A SUPPLEMENTARY MATERIAL

A.1 Notations

Table 3 summarizes some frequently used notations in this paper.

Table 3: Notations

Symbol	Description
$T_1, T_2, \ldots, T_m$	m variables
τ	the minimum number of values of each variable
θ, δ	thresholds of time and model constraints
R_e, R_s, R_{sm}	tuples aligned by EA, SA and SAMC
R_c, R_{unc}	all candidates and unchosen candidates
R_{in}, R_{out}	subsets of R_{sm} and R_{unc} to be swapped
ρ	threshold of searched subset size

A.2 Special Cases

Problem 1 defines the similarity alignment under model constraint (SAMC). Besides the time constraint in Definition 2, which restricts the similar timestamps, the model constraint in Definition 4 ensures $\Delta(r, M) \leq \delta$ for each $r \in R$.

As we introduced in Section 2, equality alignment, a natural idea of aligning tuples, is based on equal timestamps. Similarity alignment employs time constraint, to further enrich the training data. Indeed, we will prove that they are both special cases of the proposed problem. According to the similarity alignment under model constraint (SAMC), we will give the definitions of the equality alignment (EA) problem and the similarity alignment (SA) problem. An example of their relationships and the pipeline of employing the methods is illustrated in Figure 2.

The equality alignment (EA) problem enforces the same timestamps in aligned tuples, which is an intuitive alignment strategy. The definition of equality alignment is provided as follows.

Problem 2 (Equality Alignment). *Given variables $T_1, T_2, \ldots, T_m$, the* equality alignment *problem is to obtain an aligned instance R, namely R_e, such that (1) each time attribute U_i is a candidate key of R, i.e., each tuple in $T_1, T_2, \ldots, T_m$ can only be aligned once, (2) each tuple $r \in R$ has equal timestamp $r[U_1] = r[U_2] = \cdots = r[U_m]$, and (3) the number of aligned tuples $|R|$ is maximized.*

Solving the equality alignment problem is trivial. By performing join operations $R = T_1 \bowtie T_2 \bowtie \ldots \bowtie T_m$ on $U_1 = U_2 = \cdots = U_m$, it naturally maximizes the number of aligned tuples $|R|$.

The Similarity Alignment (SA) problem employs time constraint alone to filter the aligned tuples.

Problem 3 (Similarity Alignment). *Given variables $T_1, T_2, \ldots, T_m$ and time constraint θ, the* similarity alignment *problem is to obtain an aligned instance R, namely R_s, such that (1) each time attribute $U_1, U_2, \ldots, U_m$ is a candidate key of R, i.e., each tuple in $T_1, T_2, \ldots, T_m$ can be aligned once, (2) each tuple $r \in R$ satisfies the time constraint $\Theta(r) \leq \theta$, and (3) the number of aligned tuples $|R|$ is maximized.*

Rather than equal timestamps in Equality Alignment, the time constraint in Similarity Alignment, defined as time constraint $\Theta(r) \leq \theta$ in Definition 2, considers similar timestamps. It ensures that in each aligned tuple $r \in R$, the distances between any two timestamps are no greater than θ. To solve the similarity alignment problem, our proposed Candidate Generation (Algorithm 1) and Alignment Search (Algorithm 2) are also applied, by ignoring the model constraint part, as well as the pruning strategies in Algorithm 3.

Proposition 8. *Given $\delta = +\infty$, i.e., no model constraint, Problem 1 is equivalent to the similarity alignment. Together with $\delta = 0$, it is indeed the equality alignment.*

A.3 Algorithms

Algorithm 1 provides an overview of the candidate generation. Let $T_1, T_2, \ldots, T_m$ be naturally sorted on timestamps. The algorithm first specifies all the tuples $t_j \in T_j$ in the window of t_k with size θ (Line 3). Model constraint δ is adopted to filter the candidates conflicting with the model M (Line 6). As aforementioned, in the whole process, equality/similarity alignment are used to initialized preliminary R for learning a regression model M to guide the subsequent alignment (illustrated in Figure 2).

Algorithm 2 presents the Alignment Search algorithm. Line 3 initializes R_{sm} with a greedy strategy, i.e., we continuously add $r_c \in R_c$ that does not overlap with any existing $r_{sm} \in R_{sm}$. Lines 10-12 filter possible swapping by the conditions and generate all the possible aligned tuples in the time window that conform to the model M. Line 14 finally conducts the swapping.

Algorithm 3 outlines the Alignment Search algorithm with pruning strategies. Line 9 is proved by Proposition 3, to traverse R_{sm} instead of R_{unc}. Line 10 employs oa-path with extended time constraint to further filter the candidates, which is proved by Proposition 7.

Algorithm 1: Candidate Generation ($T\_List, \theta, M, \delta$)

Input: variables list $T\_List = (T_1, T_2, \ldots, T_m)$, time constraint θ, model constraint (M, δ)
Output: candidates of aligned tuples R_c

```
1 R_c ← ∅;
2 for each T_k ∈ T_list do
3 |   R_k ← {(t_1, t_2, ..., t_m) | ∀1 ≤ i ≤ m, t_i ∈ T_i, t_k[U_k] ≤
  |     t_i[U_i] ≤ t_k[U_k] + θ};
4 |   R_c ← R_c ∪ R_k;
5 for each r ∈ R_c do
6 |   if Δ(r, M) > δ then
7 |   |   R_c ← R_c \ {r}
8 return R_c;
```

A.4 Experiment Settings

A.4.1 Datasets. (1) House [1] is about electric power consumption in one household. The regression model is built on active power, reactive power and intensity attributes to predict voltage. (2) Telemetry [2] is from environmental sensor telemetry data. The regression model is built on carbon monoxide, humidity and smoke attributes

[1] https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+consumption
[2] https://www.kaggle.com/garystafford/environmental-sensor-data-132k

Algorithm 2: Alignment Search (R_c, ρ)

Input: aligned candidates R_c, threshold ρ
Output: aligned tuples R_{sm} without overlapping timestamp

$R_{sm} \leftarrow \emptyset$;
for $r_c \in R_c$ **do**
  if $\forall r_{sm} \in R_{sm}, r_c \not\asymp r_{sm}$ **then**
    $R_{sm} \leftarrow R_{sm} \cup \{r_c\}$;
local optimal $\leftarrow$ False;
while *not local optimal* **do**
  local optimal $\leftarrow$ True;
  for $2 \le p \le \rho$ **do**
    $R_{unc} \leftarrow R_c \setminus R_{sm}$;
    for $R_{out} \subseteq R_{unc}, |R_{out}| = p$ **do**
      if $\forall r_a, r_b \in R_{out}, r_a \not\asymp r_b$ **then**
        $R_{in} \leftarrow \{r_{in} \in R_{sm} | \exists r_{out} \in R_{out}, r_{out} \asymp r_{in}\}$;
        if $|R_{in}| \le p - 1$ **then**
          $R_{sm} \leftarrow R_{sm} \setminus R_{in} \cup R_{out}$;
          local optimal $\leftarrow$ False;
return R_{sm};

Algorithm 3: Alignment Search-Pruning (R_c, ρ, θ)

Input: aligned candidates R_c, threshold ρ, time constraint θ
Output: aligned tuples R_{sm} without overlapping timestamp

$R_{sm} \leftarrow \emptyset$;
for $r_c \in R_c$ **do**
  if $\forall r_{sm} \in R_{sm}, r_c \not\asymp r_{sm}$ **then**
    $R_{sm} \leftarrow R_{sm} \cup \{r_c\}$;
local optimal $\leftarrow$ False;
while *not local optimal* **do**
  local optimal $\leftarrow$ True;
  for $1 \le p \le \rho - 1$ **do**
    for $R_{in} \subseteq R_{sm}, |R_{in}| = p$ **do**
      if $\forall r_a \in R_{in}, \forall r_b \in R_{in}, \forall i \in \{1, 2, \ldots, m\}, |t_{ia} - t_{ib}| \le (2p-1)\theta$ **then**
        $R_{out} \leftarrow \{r_{out} \in R_{unc} | \exists r_{in} \in R_{in}, r_{out} \asymp r_{in}\} \cap \{r_{out} \in R_{unc} | \forall r \in R_{sm} \setminus R_{in}, r \not\asymp r_{out}\}$;
        if $|R_{out}| \ge p + 1$ **then**
          $R_{sm} \leftarrow R_{sm} \setminus R_{in} \cup R_{out}$;
          local optimal $\leftarrow$ False;
return R_{sm};

to predict temperature. (3) Water [3] is collected by sensors from water quality monitoring stations. The regression model is built on electric conductivity, water temperature and turbidity to predict water level. (4) Air quality [4] is air pollutants data from air-quality monitoring sites in Beijing. It is a high-dimensional dataset with 11 attributes, the regression model is to predict PM10 by other attributes. (5) Fuel is fuel consumption data of vehicles, collected by our partner company, motivating this study. The regression model is built on engine torque and speed to predict fuel consumption.

[3] https://www.kaggle.com/ivivan/real-time-water-quality-data
[4] https://archive.ics.uci.edu/ml/datasets/Beijing+Multi-Site+Air-Quality+Data

For House, Telemetry and Air Quality, their timestamps are all naturally aligned to serve as ground truth. We simulate the unaligned variables by introducing disturbance on timestamps [8]. For Fuel, due to the aforementioned issues like transmission delays, only about 5% data are naturally aligned. Analogously, for Water dataset, among 29k tuples, only around 13k tuples are naturally aligned. We thus consider the part of naturally aligned tuples as the ground truth of evaluation, denoted as R_{truth}.

A.4.2 Regression Models. We employ regression models including Linear Regression (LR) and XGBoost (XGB [11]), implemented by scikit-learn[5]. TabNet [6] is a more recent canonical deep tabular data learning architecture for multiple regression.

A.4.3 Alignment Methods. We compare our proposed **S**imilarity **A**lignment under **M**odel **C**onstraint (SAMC) with existing methods: (1) dynamic time warping (DTW) [4], a dynamic programming-based algorithm; (2) derivative dynamic time warping (DDTW) [22], an extension of DTW considering the local derivatives of the data; (3) canonical time warping (CTW) [36], an extension of canonical correlation analysis (CCA) for spatio-temporal alignment of human motion; (4) generalized time warping (GTW) [37, 38], which extends DTW for temporally aligning multi-modal sequences from multiple subjects; (5) trainable time warping (TTW) [23] utilizes a sinc convolutional kernel and a gradient-based optimization technique for multiple alignment. Since our scenario focuses on aligning multiple variables, for DTW, DDTW and CTW proposed only on pairs of time series, we extend them with the framework of Procrustes analysis [37] for handling multiple time series alignment.

A.4.4 Evaluation Metrics. We propose to evaluate how accurate the methods align the tuples (alignment accuracy) and how accurate the models are learned from the aligned data (model accuracy).

For alignment accuracy, we compare the instance R, aligning $T_1, T_2, \ldots, T_m$ by various algorithms, to the truth R_{truth} introduced in Section A.4.1. The alignment accuracy is given by

$$\text{F1-score} = 2 * \frac{precision * recall}{precision + recall},$$

where $recall = \frac{R_{truth} \cap R}{R_{truth}}$, and $precision = \frac{R_{truth} \cap R}{R}$.

For model accuracy, we reserve 20% naturally aligned tuples of each dataset as test data R_{test} for evaluating the learned models. For a learned regression model M, given independent variables $V_1, V_2, \ldots, V_{m-1}$ and dependent variable V_m, we employ RMSE, i.e.,

$$RMSE(M, R_{test}) = \sqrt{\frac{\sum_{r \in R_{test}} (M(r[V_1], \ldots, r[V_{m-1}]) - r[V_m])^2}{|R_{test}|}}.$$

The lower the RMSE is, the better the alignment and learned models are.

A.4.5 Implementation Details. In our **S**imilarity **A**lignment under **M**odel **C**onstraint (SAMC), the model M is initialized by **S**imilarity **A**lignment (SA), as illustrated in Figure 2. The learned model M is then utilized and updated in SAMC as described in Section 2.1.

The experiments are conducted on an Ubuntu 16.04 LTS machine with 16 2.1GHZ cores and 128 GB memory. The large-scale data alignment is executed on a cluster of 3 nodes with Apache Spark 2.2.3. Each node has 32 GB memory and 16 2.6GHz cores. Experimental code and data are anonymously available in [3].

PROOFS

A.5 Proof of Proposition 8

Given $\delta = +\infty$, i.e., no model constraint in Problem 1, which derives the similarity alignment in Appendix A.2.

When given $\theta = 0$ together, the time constraint $\Theta(r) \leq \theta = 0$ enforces the same timestamps in the aligned tuple, i.e., $r[U_1] = r[U_2] = \cdots = r[U_m]$, which is exactly required in equality alignment in Appendix A.2.

A.6 Proof of Theorem 1

We first show that the problem is in NP. Given an aligned instance R, all the three conditions can be verified in polynomial time. Condition (1) is verified by comparing the tuples with each other in $O(|R|^2)$ time. Recall that $|R| \leq \min\{|T_1|, |T_2|, \ldots, |T_m|\} = \tau$. For condition (2), the time constraint θ can be checked by traversing the tuples in $O(|R|)$ time and the model constraint δ is examined by searching the nearest neighbor for each tuple $r \in R$ in $O(|R| \log |M|)$ time. Condition (3) can simply be checked by comparing $|R|$ and κ.

Next, we will start from the scenario when $m = 3$ and build a reduction from the *maximum 3-dimensional matching* [17, 20], one of Karp's 21 NP-complete problems [21], to our alignment problem. Finally, we can show that the reduction is also adaptive to $m \geq 4$, thus prove the NP-hardness. Combining with that the problem is in NP, we could conclude the NP-completeness of the problem.

Let A, B, C be finite disjoint sets, and $P_c \subseteq A \times B \times C$, i.e., $P_c = \{(a, b, c) | a \in A, b \in B, c \in C\}$. For $p_1(a_1, b_1, c_1), p_2(a_2, b_2, c_2) \in P$, we say that p_1 and p_2 intersect on some coordinate, denoted by $p_1 \leftrightarrow p_2$, if either $a_1 = a_2$, $b_1 = b_2$ or $c_1 = c_2$. We use $P \subseteq P_c$ to denote a 3-dimensional matching if no element in P intersects with others, i.e., $\forall p_1, p_2 \in P, p_1 \leftrightarrow p_2$. The *maximum 3-dimensional matching* is to find the matching P^* among all the possible values of P with the largest amount of triples. The decision problem is, given an integer κ, to decide whether there exists a 3-dimensional matching P such that $|P| \geq \kappa$.

For an instance of the *maximum 3-dimensional matching* problem with P_c, to construct a similarity alignment problem under model constraint, we first set $\theta = +\infty$, i.e., the time constraint is satisfied by any two aligned tuples. Then, we create T_1, T_2, T_3 in our problem by assigning unique timestamps to them. In the meantime, let each $t_{1i} \in T_1$ correspond to $a_i \in A$, each $t_{2j} \in T_2$ correspond to $b_j \in B$ and each $t_{3k} \in T_3$ correspond to $c_k \in C$. Next, if $(a_i, b_j, c_k) \in P_c$, we assign $t_{1i}[V_1]$, $t_{2j}[V_2]$ and $t_{3k}[V_3]$ unique combination of values. Then, we set the regression model to satisfy $M(t_{1i}[V_1], t_{2j}[V_2]) = t_{3k}[V_3]$. Let model constraint $\delta = 0$, since each $(t_{1i}[V_1], t_{2j}[V_2], t_{3k}[V_3])$ is unique, we have $R_c = \{(t_{1i}, t_{2j}, t_{3k}) | M(t_{1i}[V_1], t_{2j}[V_2]) = t_{3k}[V_3]\}$. Therefore, we obtain $(t_{1i}, t_{2j}, t_{3k}) \in R_c$ if and only if $(a_i, b_j, c_k) \in P_c$.

Next, we will show that, for each satisfied 3-dimensional matching P, we have $|P| \geq \kappa$, if and only if the aligned instance $R = \{(t_{1i}, t_{2j}, t_{3k}) | M(t_{1i}[V_1], t_{2j}[V_2]) = t_{3k}[V_3]\}$ corresponding to P in our problem is also the set that satisfies (1) three candidate keys U_1, U_2, U_3, (2) time constraint $\Theta(r) \leq \theta$ and model constraint $\Delta(r, M) \leq \delta$ satisfied for each $r \in R$, and (3) $|R| \geq \kappa$.

First, according to the definition, we suppose $|P| \geq \kappa$. Since $\forall p_1, p_2 \in P, p_1 \leftrightarrow p_2$, for the corresponding R, we will have $\forall r_1, r_2 \in R$, $r_1 \not\asymp r_2$, thus condition (1) is satisfied. For condition (2), first recall that we construct our alignment problem by supposing the time constraint θ large enough. Moreover, we generate R_c by satisfying model constraint, and thus condition (2) is satisfied automatically. Finally, since we have $(t_{1i}, t_{2j}, t_{3k}) \in R$ if and only if $(a_i, b_j, c_k) \in P$, $|R| = |P| \geq \kappa$ holds, condition (3) is satisfied.

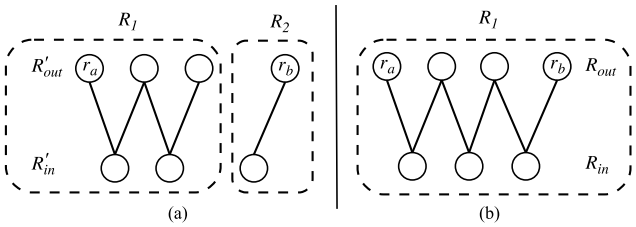

Figure 11: An example of (a) disconnected and (b) connected graphs of oa-path

Conversely, suppose that $|R|$ satisfies conditions (1), (2), (3). Following similar steps, according to the condition (1), $\forall r_1, r_2 \in R$, $r_1 \not\asymp r_2$, the corresponding P has $p_1 \leftrightarrow p_2, \forall p_1, p_2 \in P$. Next, according to condition (3), $|R| \geq \kappa$. Since $(t_{1i}, t_{2j}, t_{3k}) \in R$ if and only if $(a_i, b_j, c_k) \in P$, we will get $|P| = |R| \geq \kappa$. The conditions of the 3-dimensional matching problem are satisfied.

A.7 Proof of Proposition 2

We first show that the procedure of Problem 1 similarity alignment under model constraint is equivalent to the k-Set Packing (k-SP) problem with $m = k$. Given a ground set V and a collection $\mathcal{S}$ of sets, each of them contains k element from V, k-SP problem is to find a maximum subcollection of $\mathcal{S}$ so they are pairwise disjoint. In our scenario, let the set of all single tuples from origin data be V, and let R_c correspond to $\mathcal{S}$. The equivalence of both problems is intuitive since they both maximize the target set.

Next, our Alignment Search (Algorithm 2) follows the framework of ρ-optimal local search algorithm for k-SP problem [19, 26], which is proved to have a bounded approximation ratio. According to the proof in [19], for m-dimensional time series and parameter ρ for Alignment Search, the bounds on approximation ratio ξ are obtained.

A.8 Proof of Proposition 3

Condition (4) requires that any tuple $r_{out} \in R_{out}$ must overlap with at least a tuple in R_{in}. Assume that a tuple r_{out} does not overlap with any tuple in R_{in}. Combining with condition (3), r_{out} does not overlap with any tuple in R_{sm}. This is impossible, since the initialization of R_{sm} has ensured every tuple in R_{unc} overlaps with at least one tuple in R_{sm}. Moreover, each swap will also ensure this, since it only swaps out the tuples with conflicts. Hence, any tuple $r_{out} \in R_{out}$ must overlap with at least a tuple in R_{in}.

Condition (5) requires that any tuple $r_{in} \in R_{in}$ must overlap with at least a tuple in R_{out}. This is because Line 8 of Alignment Search algorithm ensures we traverse R_{sm} by increasing the subset size p. If $r_{in} \in R_{in}$ does not overlap with any tuple in R_{out}, in the former iteration with smaller p, the subset $R'_{in} = R_{in} \setminus \{r_{in}\}$ should already be swapped with the current R_{out}.

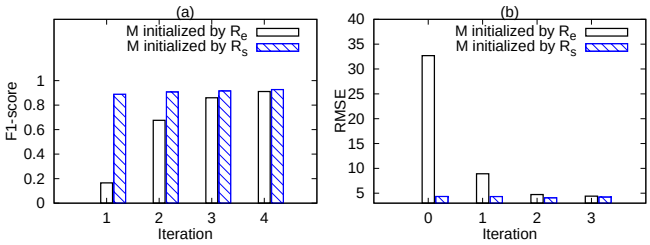

Figure 12: Alignment accuracy and model prediction performance by multiple iterations of similarity alignment under time constraint $\theta = 80$ and model constraint $\delta = 8$ on House

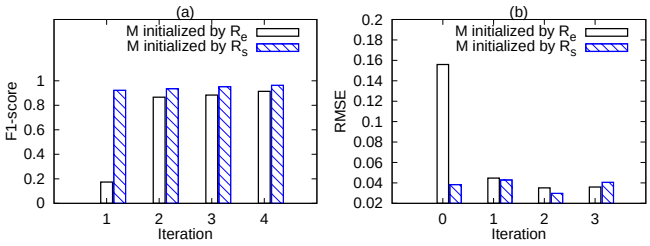

Figure 13: Alignment accuracy and model prediction performance by multiple iterations of similarity alignment under time constraint $\theta = 85$ and model constraint $\delta = 0.5$ on Water

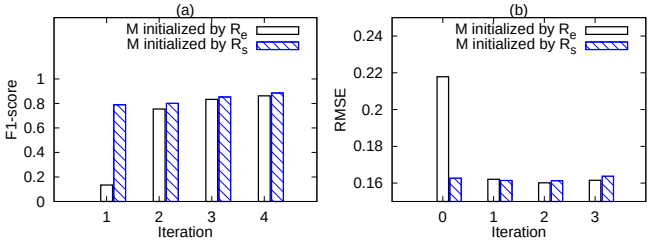

Figure 14: Alignment accuracy and model prediction performance by multiple iterations of similarity alignment under time constraint $\theta = 75$ and model constraint $\delta = 0.5$ on Telemetry

A.9 Proof of Lemma 4

For simplicity, here we slightly abuse t_{ki} to denote $t_{ki}[U_k]$, i.e., the timestamp of t_{ki}. Let t_i^{max}, t_i^{min} denote the maximum and minimum timestamps of r_i, for consecutive r_i and r_{i+1} in the oa-path. Since $r_i \asymp r_{i+1}$, r_i and r_{i+1} must overlap on some timestamp t_k, i.e., $t_{ki} = t_{k(i+1)}$. According to the time constraint in Definition 2, we have $t_{ki} \le t_i^{max} \le t_{ki} + \theta$ and $t_{k(i+1)} \le t_{i+1}^{max} \le t_{k(i+1)} + \theta$. Therefore, we obtain $t_{i+1}^{max} \le t_{k(i+1)} + \theta = t_{ki} + \theta \le t_i^{max} + \theta$, i.e., $t_{i+1}^{max} \le t_i^{max} + \theta$. By iteratively applying this formula, we have $t_l^{max} \le t_{l-1}^{max} + \theta \le t_{l-2}^{max} + 2\theta \le \cdots \le t_1^{max} + (l-1)\theta \le t_1^{min} + l\theta$. Similarly, we could also prove $t_1^{max} \le t_l^{min} + l\theta$. Hence, we obtain $t_1^{max} \le t_l^{min} + l\theta$ and $t_l^{max} \le t_1^{min} + l\theta$, indicating $|t_1^{max} - t_l^{min}| \le l\theta$ and $|t_l^{max} - t_1^{min}| \le l\theta$.

Finally, we have $|t_{k1} - t_{kl}| \le \max(|t_1^{max} - t_l^{min}|, |t_l^{max} - t_1^{min}|) \le l\theta, \forall k \in \{1, 2, \ldots, m\}$ i.e., $|t_{k1}[U_k] - t_{kl}[U_k]| \le l\theta$.

A.10 Proof of Lemma 5

In condition (2) in Section 3.3.1 and the requirement of the chosen candidates in R_{sm}, tuples in R_{out} and R_{in} should not overlap with other tuples in the same set. Therefore, if r_a and r_b are connected by a path, the tuples on path must be alternatively chosen from R_{out} and R_{in}, i.e., an oa-path.

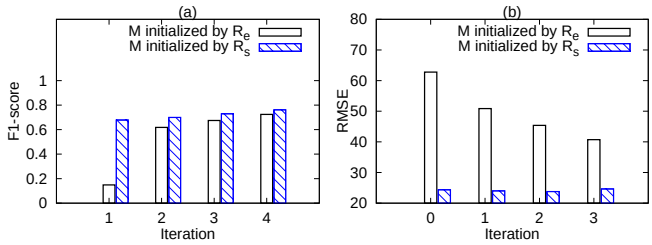

Figure 15: Alignment accuracy and model prediction performance by multiple iterations of similarity alignment under time constraint $\theta = 35$ and model constraint $\delta = 60$ on Air Quality

A.11 Proof of Lemma 6

Firstly, we will prove that an oa-path $(r_a, \ldots, r_b)$ must exist. Since R_{in} could be swapped by R_{out}, R_{in} and R_{out} should satisfy conditions (4) and (5) of Proposition 3, i.e., each tuple r in R_{out} and R_{in} overlaps with at least one other tuple. If there does not exist an oa-path $(r_a, \ldots, r_b)$, this graph is disconnected, i.e., at least two vertices of the graph are not connected by a path. Without loss of generality, we can assume that the disconnected path could be divided into z connected parts $\mathcal{R} = \{R_1, R_2, \ldots, R_z\}$, since each tuple r at least overlaps one other tuple, $\forall R_i \in \mathcal{R}$, we have $|R_i| \ge 2$. Recall that $|R_{out}| > |R_{in}| = p$, that is, there must exist $R_i \in \mathcal{R}$, $|R_i \cap R_{out}| > |R_i \cap R_{in}|$ (see Figure 11(a) for an example). Let $R'_{out} = R_i \cap R_{out}$ and $R'_{in} = R_i \cap R_{in}$. We can safely swap R'_{in} with R'_{out}, since tuples in R'_{out} do not overlap with other tuples outside R'_{in}, and $|R'_{out}| > |R'_{in}|$. Hence, we find that $|R'_{in}| < |R_{in}| = p$ could be swapped with R'_{out}. It conflicts with the assumption in Lemma 6 that the algorithm has found all possible swaps with $|R'_{in}| < p$. To conclude, an oa-path $(r_a, \ldots, r_b)$ must exist, i.e., this graph is a connected graph (see Figure 11(b) for an example).

Next, we will show the length of the oa-path P_{oa} is less than $2p - 1$. Recall that $|R_{in}| = p$. There are at most p tuples from R_{in} in the oa-path, i.e., $|R_{in} \cap P_{oa}| \le p$. Since tuples from R_{out} and R_{in} alternatively occur in the oa-path, and the start and end vertices r_a and r_b are both from R_{in}, there are at most $p - 1$ tuples from R_{out} in the oa-path, i.e., $|R_{out} \cap P_{oa}| \le p - 1$. Therefore, combining with $P_{oa} \subseteq R_{out} \cup R_{in}$, we have $|P_{oa}| = |R_{out} \cap P_{oa}| + |R_{in} \cap P_{oa}| \le 2p - 1$.

In summary, we prove that $\forall r_a, r_b \in R_{in}$, there must exist an oa-path $P_{oa} = (r_a, \ldots, r_b)$ with length $\le 2p - 1$.

A.12 Proof of Proposition 7

By combining Lemmas 4 and 6, an oa-path from r_a to r_b exists, having $|t_{ka}[U_k] - t_{kb}[U_k]| \le (2p-1)\theta, \forall k \in \{1, 2, \ldots, m\}$. That is, for any tuple pair $r_a, r_b \in R_{in}$, the differences between the timestamps of r_a and r_b are no greater than $(2p-1)\theta$.

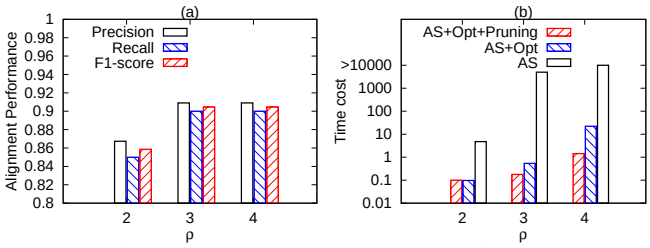

Figure 16: (a) Alignment performance and (b) time cost of SAMC by varying ρ over Water dataset with $\theta = 85$ and $\delta = 0.5$

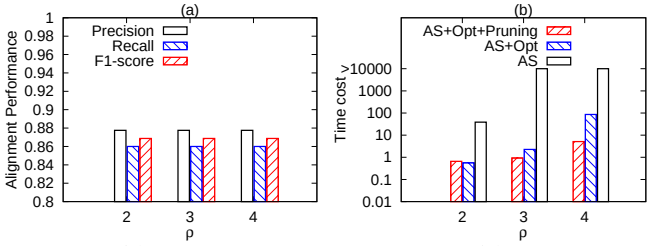

Figure 17: (a) Alignment performance and (b) time cost of SAMC by varying ρ over Telemetry dataset with $\theta = 75$ and $\delta = 0.5$

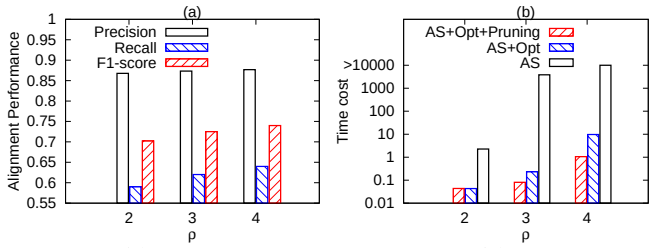

Figure 18: (a) Alignment performance and (b) time cost of SAMC by varying ρ over Air Quality dataset with $\theta = 35$ and $\delta = 60$

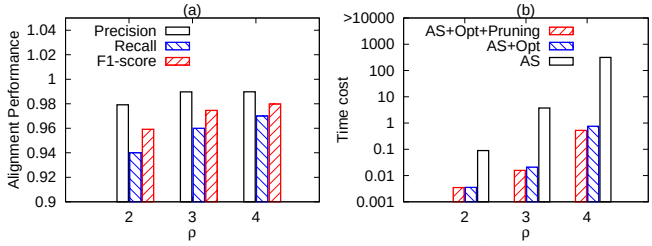

Figure 19: (a) Alignment performance and (b) time cost of SAMC by varying ρ over Fuel dataset with $\theta = 120$ and $\delta = 35$

ADDITIONAL RESULTS

A.13 Additional Results for Section 4.2.1

Figures 12-15 present the results of the proposal under various number of iterations over other datasets.

A.14 Additional Results for Section 4.2.2

Figures 16-19 resent the results of the proposal by varying ρ, the largest size of the sets we will consider for swapping, over other datasets.