列是否满足模型约束。

不同之处在于，SAMC 主要在固定的约束设置下按顺序扫描并匹配元组，而我们的方法引入了权重最大化目标，并融合了基于模型的约束，从而即使存在缺失数据或噪声数据，也能够选出高质量的对齐结果。这种设计使我们的方法在应对纯顺序策略局限性的同时，保留了高效的冲突解决机制。

我们首先生成一个元组集 R_c 作为候选集，以确保每个元组 $r \in R_c$都满足定义2和定义4中规定的时间约束和位置约束。该候选集构成一个可行搜索空间，过滤掉那些本质上违反对齐约束并会导致无效或低质量结果的组合。

通过预先构造 R_c ，我们缩小了后续优化的搜索空间并降低了计算成本，同时确保所有选定的元组在时间和位置上保持一致。基于 R_c，我们随后选取一个元组子集，并合成一条新的对齐时间序列，使权重和最大，同时满足第 2.3 节（3）和（4）中所述的模型约束和唯一性约束。

本节首先介绍一种经过剪枝的候选生成方法，以便根据时间和位置约束高效构造 R_c 。随后，我们提出一种时间序列合成算法，按照第 2.3 节（1）中所述的要求，从 R_c 中选取元组，形成最终的对齐结果。

3.1 候选生成

我们初始化一个索引集 I_c ，其中包含 m 个数字，表示候选搜索过程中当前的行索引。每一轮中，如果满足两个约束，相应的元组就会被放入候选集 R_c ，并将 I_c 中最后一个索引移至同一时间序列中的下一个 (T, V)，以确保候选集中的元组按升序排列。如果索引超出范围，我们就移动前一个索引，并根据位置约束将其后的所有索引初始化，而不是将它们移至该列的起始位置。

为剪枝搜索过程，如果不满足任何一个时间约束，考虑到每个时间序列中的时间戳单调递增这一特性，我们会直接将 I_c^{max} 和 I_c^{min} 中较靠后的一个移至下一行； I_c^{max} 表示 I_c 中时间戳最大的索引，I_c^{min} 表示时间戳最小的索引。移动后，我们按照上述相同方式初始化其后的索引。由于位置约束也具有相同特性，因此当位置约束不满足时，也可以采用类似策略。最坏情况下，候选生成的时间复杂度为 $O(nm(\beta+1)^m)$。

3.2 时间序列合成

基于生成的候选集 R_c，我们根据问题 1 中的要求 (1) 迭代选择元组，以合成新的对齐时间序列 r_a ，然后将其添加到备选序列集 R_a 中进行进一步评估。该方法的优化目标表示为 W_O。

最后，考虑到时间序列中的缺失值，我们使用具有缺失容忍能力的 MARSS [4] 时间序列模型预测每个 r_a，计算平均归一化平均绝对误差，并在满足定义 6 的结果中选择权重和最大者作为最终对齐结果 R_m。

$R_c[i]$ 是 R_c 中第 i 个候选元组。在算法 1 第 1—2 行中，我们将 ψ 初始化为标记，并将 W 初始化为当前最大的权重和。在第 3 行中，遍历所有合成情况。在第 4—7 行中，我们合成时间序列；在第 8—11 行中，检查其是否满足第 2.3 节中的要求 (1)。最后，在第 12—15 行中，我们找出权重和最大的合成结果。

权重计算的时间复杂度为 $O(|R_c|m^2)$，算法 1 的时间复杂度为$O(\sum_{i=1}^{2^{|R_c|}} s_i^2 \times m)$，其中 s_i 表示每个组合时间序列的大小。最坏情况复杂度为 $O(2^{|R_c|}|R_c|^2 m)$。

算法 1：时间序列组合(R_c, δ)

输入： 候选集 R_c，模型约束 δ
输出： 对齐结果 R_m

$\psi \leftarrow True$
$W \leftarrow -1$
对于 $i \leftarrow 1$ **到** $2^{|R_c|}-1$，**执行**
  $S \leftarrow \{\}$
  对于 $j \leftarrow 0$ **到** $K-1$，**执行**
    如果 j 是 $binary(i)=1$ 的第 j 位，**则**
      $S \leftarrow S \cup R_c[j]$
  对于 $i \leftarrow 1$ **到** $|S|$，**执行**
    对于 $j \leftarrow i$ **到** $|S|$，**执行**
      如果 $S_i \asymp S_j$，**则**
        $\psi \leftarrow False$
  如果 $\psi = True$ 且 $\Delta(S,\ M) \leq \delta$，**则**
    如果 $\sum_{r\in S} W\ (r)\ > W$ **然后**
      $R_m \leftarrow S$
      $W \leftarrow \sum_{r\in S} W(r)$
返回 R_m

4 基于加权集合打包的近似

第3节中的组合算法会产生较高的计算成本，因为其时间复杂度涉及枚举候选集中所有有效的元组组合，而候选集的规模会随着变量数量和 R_c 的大小呈指数级增长。这主要是因为该算法会将权重总和较低的组合时间序列也纳入考虑。因此，在下一节中，我们提出一种基于加权 k-集合打包结构 [2]、时间复杂度更低的对齐算法。候选生成方法与第3.1节相同。该近似方法的优化目标表示为 W_M。

在时间序列组合过程中，我们还旨在选择若干个互不冲突的元组，使其权重和最大。首先，我们通过贪心策略选择彼此不冲突的元组，初始化元组集 R_f。接下来，对于某个 $x \in R_f$，考虑满足以下条件的元组集 Q：(1)$\forall q_1$, $q_2 \in Q$, $q_1 \not\asymp q_2$ 以及 (2)$\forall q \in Q$, $q \asymp x$。

选择与 Q 中任意元组发生冲突的所有元组，构成新的元组集 I。我们找出使 $\sum_{q\in Q} W_{(q)\sum_{i\in I} W(i)}$ 最大化的 Q，然后用 Q 替换 I。如果多个集合达到相同的最大比率，则保留并探索所有这些分支。

当不存在这样的 Q 能使 $\sum_{x\in R_f} W(x)$ 增大时，该过程便会停止。然后，我们使用 MARSS [4] 模型计算所有组合时间序列的归一化损失；如果模型

约束满足，则选择权重和最大的作为对齐结果 R_m。每个分支的时间复杂度为 $O(|R_c|m^2K^m)$， K 是每个元组发生冲突的元组数量的最大权重和。

命题1. 给定输入时间序列数据和固定约束 θ、 β 和δ，基于加权集合打包的近似方法的近似比 $\xi = \frac{W_M}{W_O}$ 满足: ξ 不小于$\frac{3}{2(m+1)}$。

5 贪心与基于期望的近似

尽管第4节中的时间复杂度略有改善，但由于在元组选择过程中需要进行冲突检查，其计算成本仍然很高，并且会随着 m 呈指数增长。为了解决这一问题，本节提出另外两种更高效的基于近似的组合方法，旨在降低计算量，同时保持对齐质量。

这两种方法受到不同启发式策略的启发：一种是基于贪心的方法，通过迭代选择高权重元组；另一种是基于期望的策略，用于估计每个元组之后的潜在贡献。它们在不同数据特征下的实际适用性有所不同。具体而言，当候选权重相对均衡且缺失率较低时，基于贪心的方法效率很高且表现良好，因为它能够快速识别高权重元组。相比之下，基于期望的策略会估计后续元组的潜在贡献，因此在缺失率较高或候选权重倾斜的场景下更加稳健，因为局部决策更有可能不是最优决策。

候选生成步骤与第3.1节保持相同，确保所有候选元组 $r \in R_c$ 满足时间和位置约束。每个元组的权重仍按照定义 7 中的定义进行计算，作为两种近似策略中的关键指标。基于贪心的近似方法的优化目标记为 W_G，而基于期望的近似方法的优化目标记为 W_E。

5.1 贪心

基于贪心的策略是一种快速近似方法。它会迭代地从候选集中选取局部权重最高且可行的元组。具体而言，我们遍历 R_c 中的元组，并不断将每个元组添加到新的元组集 G中，前提是 G 在执行添加操作后满足 $\forall g_1, g_2 \in G, g_1 \asymp g_2$ 。如果向 G 中添加任何元组都会导致其无法满足这一要求，则停止此过程。随后，我们选出权重最大的 $g \in G$ ，将其作为对齐结果中的一个元组添加到组合时间序列集合 R_m 中。根据实验，值得一提的是，如果有多个元组的权重同为最大值，可以随机选取其中一个。之后，我们继续以相同的方式向 G中添加元组。

当遍历完 R_c 中的所有元组后，此过程便停止。最后，与前文所述相同，我们使用 MARSS [4]模型预测数值，并检查每个组合 R_m 是否满足模型约束 δ。

算法 2： 时间序列贪心组合 (R_c, δ)

输入： 候选集 R_c，模型约束 δ
输出： 对齐结果 R_m

$G \leftarrow \{\}$
当真 时执行
  $R_m \leftarrow \emptyset$
  对于 $i \leftarrow 0$ **到** $|R_c|$，**执行**
    如果 $\forall r \in R_m$，$R_c[i] \not\asymp r$ **则**
      如果 $G = \emptyset$ 或 $\forall g \in G$，$g \asymp R_c[i]$ **则**
        $G \leftarrow G \cup \{R_c[i]\}$
      else
        $R_m \leftarrow R_m \cup$ arg 最大值$_{g \in G} W(g)$
        $G \leftarrow \{\}$
  如果 $\Delta(R_m, M) \leq \delta$ **则**
    返回 R_m

每轮组合方法的时间复杂度为 $O(|R_c|m|G| + |R_c|log|G|)$。最坏情况复杂度为 $O(|R_c|m(4\beta+1)^m + |R_c|mlog(4\beta+1))$。

命题 2. 给定输入时间序列数据和固定约束 θ, β *and*δ，基于贪心的近似的近似比 $\xi = \frac{W_G}{W_O}$ 满足 ξ 不小于 $\frac{\lceil \frac{N}{m} \rceil \times W_{min}}{N \times W_{max}}$。$N$是精确解中在相同 R_c下对齐元组的数量。W_{max} 是理论最大权重，W_{min} 是理论最小权重。

5.2 期望

不同的是，当我们从每个元组集 G 中选取一个最大权重元组时，我们使用权重和作为衡量 $g \in G$中每个元组的指标，而不是使用它们原本的权重。具体而言，对于每个元组 $g \in G$，我们定义元组集 E，使其满足 $\forall r \in E$，$r > g \wedge r \not\asymp g$。

在 R_c中很难找到所有满足此要求的元组，而我们只需考虑 R_c 中与G中的至少一个元组冲突的元组，因为其他元组不会影响 G中的元组。

引理 1。对于每个 G，存在元组 r，其位于 R_c 中，并且仅当 $\forall i \in [1, m]$、$r[i] \leq max + \beta$ 时，与 G 中至少一个元组发生冲突，其中 最大值 $=$最大值$_{i \in [1,m],\ r' \in G} r'[i]$.

根据引理 1，时间复杂度可以大幅降低。最后，我们将集合 E 的总权重添加到 G 中的每个权重为 $W(g)$ 的元组上，找出权重最大的一个元组，并将其加入组合时间序列集合 R_m。最后，

我们检查每个组合的 R_m 是否满足模型约束 δ。每轮执行此时间序列组合方法的时间复杂度为 $O(|R_c|m(|G|+z) + |R_c|log|G|)$。$z$ 表示与 G 中至少一个元组发生冲突的集合大小。最坏情况复杂度为 $O(2|R_c|m(4\beta+1)^m + |R_c|mlog(4\beta+1))$。

命题 3. 给定输入时间序列数据和固定约束 θ、β 和 δ，基于期望的近似的近似比 $\xi = \frac{W_E}{W_O}$ 满足 ξ 不小于 $min(\frac{(\lceil \frac{N}{m} \rceil+1)\times W_{min}}{N\times W_{max}}, \frac{\lceil \frac{N}{m} \rceil \times W_{min}}{(N-1)\times W_{max}+W_{min}})$。~~$\bar{N}$ 是精确解中具有相同 R_c 的对齐元组数量。~~

6 实验

本节将第 5 节的方法与现有方法进行比较：用于多变量对齐的 Procrustes 分析框架扩展方法 DTW [7], DDTW [5] 和 CTW [16][17]，以及在真实世界数据集 Telemetry、Household、Water 和 Air 上应用的 GTW [17,18],TTW [6] 和 SAMC [3][5]；这些数据集的缺失率由 MCAR 生成，范围为 10% 至 %。Telemetry 和 Water 各包含 4 个时间序列，共有 5000 个元组；Household 包含 4 个时间序列，共有 6839 个元组；Air 包含 11 个时间序列，共有 1000 个元组。我们通过对齐元组总数和对齐准确率评估每种方法。

我们采用 F1分数，通过将恢复出的变量对与真实值进行比较来评估对齐准确率。R_{truth} 表示原始完整数据中真实的（T，V）配对集合，R 表示最终在 R_m 中对齐的（T，V）配对集合，召回率为 $= \frac{|R_{\text{truth}} \cap R|}{|R_{\text{truth}}|}$，精确率为 $= \frac{|R_{\text{truth}} \cap R|}{|R|}$。

6.1 与现有方法的比较

我们使用 Timer [8] 和 TimesNet [14]进行缺失值插补，并采用线性插值填补缺失时间戳，因为其他基线方法无法直接处理不完整输入。

由于第3节和第4节中方法的时间复杂度较高（如表 1 所示），我们仅对第5节提及的方法进行实验。对于缺失率从 10% 到 40% 的四个数据集，我们提出的方法均能持续取得更高的对齐准确率。具体而言，随着缺失率增加，基于贪心的策略性能会下降。这主要是因为该策略高度依赖候选元组的单独权重，在信息稀疏时容易受到误导。相比之下，基于期望的策略在缺失率不断上升时表现更为稳定，但可能需要付出更高的计算成本。

表2比较了 SAMC [3] 与我们的两种方法对齐的元组数量。此处我们仅将我们的方法与 SAMC 进行比较，因为基于 DTW 的方法始终会对齐与输入数据大小相同的元组。随着缺失率上升，SAMC 可能会对齐更多元组，这是因为

[5] https://github.com/fangfcg/SAMC

数据集	缺失	DTW		DDTW		CTW		GTW		TTW		SAMC		贪婪期望	
		Timer	TimesNet	Timer	TimesNet	Timer	TimesNet	Timer	TimesNet	Timer	TimesNet	Timer	TimesNet		
遥测	10%	0.527	0.527	0.215	0.213	0.261	0.254	0.197	0.201	0.188	0.167	0.837	0.830	**0.990**	**0.996**
	20%	0.509	0.514	0.197	0.192	0.256	0.247	0.316	0.185	0.263	0.229	0.853	0.853	**0.993**	**0.993**
	30%	0.505	0.499	0.181	0.178	0.263	0.280	0.232	0.346	0.499	0.499	0.841	0.842	**0.982**	**0.998**
	40%	0.487	0.488	0.174	0.161	0.271	0.284	0.195	0.171	0.527	0.309	0.883	0.873	**0.964**	**0.997**
Household	10%	0.158	0.159	0.143	0.143	0.253	0.249	0.302	0.247	0.393	0.004	0.956	0.956	**0.998**	**0.998**
	20%	0.149	0.148	0.126	0.127	0.236	0.246	0.689	0.856	0.002	0.003	0.960	0.961	**0.994**	**0.998**
	30%	0.137	0.197	0.116	0.124	0.167	0.174	0.406	0.529	0.085	0.590	0.964	0.964	**0.986**	**0.998**
	40%	0.128	0.143	0.117	0.119	0.058	0.062	0.300	0.195	0.187	0.113	0.963	0.964	**0.972**	**0.998**
水	10%	0.168	0.168	0.106	0.109	0.079	0.076	0.105	0.120	0.007	0.001	0.941	0.942	**0.992**	**0.995**
	20%	0.188	0.158	0.101	0.103	0.040	0.064	0.129	0.114	0.014	0.008	0.907	0.914	**0.994**	**0.997**
	30%	0.005	0.003	0.098	0.099	0.055	0.058	0.213	0.138	0.016	0.047	0.913	0.926	**0.985**	**0.997**
	40%	0.003	0.014	0.096	0.102	0.057	0.064	0.140	0.149	0.001	0.002	0.904	0.907	**0.969**	**0.998**
Air	10%	0.170	0.229	0.110	0.118	0.064	0.081	0.290	0.201	0.098	0.048	0.924	0.924	**0.979**	**0.980**
	20%	0.222	0.200	0.104	0.113	0.063	0.070	0.286	0.386	0.123	0.184	0.907	0.907	**0.982**	**0.984**
	30%	0.164	0.155	0.098	0.111	0.052	0.059	0.134	0.256	0.065	0.052	0.914	0.914	**0.988**	**0.989**
	40%	0.193	0.161	0.096	0.117	0.054	0.067	0.439	0.363	0.032	0.049	0.907	0.907	**0.986**	**0.989**

表 1：使用时间序列插补模型 Timer 和 TimesNet，在四个缺失数据集上与现有六种方法进行的 F1分数性能比较。每个数据集的缺失率范围为 10% 至 40%。

数据集	缺失	SAMC		贪心期望	
		Timer	TimesNet		
遥测	10%	4885	4879	4946	**4968**
	20%	4899	4902	4946	**4978**
	30%	4927	4932	4905	**4985**
	40%	4953	4954	4847	**4990**
Household	10%	6742	6742	6811	**6820**
	20%	6759	6762	6788	**6829**
	30%	6769	6768	6744	**6831**
	40%	6778	6781	6672	**6833**
水	10%	4874	4879	4964	**4986**
	20%	4791	4810	4955	**4992**
	30%	4830	4859	4924	**4996**
	40%	4816	4805	4866	**4996**
Air	10%	942	942	962	**965**
	20%	936	936	966	**973**
	30%	938	938	976	**978**
	40%	936	936	978	**986**

表2：使用时间序列插补模型 Timer 和 TimesNet，在四个缺失数据集上对比 SAMC、贪心和期望方法的对齐元组。每个数据集的缺失率为 10% 至 40%。

时间戳插补可能会使更多元组满足自身的时间约束。从该表可以看出，缺失率较低时，基于贪心的策略能够对齐更多元组；但随着缺失率上升，其性能会急剧下降。而基于期望的策略无论缺失率如何，始终能够对齐更多元组。这主要是因为基于期望的方法会考虑后续的潜在贡献，并且我们对每个元组权重的定义有助于提升位置一致性。

附录B中的图3展示了缺失率为20%时，Household数据集规模从1k到6k变化时的对齐准确率。其他方法可能会受到数据集规模的较大影响，而我们的方法在所有数据规模下都表现更好。

6.2 参数确定

本节首先分析时间约束 θ、位置约束 β、模型约束 δ 以及其他权重因子对贪心方法和期望方法对齐性能的影响。然后，我们介绍在没有真实值的情况下自动确定约束和权重因子的方法。

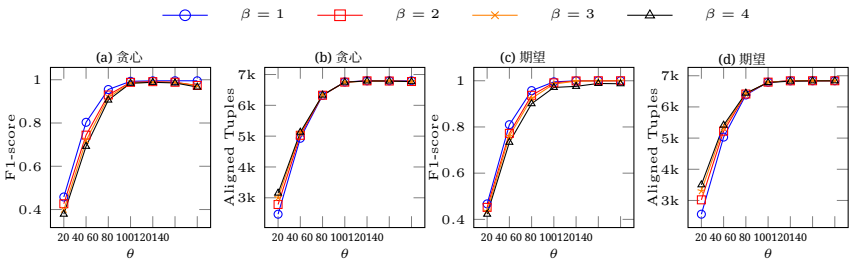

图 1：在缺失率为 20% 的 Household 数据集上，各组 θ 在确定的 $k_1 = 3$、$k_2 = 2$ 和$b = 1$、 $c = 1$ 下的对齐性能。

变化 θ 和 β。 固定模型约束 δ 并设置权重因子集合k_1， k_2， b， c时，在缺失率为 20% 的 Household 数据集上，位置约束 β从 1 变化到 4，时间约束θ从 20 变化到 140。图 1 展示了各组约束下的性能。随着θ 增大，两种方法的表现都变得更好，这并不令人意外，因为满足时间约束的候选元组更多了；但当 θ 过大时，性能会略有下降。这主要是因为候选集中存在更多具有误导性的元组。此外，当 β 变化时，两种方法的表现都很稳定。

确定 θ 和 β。为了确定时间约束 θ，我们遵循 SAMC[3], ，收集记录于相同语义时间、但因网络或硬件延迟而产生差异的变量对之间的非缺失时间戳差值

。随后，我们分析这些差值的分布，并将θ 设为一个较高百分位数阈值，例如第95百分位数，以容忍大多数观测到的延迟。

对于我们所提出的方法中特有的位置约束 β，我们根据候选元组集的结构对其进行估计。首先，我们指定一个初始下界 β'，并使用第3.1节中描述的方法构造候选集 R_c 。然后，我们观察候选集中每个元组内任意两个值之间的行索引距离分布。最后，在满足 $\beta > \beta'$ 的条件下，我们选择该分布的第80百分位数作为位置约束 β。这确保大多数紧凑且结构一致的元组得以保留，同时过滤掉位置间隔过大的元组。

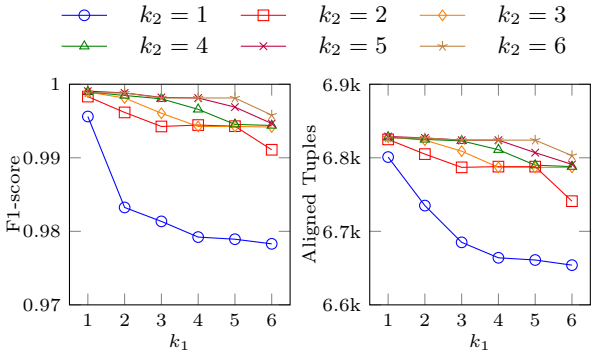

图2：在具有20%缺失的数据集Household上，对于每个 k ，使用已确定的 $\theta = 90$、$\beta = 1$ 和 $b = 1$、 $c = 1$ 时，基于贪心的策略的对齐性能。

改变 δ 和 k_1， k_2。 δ 用于过滤其中一致性不足的对齐结果，因此不同的 δ 不会对对齐性能产生较大影响，包括F1分数和对齐元组的数量。在固定模型约束 δ、时间约束 θ、位置约束 β 和 b， c的情况下，图2展示了在具有20%缺失的数据集Household上，贪心策略的性能。贪心策略的性能会受到 k_1 和 k_2的影响，但始终能够保持在合理范围内，即F1分数不低于0.97，对齐元组不少于6.6k。与贪心策略相比，期望策略表现得更加稳健，F1分数不低于0.99，对齐元组不少于6.8k。

确定 δ 和 k_1， k_2， b， c。 首先，我们可以直接给出 $b = 1$ 和 $c = 1$，因为这两个权重因子的设计目的是确保没有任何元组的权重为零。对于每一对（k_1， k_2），无论模型约束如何，我们都可以得到若干对齐结果。我们使用 Δ 对每个结果进行评估，该指标在

定义 6。具体而言，我们观察相同（k_1，k_2）下每个对齐结果的 $\bar{\Delta}$ 平均值，并选择 $\bar{\Delta}$ 最小的结果。然后，我们选择相应的（k_1，k_2）作为权重因子，并得到模型约束 $\underline{\delta} = \underline{\Delta}$。

7 结论

本文提出了一种无需插补即可对齐不完整多变量时间序列的基于约束的框架。该方法应用时间、位置和模型约束，以确保对齐的有效性和一致性。为应对精确对齐所带来的计算挑战，我们开发了三种在效率与准确性之间取得平衡的近似算法。在真实世界数据集上的实验表明，我们的方法优于现有基线方法，尤其是在缺失率较高的情况下；同时，贪心和期望策略在保持出色性能的同时，也提供了良好的实际可扩展性。

参考文献

1. Bastos, A.F., Santoso, S., Krishnan, V.K., Zhang, Y.：基于机器学习的配电网电压预测与传感器布设。技术报告，国家可再生能源实验室（NREL）（2020） 2. Chandra, B., Halldórsson, M.M.：贪心局部改进与加权集合打包近似。算法期刊 **39**(2)，223–240（2001） 3. Fang, C., Song, S., Mei, Y., Yuan, Y., Wang, J.：关于回归中的元组对齐。载于：第28届ACM SIGKDD知识发现与数据挖掘会议（KDD）论文集。第336–346页（2022） 4. Holmes, E.E., Ward, E.J., Wills, K.：MARSS：用于分析时间序列数据的多元自回归状态空间模型。R期刊 **4**(1)，11–19（2012年） 5. Keogh, E.J., Pazzani, M.J.：导数动态时间规整。载于：2001年SIAM国际数据挖掘会议（SDM）论文集。第1–11页。工业与应用数学学会（2001） 6. Khorram, S., McInnis, M.G., Provost, E.M.：可训练时间规整：在连续时间域中对齐时间序列。载于：IEEE国际声学、语音与信号处理会议（ICASSP）论文集。第3502–3506页（2019） 7. Li, S.Z., Jain, A.K.：动态时间规整（DTW）。载于：《生物特征识别百科全书》，第231页。施普林格美国（2009） 8. Liu, Y., Zhang, H., Li, C., Huang, X., Wang, J., Long, M.：Timer：生成式预训练变换器是大时间序列模型。载于：第41届国际机器学习会议（ICML）论文集。第235卷，第32369–32399页（2024） 9. Marx, S.E., Luck, J.D., Pitla, S.K., Hoy, R.M.：控制器局域网（CAN）总线数据采集的多种硬件/软件解决方案及转换方法比较。农业中的计算机与电子技术 **128**，141–148（2016年） 10. Philipp, P., Altmannshofer, S.：针对具有未同步时钟的网络化控制系统的新型滚动时域估计器方法的实验验证。载于：美国控制会议（ACC）论文集。第4939–4944页。IEEE（2012年） 11. Silva, I., Moody, G., Scott, D.J., Celi, L.A., Mark, R.G.：预测重症监护室患者的院内死亡率：PhysioNet/心脏病学计算挑战赛2012。载于：心脏病学计算。第39卷，第245页（2012年） 12. Tong, Y., She, J., Ding, B., Wang, L., Chen, L.：空间众包中的在线移动微任务分配。载于：2016年IEEE第32届国际数据工程会议（ICDE）论文集。第49–60页（2016年）。https://doi.org/10.1109/ICDE. 2016.7498228 13. Wang, J., Du, W., Yang, Y., Qian, L., Cao, W., Zhang, K., Wang, W., Liang, Y., Wen, Q.：多元时间序列插补的深度学习：综述（2025），https://arxiv.org/abs/2402 04059 14 . . Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M.：TimesNet：面向通用时间序列分析的时间二维变化建模（2022），https://arxiv.org/ abs/2210.02186 15. Ye, M., Yi, X., Jiao, S.：通过参数匹配实现卡车装载式混凝土泵的能量优化。Energy Procedia **88**，574–580（2016年） 16. Zhou, F., la Torre, F.D.：用于人类行为对齐的典型时间规整。载于：神经信息处理系统会议（NeurIPS）论文集。第2286–2294页。Curran Associates, Inc.（2009）

17. Zhou, F., la Torre, F.D.：用于人体运动多模态对齐的广义时间扭曲。载于：IEEE计算机视觉与模式识别会议（CVPR）论文集。第1282–1289页。IEEE计算机学会（2012年） 18. Zhou, F., la Torre, F.D.：广义典范时间扭曲。IEEE模式分析与机器智能汇刊 **38**(2)，279–294（2016年）

A 证明

A.1 命题 1 的证明

时间序列组合过程等价于带有 $m = k$ 的加权k-集合打包（k-SP）问题。给定一个全集 V 和一个集合族 S，其中每个集合都包含来自 V 的 k 个元素，加权k-SP问题旨在从 S 中找到一个总权重最大的子集合族，使这些集合两两不相交。在我们的场景中，令原始数据中的所有单个元组组成的集合为 V，并令R_c 对应于 S。这两个问题是等价的，因为它们都旨在最大化权重和。我们的时间序列组合过程遵循加权集合打包近似 [2], 的框架，该框架已被证明具有有界的近似比。

A.2 命题 2 的证明

采用贪心策略时，对于每个 G，我们有 $\forall g_1,\ g_2 \in G,\ g_1 \asymp g_2$，因此 R_c 中的每个元组 r 都满足 $r \in R_m$ 或 $\exists r_m \in R_m,\ r_m \asymp r$。换言之，$G$ 中必定至少有一个与之冲突的元组。假设精确解包含 N 个元组，记为 $T = \{T_1, T_2 ... T_N\}$。根据问题陈述中的原则 (1)，我们有 $\forall t_1,\ t_2 \in T,\ t_1 \not\asymp t_2$，因此近似解中的每个元组最多会与 T 中的 m 个元组冲突，因为在同一列中，每个元组最多只能与 T 中的一个元组冲突。因此，我们证明了 $|R_m| \geq \lceil \frac{N}{m} \rceil$。

我们有

$$\begin{cases} W_{max} = \dfrac{k_1 \times p_{max} + b}{k_2 \times d_{min} + c} \\ W_{min} = \dfrac{k_1 \times p_{min} + b}{k_2 \times d_{max} + c} \end{cases} \tag{6}$$

p_{max}、p_{min} 分别是对齐元组中的最大和最小配对，d_{max}、d_{min}分别是对齐元组中的最大和最小总距离。

为了构造最小近似权重，我们将 T 划分为若干个集合，每个集合包含 m 个元组。对于每个集合 $S = \{S_1, S_2 ... S_m\}$，我们在候选集中构造两个元组 $m_1,\ m_2$，使其满足：(1) $\forall s \in S,\ m_1,\ m_2 < s$；(2)$m_1 \asymp m_2$；(3) $\forall s \in S,\ m_1 \asymp s$和 $m_2 \not\asymp S_1$；(4) $W(m_1) = W(m_2) = W_{min}$和$\forall s \in S,\ W(s) = W_{max}$。在这种情况下，我们可以得到最小权重和。

因此，最小近似率界保证为 $\frac{\lceil \frac{N}{m} \rceil \times W_{min}}{N \times W_{max}}$，这就是理论最小值。

A.3 引理 1 的证明

对于每个元组集 G，其在所有列中的最大值记为 max，表示为$\forall g \in G,\ i \in [1, m],\ g[i] \leq max$。如果存在任意 $r[i] > max + \beta$，那么根据定义4，$\forall i,\ j \in [1, m],\ |I[(T_i, V_i)] - I[(T_j, V_j)]| \leq \beta$，$r$ 在所有列中的最小最大值为 $max + \beta + 1$，其最小值为 $max + 1$。因此，$\forall i \in [1, m],\ r[i] \in [max + 1, max + \beta + 1]$。所以，始终存在 $r \not\asymp g$。

A.4 命题 3 的证明

将期望近似的对齐结果记为 $R_m = \{E_1, E_2 ... E_l\}$。与贪心策略相同，我们也保证期望近似能够对齐不少于 $\lceil \frac{N}{m} \rceil$个元组。我们假设可以得到权重$W(E_1) = W(E_2) = ... = W(E_{l-1}) = W_{min}$，这就是理论最小值。对于最后一个元组 E_l，有以下两种情况：

1. E_l 位于最后一个 G中。

只有当 E_l 与 S中的一个元组发生冲突时，才会出现这种情况。如果最终选择了 E_l，那么必定有 $W(E_l) \geq W(S)$。当 $W(E_l) = W(S) = W_{min}$ 时，近似值达到最小。因此，我们得到界保证$\frac{\lceil \frac{N}{m} \rceil \times W_{min}(N-1) \times W_{max} + W_{min}}{}$.

2. E_l 不在最后一个 G中。

根据 5，至少存在一个元组 $f_1 \in R_c$，它与至少一个元组$f_2 \in G$不冲突。因此，如果最终选择 E_l，那么必定有 (1) $W(E_l) \geq 2W_{min}$ 或 (2) $\exists f \in R_c$, $f > E_l$ 和 $f \not\asymp E_l$。因此，在这两种情况下，我们都能得到最小近似率界保证 $(\frac{\lceil \frac{N}{m} \rceil + 1) \times W_{min}}{N \times W_{max}}$。

B 实验

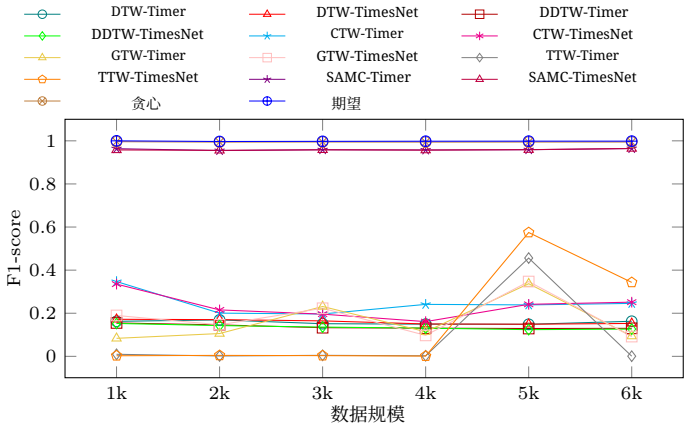

图3：在 Household 数据集缺失 20% 的情况下，对比现有对齐方法在对齐结果F1分数上的表现。