Supplementary

1 The pseudo codes

Algorithm 1: CreateFollowingNetwork

```
input  : A time series set Q = {Q_1, ..., Q_n} and a threshold σ
output : A n × n adjacency matrix E
E_{i,j} = 0, ∀i, j ∈ {1, ..., n};
for i ← 1 to n do
    for j ← i + 1 to n do
        U ← Q_i and W ← Q_j;
        P_{U,W} ← DTW(U, W) ;
        if s(P_{U,W}) ≥ σ then
            E_{j,i} = |s(P_{U,W})|;
        else if s(P_{U,W}) ≤ −σ then
            E_{i,j} = |s(P_{U,W})|;
        else
            E_{i,j} = 0;
        end
    end
end
```

Algorithm 2: CreateDyFollowingNetwork

```
input  : A time series set U, ω, δ, and σ
output : A n × n × t* adjacency matrix E*.
K ← (t* − ω)/δ ;
for i ← 1 to K do
    /* current time interval */
    w(i) = [(i − 1) × δ, (i − 1) × δ + ω] ;
    /* SubTimeSeries(U, w(i)) returns all sub time series in U within the interval w(i) */
    Q ←SubTimeSeries(U, w(i));
    E ←CreateFollowingNetwork(Q, σ) ;
    /* Set all edges within the time interval [(i − 1) × δ, i × δ] to be similar */
    E*_{t∈[(i−1)×δ, i×δ]} ← E ;
end
Q ←SubTimeSeries(U, [K × δ, t*]);
E ←CreateFollowingNetwork(Q, σ) ;
E*_{t∈[K×δ, t*]} ← E ;
```

2 Coordination measure

Given a set of time series $\mathcal{U}$, a set of clusters $\mathcal{C} = \{H_1, \ldots, H_n\}$ such that $\bigcup_k H_k = \mathcal{U}$, we define a cluster membership indicator $\delta_{i,j} = 1$ if time series U_i and U_j belong to the similar cluster, otherwise it is zero. **The average coordination measure** Ψ of a set of clusters $\mathcal{C}$ is defined as follows:

$$\Psi(\mathcal{C}) = \frac{\sum\limits_{U_i,U_j \in \mathcal{U}, U_i \neq U_j} \mathrm{sim}_{max}(U_i, U_j)\delta_{i,j}}{\sum\limits_{U_i,U_j \in \mathcal{U}, U_i \neq U_j} \delta_{i,j}}. \tag{2.1}$$

Note that $\Psi \in [0, 1]$. If Ψ is close to one, then all the time series within the same cluster are highly similar, with some time delay. This implies there exists a high degree of coordination within each clusters in this case. On the contrary, $\Psi \approx 0$ implies no coordination, on average.

Theorem 2.1. *Given a set of time series $\mathcal{U}$ containing a set of σ-faction $\mathcal{F} = \{F_1, \ldots, F_n\}$ where $\bigcup_{F_i \in \mathcal{F}} F_i = \mathcal{U}$, then, for all possible sets of clusters, $\mathcal{F}$ maximizes the average coordination measure Ψ.*

Proof. Reminding that for all pairs U_i, U_j within any similar faction F, $\mathrm{sim}_{max}(U_i, U_j) \geq \sigma$. Hence, $\Psi(\mathcal{F}) \geq \sigma$.

Case 1: let $H, J \in \mathcal{F}$, if we modify $\mathcal{F}$ by exchanging any time series $U_H \in H$ with $U_J \in J$ and call it $\mathcal{C}$, then we have:

$$\Psi(\mathcal{F}) - \Psi(\mathcal{C}) = \frac{S + S'}{\sum_{U_i,U_j \in \mathcal{U}, U_i \neq U_j} \delta_{i,j}}.$$

$$S = \sum_{U_i \in H \setminus \{U_H\}} \Big(\mathrm{sim}_{max}(U_i, U_H) - \mathrm{sim}_{max}(U_i, U_J)\Big)$$

$$S' = \sum_{U_i \in J \setminus \{U_J\}} \Big(\mathrm{sim}_{max}(U_i, U_J) - \mathrm{sim}_{max}(U_i, U_H)\Big)$$

For any $U_i \in H$, $\mathrm{sim}_{max}(U_i, U_H) \geq \sigma$ since $U_H \in H$.

Algorithm 3: FindFactionsAndInitiators

```
input  : An adjacency matrix E* of dynamic
         network
output : A time series of faction sets F*, and
         a time series of initiator sets L*
for i ← 1 to t* do
    /* Get a matrix at time t = i          */
    E ← E*_{t=i} ;
    /* FindInitiators(E) returns all
       nodes which have zero outgoing
       degree                              */
    L ←FindInitiators(E) ;
    F = ∅ ;
    for l ∈ L do
        /* FindReachNodeFrom(E, l)
           returns all nodes which have
           any directed path to l          */
        F_l ←FindReachNodeFrom(E, l) ;
        F = F ∪ {F_l}
    end
    F*_{t=i} = F and L*_{t=i} = L
end
```

In contrast, because $U_J \notin H$, then $\text{sim}_{max}(U_i, U_J) < \sigma$, which implies $S > 0$. $S' > 0$ for a similar reason. Therefore, $\Psi(\mathcal{F}) - \Psi(\mathcal{C}) > 0$.

Case 2: if we create $\mathcal{C}$ from $\mathcal{F}$ by spiting a cluster $H \in \mathcal{F}$ to be $H_1 \subset H$ and $H_2 = H \setminus H_1$, then we have:

$$\Psi(\mathcal{F}) - \Psi(\mathcal{C}) = \frac{\sum_{U_i \in H_1, U_j \in H_2} \text{sim}_{max}(U_i, U_j)}{|H_1||H_2|} \geq \sigma.$$

Case 3: we create $\mathcal{C}$ from $\mathcal{F}$ by merging any cluster $H \in \mathcal{F}$ with any $J \in \mathcal{F}$ such that $H \neq J$ to be H'. So, let

$$\Psi(\mathcal{F}) = \frac{X_{\mathcal{F}}}{S_{\mathcal{F}}} \geq \sigma,$$

then

$$\Psi(\mathcal{C}) = \frac{X_{\mathcal{F}} + \sum_{U_i \in H, U_j \in J} \text{sim}_{max}(U_i, U_j)}{S_{\mathcal{F}} + |H||J|}.$$

By merging H and J, we introduce pairs of time series across H and J to Equation 2.1 such that $\text{sim}_{max}(U_i, U_j) < \sigma$ since these pairs are not belong to the same faction. These pairs decrease the average of $X_{\mathcal{F}}$, which implies $\Psi(\mathcal{F}) > \Psi(\mathcal{C})$.

Since we shown that no matter how we edit $\mathcal{F}$, the average coordination measure Ψ cannot increase, therefore, $\mathcal{F}$ maximizes the average coordination measure.

3 Time complexity

Let n be a number of time series, ω be a time window, δ be a shifting factor (we use $\delta = 0.1\omega$), and t^* be a total length of time series. By deploying DTW Sakoe Chiba band technique [2] setting δ as a band limitation, the time complexity of computing a following network is $\mathcal{O}(n^2 \times \omega \times \delta)$. Since we need warping paths, not a distance, the upper/lower bounds tricks which are used to speed up DTW found in the time series literature cannot be applied here. The number of following networks we need to compute is $\frac{t^*}{\delta}$. In total, the time complexity of our framework is $\mathcal{O}(n^2 \times \omega \times t^*)$. Additionally, we might explore k candidates of ω in order to find the optimal ω. Since k is a constant, the asymptotic time complexity of our framework also remains the same. This expensive cost is unavoidable and it makes our framework hard to be a scalable framework.

4 Comparison method

From the best of our knowledge, there is no existing methods dealing with the FACTION INITIATOR INFERENCE PROBLEM. The closest method that we can compare against is the flock model [1,3]. We compared our framework against Volatility Collective Behaviors Model [3], which has an assumption that all members in a similar group move toward the similar direction on a non-linear trajectory. Hence, we modified the FLOCK framework to make it work in our setting as a baseline of comparison. In stead of using DTW to build following networks, we created FLOCK following networks. According to the work in [1], the time series A follows the time series B at any time step t if the angle of their direction vector from time $t-1$ to t is less than the threshold β as well as B is in the front of A with respect to B's direction, as well as A and B must have their distance less than the threshold γ. The FLOCK following networks are built for all time steps. The rest of FLOCK framework is similar to our framework. We set the FLOCK parameters such that it can perform the best.

References

[1] M. Andersson, J. Gudmundsson, P. Laube, and T. Wolle. Reporting leaders and followers among trajectories of moving point objects. *GeoInformatica*, 12(4):497–528, 2008.

[2] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978.

[3] T. E. Will. Flock leadership: Understanding and influencing emergent collective behavior. *The Leadership Quarterly*, 27(2):261–279, 2016.