the example in Fig. 2, a dotted circle denotes a cluster. Given $M = 3, K = 4, L = 2$, and $G = 2$, then $O = \{o_4, o_5, o_6\}$ is a co-movement pattern. Specifically, for $T = \langle 3, 4, 6, 7\rangle$, the following hold: (i) o_4, o_5, and o_6 belong to the same cluster at times 3, 4, 6, and 7; (ii) $|O| = 3$; (iii) $|T| = 4$; and (iv) T is 2-consecutive and 2-connected.

In a setting with streaming trajectories, GPS records are produced continuously over time. Therefore, we define the notion of streaming trajectory below.

DEFINITION 5. **(Streaming Trajectory)**. *A streaming trajectory is an unbounded ordered sequence of GPS records, i.e.,* $o = \langle r_1, r_2, ...\rangle$.

In Fig. 2, o_1 to o_8 are streaming trajectories. A streaming trajectory is unbounded, i.e., the next GPS record and the total length of the trajectory are unknown in advance, which makes real-time co-movement pattern mining more difficult. Here, "real-time" means being able to process the data, and show it in results as soon as it arrives. For the setting of stream processing, we introduce the notion of a snapshot and the real-time co-movement pattern mining.

DEFINITION 6. **(Snapshot)**. *A snapshot* $S_t = \{o_1.l, o_2.l, ..., o_n.l\}$ *contains all the locations of a trajectory set* $\{o_1, o_2, ..., o_n\}$ *at time* t.

For simplicity, we use $\{o_1, o_2, ..., o_n\}$ to represent $\{o_1.l, o_2.l, ..., o_n.l\}$. In Fig. 2, there exist eight snapshots.

DEFINITION 7. **(Real-time Co-movement Pattern Mining)**. *Given parameters* M, K, L, *and* G *that defines a general co-movement pattern, real-time co-movement pattern mining finds all co-movement patterns in the snapshot set* $S = \{S_1, S_2, ..., S_t\}$, *where* t *is the current time* t.

As an example, if the current time is 5, $\{o_4, o_5\}$ and $\{o_6, o_7\}$ are *CP*(2, 4, 2, 2) patterns where $T = \langle 2, 3, 4, 5\rangle$. However, no *CP*(3, 4, 2, 2) pattern exists until time 7, where $\{o_4, o_5, o_6\}$ qualifies with $T = \langle 3, 4, 6, 7\rangle$.

3.2 DBSCAN

DBSCAN [9] is a popular density-based clustering method. It relies on two parameters to characterize density or sparsity, i.e., a positive real value ϵ and a positive integer $minPts$. Next, we introduce the definitions of core point and density reachable point.

DEFINITION 8. **(Core Point)** *A location* u *is a core point if at least* $minPts$ *locations* v *satisfy* $d(u, v) \leq \epsilon$, *where* $d(u, v)$ *denotes the distance between* u *and* v.

DEFINITION 9. **(Density Reachable Point)** *A location* u *is density reachable from location* v *if there exist a sequence of locations* $x_1, x_2, ..., x_t (t \geq 2)$ *such that* (*i*) $x_1 = v$ *and* $x_t = u$; (*ii*) x_i $(1 \leq i < t)$ *are core points; and* (*iii*) $d(x_i, x_{i+1}) \leq \epsilon$ $(1 \leq i < t)$.

Based on Definitions 8 and 9, a cluster is formed by a set of core points and their density reachable points. At time 3 in Fig. 2, given the ϵ shown in the figure and $minPts = 3$, o_3, o_4, o_5, o_6, and o_7 are core points, while o_2 and o_8 are density reachable points. Thus, a cluster $\{o_i | 2 \leq i \leq 8\}$ is formed. By scanning the whole data set, we can find all clusters.

3.3 Range Join

According to Definition 8, to determine whether u is a core point, we need to find all locations v in each snapshot S_t with their distances to u satisfying $d(u, v) \leq \epsilon$. A range query can be employed to find core points.

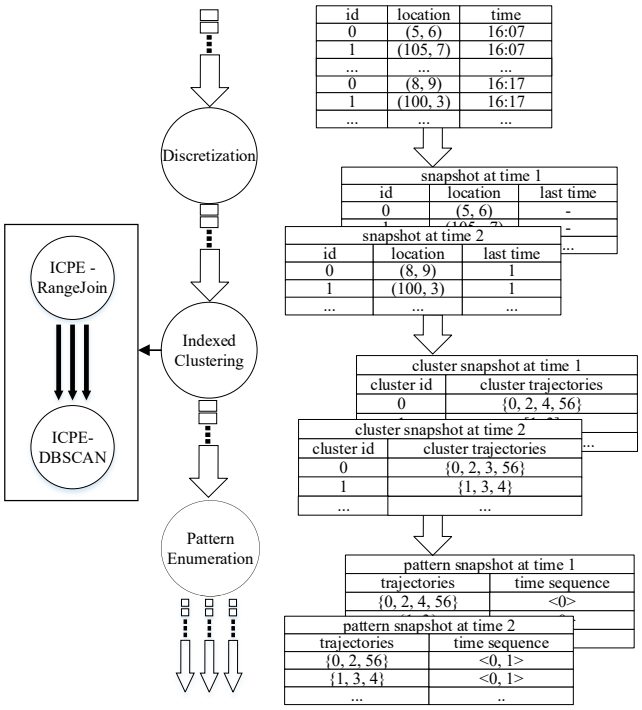

Figure 3: Indexed Clustering and Pattern Enumeration (ICPE)

DEFINITION 10. **(Range Query)** *Given a set* O *of locations, a threshold* ϵ, *and a query location* u, *a range query finds all locations* v *in* O *for* u *with their distances to* u *no larger than* ϵ, *i.e.,* $RQ(u, \epsilon) = \{(u, v) | d(u, v) \leq \epsilon, v \in O\}$.

We use the L_1-norm to measure the distance between two locations, although it is easy to also support other distance functions. A range query $RQ(u, \epsilon)$ retrieves all locations v located in the range region $([u.x - \epsilon, u.x + \epsilon], [u.y - \epsilon, u.y + \epsilon])$, e.g., the red square in Fig. 2. Thus, at time 1, $RQ(o_6, \epsilon) = \{(o_6, o_5), (o_6, o_7)\}$.

For clustering, we have to check every object o in S_t to determine whether o is a core point, i.e., we perform a range query for every object o. Therefore, a range join can be used in the first step of DBSCAN in order to improve efficiency.

DEFINITION 11. **(Range Join)** *Given a set* O *of locations and a threshold* ϵ, *a range join finds all location pairs in* O *with their distances no larger than* ϵ, *i.e.,* $RJ(O, \epsilon) = \{(u, v) | d(u, v) \leq \epsilon, u \in O, v \in O\}$.

For example, in Fig. 2, given a set of locations at time 1 (i.e., $O = \{o_1, o_2, ..., o_8\}$) and a threshold ϵ, $RJ(O, \epsilon) = \{(o_1, o_2), (o_3, o_4), (o_5, o_6), (o_6, o_7)\}$.

4. OVERVIEW OF CO-MOVEMENT PATTERN DETECTION

In this section, we present an overview of co-movement pattern detection over streaming trajectories. Fig. 3 shows the framework and the processing flow, termed as Indexed Clustering and Pattern Enumeration (ICPE). ICPE takes streaming trajectories as input.

First, it uses window operations to transform the streaming trajectories into snapshots, as discussed in Section 3.1. For example, in Fig. 3, the streaming trajectories are transformed into snapshots, i.e., a snapshot at time 1, a snapshot at time 2, and so on.

Second, ICPE performs index-based clustering based on RangeJoin and DBSCAN, to be detailed in Section 5. When a new snapshot arrives, ICPE detects the clusters of the trajectories that move