	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
subtask 1 for o_1	$\{o_2\}$	$\{o_2\}$	ø	ø	$\{o_2\}$	ø	$\{o_2,o_3\}$	ø
subtask 2 for o_2	ø	ø	$\{o_3,o_4,o_5,o_6,o_7,o_8\}$	ø	ø	$\{o_3,o_4,o_5,o_6\}$	$\{o_3\}$	ø
subtask 3 for o_3	$\{o_4\}$	$\{o_4,o_5\}$	$\{o_4,o_5,o_6,o_7,o_8\}$	ø	ø	$\{o_4,o_5,o_6\}$	ø	ø
subtask 4 for o_4	ø	$\{o_5\}$	$\{o_5,o_6,o_7,o_8\}$	$\{o_5,o_6,o_7\}$	$\{o_5\}$	$\{o_5,o_6\}$	$\{o_5,o_6,o_7\}$	$\{o_5,o_6,o_7\}$
subtask 5 for o_5	$\{o_6,o_7\}$	ø	$\{o_6,o_7,o_8\}$	$\{o_6,o_7\}$	ø	$\{o_6\}$	$\{o_6,o_7\}$	$\{o_6,o_7\}$
subtask 6 for o_6	$\{o_7\}$	$\{o_7\}$	$\{o_7,o_8\}$	$\{o_7\}$	$\{o_7\}$	ø	$\{o_7\}$	$\{o_7\}$
subtask 7 for o_7	ø	ø	$\{o_8\}$	ø	ø	$\{o_8\}$	ø	ø
subtask 8 for o_8	ø	ø	ø	ø	ø	ø	ø	ø

η

Figure 7: Example of Id-based Partitioning for Fig. 2

time	3	4	5	6	7	8	
o_5	1	1	1	1	1	1	√
o_6	1	1	0	1	1	1	√
o_7	1	1	0	0	1	1	√
o_8	1	0	0	0	0	0	×
$\{o_5, o_6\}$	1	1	0	1	1	1	√
$\{o_5, o_7\}$	1	1	0	0	1	1	√
$\{o_6, o_7\}$	1	1	0	0	1	1	√
$\{o_5, o_6, o_7\}$	1	1	0	0	1	1	√

Figure 8: Bit Compression on $P_3(o_4)$

6.1 Baseline

We adapt the state-of-the-art distributed co-movement pattern detection method (i.e., SPARE [10]) on historical trajectories as the baseline algorithm. We note that SPARE uses a star partitioning scheme for historical trajectories. However, this partitioning cannot be applied to streaming trajectories, because we do not know which trajectories are related in advance, and they thus cannot be distributed to the same partition at the beginning. Instead, we present an id-based partitioning technique.

ID-based Partitioning Technique. A Flink subtask is created for each trajectory o with the *key* equaling to the trajectory id $o.id$. We use partition $P_t(o)$ to denote the set of trajectories distributed to subtask $o.id$ at time t. In order to avoid duplications, $P_t(o)$ contains other trajectories (except for o) in the same cluster with their ids larger than $o.id$. Note that, at different times t_i, partitions $P_{t_i}(o)$ will be sent to the same subtask $o.id$ for processing.

Fig. 7 shows all the partitions for Fig. 2, where a subtask is created for each of 8 trajectories. At time 1, the cluster snapshot is $\{(o_1, o_2), (o_3, o_4), (o_5, o_6, o_7)\}$, which results in partitions: $P_1(o_1) = \{o_2\}$ for subtask 1, $P_1(o_2) = \emptyset$ for subtask 2, $P_1(o_3) = \{o_4\}$ for subtask 3, $P_1(o_4) = \emptyset$ for subtask 4, $P_1(o_5) = \{o_6, o_7\}$ for subtask 5, $P_1(o_6) = \{o_7\}$ for subtask 6, $P_1(o_7) = \emptyset$ for subtask 7, and $P_1(o_8) = \emptyset$ for subtask 8.

Based on the significance constraint M, the lemma below is used to find the valid clusters for a partition.

LEMMA 3. *Given a cluster C and a significance constraint M, if $|C| < M$, then C can be discarded.*

PROOF. The proof is simple due to the significance constraint M, and it is thus omitted. □

As an example, in Fig. 2, if $M = 3$, the clusters $\{o_1, o_2\}$ and $\{o_3, o_4\}$ at time 1 can be discarded.

Pattern Enumeration. For each partition $P_t(o)$ at time t, we first enumerate all possible combinations of trajectories, and then find the valid time sequence for each combination. Specifically, we first initialize all possible patterns $O \subseteq P_t(o) \cup \{o\}$, where $|O| \geq M$. Note that, pattern enumeration on partition $P_t(o)$ should include o. For simplicity, the pattern enumeration on partition $P_t(o)$ removes o because o is a common trajectory. Considering the example in Fig. 7, given a partition $P_1(o_5) = \{o_6, o_7\}$ and $M = 2$, the possible patterns include $\{o_6\}$, $\{o_7\}$, and $\{o_6, o_7\}$, where o_5 is a comment element and is omitted in the patterns.

Pattern Verification. Next, we determine whether each pattern O enumerated in $P_t(o)$ is valid, i.e., we try to find the valid time sequence T for O in the partitions $P_i(o)$ $(i \geq t)$. More specifically, for a pattern O, T is first initialized to $\{t\}$. If O also exists in $P_{t'}(o)$ at the next time t', then $T = T \cup \{t'\}$. If T satisfies the K, L, and G constraints in Definition 4, then O is valid. As proved in [10], no valid pattern is missed if every η snapshots are verified.

LEMMA 4. $\eta = (\lceil \frac{K}{L} \rceil - 1) \times (G-1) + K + L - 1$ *guarantees that no valid pattern is missed.*

PROOF. The proof can be found elsewhere [10]. □

Hence, for pattens enumerated in $P_t(o)$, we need to use η snapshots $P_i(o)$ $(t \leq i \leq t+\eta-1)$ to determine whether they are valid. For example, in Fig. 7, if $K = 4$ and $G = L = 2$, then $\eta = 6$, and thus, we use $P_i(o)$ $(1 \leq i \leq 6)$ to verify the patterns enumerated in $P_1(o)$. Although η snapshots are being processed at the same time, this does not mean that our methods are batch methods. This processing is simply necessary because multiple snapshots are needed for verifying the validity according to Definition 4.

Based on the consecutive constraint L and the connection constraint G, two lemmas [10] can be used to avoid unnecessary verifications when finding valid patterns.

LEMMA 5. *Given a pattern O enumerated in partition $P_t(o)$, a consecutive constraint L, and a time sequence T obtained before the current time t', assume that the last time segment T_l of T satisfies $|T_l| < L$. If $O \subseteq P_{t'}(o)$ and $t' - max(T) \neq 1$, then O can be discarded.*

PROOF. The proof is straightforward due to $T = T \cup \{t'\}$ does not satisfy consecutive constraint L. □

For instance, in Fig. 7, considering a pattern $O = \{o_2\}$ enumerated in $P_1(o_1)$, we can get $T = \langle 1, 2, 5 \rangle$ before the current time $t' = 7$. Given $L = 2$, the length of the last time segment $T_l = \{5\}$ in T is smaller than L. In addition, as $O \subseteq P_7(o_1)$ and $t - max(T) = 7 - 5 = 2 > 1$, $O = \{o_2\}$ can be discarded. This holds because $T = \langle 1, 2, 5, 7 \rangle$ does not satisfy the consecutive constraint L.

LEMMA 6. *Given a pattern O enumerated in partition $P_t(o)$, a connection constraint G, and a time sequence T obtained before the current time t', if $O \subseteq P_{t'}(o)$ and $t' - max(T) > G$, then O can be discarded.*

PROOF. For a time sequence T obtained before the current time t', if $O \subseteq P_{t'}(o)$ and $t' - max(T) > G$, then $T \cup \{t'\}$ does not satisfy constraint G, and thus, O can be discarded. □

For example, in Fig. 7, considering a pattern $O = \{o_4\}$ enumerated in $P_1(o_3)$, we can get $T = \langle 1, 2, 3 \rangle$ before the current time $t' = 6$. If $G = 2$, $O \subset P_6(o_3)$ and $t' - max(T) = 6 - 3 = 3 > 2$, then O can be discarded according to Lemma 6.

Based on Lemmas 3 to 6, we present Baseline algorithm, with the pseudo-code shown in Algorithm 3. It takes as inputs a partition $P_t(o) = \{o_i | 1 \leq i \leq |P_t(o)|\}$ and four constraints (M, K, L, G). First, Baseline initializes an empty list H, and computes $\eta = (\lceil \frac{K}{L} \rceil - 1) \times (G-1) + K + L - 1$ (line 1). Then, it enumerates all possible