Article

FINNCH: Cooperative Pursuit Navigation for a Pursuer Team to Capture a Single Evader in Urban Environments

Xiayin Lou, Min Sun *, Hanjun Yang and Shihao Yang

Institute of Remote Sensing and Geographic Information Systems, Peking University, 5 Summer Palace Road, Beijing 100871, China; xiayin_lou@stu.pku.edu.cn (X.L.); hanjuny@stu.pku.edu.cn (H.Y.); yangshihao@stu.pku.edu.cn (S.Y.)
* Correspondence: sunmin@pku.edu.cn

Abstract: The development of a cooperative pursuit strategy for capturing escaping criminals or dangerous animals in urban public safety emergencies is becoming increasingly in demand. An ideal strategy should consider both the encirclement needed to prevent criminals from evading and the distance that pursuers need to move. This article proposes a fine-grained navigation network-based cooperative hunting (FINNCH) method. A fine-grained navigation network is created to provide detailed information about the traversability in urban areas. Three interaction rules inspired by biological behaviors in nature are introduced to achieve dynamic cooperation between pursuers. An heuristic search strategy is used to guide the pursuers toward potentially good directions, which consequently reduces the search effort. Two spatial constraints, namely, the direction centrality constraint (DCC) and encirclement distance constraint (EDC), are then constructed to evenly distribute the pursuers around the evader. Several experiments are conducted to evaluate the effectiveness and efficiency of the proposed method. The results show that FINNCH can provide navigation for multiple pursuers in complex urban environments comprised of roads, meadows, trees, water, and buildings. These findings point to the promising future of FINNCH for practical applications.

Keywords: public security emergency; cooperative pursuit; fine-grained navigation

1. Introduction

Public security emergencies, such as the escape of a criminal suspect or a dangerous animal, occasionally occur in urban areas. To respond to these emergencies, organizing a team of pursuers (e.g., police officers, firefighters, or robots) to collaboratively encircle and capture these evaders is necessary. With the development of various visual sensors, from network cameras installed in shopping malls and residential communities to unmanned aerial vehicles (UAVs) used to actively acquire aboveground topography information [1], these emergencies can be handled more efficiently. However, coordinating pursuers to capture an evader in an urban environment remains a challenge for two main reasons. First, pursuers usually do not have good knowledge of the surrounding environments near emergencies, so proper guides are needed. Existing navigation software is based on road networks, so may fail to find routes in regions without roads, such as squares, parks, residential communities, etc. To tackle this problem, the current research team proposed the TraV method [2], which combines remote sensing images with geographic information to generate a fine-grained navigation network, thus enabling the navigation software to provide feasible routes to reach any location in an urban area. Second, there are few cooperative pursuit strategies designed for urban areas. Related algorithms focus on planning paths for a team of robots in an ideal bounded area. For example, some algorithms [3–5] assume that robots are in a space without any obstacles, while others [6–9] merely set some simple obstacles that are far from identical to real-world situations. They
rarely consider real geographic environments or explore how to serve human beings and robots in emergency events.

Inspired by previous research [4,10], a decentralized method is developed to address these challenges: fine-grained navigation network-based cooperative hunting (FINNCH). It is developed on the basis of a fine-grained navigation network, which describes the traversability in detail. The core idea of the FINNCH method is fourfold:

1. Obstacles in an urban environment (e.g., buildings and water bodies) are much more complex than those in an experimental environment, and they largely narrow the traversable spaces in an urban area. Based on three dynamic interaction rules for chasing-team members proposed by Fu et al. [4] (for more details, please see Section 4), a warmup strategy is added, in which pursuers maintain a relatively small distance between each other at the early stage so that they can more easily find reasonable routes in narrow traversable spaces.

2. In actual environments, the traversability of different areas varies due to the uneven vegetation distribution and ground roughness. Therefore, taking the local traversability cost into account will be helpful for finding a better route. In this article, an heuristic search strategy is introduced that considers the local traversability and uses the cost to the evader as prior information to accelerate the search for cooperative pursuit routes.

3. During the cooperative pursuit, performing a proper encirclement is the key to a successful capture. To guide pursuers to distribute evenly in the space around the evader, two spatial constraints, namely, the direction centrality constraint (DCC) and the encirclement distance constraint (EDC), are proposed to better encircle the evader.

4. By combining dynamic interactions, an heuristic search strategy, and spatial constraints, a decentralized action strategy is constructed with artificial potential functions. This strategy can help the pursuer team properly surround the evader while guaranteeing that the distance traversed by each pursuer remains relatively short.

The remainder of this article is structured as follows. Section 2 provides a formal description of the problem. Next, a review of related research on cooperative pursuit is conducted in Section 3. Section 4 describes the core idea of FINNCH and an independent hunting strategy for each pursuer based on a fine-grained navigation network. In Section 5, taking a residential community in a city in northwestern China as an example, experiments are conducted to validate the performance of the proposed method in offering cooperative pursuit navigation for multiple pursuers. In Section 6, the limitations of the proposed method and future work are discussed. Finally, in Section 7, the conclusions are drawn.

2. Problem Definition

Cooperative pursuit navigation for multiple pursuers capturing a single evader in urban environments can be formulated as follows. Within a square-shaped area of a city, there exist $k$ pursuers capturing a single evader. Given the existing telecommunication and UAV remote sensing technology, it is assumed that any pursuer can obtain the locations of other pursuers and the evader at any moment, and the evader can detect the pursuers when they are within the evader’s sensitive radius. The capture condition is that $k$ pursuers arrive simultaneously at a distance no greater than $r_c$ from the evader. Suppose that the area for cooperative pursuit is a two-dimensional planar space represented as $D$. At step $t$, the location of the evader is $l^e(t)$ ($e$ refers to evader), the location of the $i$th pursuer is $l^p_i(t)$ ($p$ refers to pursuer), and the threshold distance for successfully capturing the evader is $r_c$. The capture condition is then defined as:

$$\max_i |l^e(t_c) - l^p_i(t_c)| \leq r_c, i \in \{1, 2, \ldots, k\}, l^e(t), l^p_i(t) \in D$$ (1)
where \( t_c \) is the capture time. At the initial step \( t_s \), the locations of the pursuers and the evader should satisfy the following condition:

\[
d_{\text{min}}(t_s) > r_c
\]

where \( d_{\text{min}}(t) \) refers to the minimum distance between the pursuer team and evader, which is defined as:

\[
d_{\text{min}}(t) = \min_{i=1\ldots k} \{|l'_i(t) - l^e(t)|\}
\]

To achieve cooperation among a pursuer team, the action of any pursuer is expected to be determined according to the locations of other pursuers, as well as local traversability. Because of the difference in spatial contexts, the hunting strategy for each pursuer varies. At step \( t \), the process by which the \( i \)th pursuer decides the location at the next step can be expressed as:

\[
l'_i(t + 1) = \mu_i(l^e(t), l^p_1(t), l^p_2(t), \ldots, l^p_k(t), \mathcal{G}), i \in \{1,2,\ldots,k\}
\]

where \( \mu_i \) is the independent hunting strategy for the \( i \)th pursuer, \( \mathcal{G} = (V,E,w) \) is a fine-grained navigation network that represents the ground traversability, \( V \) is a set of vertices, \( E \) is a set of edges that connect vertices, and \( w \) is the non-negative traversability costs of edges. Equation (4) is a recursive function, in which the state of each pursuer at step \( t + 1 \) is computed from the previous moment \( t \). Via iterative computation, the capture condition (Equation (1)) can be satisfied at a finite step \( t_c \). In practical applications, the locations obtained from the successive calculations of Equation (4) will form routes that can provide guidance for all pursuers.

3. Related Works

3.1. Cooperative Pursuit

Cooperative pursuit is essentially a process in which a team of pursuers works together to capture one or more evaders. Therefore, it is rather important for a pursuer team to have a feasible strategy. In some emerging fields, such as robotics and aerospace, cooperative pursuit is generally abstracted as a pursuit–evasion game (PEG) problem.

The PEG problem was first introduced by Issacs [11]. In his seminal work, the dynamics of the pursuers and the evader in a 2D plane were mathematically described by differential equations [12]. A value function was then defined to represent the capture time, which the pursuers attempt to minimize and the evader tends to maximize [13]. In this way, the PEG problem can be transformed into a so-called Hamilton–Jacobi–Issacs (HJI) partial differential equation (PDE), which can be used to yield optimal routes for the pursuers to minimize the time to capture. Solutions to the HJI PDE are generally computed either by employing the characteristic method via backward integration from a known terminal condition [14] or by using numerical approximation methods in which the continuous state space is discretized [15]. Nevertheless, regular gridding has difficulties in representing obstacles with complex geometries, such as buildings and water bodies. In addition, the computational cost will increase exponentially as the number of pursuers grows.

To overcome the computational complexity of HJI approaches, many studies have presented Voronoi-based methods [3,5,16]. As illustrated in Figure 1, this type of method uses Voronoi decomposition to divide the whole domain into multiple convex polygons with respect to agent locations. Then, the pursuers try to cooperatively minimize the area of the Voronoi cell to which the evader belongs. Each agent influences the Voronoi cell of another agent only by the shared Voronoi boundary. As a result, each agent computes its action independently given the locations of other agents. To enable the pursuers to chase the evader in an obstacle-cluttered environment, Pierson and Rus [17] developed an improved Voronoi decomposition, namely, the obstacle-aware Voronoi cell (OAVC). However, this method allows the evader to pass through the obstacles, which may create a
barricade that the pursuers fail to navigate around, and collision avoidance between pursuers is not considered. The obstacle-aware buffered Voronoi cell (OABVC) proposed by Tian et al. [8] further improved the OAVC: a security domain was added for each pursuer, which prevents the pursuers from being too close to one another. Although some researchers have explored cooperative pursuit in an obstacle-cluttered environment, the obstacles designed in these studies were relatively simple in comparison with those in the real world. Moreover, the impact of local traversability was not considered.

![Figure 1](image_url)  
**Figure 1.** The capturing process of three pursuers and a single evader E, where the Voronoi cell of E is indicated by the gray shade.

Compared with HJI approaches, Voronoi-based methods can transform the PEG from a high-dimensional PDE problem to a low-dimensional local area minimization problem [18]. Furthermore, the decentralized strategy adopted by Voronoi-based methods greatly lowers the computational consumption.

In a decentralized strategy, each pursuer has its own local knowledge of the world and can decide its future actions by independently taking into account the locations of other pursuers and the evader [19]. Hence, this strategy typically has a lower computational burden and higher flexibility. Many studies have validated the effectiveness of the decentralized strategy [20–23]. In this work, a decentralized strategy is introduced in the proposed FINNCH method, in which each pursuer moves independently while cooperating with the other pursuers.

The aforementioned studies generally presumed that all the agents are located in a bounded environment without obstacles or just with simple obstacles. In contrast to the experimental settings, pursuers in an urban environment may confront complicated geographic contexts (such as buildings, waters, forests, etc.), but a large component of urban areas (such as meadow, bushes, etc.) is actually traversable. In a public safety emergency, to quickly catch the evader, pursuers can directly traverse these traversable areas. Consequently, a fine-grained representation of urban environments is highly required to generate reasonable and low-cost routes for cooperative pursuit navigation. However, due to the complexity and uncertainty of urban areas, modeling the environments is difficult [24]. In this study, the FINNCH method is developed based on the fine-grained navigation network constructed from the TraV method.

Motivated by these observations, an urban environment-oriented, distributed method composed of dynamic interactions between pursuers, an heuristic search strategy, and spatial constraints is proposed. The proposed method is elaborated in Section 4.

### 3.2. Fine-Grained Navigation Networks

In urban areas, fine-grained navigation becomes increasingly important in the fields of precise delivery, public security, firefighting, and emergency medical care, etc. However, current navigation software, such as Google Maps, depending on road networks, often fails to offer more accurate routes or makes a detour in regions where road networks are absent on grounds without roads (GWRs), such as squares, parks and residential communities, etc. To tackle this problem, in [2], we proposed the TraV method for the construction of fine-grained navigation networks for urban environments (see Figure 2). First,
remotely sensed urban scene imagery is employed to estimate the traversability of GWRs. Then, the urban areas are spatially partitioned into meaningful atomic regions using superpixel segmentation [25]. Next, a computational model of traversability that can represent the traversability within each region and between any two adjacent regions is built to construct the navigation mesh of GWRs (see Figure 2a). Finally, a fine-grained navigation network is created through integrating existing urban road networks and a GWR navigation mesh (see Figure 2b). By building such a fine-grained navigation network, we can provide both coarse guidance at global level and fine navigation at local level.

![Figure 2](image)

**Figure 2.** (a) The TraV method for constructing a fine-grained navigation mesh in GWRs. (b) An overview of a fine-grained navigation network. The upper layer is the GWR navigation mesh, and the lower layer is urban road networks. The two layers are connected with switching nodes.

4. Method

An overview of the proposed method is provided in Figure 3. Based on the fine-grained navigation network (Figure 3a), FINNCH retrieves all adjacent vertices as candidates according to the current location of the i-th pursuer (Figure 3b). It then uses (1) dynamic interactions (Figure 3c) to make the pursuers collaboratively chase the evader, (2) geographic heuristics (Figure 3d) considering both the cumulative cost from the start and the predicted cost to the goal to speed up the search, and (3) spatial constraints (Figure 3e) to improve the encirclement to help the i-th pursuer decide its next location and update its route (Figure 3f).

In this section, three dynamic interaction rules between pursuers are first described: the target pursuit force $v_i^{par}$, the distance-keeping force $v_i^{keep}$, and the dynamic cooperation force $v_i^{coop}$. The calculation of the two spatial constraints DCC and EDC is then explained. Next, the dedicated hunting strategy for each pursuer is provided. Finally, an iterative procedure is developed to solve the optimal routes for a pursuer team to cooperatively encircle and capture a single evader. The proposed FINNCH method was partially inspired by the bioinspired cooperative control method developed by Fu et al. [4]. Nevertheless, as described in Section 1, existing methods are mainly designed for robots in relatively simple environments, whereas the focus of this work is on providing both humans and robots with a cooperative strategy to capture an evader in complex urban areas.
Figure 3. An overview of the proposed method. The areas in red are obstacles. FINNCH retrieves adjacent vertices on the fine-grained navigation network, then uses a hunting strategy combining three components (dynamic interactions, geographic heuristic, and spatial constraints) to decide the next location.

4.1. Target Pursuit Force

During the pursuit, since surveillance devices such as UAV can easily track the location of the evader from the air, we assume a pursuer always knows the location of the evader so that it can approach the evader quickly. Here, a simple model is used to represent the attractiveness of the evader to the pursuer, which is expressed as:

\[
v_i^{pur} = \frac{d_{e,i}}{|d_{e,i}|} = \frac{l^e - l_i^P}{|l^e - l_i^P|}
\]  

(5)

where \(v_i^{pur}\) is the target pursuit force, \(d_{e,i} = l^e - l_i^P\) is the displacement vector between the \(i^{th}\) pursuer and evader, \(l^e\) is the location of the evader, and \(l_i^P\) is the location of the \(i^{th}\) pursuer.

4.2. Distance-Keeping Force

To better encircle the evader, a certain distance should be maintained between the pursuers to avoid collision. Thus, there is a “short-range repulsive force” [10] to prevent pursuers from being too close, which can be calculated as:

\[
v_i^{keep} = U \left[ \sum_{j \neq i} d_{i,j} \cdot H(r_{keep} - |d_{i,j}|) \right]
\]  

(6)

where \(v_i^{keep}\) is the distance-keeping force, \(d_{i,j} = l_i^P - l_j^P\) is the displacement vector between the \(i^{th}\) and \(j^{th}\) pursuers, \(r_{keep}\) is the minimum distance between pursuers, and \(H(\cdots)\) is the Heaviside step function. When the distance between pursuers \(|d_{i,j}|\) is greater than \(r_{keep}\), the repulsive force is equal to zero. When this distance \(|d_{i,j}|\) is less than or equal to \(r_{keep}\), there is a repulsive force to widen the distance between pursuers. \(U(\cdots)\) is a function that converts a vector into a dimensionless one. It is noted that the function of \(U(\cdots)\) remains the same in the following expressions.
4.3. Dynamic Cooperation Force

When approaching the evader, the pursuers should gradually shrink their encirclement, making it difficult for the evader to escape. Specifically, when the pursuer team is far from the evader, they do not get too close to one another. When this team is near the evader, however, they actively shorten the distances between each other to capture the evader. To represent the dynamic cooperation between pursuers, the “dynamic long-distance repulsive force” is introduced, which refers to the work of Fu et al. [4] (see Figure 4). Each pursuer is subject to the repulsive forces from other pursuers, of which the strength is related to the minimum distance between the pursuers and the evader. For the $i$th pursuer, its dynamic cooperation force $v_i^{coop}$ is defined as:

$$v_i^{coop} = \eta(d_{\text{min}}(t)) \times U \left[ \sum_{j\neq i} d_{i,j} \left( \frac{1}{|d_{i,j}|} - \frac{1}{|d_{i,j}|^2} \right) \right]$$

(7)

where $\eta(d)$ is a decay function in which the strength of the repulsive force weakens as the distance between the pursuers and the evader shortens. $d_{\text{min}}(t)$ is the minimum distance between the pursuer team and the evader, and $d_{i,j} = l_i^p - l_j^p$ is the displacement vector between the $i$th and $j$th pursuers. The definition of $\eta(d)$ is given as:

$$\eta(d) = \begin{cases} S_{\text{max}}, & d \geq r_{\text{thres}} \\ \frac{S_{\text{max}}}{r_{\text{thres}}}, & d < r_{\text{thres}} \end{cases}$$

(8)

where $S_{\text{max}}$ is the maximum strength of the repulsive force and $r_{\text{thres}}$ is the threshold distance at which the strength of the repulsive force starts to reduce. When $d < r_{\text{thres}}$, the strength decays as the distance $d$ decreases, and when $d \geq r_{\text{thres}}$, the strength plateaus at the maximum. It is worth noting that a linear decay function is used to improve the computational efficiency compared to the nonlinear decay function used in a previous study [4].

![Figure 4. The dynamic cooperation force among a team of pursuers.](image)

4.4. Warmup Strategy

In theory, a pursuer team gradually spreads out to perform an encirclement. However, if the starting points of the pursuers are too close, two repulsive forces (distance-keeping force and dynamic cooperation force) can be very strong, so they may scatter too early. Moreover, the presence of a large number of obstacles in urban areas, such as buildings, relatively narrows the traversable spaces available to the pursuers. These two problems will make it hard to find routes for multiple pursuers in a complex urban environment. In the face of these two problems, a warmup strategy is proposed, the main idea of
which is to let the strength of the two repulsive forces increase from a small initial value and reach the maximum value after \(T_w\) steps. In this way, we can simulate the cooperative pursuit in the realistic situation. The warmup strategy can be represented as:

\[
\lambda(t, T_w) = \begin{cases} 
\lambda_0 + (1 - \lambda_0) \cdot \frac{t}{T_w}, & t \leq T_w, \lambda_0 \in [0, 1] \\
1.0, & t > T_w 
\end{cases}
\]

(9)

where \(\lambda_0\) is the initial strength of the repulsion and \(T_w\) is the warmup step at which the repulsion reaches its maximum. When step \(t \leq T_w\), the strength increases gradually as the pursuit proceeds, and when \(t > T_w\), the strength always stays at 1.0.

4.5. Direction Centrality Constraint (DCC)

During the pursuit, when the evader observes the pursuers, it will adopt an escape strategy. To reduce the probability of the successful escape of the evader, the pursuers should be evenly distributed in all directions around the evader. As a result, the DCC is introduced, which measures the difference in the directional distribution [26] and uses it as a penalty term to constrain the routes of the pursuers. The DCC is defined as follows:

\[
\rho(t) = \frac{k}{4(k - 1)\pi^2} \sum_{i=1}^{k} \left(\alpha_i - \frac{2\pi}{k}\right)^2
\]

(10)

As shown in Figure 5a, centered on the evader, \(k\) pursuers form \(k\) angles \(\alpha_1, \alpha_2, \ldots, \alpha_k\). In a 2D space, the condition \(\sum_{i=1}^{k} \alpha_i = 2\pi\) holds. When these angles are equal, the DCC reaches its minimum value 0. When one angle is \(2\pi\) while others are 0, the DCC value is maximized as 1.

![Figure 5](image)

4.6. Encirclement Distance Constraint (EDC)

Although the DCC can guide the pursuers to be evenly distributed in every direction, there will be a loophole in the encirclement if the distance between each pursuer and the evader differs considerably. To improve the encirclement, the EDC is further introduced, which measures the distribution of the distances between the pursuers and the evader (see Figure 5b). Let the distances between all pursuers and the evader be \(|d_{e,1}|, |d_{e,2}|, \ldots, |d_{e,k}|\), where \(d_{e,i}\) is the displacement vector between the \(i\)th pursuer and the evader. These distances can then be standardized as follows.

\[
\beta_i = \frac{|d_{e,i}| - \min_j{|d_{e,j}|}}{\max_j{|d_{e,j}|} - \min_j{|d_{e,j}|}}, \quad i \in \{1, 2, \ldots, k\}
\]

(11)
Here, max–min standardization is adopted. The EDC is defined as the variance of these standardized distances, which can be computed as:

\[
\psi(t) = \frac{1}{k} \sum_{i=1}^{k} (\beta_i - \mu)^2, \quad \mu = \frac{1}{k} \sum_{i=1}^{k} \beta_i
\]

where \( \mu \) is the mean of these standardized distances.

4.7. Independent Hunting Strategy for Each Pursuer

At any moment, each pursuer decides its next move based on its spatial relationships with respect to other pursuers and the evader. In a fine-grained navigation network \( \mathcal{G} = (V, E, w) \), at step \( t \), the \( i \)th pursuer is located at \( l_t^i \), and it has \( M \) candidate vertices \( \{l_j^i(t), j \in \{1,2,\ldots,M\}\} \) to move on at the next step. The corresponding costs for all candidate vertices \( \{\theta_j(t), j \in \{1,2,\ldots,M\}\} \) will then be calculated. Finally, the \( i \)th pursuer will choose the vertices with the lowest cost as the best location at step \( t + 1 \). The procedures for determining the best location are as follows.

1. At step \( t \), the \( i \)th pursuer is subjected to three different forces: the target pursuit force \( v_i^{\text{pur}}(t) \), the distance-keeping force \( v_i^{\text{keep}}(t) \), and the dynamic cooperation force \( v_i^{\text{coop}}(t) \). These three forces act together to form a combined force, which is given by:

\[
v_i^{\text{comp}}(t) = v_i^{\text{pur}}(t) + \lambda^{\text{keep}}(t) \cdot v_i^{\text{keep}}(t) + \lambda^{\text{coop}}(t) \cdot v_i^{\text{coop}}(t)
\]

where \( \lambda^{\text{keep}} \) and \( T_w^{\text{keep}} \) are the initial strength and warmup steps of the distance-keeping force, respectively, and \( \lambda^{\text{coop}} \) and \( T_w^{\text{coop}} \) are the initial strength and warmup steps of the dynamic cooperation force, respectively. Regarding the \( m \)th candidate vertex \( l_m^n \), the displacement vector between it and the \( i \)th pursuer is \( v_i^{\text{tri}} = l_m^n - l_t^i \). The normalized cosine similarity (NCS) \( \cos S_{im}(t) \) between \( v_i^{\text{tri}} \) and \( v_i^{\text{comp}}(t) \) can then be calculated:

\[
\cos S_{im}(t) = 1 - \frac{\phi(v_i^{\text{tri}}, v_i^{\text{comp}}(t)) + 1}{2}
\]

where \( \phi(v_1, v_2) \) is a function that calculates the cosine value of the angle between two vectors.

2. The normalized heuristic value (NHV) \( H_{im}(t) \) is computed for the \( m \)th candidate vertex \( l_m^n \) at step \( t \). The NHV estimates the Euclidean distance between the candidate vertex and the location of the evader \( l_{\text{goal}} \) to provide directional guidance. It can be computed as:

\[
H_{im}(t) = \frac{|l_m^n - l_{\text{goal}}|}{SL}
\]

where \( SL \) is the side length of the pursuit area.

3. The normalized accumulated cost (NCS) \( G_{im}(t) \) is computed for the \( m \)th candidate vertex \( l_m^n \) at step \( t \). The NCS represents the exact cost of the route of the \( i \)th pursuer's route from the starting point \( l_i^{\text{start}} \) to the candidate vertex. The NCS value at step \( t \) can be calculated from that at step \( t - 1 \):

\[
G_{im}(t) = G_{im}(t - 1) + \frac{\text{cost}_m}{SL}
\]

where \( \text{cost}_m \) is the traversal cost from the location \( l_t^i \) of the \( i \)th pursuer to the candidate vertex \( l_m^n \).

4. The DCC \( \rho_{im}(t) \) and EDC \( \psi_{im}(t) \) are computed for the \( m \)th candidate vertex \( l_m^n \) at step \( t \) according to Sections 4.5 and 4.6, respectively.

5. For the \( m \)th candidate vertex, its cost \( \theta_{im}(t) \) is the weighted sum of the previously mentioned five components. \( \theta_{im}(t) \) can be computed as:
\[
\theta_{i,m}(t) = w_1 \cdot \cos \theta_{i,m}(t) + w_2 \cdot H_{i,m}(t) + w_3 \cdot G_{i,m}(t) + w_4 \cdot \rho_{i,m}(t) + w_5 \cdot \psi_{i,m}(t)
\]  
(17)

where \( W = \{w_i, i \in \{1,2,3,4,5\} \} \) are the weights for these components, and they satisfy the following condition.

\[
\sum_{i=1}^{5} w_i = 1
\]  
(18)

6. According to the preceding steps, the costs for all candidate vertices are computed, and the \( \hat{m} \)th candidate vertex \( l^c_{\hat{m}} \) with the lowest cost is selected, as presented by the following equation.

\[
\hat{m} = \arg \min_{j=1,\cdots,k} \theta_{i,j}(t)
\]  
(19)

4.8. Escape Strategy for the Evader

As the pursuers approach, the evader will detect them and try to shake off the chase. Hence, we design a simple escape strategy for the evader. Generally, the evader has a sensitive radius \( r_{\text{sens}} \) and acts differently according to different levels of risk posed by the pursuers. When the pursuers are far away from the evader, that is, \( d_{\text{min}}(t) > r_{\text{sens}} \) the evader will choose the best exit and find the shortest route to it. When there is a pursuer within the evader’s sensitive radius, namely, \( d_{\text{min}}(t) \leq r_{\text{sens}} \), the evader will prioritize shaking off the encirclement. During the process of escaping encirclement, the evader will adopt two actions. The first is exit pursuit force \( v^e_{\text{pur}} \), which leads the evader to head for the exit. The second is direct escape force \( v^\text{direct} \) (see Figure 6), in which all the pursuers exert repulsive forces on the evader. The direct escape force is formed as:

\[
v^\text{direct} = U \left( \sum_{i=1}^{k} \frac{r_i}{|v_i|^2} \right) = \left[ \sum_{i=1}^{k} \left( \frac{l^e - l^p_i}{|l^e - l^p_i|} \right) \right]
\]  
(20)

in which \( r_i = l_e - l^p_i \) is the repulsive force posed by the \( i \)th pursuer, and \( r_i \) is divided by the square of its length. \( U(\cdots) \) is a function that unitizes a vector. This represents the fact that nearer pursuers pose a higher risk to the evader. During the pursuit, the evader will switch between these two actions in order to get rid of the pursuers.

![Figure 6. Direct escape force to the evader from the pursuers.](image)

4.9. Solution Algorithm

Section 4.7 described in detail the strategy of the \( i \)th pursuer in choosing its next action at step \( t \). To clearly illustrate how to compute the optimal routes for multiagent cooperative pursuit, the pseudocode is provided in Algorithm 1, in which \( \text{len}() \) denotes the function that calculates the number of elements in a set. The time complexity of this algorithm is \( O(n^3) \). In reality, because the area of an urban block is limited, the upper bound of computational cost and time is also limited.
At the initialization stage of the algorithm, a fine-grained navigation network $G = (V, E, w)$ is generated using the TraV method, step $t$ is set to 0, and the route for each pursuer is set to empty. After the pursuit begins, at every step, each pursuer acquires its neighboring vertices, which serve as candidates for the next location. Then, the best candidate location is selected according to the procedures described in Section 4.7. Next, the route of each pursuer is updated by adding this selected location. When all pursuers are within the threshold radius $r_C$ with the evader as the center point, the computation of routes for cooperative pursuit is terminated.

**Algorithm 1:** The FINNCH method for cooperative pursuit with a team of pursuers

Inputs: A team of pursuers $P = \{P_i, \forall i\}$; an evader $E$; threshold distance to end hunting $r_C$

Outputs: Routes for pursuers $R = \{R_i, \forall i\}$

Initialization: Generate a fine-grained navigation network $G = (V, E, w)$; set $t = 0$ and empty routes $R = \{R_i, \forall i\}$ for pursuers.

while CheckCapture($P, E, r_C$, $t$) $\neq$ true do  
  check if all pursuers are within the threshold distance $r_C$
  for $i = 1$ to $len(P)$ do
  Acquire all candidates $\{l_i^j, \forall j\}$ adjacent to the current location of the $i$th pursuer $P_i$ in $G$
  Compute corresponding scores $\{\theta_{i,j}(t), \forall j\}$ for all candidates according to Equation (17)
  Select the best candidate location $l_{i,m}^c$ that has the lowest score value.
  Update the location of the $i$th pursuer $P_i$ to the selected location $l_{i,m}^c$.
  Add the selected location $l_{i,m}^c$ to the route $R_i$ of the $i$th pursuer $P_i$.
  end for
  Set $t = t + 1$
end while

5. Experiments and Results

5.1. Experimental Area and Dataset

The experimental area is a square region with a side length of 617.7 m and an area of about 38 ha, which is located in a city in northwestern China (see Figure 7). The main data used are comprised of two parts: (1) a UAV remote sensing image of the experimental area, and (2) vector datasets for the corresponding area, from road networks to buildings. The UAV remote sensing image was captured in April 2021 with a spatial resolution of 0.2 m. The land cover types in this image include buildings, roads, vegetation, impervious ground, and water surfaces. In general, the vegetation in urban areas includes shrubs, trees, and lawns, but in our experimental area (see Figure 7b), the type of vegetation is mainly trees. All the vector datasets are in the shapefile format.

![Figure 7](image-url)  
Figure 7. (a) The experimental area location, (b) the corresponding remote sensing image, and (c) the vector datasets. The remote sensing image was captured by a UAV in April 2021.
As shown in Figure 8, based on the UAV remote sensing image and vector datasets of the experimental area, a fine-grained navigation network was constructed with the improved TraV method. Generally, geographic objects like vegetation and water surfaces transform with different seasons, whereas buildings and roads do not change much. Considering that the update of vector datasets is relatively slow and some buildings and roads may have changes, remote sensing images play a dominant role in the improved TraV method, in which remote sensing images are used to supplement and correct the geographic information. Hence, the newest remote sensing images can be obtained by UAVs to continuously update the real traversability of the experimental area. In this way, we can always compute reasonable routes for pursuers.

![Image](image_url)

**Figure 8.** The constructed fine-grained navigation network of the experimental area. Colored areas are obstacles such as buildings.

### 5.2. Implementation

All the experiments were performed on a MacBook Pro laptop with an Intel® Core™ i7-9750H processor clocked at 2.6 GHz with 32 GB, 2667 MHz DDR4 RAM. The proposed FINNCH algorithm was implemented in C++ and Python using the Computational Geometry Algorithms Library (CGAL) for various vector operations, the Boost Graph Library (BGL) for the manipulation of the fine-grained navigation network, and OpenCV for visualization.

The parameters of the TraV algorithm were set to fixed values because the surrounding environments generally remain unchanged during pursuit. Therefore, only the parameters of the proposed FINNCH algorithm are discussed. The parameters were determined based on the experience and the realistic conditions. The descriptions and corresponding values of related parameters used in the experiments are reported in Table 1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_{\text{keep}}$</td>
<td>10 m</td>
<td>Minimum distance between pursuers</td>
</tr>
<tr>
<td>$r_c$</td>
<td>15–18 m</td>
<td>Threshold distance for successfully capturing the evader</td>
</tr>
<tr>
<td>$r_{\text{thres}}$</td>
<td>20 m</td>
<td>Threshold distance at which the repulsive force begins to decrease</td>
</tr>
<tr>
<td>$S_{\text{max}}$</td>
<td>0.7</td>
<td>Maximum strength of the repulsive force</td>
</tr>
<tr>
<td>$w_1$</td>
<td>0.2</td>
<td>Weight for the normalized cosine similarity $C_oS_{1m}(t)$</td>
</tr>
<tr>
<td>$w_2$</td>
<td>0.4</td>
<td>Weight for the normalized heuristic value $H_{lm}(t)$</td>
</tr>
<tr>
<td>$w_3$</td>
<td>0.2</td>
<td>Weight for the normalized accumulated cost $G_{lm}(t)$</td>
</tr>
<tr>
<td>$w_4$</td>
<td>0.1</td>
<td>Weight for the direction centrality constraint $\rho_{l,m}(t)$</td>
</tr>
<tr>
<td>$w_5$</td>
<td>0.1</td>
<td>Weight for the encirclement distance constraint $\psi_{l,m}(t)$</td>
</tr>
<tr>
<td>$\lambda_{\text{keep}}^0$</td>
<td>0.5</td>
<td>Initial strength of the distance-keeping force</td>
</tr>
<tr>
<td>$r_{\text{keep}}^w$</td>
<td>50</td>
<td>Maximum warmup steps for the distance-keeping force</td>
</tr>
<tr>
<td>$\lambda_{\text{coop}}^0$</td>
<td>0.8</td>
<td>Initial strength of the dynamic cooperation force</td>
</tr>
<tr>
<td>$r_{\text{coop}}^w$</td>
<td>100</td>
<td>Maximum warmup steps for the dynamic cooperation force</td>
</tr>
<tr>
<td>$r_{\text{sens}}$</td>
<td>60 m</td>
<td>Sensitive radius of the evader</td>
</tr>
</tbody>
</table>

5.3. Three Pursuers versus One Mobile Evader

The FINNCH method was evaluated in terms of the cooperative pursuit of a mobile evader (ME) in the real world. In this set of experiments, starting positions of the evader and the pursuers were defined, and the initial distance between the pursuer team and the evader in each experiment exceeded 120 m. Figure 9 illustrates the results under four experimental conditions, from which the following can be seen.

1. It was much more difficult to keep the pursuers evenly distributed around the ME compared with the stationary evader. However, the proposed FINNCH method was still able to secure the formation of a fair encirclement. As shown in Figure 9a,c,d, though there were many obstacles nearby, the encirclement formed by the pursuer team left little space for the evader to escape.

2. The FINNCH method can reduce the distance between the pursuers to pass through areas where the distribution of obstacles is dense, and the traversable spaces are relatively small; when entering an open area, the pursuer team spreads out to encircle and capture the evader (see Figure 9a,d).

3. When the initial distance between the pursuer team and the evader is large (see Figure 9c,d), the pursuers must spend more time shortening their distance from the evader. This may make it difficult for the pursuer team to achieve a sufficiently large encirclement.

4. As illustrated in Figure 9c,d, the proposed FINNCH method still performed well in long-distance cooperative pursuit.
Figure 9. The results of cooperative hunting with three pursuers capturing an ME. The evader moves from southeast to northwest: (a) pursuers start together from the southeastern corner, (b) pursuers set off from the northeast, southeast and southwest respectively. The evader moves from northeast to southwest: (c) pursuers begin chasing from the northwest, northeast and east, respectively, (d) pursuers start from the northeastern corner.

5.4. Comparison with Grid-Based Method

We then compare the proposed FINNCH with a grid-based method, such as in the work by Zhang et al. [27], in which three different initial configurations for the pursuer team and evader are adopted. In the grid-based method, the whole space is divided into a host of square grids. The qualitative and quantitative comparisons between FINNCH and grid-based method are demonstrated in Figure 10 and Table 2 respectively, from which the following can be seen.

1. Grid-based method fails to capture the evader (see the first row of Figure 10), whereas FINNCH successfully finds feasible routes for multiple pursuers (see the first and second row of Figure 10).

2. From the view of encirclement, the proposed FINNCH method also outperforms the grid-based method. Here, we use two encirclement indicators, namely, direction centrality metric (DCM) and encirclement distance metric (EDM), to help evaluate the encirclement (these two indicators are calculated in the same way as DCC and EDC described in Sections 4.5 and 4.6). Lower values of these two indicators suggest the pursuers can encircle the evader more effectively, making it hard to escape. It can be
seen that FINNCH generally has lower values in terms of both DCM and EDM (see “Encirclement Indicators” in Table 2.

3. As displayed in Table 2, the routes computed by FINNCH are significantly shorter than those computed by the grid-based method. Especially in Figure 10a, the average pursuit distance of FINNCH (1863.87 m) is around a third that of the grid-based method (4864.14 m). This indicates that FINNCH can shorten the distance traveled by each pursuer, thus accelerating the pursuit.

4. In addition, the proposed FINNCH method fully exploits the fine-grained navigation network. To be specific, in areas with roads, this method will prioritize the use of existing roads to find a route; in areas without obvious roads, this method will traverse terrain that is easily accessible. However, grid-based method merely considers the obstacles. Hence, the results for FINNCH are more reasonable and in accordance with the understanding of reality.

Figure 10. Qualitative comparison between FINNCH and grid-based method. (a) Pursuers start from the southeastern corner; the evader moves from southeast to northwest. (b) Pursers set off from the southwest, southeast and east respectively; the evader moves from southeast to northwest. (c) Pursuers begin chasing from the northwest; the evader moves from northwest to southeast.

Table 2. Quantitative comparison between FINNCH and grid-based method.

<table>
<thead>
<tr>
<th>Method</th>
<th>Id</th>
<th>Time (s)</th>
<th>Encirclement Indicators</th>
<th>Average Pursuit Distance (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>DCM</td>
<td>EDM</td>
</tr>
<tr>
<td>Grid-based method</td>
<td>a</td>
<td>0.226</td>
<td>0.572</td>
<td>0.221</td>
</tr>
<tr>
<td></td>
<td>b</td>
<td>0.333</td>
<td>0.553</td>
<td>0.168</td>
</tr>
<tr>
<td></td>
<td>c</td>
<td>0.168</td>
<td>0.298</td>
<td>0.208</td>
</tr>
<tr>
<td>FINNCH</td>
<td>a</td>
<td>0.105</td>
<td>0.444</td>
<td>0.182</td>
</tr>
<tr>
<td></td>
<td>b</td>
<td>0.115</td>
<td>0.137</td>
<td>0.184</td>
</tr>
<tr>
<td></td>
<td>c</td>
<td>0.139</td>
<td>0.248</td>
<td>0.168</td>
</tr>
</tbody>
</table>
6. Discussion
6.1. Number of Pursuers

The efficiency of the developed FINNCH method with different numbers of pursuers (two to six) was tested. Only the number of pursuers was changed, while the other parameters remained unchanged. Here, we adopted four different initial position configurations for the pursuer team and evader. The corresponding results are exhibited in Figure 11. Considering that the experimental area was relatively small, the number of pursuers was only increased from two to six. A larger number did not seem to conform to reality, so no further increase was made. The curves in Figure 11 indicate that as the number of pursuers increased, the time required for the pursuit also increased. However, the maximum time was no more than 0.5 s, which is still acceptable in practical applications.

It is evident that the computation time increased at different rates under different initial settings. This occurred because the complexity of obstacles between the pursuer group and the evader was different in different experiments. When the number of obstacles around the evader was large and the geometric shapes of these obstacles were complicated, more time was needed to find proper routes for all pursuers.

![Figure 11](image)

**Figure 11.** The path-planning time as a function of the number of pursuers under four different initial configurations.

Cooperative pursuit for multiple pursuers is a rather complex problem. In the field of robotics, even if an ideal space is given, achieving a successful pursuit within a limited time is difficult, so the pursuit domain is usually bounded. The situation in the real world is much more complicated than that in an artificially designed space. In this study, a residential community was used as the experimental area, in which there were mixed landcover types. It was found that some pursuers failed to reach the vicinity of the evader at the same time as the other pursuers. This case was regarded as a failed pursuit. The relationship between the success rate of the pursuit and the number of pursuers was studied. The greater the number of pursuers, the higher the percentage of successful pursuits (see...
Figure 12). Obviously, more pursuers can form a larger encirclement, thus making it easier to capture the evader.

![Figure 12](image)

**Figure 12.** (a) The initial positions of the pursuer team were generated in the white region and that of the evader was generated in the blue region. (b) The success rate of cooperative pursuit as a function of the number of pursuers.

### 6.2. Influences of Important Parameters

The developed FINNCH algorithm is significantly influenced by four hyperparameters, namely, $S_{\text{max}}$, $r_{\text{thresh}}$, $r_{\text{keep}}$, and $W$, which are manually determined. To obtain optimal routes for cooperative pursuit, fine-tuning is demanded when executing the algorithm. This section briefly discusses the influences of these parameters on cooperative pursuit and provides basic guidance in choosing proper hyperparameters.

$S_{\text{max}}$ controls the maximum strength of the repulsive force. As the key parameter in the FINNCH method, $S_{\text{max}}$ affects the formation of the encirclement mainly by influencing the distance between pursuers. If $S_{\text{max}} = 0$, dynamic cooperation does not exist and the encirclement cannot be formed. To determine how $S_{\text{max}}$ impacts the results, the FINNCH algorithm was run with different values of $S_{\text{max}} = 0.3, 0.7, 0.8$. As illustrated in Figure 13, if $S_{\text{max}} = 0.3$, there is a weak dynamic cooperation force between the pursuers, i.e., they cannot perform an encirclement large enough to capture the evader. If $S_{\text{max}}$ is too large, such as 0.8, the pursuers may be too dispersed spatially, leading to two possible outcomes: (1) the routes computed are too long, and (2) routes cannot be found. Hence, selecting an appropriate $S_{\text{max}}$ value is very important to balance the encirclement and the route length. In addition, as the number of pursuers increases, a smaller $S_{\text{max}}$ value contributes to better performance, as revealed in the experiments described in Section 6.1. In a limited geographic space, a greater number of pursuers means that the search space per pursuer is reduced, so a smaller distance between the pursuers must be maintained to perform the encirclement.
Figure 13. The results of cooperative pursuit using the FINNCH algorithm with different $S_{\text{max}}$ values: (a) $S_{\text{max}} = 0.3$, (b) $S_{\text{max}} = 0.7$, and (c) $S_{\text{max}} = 0.8$. Red dots stand for the starting points of pursuers, green dots stand for the capturing points, and colored solid lines represent the routes of pursuers.

The parameter $r_{\text{thres}}$ is the threshold radius size at which repulsion begins to decay. The larger the value of $r_{\text{thres}}$, the earlier the pursuer group turns from encirclement to capture. If $r_{\text{thres}} = 0$, the dynamic cooperation force stays at maximum, so the pursuers will start to narrow the encirclement at a later stage. An optimal $r_{\text{thres}}$ might be inferred by parameter tuning, thus striking a good balance between encirclement and capturing. An example is provided to illustrate how cooperative pursuit results change with different values of $r_{\text{thres}}$. The three subfigures in Figure 14 present the cooperative pursuit results with $r_{\text{thres}} = 30, 100, \text{ and } 160$, respectively. When $r_{\text{thres}}$ was small, the pursuers traveled a greater distance, but were able to form a better encirclement. With a relatively large value of $r_{\text{thres}} = 160$, the pursuer team began to reduce the distance between members when they were far away from the evader, thus failing to perform an effective encirclement and increasing the likelihood of the successful escape of the evader.

Figure 14. The results of cooperative pursuit using the FINNCH algorithm with different $r_{\text{thres}}$ values: (a) $r_{\text{thres}} = 30$, (b) $r_{\text{thres}} = 100$, and (c) $r_{\text{thres}} = 160$. Red dots stand for the starting points of pursuers, green dots stand for the capturing points, and colored solid lines represent the routes of pursuers.

The parameter $r_{\text{keep}}$ controls the minimum distance between pursuers. During the pursuit, maintaining a certain distance can expand the search space of the pursuer team, thus helping to improve the efficiency of the pursuit. $r_{\text{keep}}$ has a significant effect on the entire cooperative pursuit process. The dataset was used to show how the results change with the increase of the value of $r_{\text{keep}}$. Figure 15 exhibits the cooperative pursuit results with $r_{\text{keep}} = 0, 20, \text{ and } 40$, respectively. When $r_{\text{keep}}$ was 0, no distance-keeping force existed, and the routes of pursuers partially overlapped. When $r_{\text{keep}}$ was relatively large ($r_{\text{keep}} = 40$), the pursuers maintained a great distance between each other and had to...
traverse a much greater distance. Moreover, \( r_{\text{keep}} \) does not matter substantially for the distribution of capture points around the evader.

![Figure 15](image-url)

**Figure 15.** The results of cooperative pursuit using the FINNCH algorithm with different \( r_{\text{keep}} \) values: (a) \( r_{\text{keep}} = 0 \), (b) \( r_{\text{keep}} = 20 \), and (c) \( r_{\text{keep}} = 40 \). Red dots stand for the starting points of pursuers, green dots stand for the capturing points, and colored solid lines represent the routes of pursuers.

The hyperparameter \( W \) determines the importance of different components in the independent hunting strategy of each pursuer. Appropriate weights can reduce the distance that the pursuer must traverse while maintaining a good encirclement. In this study, it was found that a high weight of the NHV results in better performance. However, the pursuit may be trapped in the local minima if higher weights are given to the DCC and EDC. In practice, the weight of each component is determined by experience and the actual conditions of the experimental area.

6.3. Application

In this work, our aim is to provide multiple police officers or robots with route guidance to catch a dangerous person (i.e., murder, terrorist, etc.) or animal (i.e., lion, tiger, etc.) that wants to escape. To handle such a threat, a team instead of a single individual is definitely needed to increase the success rate of pursuit and lower the possibility of causing public security incidents. In general, familiarizing members with a complex urban environment (such as a residential community) in a short time is very hard, so route guidance is necessary for police officers or robots. Our method can guide them to reach appropriate locations to encircle the evader, thus catching the evader. In real-life scenarios, our method merely offers routes for police officers as a reference—they are not required to follow the route exactly.

6.4. Limitations and Future Work

Although the proposed FINNCH method has demonstrated promising results in the given experimental area for solving the cooperative pursuit problem, the limitations of the proposed method must be acknowledged, as discussed herein. As displayed in Figure 16, it was found that the routes computed by the FINNCH method fell into local optima (marked by orange rectangles). There are three potential reasons. First, the artificial potential functions used can cause the pursuers to easily fall into local optima. Second, the obstacles in urban environments are complex, and the paths linking areas are narrow; thus, it is difficult to find proper routes for all pursuers. Finally, when the evader is near an obstacle, to make the pursuers distribute evenly around the evader, some pursuers may be blocked by this obstacle. Two measures can be taken to solve this problem: the first is to adjust parameters, such as \( S_{\text{max}} \), to reduce the distance between the pursuers so that they can travel through narrow channels; the other is to add weights for the heuristic search strategy to help the pursuers get rid of the local optima.
Moreover, two potential directions are worthy of future exploration. First, the experimental area in the present work was merely confined to a residential neighborhood, so extending the proposed method to a larger urban area should be considered. Second, the proposed method can also be applied to other domains. For example, a group of unmanned ground vehicles (UGVs) can navigate in an unknown urban environment with the FINNCH method to collaboratively capture an evader.

![Figure 16](image.png)

**Figure 16.** Issues in the optimal routes computed by the FINNCH method. (a) A pursuer is trapped in a corner of a building. (b) A pursuer fails to bypass the building to catch the evader. Red dots stand for the starting points of pursuers, green dots stand for the capturing points, and colored solid lines represent the routes of pursuers.

### 7. Conclusions

In this study, a novel method was developed to compute routes for cooperatively capturing a single target on the basis of a fine-grained navigation network. The fine-grained navigation network is introduced to offer a detailed representation of local traversability. Inspired by some biological behaviors in nature, dynamic cooperation between the pursuers is implemented using artificial potential functions. An heuristic search strategy is adopted to speed up the search. Furthermore, two spatial constraints, namely, the DCC and EDC, are created to guide the pursuers to be evenly distributed around the evader. Then, a decentralized strategy is utilized to generate the routes for the pursuers. Taking a city in northwestern China as a case, a series of experiments were conducted to validate the proposed method. The results show that the proposed method performs well in addressing the problem of multiagent cooperative pursuit in urban environments. The proposed FINNCH method strikes a balance between the encirclement preventing the escape of the evader and the distance traversed by the pursuers. This new approach can support police officers to more quickly and cooperatively capture evaders in urban environments.

**Author Contributions:** Conceptualization, X.L. and M.S.; methodology, X.L. and M.S.; software, X.L.; validation, H.Y.; formal analysis, X.L. and H.Y.; resources, M.S.; data curation, S.Y.; writing—original draft preparation, X.L.; writing—review and editing, M.S.; visualization, H.Y. and S.Y.; supervision, M.S.; project administration, M.S.; funding acquisition, M.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Data are unavailable due to privacy.

**Acknowledgments:** The authors would like to thank the generous help from Liuyun Duan. This work was supported by the High-Performance Computing Platform of Peking University.
Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design, collection, analyses, data interpretation, or writing of the manuscript, or in the decision to publish the results.

References


**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.