

Autonomous environment-adaptive microrobot swarm navigation enabled by deep learning-based real-time distribution planning

Lidong Yang^{1,†}, Jialin Jiang^{1,†}, Xiaojie Gao², Qinglong Wang¹, Qi Dou^{2,5,6}, and Li Zhang^{1,3,4,5,6,*}

¹Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong.

²Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong.

³Department of Surgery, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong.

⁴Chow Yuk Ho Technology Centre for Innovative Medicine, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong.

⁵T-Stone Robotics Institute, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong.

⁶Multi-Scale Medical Robotics Center (MRC), InnoHK, Hong Kong Science Park, Shatin, N.T., Hong Kong..

[†]Lidong Yang and Jialin Jiang contributed equally to this work.

^{*}corresponding author: Li Zhang (lizhang@mae.cuhk.edu.hk.)

ABSTRACT

Navigating a large swarm of micro-/nanorobots is critical for potential targeted delivery/therapy applications owing to the limited volume/function of a single microrobot, and microrobot swarms with distribution reconfigurability can adapt to environments during navigation. However, current microrobot swarms lack the intelligent behavior to autonomously adjust their distribution and motion according to environment change. Such autonomous navigation is challenging which requires the real-time appropriate decision-making capability of the swarm for unknown and unstructured environments. Here, to tackle this issue, we propose a framework that defines different autonomy levels for environment-adaptive microrobot swarm navigation and designs corresponding system components for each level. To realize high autonomy levels, real-time autonomous distribution planning is a key capability for the swarm, regarding which we show that deep learning (DL) is an enabling approach that lets the microrobot swarm learn optimal distributions in extensive unstructured environment morphologies. For real-world demonstration, we study the reconfigurable magnetic nanoparticle swarm and experimentally demonstrate the autonomous swarm navigation for targeted delivery and cargo transport in environments with channels or obstacles. This work could bring computational intelligence to micro-/nanorobot swarms, enabling them to autonomously make appropriate decisions during navigation in unstructured environments.

Introduction

Swarm robotics is an important subfield of robotics¹, which seeks strategies to deploy large groups of robots that can cooperatively perform a task. To date, researchers have widely studied robot swarms at centimeter scales^{2,3} and demonstrated capabilities that cannot be accomplished by individual robots, such as directional motion⁴, manipulation⁵, and challenging terrodynamic tasks⁶. With broad application scenarios and theoretical/technical challenges, swarm robotics is receiving extensive attention⁷.

Especially, swarm robotics at micro-/nanoscale has recently been an emerging field with high impact^{8–11}. As a single

microrobot has a small size and volume, its capabilities may meet critical limitations, such as delivery^{12,13}, manipulation¹⁴, environment cleaning^{15,16}, and *in-vivo* tracking^{17–19}, where swarm control is the necessary technology. It is inspiring that recent technology can integrate actuators into microrobots²⁰, but the full integration of actuation, communication, battery, and control counterparts into artificial micro-/nanorobots has not been realized that hinders the swarm navigation via individual microrobot control. As a result, swarm navigation of simple tiny agents, e.g., spherical particles, actuated by external global energy fields is currently the feasible approach. The simple microagent structure also makes it easy for low-cost batch fabrication and functionalization, such as drug loading²¹ and therapy²². However, the global field actuation of thousands to millions of micro-/nanorobots, including acoustic^{8,23,24}, magnetic^{10,25–31}, electric³², and optical³³ fields, results in a challenging under-actuated swarm control problem. Such swarm control relies on the global field-induced gradients/forces^{8,10,24,28,33}, fluidic forces^{25,27,30}, or agent-agent interaction forces^{29,32} to assemble and form swarm patterns. Then, the microrobot swarms can perform on-demand navigation as an entity via field regulation.

In order to navigate in complex and unstructured environments (e.g., narrow branched channels), microrobot swarms should have reconfigurability to adjust their distributions accordingly. Although their under-actuation nature restricts the morphology changing capability, recently, it has been shown that the pattern configuration can be changed by switching the mode of the global actuation field^{10,24,26}. Moreover, the swarm distribution, e.g., swarm shape and orientation, can also be tuned by field parameter control^{10,24,26,29,34}. Such reconfigurable microrobot swarms have high adaptability to working environments, emerging as active matter with physical intelligence³⁵. However, to date, these adaptions rely on manual inputs of the operator who observes the swarm/environment and controls the field parameters. Hence, current microrobot swarms lack the intelligent behavior to autonomously make decisions (i.e., computational intelligence) when interacting with working environments. In fact, to well maneuver the reconfigurable swarms, human operators need extensive training to understand the swarm mechanisms and characteristics. As microrobot swarms may consist of thousands, even millions of basic elements that are loosely interacted, inappropriate manual control actions could break the stability of the swarm pattern or cause obstacle collisions that fail the navigation. In addition, it is overwhelming to simultaneously plan and control multiple swarm parameters (e.g., swarm shape, orientation, and motion). Therefore, autonomy and thus computational intelligence is critical for microrobot swarms, which enables them to autonomously make timely and optimal decisions during adaptive navigation in unstructured environments for task execution³⁶, e.g., targeted delivery and micromanipulation. This step is necessary to make microrobot swarms evolve toward intelligent matter³⁷.

Autonomy levels and system design

In this work, we are motivated to address the problem of autonomous adaptive microrobot swarm navigation in unstructured environments. As one possible framework, we propose five levels of autonomy (Fig. 1a). Each level has increasing autonomy whose characteristics are provided by description and schematic. The autonomy is realized externally, and key external system components required for each level are summarized in Fig. 1b.

Level 0: Manual navigation. This level is state-of-the-art and has no autonomy. A human operator observes the swarm and its working environment via an imaging system (e.g., optical microscopes). Swarm parameters, including the swarm shape, orientation, and position, are simultaneously planned for navigation without obstacle collision. To reach the target following the planned swarm trajectory and distribution (the minimal swarm bounding box (MSBB)), the operator controls the actuation field to correctly regulate these swarm parameters. This level only requires a field generation system and an imaging system, but the operator must well understand the mechanisms and characteristics of the microrobot swarm and have extensive experience to avoid wrong control actions that may fail the swarm navigation, for example, obstacle collisions and swarm pattern instability.

Level 1: Automated swarm control. With this level, the operator focuses on the navigation planning, i.e., the motion trajectory and swarm distribution, that suitably leads the swarm to the target. The exhausting swarm control task is automated by the external computer system so that the operator does not need a deep understanding of the swarm reconfiguration and motion principles. In addition, automated control can maintain high precision and swarm stability. Regarding the automated

a

Autonomy level	Description	Schematic
Level 0	Manual navigation. An operator observes the swarm and plans its future states. The actuation field is then manually controlled to accomplish the navigation to the target.	
Level 1	Automated swarm control. The microrobot swarm automatically follows the given trajectory and distribution by automated field regulation.	
Level 2	Autonomous trajectory tracking. The microrobot swarm autonomously navigates along the given trajectory with real-time distribution planning and automated control.	
Level 3	Autonomous target reaching. The microrobot swarm autonomously navigates to the given target with real-time trajectory and distribution planning and automated control.	
Level 4	Fully autonomous navigation. The microrobot swarm autonomously navigates in an unknown environment for task execution.	

• Notes: the red color marks manually determined parameters, and the green color marks autonomously determined parameters.

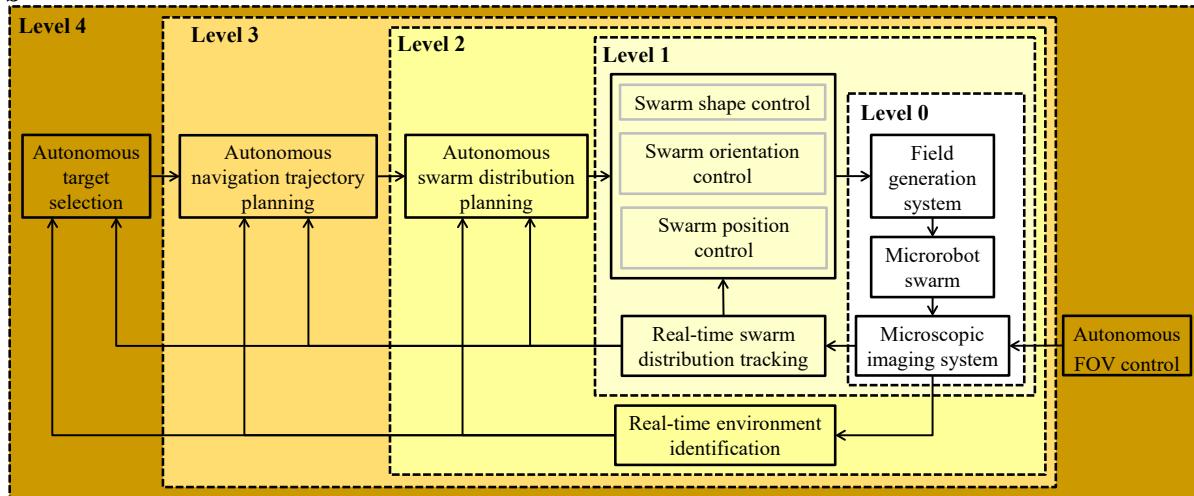
b

Figure 1. Autonomy levels of environment-adaptive microrobot swarm navigation. With increasing of the autonomy level, less professional skills and operation actions are required for the human operator, while more system components should be designed to increase the decision-making capability of the swarm. Autonomy levels higher than 1 are challenging, for which microrobot swarms must make real-time appropriate distribution decisions in unstructured environments. **a**, Descriptions and schematics for the five autonomy levels. **b**, System components required for each autonomy Level. Five dashed rectangular boxes with different colors represent the five autonomy Levels, and the external system components required for each Level are enclosed in the corresponding rectangular box.

69 swarm control, we propose three controllers for the swarm shape, orientation, and position, respectively, which are quantified
 70 by the shape ratio $R_s = \frac{a_s}{b_s}$ (a_s the long edge and b_s the short edge of the MSBB), orientation angle α_s (the intersection angle
 71 between the long edge of the MSBB and x -axis), and the swarm center position P_s , respectively. As the shape deformation of a
 72 microrobot swarm is usually governed by multiple forces, such as the field-induced force, fluidic force, and the interactive
 73 agent-agent force, such deformation processes may be nonlinear and time-delayed that makes the controller design non-trivial.
 74 Here, we propose the fuzzy logic-based controller, which can translate the knowledge and experience of skilled operators into
 75 the control design³⁸. Unlike single microrobot control, the assembled swarm pattern may deform and even split when subject to
 76 excessive translational and rotational motion. Similarly, we propose to integrate such constraints into the swarm orientation
 77 controller via experience-derived fuzzy control. In addition, this level requires real-time swarm distribution tracking, and we
 78 adopt the statistics-based method³⁹, in which 360 uniformly distributed boundary points of the swarm distribution are used to
 79 optimally fit the MSBB. Detailed design and implementation of these system components are presented in Methods. Since this
 80 level requires preset swarm distributions and trajectory, it is only applicable for static environments.

81 **Level 2: Autonomous trajectory tracking.** This level requires the microrobot swarm to autonomously determine its
 82 distribution on the next trajectory point in real time, according to its current distribution and working environment. The operator
 83 only needs to set the navigation trajectory, which further simplifies the task and reduces the workload. In addition to the system
 84 components of Level 1, autonomous swarm distribution planning is required, which is one of the major challenges compared
 85 with single microrobot navigation where the microrobot can be simply treated as a spherical particle⁴⁰. Here, to provide a
 86 solution for this key issue, we designed a DL-based approach.

To let the swarm make optimal distribution decisions, i.e., R_s and α_s , on the next trajectory point, we formulated it as an
 optimization problem:

$$[R_s, \alpha_s] = \operatorname{argmin} \left\{ -\frac{1}{N} \sum_{i=1}^N (||b_i - \mathcal{O}||) + w_1 \cdot (R_s - R_{\text{current}}) + w_2 \cdot ||\alpha_s - \alpha_{\text{fwd}}|| \right\} \quad (1)$$

87 where b_i is the i -th of the N uniformly distributed boundary points of the MSBB; R_{current} is the current swarm shape ratio; \mathcal{O}
 88 denotes the obstacle regions; α_{fwd} is the forward direction from current swarm position to next trajectory point; w_1 and w_2 are
 89 positive constant weights. The three terms in Eq. (1) aim to maximize the average distance between the swarm distribution and
 90 the obstacles for safe navigation, minimize the swarm shape ratio for agile motion, and minimize the difference between the
 91 future swarm orientation and the future forward motion direction for fast and smooth navigation, respectively. The solution
 92 of Eq. (1) minimizes the weighted sum of the three terms. Users can also modify the optimization terms according to the
 93 properties of specific microrobot swarms. For example, if the pattern and motion of the swarm are robust to the close interaction
 94 with obstacles, one can increase w_1 and w_2 to reduce the weight of obstacle avoidance. When navigating in unstructured
 95 environments, Eq. (1) cannot be analytically solved, and the numerical gradient-based searching method may converge to a
 96 local minimum. Thus, the traversal-based optimization method (Supplementary Note 1) should be used to plan the globally
 97 optimal swarm distribution, which, however, may not fulfill the real-time requirement. Herein, we sought to train deep neural
 98 networks (DNNs) to imitate the optimal solutions of Eq. (1) obtained by the optimization method, after which the DNNs were
 99 implemented for real-time decision-making. Fig. 2a-b illustrate the working pipeline and structures of the two DNNs, i.e., the
 100 DNN for swarm shape planing (SSP-DNN) and the DNN for swarm orientation planning (SOP-DNN). First, the SSP-DNN
 101 takes the current navigation state as its input that contains the current swarm distribution, surrounding environment, and next
 102 trajectory point, which correspond to the three channels of the input color image. Then, an intermediate swarm distribution can
 103 be obtained by moving the swarm to the next trajectory point with its shape ratio and orientation angle being the output of the
 104 SSP-DNN and α_{fwd} , respectively. Finally, the SOP-DNN rectifies the swarm orientation angle in the intermediate state, and
 105 thus the swarm distribution on the next trajectory point is determined. Details are explained in Methods.

106 For the DNN training, there is no public dataset. To make the method easy to implement and have robustness to
 107 difference/randomization between the training data and the real environment, we propose to tailor a simulation engine to

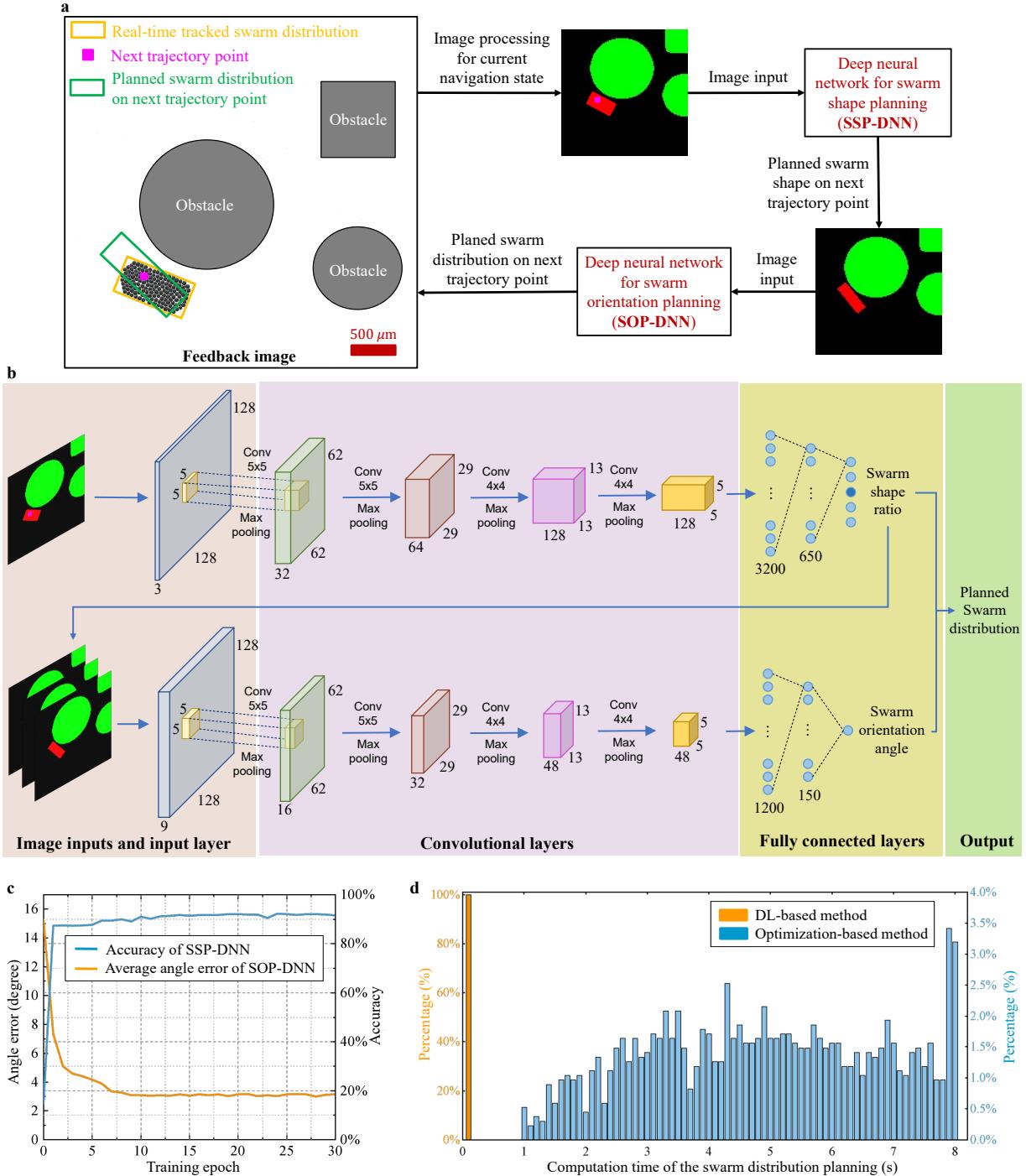


Figure 2. The real-time autonomous swarm distribution planning method based on DL. **a**, Working pipeline of the method. **b**, Structures of the two DNNs. **c**, Training convergence of the two DNNs. **d**, Computation time comparison of the proposed DL-based method and the traversal optimization-based method for swarm distribution planning, in which the statistics of the computation time of 1 K random examples are plotted.

generate a large dataset based on basic features of the real environment. Moreover, to reduce the domain gap between the images in the simulation and the images captured by the imaging system, before sending the captured image to the DNN, the real-time environment identification component should filter noise and segment the obstacle regions. Details of the training and applications processes are described in Supplementary Note 2. The generated dataset contains 50 K navigation scenarios

with different environment morphologies or swarm navigation states, 100 of which are illustrated in Supplementary Fig. 2. Validation results (Fig. 2c) by 5 K additional navigation scenarios showed that, after 30 training epochs, the accuracy of the SSP-DNN and the angle error of the SOP-DNN are 90.7% and 3.2°, respectively. Fig. 2d plots the distribution diagram of computation time for 1 K cases, showing that the proposed DL-based method keeps the computation time less than 0.1 s, which achieves ten times (at least) faster processing speed compared with the traversal-based optimization method. We also studied the influence of the dataset size on the planning accuracy (Supplementary Note 3). With the DL-based method, the microrobot swarm is enabled with the real-time optimal distribution planning capability in unstructured environments. Validations for different environment morphologies, including blank, channel, and obstacle environments, are included in Supplementary Video 1, showing the planning robustness. This level is still only applicable for static environments, as it requires a preset navigation trajectory.

Level 3: Autonomous target reaching. Utilizing the autonomy of this level, the operator only needs to set the navigation target, then all the other swarm planning and control tasks are autonomously accomplished. That is, it requires no professional skills of the operator, but the swarm should perform the autonomous real-time trajectory planning, which is a key feature of autonomous navigation. As traditional trajectory planning methods, e.g., A* and RRT*,⁴¹ are designed for single robot navigation, they cannot be directly adopted for the microrobot swarm navigation. The specific requirements and the proposed RRT*-based method are presented in Methods. Other planning methods satisfying these requirements can also be used in the autonomy framework. Integrating with this system component, the microrobot swarm can autonomously and quickly re-plan its trajectory when the navigation target is dynamically modified, meaning that this level can be applied to dynamic environments.

Level 4: Fully autonomous navigation. After a task is given, e.g., targeted delivery to a region, the microrobot swarm with this autonomy level can autonomously accomplish the navigation without human intervention. This level further requires the autonomous target selection based on real-time analysis of the working environment and current swarm position. Moreover, to achieve long-distance navigation, the field of view (FOV) of the imaging system should be autonomously controlled. Apparently, this level can be applied to dynamic environments. The autonomous target selection and FOV control algorithms are included in Methods.

Reconfigurable magnetic nanoparticle swarm: a case study

To investigate how those autonomy levels and corresponding system components could be realized for the real-world microrobot swarm navigation, we took the magnetic microswarm as a case study which consists of millions of Fe₃O₄ nanoparticles with diameters around 400 nm (Supplementary Note 4). Compared with swarm navigation of traditional large-scale robots, the autonomous navigation of such microrobot swarms has three unique challenges as summarized in Supplementary Note 5.

Reconfiguration and environment adaptability

As shown in Fig. 3(a), three swarm pattern configurations can be triggered by applying three programmed magnetic fields. The spreading swarm (SS), ribbon-like swarm (RS), and vortex-like swarm (VS) are generated by a three-dimensional (3D) dynamic field⁴², an oscillating field²⁹, and a rotating field²⁵, respectively. Different swarm configurations can be reversibly transformed by switching field forms (Fig. 3(b) and Supplementary Video 2). The three configurations have distinct properties and thus different functions. As depicted in Fig. 3(c), they have three potential application scenarios: micromanipulation, targeted delivery, and targeted therapy.

The adaptive swarm navigation relies on the shape deformation and motion (translational and rotational) of the RS. We control the swarm shape ratio R_s by tuning the field ratio R_f (defined in Supplementary Fig. 5). The RS performs translational motion when the oscillation field plane has a nonzero pitch angle γ_f . The swarm orientation angle α_s follows the rotation of the field oscillating direction. We designed a field regulation principle that can independently control α_s and swarm motion direction θ_s via tuning the field orientation angle α_f and motion-mapped field angle θ_f , respectively. Details are presented in Methods and Supplementary Video 3. From the motion characterization experiments (Supplementary Fig. 6-7), we found: (1) a

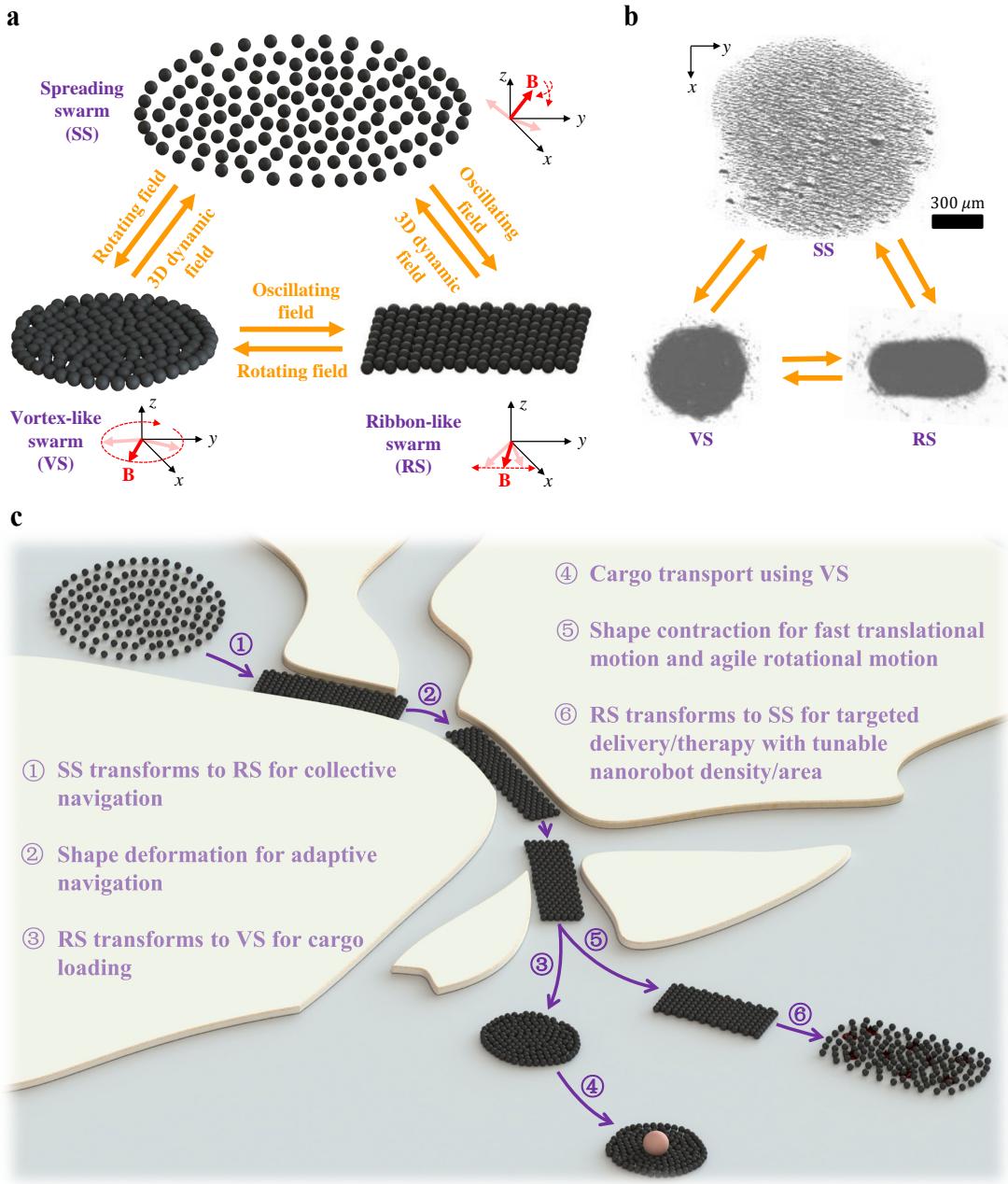


Figure 3. The reconfigurable magnetic nanoparticle swarm and its adaptive navigation for task execution. **a**, Schematic of the three swarm configurations and the transformation between them. Fe_3O_4 nanoparticles in the swarm are magnetized after applying the magnetic field **B**. **b**, Experimental results of the three swarm configurations and their swarm pattern transformation (Supplementary Video 2). The RS is generated by the interactive agent-agent magnetic forces, endowing it with the fast shape deformation capability and the resilience to disturbance (e.g., flows and boundaries). Thus, the RS is suitable for collective navigation with fast distribution adaption. The VS is formed by inward fluidic vortex force that could be utilized for cargo loading and transport. However, the relatively slow shape deformation and sensitiveness to environment inhomogeneity (e.g., boundaries) make it not suitable for adaptive navigation. Targeted delivery to a region is the unique function of the SS. Its distribution area increases with the actuation time of the 3D dynamic field so that the swarm distribution area can fit the targeted region by controlling the field application time. The pros and cons of the three swarms are summarized in Supplementary Table I. **c**, The adaptive navigation of magnetic nanoparticle swarm in unstructured environments for micromanipulation or targeted delivery/therapy using different swarm configurations.

154 smaller shape ratio results in a faster motion speed; (2) the swarm moves faster when the intersection angle between the swarm
155 orientation and the swarm motion direction becomes smaller; (3) the swarm pattern with a larger shape ratio splits more easily
156 during rotation. Therefore, to have fast translational motion and agile rotational motion, the RS should keep its shape ratio as
157 small as possible. For fast navigation, it is also desired that the swarm orientation keeps close to the motion direction. The two
158 criteria were used to optimize the swarm distribution planning, as in Eq. (1).

159 **Experimental demonstration**

160 In this part, we will demonstrate each autonomy Level. The purposes are to test if these system components still can work well
161 when more of them are integrated for a higher autonomy Level and to show how the skill requirements and workload of the
162 operator could be reduced when the autonomy Level is enhanced.

163 ***Manual swarm control (Level 0) vs. automated swarm control (Level 1)***

164 To test the skill requirements for microrobot swarm navigation control, novice operators were instructed to navigate the RS
165 to follow five preset targets in sequence (Fig. 4a). These navigation targets have different positions, orientations, and shape
166 ratios so that the operator needs to simultaneously control the swarm shape ratio, orientation, motion direction, and motion
167 speed by regulating four field parameters: R_f , θ_f , α_f and γ_f . Owing to the low-frequency and non-optimal manual control, the
168 swarm cannot perfectly navigate to the targeted distributions. The swarm also spent long time for shape deformation and did
169 not navigate with the shortest path. More critically, without a deep understanding of the swarm mechanism, novice operators
170 may easily undergo failure. One common scenario is shown in Fig. 4a, where a large swarm shape ratio is desired for Target #3
171 and #4. In such cases, too fast rotation would lead to swarm splitting (Supplementary Video 4). More common failure cases are
172 shown in Supplementary Note 6 and Supplementary Fig. 8.

173 For comparison, we implemented the automated control. Fig. 4b and Supplementary Video 4 illustrate the navigation
174 results, showing that the high-frequency (10 Hz) and parallel multi-parameter automated control enabled the accurate navigation
175 to the five targeted distributions. Notably, the swarm orientation controller can generate the fastest constrained rotation motion
176 without swarm splitting. Results revealed that, with the automated control, requirements of skill and workload for the operator
177 are reduced, while the navigation accuracy and swarm stability are increased.

178 ***Trajectory tracking under obstacles (Level 2)***

179 In this experiment, a 'CU'-shaped trajectory composed of dense trajectory points was preset for the swarm to track, around
180 which three static obstacles suppress the available navigation space (Fig. 4c). The objectives were (1) to test the real-time
181 experimental performance of the autonomous swarm distribution planning method and (2) to test the navigation feasibility and
182 accuracy after integrating the autonomous swarm distribution planning and automated control. Moreover, to assess the planning
183 intelligence, two experiments with different nanoparticle amounts are conducted.

184 Navigation results (Supplementary Video 5, Fig. 4c-d) showed that, although the training dataset does not contain this
185 working environment, the swarm still planned appropriate distributions, validating the robustness to environment morphology
186 changing. Meanwhile, the swarm automatically followed the planned distribution, which validated the compatibility between the
187 distribution planning and automated control. Not only the successful adaptive navigation under obstacles, the high consistency
188 between the navigation trajectory and the preset one also validated the accuracy. Furthermore, it is noteworthy that when the
189 swarm contains different nanorobot amounts, it will adjust the distribution strategy accordingly. The swarm kept its shape ratio
190 as small as possible under the condition of successful obstacle avoidance. This intelligent behavior was obtained during the
191 imitation learning process so that the swarm can maintain agile rotational and fast translational motion. As demonstrated, the
192 swarm containing fewer nanorobots exhibited smaller shape ratios when encountering obstacles, resulting in faster navigation
193 (Fig. 4e).

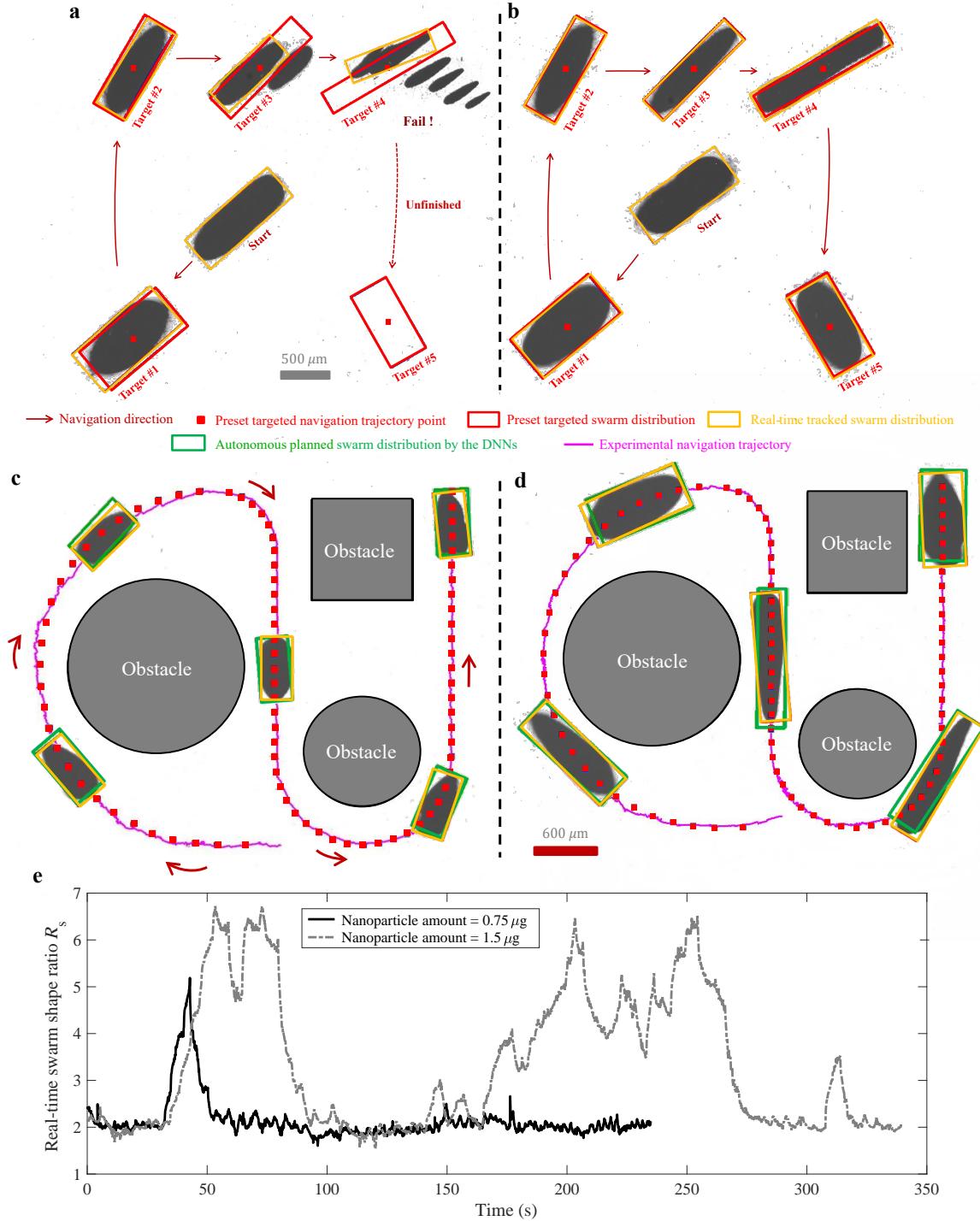


Figure 4. Experimental demonstrations of navigation autonomy Levels from 0 to 2. **a**, The RS was manually controlled to navigate to five preset targets in sequence, which corresponds to Level 0. When the swarm shape ratio is large (5 at Target #3 and 6 at Target #4), the difficulty of simultaneous shape, motion, and orientation control increases, even failures happened: the swarm split due to the excessive rotation control. **b**, The navigation task was the same as that in **a**, but manual control was replaced by automated control, i.e., autonomy Level 1. Navigation accuracy was enhanced, and the swarm maintained stability during the navigation. **c** and **d**, Comparative demonstrations of autonomy Level 2, in which the swarms with different nanoparticle amounts ($0.75 \mu\text{g}$ for **c** and $1.5 \mu\text{g}$ for **d**) navigated along the 'CU'-shaped trajectory under obstacles. The field pitch angle was fixed at 3° for both cases. **e**, The real-time swarm shape ratios during the experiments in **c** and **d**.

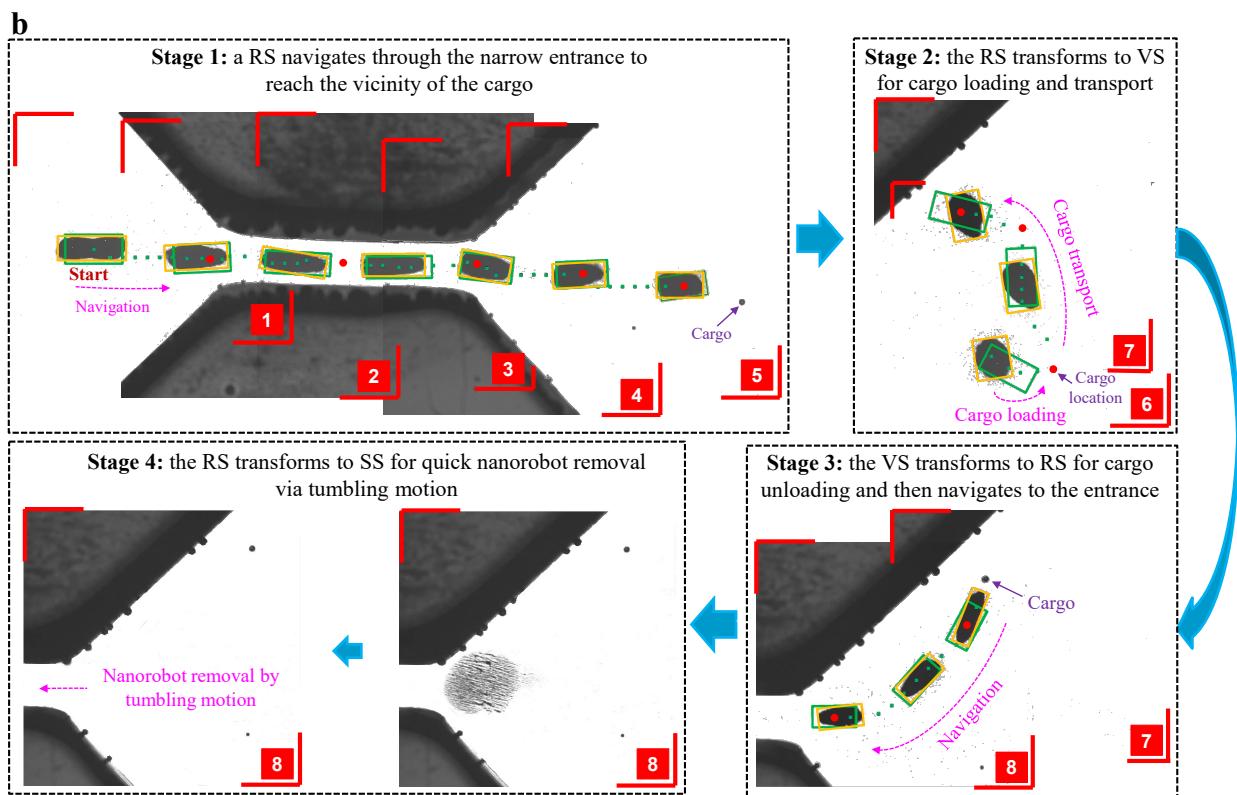
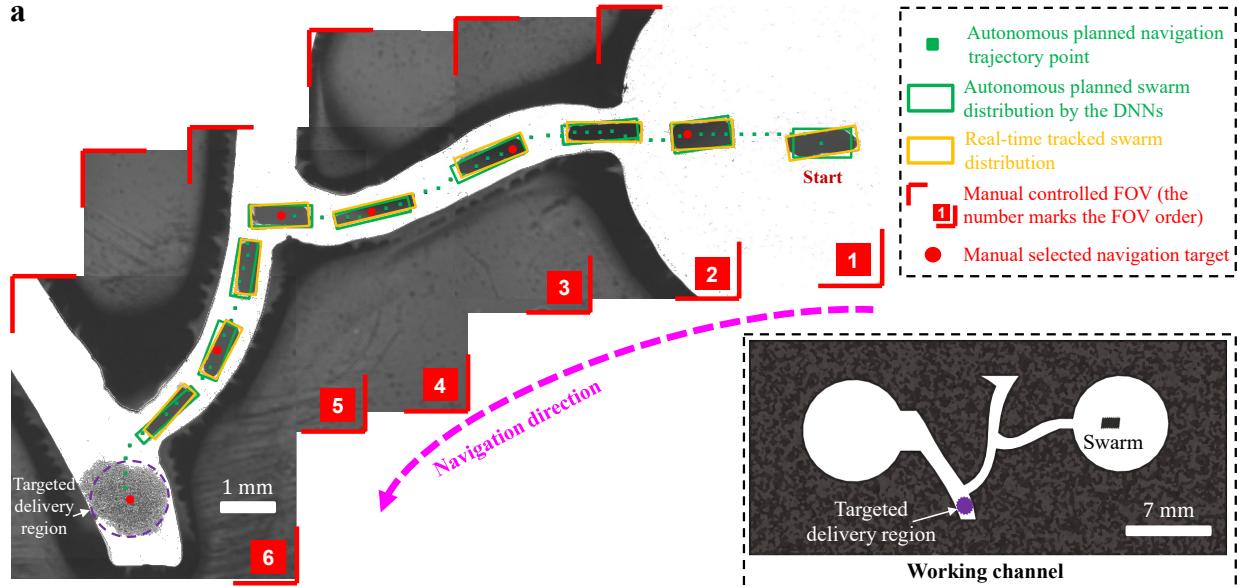


Figure 5. Task execution using autonomy Level 3. **a**, Targeted delivery of a swarm of magnetic nanorobots ($1.5 \mu\text{g}$) to a region in a channel environment. To accomplish the task, the operator only needs to select the target position in each FOV and then controls the FOV when the swarm reaches the target. Repeat the two operations until the swarm reaches the final delivery target. Field pitch angle γ_f was set as 3° for translational motion. Finally, the RS transforms to SS via field switching to cover the desired region. The bottom inset shows the fabricated acrylic channel environment. **b**, Cooperative micromanipulation in confined space using the reconfigurable magnetic nanorobot swarm. This task has four stages, requiring the cooperation of the three swarm configurations. To accomplish such a relatively complex task, the operator would benefit from the high autonomy level of the swarm, which could largely reduce the skill requirements and workload.

194 **Targeted delivery to a region in channel environment (Level 3)**

195 This experiment aims to demonstrate the delivery of a swarm of nanorobots to a targeted region through vessel-like environments,
196 for which we fabricated branched channels with varying widths and curvatures (Fig. 5a). Using autonomy Level 3, novice
197 operators who do not understand the swarm mechanisms can still easily perform this task. They only need to select the targeted
198 position in the current FOV, and when the swarm reaches the target, they adjust the FOV to approach the final target. After the
199 swarm reaches the final targeted position, the RS transforms to SS to cover the desired region.

200 Fig. 5a and Supplementary Video 6 show the experimental results. When the novice operator selected one navigation target,
201 the swarm autonomously planned its trajectory points and corresponding distributions leading it to the target without collision.
202 Since a safety distance against the wall is considered in the autonomous trajectory planning, the trajectory points are located
203 near the channel center, allowing for smooth navigation. The automated control worked well together with these planning
204 decisions such that the swarm could appropriately adjust its shape, orientation, and position to navigate in the narrow channels.
205 After changing the FOV five times, the swarm finally reached the delivery target, which then transformed to SS to cover the
206 desired region. Further, to validate the navigation capability in highly curved environments with sharp turns and curved narrow
207 channel environments, we also conducted simulations and experiments. Details are included in Supplementary Note 7 and
208 Supplementary Video 7.

209 **Cooperative micromanipulation in confined space (Level 3)**

210 In addition to the targeted delivery, a reconfigurable nanorobot swarm can also be utilized for micromanipulation in a confined
211 workspace. We performed this task by creating a space with a narrow entrance, to passing through which the swarm must
212 reconfigure its distribution.

213 This task consists of four stages, as illustrated in Fig. 5b. At Stage 1, the operator repeated the manual FOV control and
214 target selection until the swarm reached the vicinity of the cargo (a 100 μm polystyrene microbead). At Stage 2, the RS
215 transformed to VS by switching to the rotating field, following which the operator should select the cargo location as the next
216 navigation target such that the cargo can be loaded into the swarm via inward fluidic vortex force. The cargo may not be
217 successfully loaded in one operation. However, as the autonomous trajectory planning and distribution planning are executed in
218 real time, the operator can dynamically adjust the target until the cargo is loaded. Then, by similar operations in Stage 1, the
219 cargo was transported to the targeted location. At Stage 3, the VS was transformed back to RS. Due to the outward fluid flow
220 of the RS, the cargo was unloaded. Meanwhile, the RS navigated to the entrance. Finally, at Stage 4, the nanorobot swarm
221 quickly moved out of the confined space via tumbling motion. To avoid blocking the entrance by large nanorobot clusters, the
222 RS should transform to SS before the removal. Benefiting from the high autonomy level, the mental and physical workloads of
223 the operator were largely reduced when performing such a relatively complex task. The task execution process is included in
224 Supplementary Video 8.

225 **Fully autonomous exploration in channel environment (Level 4)**

226 Although autonomy Level 3 has dramatically reduced the human workload, the appropriate target selection and FOV control
227 still rely on the experience of the operator. Therefore, we sought to validate if it is feasible to let the swarm make full decisions
228 when navigating in unknown and unstructured environments. To increase the navigation difficulty, we fabricated a channel
229 environment with 90° sharp turns and diminishing passable width (Fig. 6a). A channel branch with a dead-end was used to test
230 the intelligence of the swarm when facing such cases.

231 In this experiment, 1.5 μg magnetic nanoparticles were added to the working environment. After transforming to RS, the
232 swarm started to explore the environment with full autonomy. Results (Fig. 6a, Supplementary Video 9) showed that the
233 navigation targets and FOVs were appropriately controlled, which continuously led the swarm to unexplored locations. The
234 advantages of such autonomous control lie in the high-frequency, quantitative, and reliable actions, which can avoid wrong
235 operations in manual control, for example, the swarm is missed when controlling the FOV. Notably, the sharp turns were
236 well overcome, and the swarm can intelligently adjust its navigation direction when encountering the dead-end, and it also

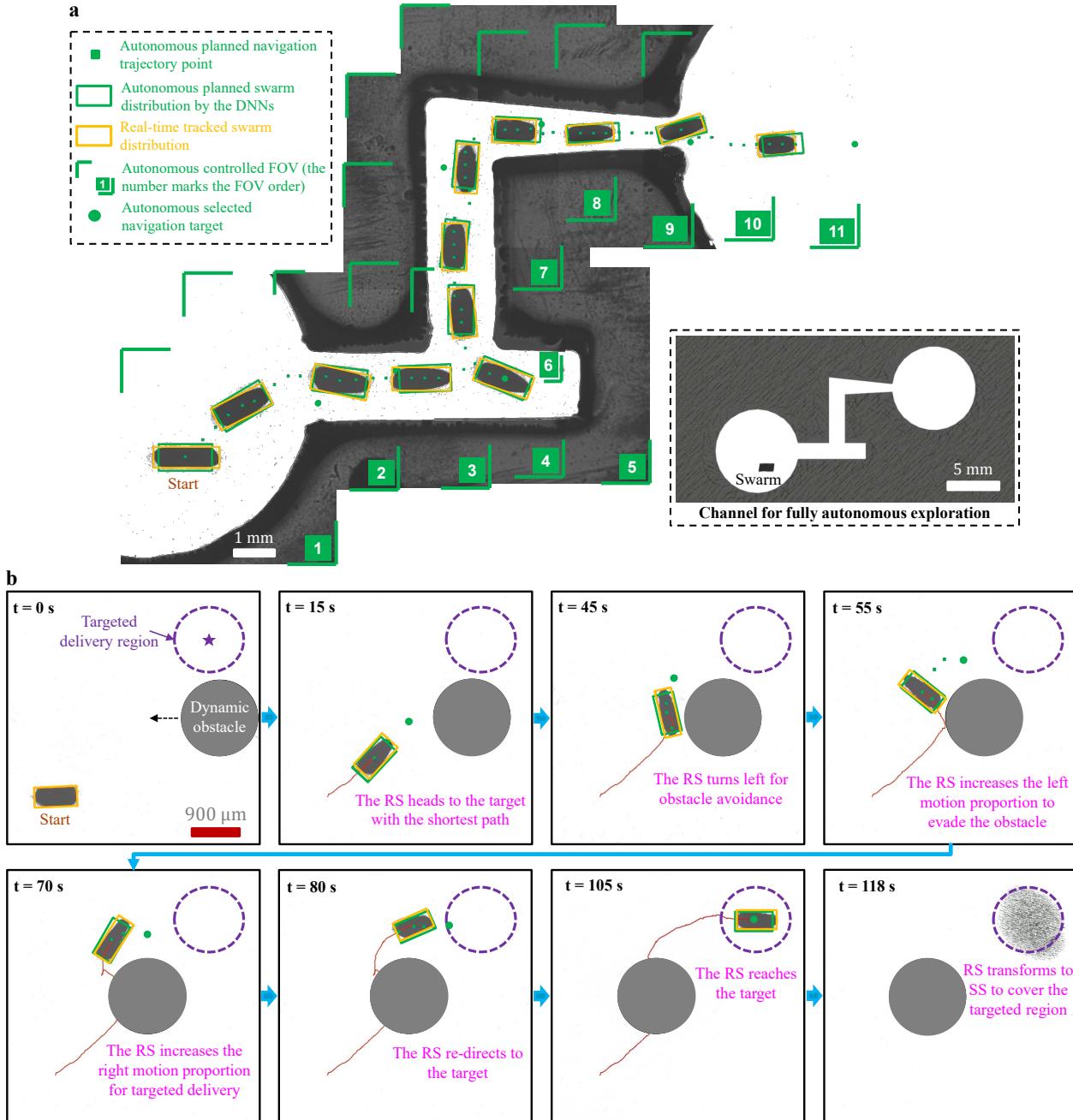


Figure 6. Experimental demonstrations of autonomy Level 4. **a**, Fully autonomous microrobot swarm exploration in an unknown channel environment. The 90° sharp turns, dead-end channel branch, and diminishing channel width were designed to test the autonomous capability of the swarm when handling relatively complex situations. The bottom inset shows the fabricated acrylic channel environment. **b**, Fully autonomous swarm delivery to a targeted region under dynamic obstacles. This figure contains eight time instants during the process, in each of which the top-left corners mark the timestamps. The brown line indicates the past path of the microrobot swarm. Without human intervention, the swarm successfully reached the target with intelligent swarm distribution planning, and it finally covered the desired region by transforming to SS. Field pitch angle γ_f was set as 4° for the two tasks.

237 appropriately adjusted its distributions to pass through the narrow channel segment.

238 **Fully autonomous targeted delivery under dynamic obstacles (Level 4)**

239 Leveraging the real-time feature of the framework, it is feasible for the swarm to execute fully autonomous tasks in dynamic
240 environments. This experiment aims to validate this capability, where a swarm of nanorobots is demanded to be delivered to a
241 targeted region. Between the swarm start location and the delivery target, a dynamic circular obstacle (diameter: 1.35 mm,
242 speed: 25 μ m/s) would hinder the swarm navigation (Fig. 6b).

243 The autonomous delivery process is included in Supplementary Video 10, and Fig. 6b illustrates eight time instants. Before
244 entering the vicinity of the obstacle, the RS chose a straight and shortest path to the target. As the navigation targets were
245 selected beyond the current swarm position, the swarm can adjust its motion in advance to avoid obstacle collision. With the
246 continuous motion of the obstacle, the swarm increased its left motion proportion to evade the obstacle. When the swarm
247 had got rid of the obstacle, it turned right to reach the delivery target with the shortest path, and the delivery task was finally
248 finished by transforming to SS for desired region coverage. Interestingly, during this task, unlike in Fig. 5a and Fig. 6a that
249 the swarm had to exhibit elongation to pass narrow channels, the swarm did not face such environments and always kept the
250 smallest shape ratio. As a result, the swarm maintained the fastest translational motion and the most agile rotational motion.
251 This intelligent behavior was obtained during the imitation learning process, which enabled the microrobot swarm to make
252 appropriate decisions for different working environments.

253 **Discussion**

254 Autonomous and intelligent swarm control of micro-/nanorobots is critical for microrobotic theory and application. In this study,
255 to realize the autonomy for microrobot swarm navigation in unstructured environments, we defined five general autonomy levels
256 and designed corresponding system components. We enabled the real-time autonomous swarm distribution planning by DL.
257 After learning, the swarm can make optimal distribution decisions in real time when navigating in unstructured environments.
258 Our techniques were validated by demonstrating each autonomy level using the reconfigurable magnetic nanoparticle swarm.
259 Together, this work represents a step towards the intelligent microrobot swarms that can autonomously make planning and
260 control decisions to adapt to their working environments. The framework we proposed may provide a prototypical paradigm to
261 introduce computational intelligence to microrobot swarms. Following this work, there are three important topics that need
262 further investigation:

- 263 • **Transfer to other microrobot swarms or environments.** The presented methods can be transferred to other microrobot
264 swarms if they have controllable shape, orientation, and motion, such as the reconfigurable microparticle swarm²⁴ and
265 the elliptical vortex-like nanoparticle swarm³⁴. We took the latter one as an example, and the navigation results in
266 different environments are illustrated in Supplementary Fig. 11 and Supplementary Video 11. The DL-based method
267 could also be transferred to environments composed of other features by following the training and application processes
268 in Supplementary Fig. 1. In the future, the automated swarm control strategies utilizing both magnetic torques and
269 forces could facilitate the navigation on 3D non-planar surfaces with strong environment uncertainties, e.g., flow. For
270 the 3D tracking, multi-view feedback is required, e.g., biplane fluoroscopes. After the 3D navigation and tracking of
271 reconfigurable microrobot swarms are realized, investigating how to transfer the methods to such 3D cases is an important
272 topic. As the RRT*-based trajectory planning method can be extended to 3D⁴³ and the DL-based method can learn 3D
273 features⁴⁴, these methods are promising to be transferred to 3D scenarios.
- 274 • **Integration with medical imaging modalities.** Biomedical applications may require medical imaging modalities. To
275 justify the effectiveness of the presented methods in such cases, we further implemented ultrasound imaging and x-ray
276 fluoroscopy (Please refer to Supplementary Note 8 for details). The preliminary experimental results showed that the
277 nanoparticle swarm can be tracked using our method. The DL-based planning method was also proved effective under
278 both ultrasound imaging and x-ray fluoroscopy. In the future, to deal with more complex scenarios under medical
279 imaging modalities, DL-based segmentation methods^{45,46} together with domain generalization methods⁴⁷ should be

further studied for accurate and reliable environment segmentation. In addition, the integration of the entire system and methods for medical imaging-guided autonomous environment-adaptive microrobot swarm navigation should be investigated.

- Simultaneous autonomous navigation of multiple microrobot swarms.** In the current strategy, we gather sufficient nanoparticles into the swarm of interest before the autonomous navigation. Then, the system only considers this microrobot swarm for navigation. At present, the independent shape and motion control of multiple magnetic nanoparticle swarms still remains a challenge. After feasible methods are realized for this issue, it would be an interesting topic to study the parallel execution of the proposed framework for every swarm and to realize the simultaneous environment-adaptive navigation of multiple microrobot swarms.

Methods

This research complies with all relevant ethical regulation, and The Joint Chinese University of Hong Kong-New Territories East Cluster Clinical Research Ethics Committee approved the study protocol (CREC Ref. No. 2020.384). Refer to the Supplementary Note 8 for details regarding ethics approval, recruitment and consent.

Field regulation principles

Oscillating field for the RS: The fast shape deformation and resilient swarm pattern to disturbances make the RS suitable for environment-adaptive navigation. We designed a field regulation method to decouple its orientation control and motion direction control. The basic oscillating field for RS actuation is described by

$$\mathbf{B}_R = [B_x, B_y, B_z]^T = [a_f \cdot \sin(2\pi\omega t), b_f, 0]^T \quad (2)$$

where ω and t represent the field frequency and time, respectively. The field ratio $R_f = a_f/b_f$, as defined in Supplementary Fig. 5. The field strength is computed by $B = \sqrt{a_f^2 + b_f^2}$. Three matrix rotation operations are designed to transform the basic field to a new field \mathbf{B}_R^* with three independent field parameters: the field orientation angle α_f , the field pitch angle γ_f , and the motion-mapped field angle θ_f as:

$$\mathbf{B}_R^* = \mathbf{R}(\theta_f) \cdot \mathbf{R}(\gamma_f) \cdot \mathbf{R}(\alpha_f) \cdot \mathbf{B}_R \quad (3)$$

where

$$\mathbf{R}(\alpha_f) = \begin{bmatrix} \cos \alpha_f & -\sin \alpha_f & 0 \\ \sin \alpha_f & \cos \alpha_f & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{R}(\gamma_f) = \begin{bmatrix} \cos \gamma_f & 0 & \sin \gamma_f \\ 0 & 1 & 0 \\ -\sin \gamma_f & 0 & \cos \gamma_f \end{bmatrix}, \mathbf{R}(\theta_f) = \begin{bmatrix} \cos(\theta_f - \alpha_f) & -\sin(\theta_f - \alpha_f) & 0 \\ \sin(\theta_f - \alpha_f) & \cos(\theta_f - \alpha_f) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

By Eq. (3) and (4), α_f and θ_f can be tuned to independently control the swarm orientation angle α_s and swarm motion direction angle θ_s , respectively. Regarding swarm motion velocity, a larger field pitch angle γ_f results in a faster motion velocity, but a too large γ_f , e.g., 8° , would cause instability of the swarm pattern.

Rotating field for the VS: The inward fluidic attractive force of the VS is useful for cargo loading and transport. The VS is actuated by a rotating field whose expression in the global coordinate frame is

$$\mathbf{B}_V^* = \begin{bmatrix} B_x^* \\ B_y^* \\ B_z^* \end{bmatrix} = \begin{bmatrix} -B \cdot \cos \gamma_f \cdot \cos \theta_f \cdot \cos(2\pi\omega t) - B \cdot \sin \theta_f \cdot \sin(2\pi\omega t) \\ -B \cdot \cos \gamma_f \cdot \sin \theta_f \cdot \cos(2\pi\omega t) + B \cdot \cos \theta_f \cdot \sin(2\pi\omega t) \\ B \cdot \sin \theta_f \cdot \cos(2\pi\omega t) \end{bmatrix} \quad (5)$$

where the inputs γ_f and θ_f are for swarm velocity control and swarm motion direction control, respectively. Under the actuation

298 of such fields, magnetic nanoparticle chains would rotate accordingly and then assemble by the fluidic vortex attractive force.

299 **3D dynamic field for the SS:** After the collective swarm navigation to the target, the SS is triggered to let the swarm cover
300 the desired region. The mathematical expression of the 3D dynamic field to generate SS is

$$\mathbf{B}_S^* = \begin{bmatrix} B_x^* \\ B_y^* \\ B_z^* \end{bmatrix} = \begin{bmatrix} -B \cdot \sin(120 \cdot \text{Ceil}(5t)) \cdot \sin(2\pi\omega t) \\ B \cdot \cos(120 \cdot \text{Ceil}(5t)) \cdot \sin(2\pi\omega t) \\ B \cdot \cos(2\pi\omega t) \end{bmatrix} \quad (6)$$

299 where the function $\text{Ceil}(a)$ outputs the closest integer not smaller than a . This field disassembles the magnetic nanoparticle
300 swarm by segmenting the particle chains and making use of the chain-chain repulsive magnetic forces⁴². One can tune the
301 swarm distribution area by controlling the field application time t .

302 The corresponding field strengths to generate the SS, RS, and VS are 12 mT, 10 mT, and 8 mT, respectively. The
303 corresponding field frequencies for the SS, RS, and VS are 20 Hz, 15 Hz, and 8 Hz, respectively.

304 DNNs for autonomous swarm distribution planning

305 **Network structure:** As shown in Fig. 2, the deep neural network for swarm shape planning (SSP-DNN) or the deep neural
306 network for swarm orientation planning (SOP-DNN) contains one input layer, four convolutional layers, and two fully connected
307 layers. To let the SSP-DNN learn the swarm shape planning on the next trajectory point, the input to the SSP-DNN is an RGB
308 image with a dimension of $128 \times 128 \times 3$, which is obtained by image processing of the raw image from the feedback camera.
309 In the processing algorithm, current swarm distribution, next trajectory point, passable space in the working environment, and
310 obstacles in the working environment are represented by different colors. Unlike the SSP-DNN, the input of SOP-DNN consists
311 of three RGB images (each dimension: $128 \times 128 \times 3$) with different rotation angles, in order to let the DNN better perceive the
312 difference of swam rotation motion. On each image input, the swarm distribution with the output swarm shape by SSP-DNN is
313 overlaid on the next trajectory point. The orientation angle of the swarm on the second image is set as the forward direction
314 from the current position to the next trajectory point, treated as the base angle; the orientation angles of the swarm on the first
315 and third input images are obtained by rotating the base angle by 15° clockwise and counterclockwise, respectively. Going into
316 the DNNs, the first two convolutional layers of the SSP-DNN (SOP-DNN) employ a 5×5 convolutional kernel with a stride of
317 1 and output 32 (16) and 64 (32) channels, respectively. The next two convolutional layers of the SSP-DNN (SOP-DNN) apply
318 a 4×4 kernel with a stride of 1 and both output 128 (48) channels. Each convolutional layer is followed by a max-pooling
319 layer with 2×2 kernel and a stride of 2 and then a batch normalization layer. The feature maps generated by the convolutional
320 layers are then flattened to a one-dimensional vector and fed into a fully connected layer. The SSP-DNN generates probabilities
321 of choosing five pre-defined values: 2, 3, 4, 5, 6, which represents five swarm shape ratio candidates at the next trajectory point.
322 As for the SOP-DNN, it will directly output a single value ranging from -45° to 45° , representing the rotation angle toward the
323 base angle.

324 **Training details:** To train and test the DNNs, we built a simulation platform written by Python. The dataset for the working
325 environment was created by SolidWorks using the straight line and curve line tools to make it contain various morphologies
326 and further augmented by resizing, rotating, and mirroring operations. The generated dataset contains 50 K images, 100 of
327 which are shown in Supplementary Fig. 2. The dataset was split into 9:1 for training and testing, respectively. For each training
328 iteration, the minimal swarm bounding box (MSBB) indicating the current swarm distribution and the next trajectory point
329 were randomly located at the feasible region in one environment image. The learning targets of DNNs, i.e., the optimal values
330 of the shape ratio and orientation angle of swam distribution, were solved using the traversal-based optimization method. The
331 two DNNs were optimized using the stochastic gradient descent (SGD) optimizer with a learning rate $5e^{-5}$ to minimize the
332 cross-entropy loss function. We trained the two DNNs for 30 epochs with convergence having been reached, and the minibatch
333 size was set as 20. The training algorithm is programmed using python based on the pytorch framework. All training and
334 computation were conducted on a computer running Ubuntu 18.0 system, whose CPU and GPU are AMD Ryzen 3950X @

335 3.5GHz and RTX 3070 with 8GB video memory, respectively.

336 **Real-time swarm distribution tracking**

337 Due to the under-actuation of microrobot swarms, they usually navigate in a collective manner, i.e., the assembled swarm
338 patterns. To robustly track the swarm distribution and then extract the swarm position, swarm shape ratio, and swarm orientation
339 angle, we adopted a statistics-based method³⁹ that uses 360 boundary points of the swarm for distribution fitting. In this work,
340 unlike the ellipse fitting³⁹ designed for the VS, we aim to fit the MSBB that can be applied for other types of microrobot
341 swarms. To this end, we adopted the optimal fitting method in ref.⁴⁸ to obtain the MSBB. Based on the MSBB, the quantitative
342 swarm distribution parameters can be extracted. As shown in Fig. 4-6 and Supplementary Videos, the swarms' distributions
343 were well tracked in real time. Even the swarm split to several small swarms, the method still can track the largest one among
344 them (Fig. 4a).

345 **Automated swarm control algorithms**

346 To make the swarm automatically move to the planned swarm distributions, we designed three automated swarm control
347 algorithms based on fuzzy logic control. The closed-loop feature provides robustness to uncertainties from swarm models and
348 environments.

349 **Swarm shape control:** Shape deformation of microrobot swarms is critical for environment-adaptive navigation. However,
350 an explicit and precise model cannot be established for the deformation. Firstly, the relationship between the field ratio
351 R_f and swarm shape ratio R_s has strong uncertainty, depending on many factors, such as the swarm organization state, the
352 microrobot amount in the swarm, the varying fluidic flow when the swarm moves, etc. Secondly, as shown in Supplementary
353 Fig. 6, the swarm deformation has a significant time delay, which would vary depending on current swarm states, e.g., particle
354 concentration and aspect ratio. Considering these facts, we propose to use the fuzzy logic control, which does not require an
355 explicit mathematical model of the swarm deformation and can leverage the control knowledge and experience of skilled human
356 operators. Since the fuzzy logic control allows for multiple-input-multiple-output schemes, it has flexibility to be applied for
357 different microrobot swarms that may have different key swarm parameters and characteristics for the deformation³⁴.

Regarding the ribbon-like nanoparticle swarm (RS) in this work, after extensive experiments, we identified two key parameters for its deformation control. (1) $v(R_s(t))$ —the changing speed of the swarm shape ratio $R_s(t)$, which is calculated by

$$v(R_s(t)) = \frac{1}{v_{max}} ||R_s(t) - R_s(t - 0.2)|| \quad (7)$$

where v_{max} normalizes $v(R_s(t))$ to the value range of [0, 1]. The operation skill is that, when the RS performs fast shape changing, i.e., $v(R_s(t))$ is large, one should reduce the control gain to slow down the control action variation and thus to accelerate the reaching speed to a swarm equilibrium. Otherwise the fast nanoparticle re-organization in the swarm may cause instability of the RS, especially when the RS moves or rotates. (2) $c(R_s(t))$ —the tracking closeness of $R_s(t)$, which is defined by

$$c(R_s(t)) = \begin{cases} 1 - \frac{||R_s(t) - R_s^*(t)||}{R_s^*(t)} & , \frac{||R_s(t) - R_s^*(t)||}{R_s^*(t)} \leq 1 \\ 0 & , \frac{||R_s(t) - R_s^*(t)||}{R_s^*(t)} > 1 \end{cases} \quad (8)$$

where $R_s^*(t)$ is the targeted swarm shape ratio to be tracked. $c(R_s(t))$ also has a normalized value range of [0, 1]. The operation skill is that, considering the long time delay for the shape deformation, the control integration should be activated only for a narrow range of $c(R_s(t))$, that is, [0.8, 0.9], in order to avoid tracking overshoot. In addition, to accelerate the tracking convergence, the control action should be aggressive when $c(R_s(t))$ is small. Whereas, to avoid tracking oscillation, control action should be mild when $c(R_s(t))$ is large. Based on the knowledge and analysis above, we designed a fuzzy logic-based

Proportional-Integral (PI) controller for the swarm shape control:

$$R_f(t) = W_p \cdot K_p(t) \cdot (R_s^*(t) - R_s(t)) + \frac{W_i}{T_s} \sum_{t=0}^l \{K_i(t) \cdot (R_s^*(t) - R_s(t))\} \quad (9)$$

where $K_p(t) \in [1, 10]$ and $K_i(t) \in [1, 10]$ are the proportional gain and integral gain, respectively, determined by the fuzzy logic controller. $T_s = 0.2$ s is the sampling time. W_p and W_i are the constant weights for the control gains set as 0.1 and 0.02, respectively. The corresponding fuzzy control design for $K_p(t)$ and $K_i(t)$ based on the human operation skills are shown in Supplementary Fig. 14a-c and 15a-d. Then, by the centroid defuzzification method, the obtained input-output relationships of the two fuzzy logic controllers are illustrated in Supplementary Fig. 14d and 15e.

Swarm position control: The swarm position controller was designed to let the swarm automatically follow the navigation trajectory. In this work, the field pitch angle γ_f is fixed during motion so that the swarm position control is simplified as motion direction control. Denote the current swarm position and the position of the next trajectory point as $P_s = [x_s, y_s]^T$ and $P_s^* = [x_s^*, y_s^*]^T$, respectively, then the controller was designed as

$$\theta_f(t) = \theta_f(t - 0.2) + K_p \cdot \text{Atan2d}(y_s^* - y_s, x_s^* - x_s) \quad (10)$$

where K_p is a constant controller gain set as 1.05 by trial-and-error tuning via experiments, and the function $\text{Atan2d}(y, x)$ outputs the arc tangent of y/x with the unit of degree.

Swarm orientation control: It is also necessary for the RS to tune its orientation to adapt to the complex environment. Since the microrobots in the swarm are loosely coupled, the RS would split when subject to inappropriate rotation control actions. For this control issue, we designed a fuzzy logic-based controller based on two operation skills: (1) a larger swarm shape ratio should have a smaller upper limit of the field rotation angle. If exceeding this limit, the swarm will split. (2) A higher swarm shape changing speed, i.e., the swarm is not stable, should correspond to a smaller upper limit of field rotation angle in order to avoid swarm splitting. Based on these knowledge and experience, the controller was designed as:

$$\alpha_f(t) = \alpha_f(t - 0.2) + \text{Sgn}(\alpha_s^*(t) - \alpha_s(t)) \cdot \text{Min}(|K_p \cdot (\alpha_s^*(t) - \alpha_s(t)) + K_i \cdot \sum_{t=1}^l \{\alpha_s^*(t) - \alpha_s(t)\}|, \alpha_{\text{lmt}}) \quad (11)$$

where $\alpha_s^*(t)$ is the targeted swarm orientation, and functions $\text{Sgn}(a)$ and $\text{Min}(a, b)$ output the sign of a and the smaller one between a and b , respectively. K_p and K_i are constant controller gains set as 1.1 and 0.005 after careful trial-and-error tuning. The corresponding fuzzy control design and the input (R_s and $v(R_s(t))$)-output (α_{lmt}) relationship are illustrated in Supplementary Fig. 16. The other issue that should be considered for swarm orientation control is that the navigation speed would decrease when the orientation angle has a large deviation from the motion direction angle (Supplementary Fig. 7). Thus, during the navigation, the targeted swarm orientation angle $\alpha_s^*(t)$ in Eq. (11) was modified as

$$\alpha_s^*(t) = \alpha'_s + K \cdot [\theta_f(t) - \alpha'_s] \quad (12)$$

where α'_s is the planned swarm orientation angle at the next trajectory point, and the constant $K \in [0, 1]$ can be flexibly tuned according to the working environment. For example, K is set as a large value (e.g., 0.95) for open environments, e.g., Fig. 4, to make the swarm orientation approaches its motion direction to maintain relatively fast navigation speed. While, K is set as a small value (e.g., 0.05) for confined environments, for example, the channel environment in Fig. 5, to make the swarm orientation approaches its orientation on the next trajectory point to avoid collisions with walls.

Autonomous trajectory planning

A feasible trajectory planning method for microrobot swarm navigation should fulfill four specific requirements: (1) it should be applicable for unstructured environments; (2) the planned trajectory points should have a near-uniform distribution, because

373 a constant distance between the swarm and the next trajectory point is used in the DNN training; (3) a distance between the
 374 obstacles and the swarm trajectory points should be ensured to let the swarm pass through; (4) the planning time should be
 375 sufficiently short for real-time use. As traditional trajectory planning methods are designed for single robot navigation, they
 376 may not fulfill the four requirements, as illustrated in the comparison study in Supplementary Note 9. According to these
 377 requirements, we designed a new RRT*-based method. With three major modifications (summarized in Supplementary Note
 378 10), this method fulfills all the four requirements.

The overall trajectory planning process proposed in this work is summarized as **Algorithm 1**.

Algorithm 1 The proposed autonomous trajectory planning algorithm. $\text{Length}(\mathbf{T})$ returns the number of elements of the set \mathbf{T} .

```

1: Input:  $P_s$ ,  $P_g$ ,  $\mathcal{E}$ ,  $D_{sc}$ ,  $\delta$ ,  $R_{near}$ , and  $D_{sf}$ .
2:  $\mathbf{v}_1.x = x_s$ ,  $\mathbf{v}_1.y = y_s$ ,  $\mathbf{v}_1.cost = \sqrt{(\mathbf{v}_1.x - x_g)^2 + (\mathbf{v}_1.y - y_g)^2}$ ,  $\mathbf{v}_1.cucost = 0$ ,  $\mathbf{T}(1) = \mathbf{v}_1$ 
3:  $i = 2$ ,  $\mathbf{v}_{\text{lead}} = \mathbf{v}_1$ 
4: while  $i \leq Inf$  do
5:    $[\mathbf{v}_{\text{can}}.x, \mathbf{v}_{\text{can}}.y, \mathbf{v}_{\text{can}}.cost] = UniExpand(\mathbf{T}, \mathcal{E}, P_g, \mathbf{v}_{\text{lead}}, D_{sc})$ 
6:   if  $\mathbf{v}_{\text{can}} \neq void$  then
7:      $\mathbf{v}_i = \mathbf{v}_{\text{can}}$ 
8:   else
9:      $[\mathbf{v}_{\text{can}}.x, \mathbf{v}_{\text{can}}.y, \mathbf{v}_{\text{can}}.cost] = RandExpand(\mathbf{T}, \mathcal{E}, P_g, D_{sc})$ 
10:     $\mathbf{v}_i = \mathbf{v}_{\text{can}}$ 
11:  end if
12:   $\mathbf{v}_{\text{lead}} = \mathbf{v}_i$ ,  $\mathbf{T}(i) = \mathbf{v}_i$ 
13:   $\mathbf{I}_{\text{near}} = FindNeighbor(\mathbf{v}_i, \mathbf{T}, R_{near})$ 
14:   $[\mathbf{v}_i.parent, \mathbf{v}_i.cucost] = ChooseParent(\mathbf{v}_i, \mathbf{T}, \mathbf{I}_{\text{near}})$ 
15:  for  $k = 1$  to  $\text{Length}(\mathbf{I}_{\text{near}})$  do
16:     $[\mathbf{T}(\mathbf{I}_{\text{near}}(k)).parent, \mathbf{T}(\mathbf{I}_{\text{near}}(k)).cucost] = ReWire(\mathbf{v}_i, \mathbf{T}, \mathbf{I}_{\text{near}}(k))$ 
17:  end for
18:  if  $\sqrt{(\mathbf{v}_i.x - x_g)^2 + (\mathbf{v}_i.y - y_g)^2} < \delta$  then
19:     $\mathbf{v}_{\text{reach}} = \mathbf{v}_i$ 
20:    Break while
21:  else
22:     $i = i + 1$ 
23:  end if
24: end while
25:  $\mathbf{T}_{\text{trajectory}} = ExtractTra(\mathbf{T}, \mathbf{v}_i, P_g)$ 
26: for  $m = 2$  to  $\text{Length}(\mathbf{T}_{\text{trajectory}}) - 1$  do
27:    $[\mathbf{T}_{\text{trajectory}}(m).x, \mathbf{T}_{\text{trajectory}}(m).y] = SafeDist(\mathcal{E}, \mathbf{T}_{\text{trajectory}}, D_{sf})$ 
28: end for
29: Output:  $\mathbf{T}_{\text{trajectory}}$ 
```

379

The 'tree' denoted by \mathbf{T} explores the environment \mathcal{E} from the current swarm position P_s . The number of the tree vertexes (\mathbf{v}_i , $i = 1, 2, \dots$) gradually grows until reaching the targeted position P_g within a small distance δ . Unlike traditional RRT* that the exploration distances between vertexes are random, to generate near-uniform trajectory points for the swarm navigation, we make the distribution of the tree vertexes uniform by the *UniExpand()* algorithm. It expands the tree, from the leading vertex \mathbf{v}_{lead} , in the directions of $d_{\text{pre}} + 45^\circ \cdot k$ ($k = -2, -1, 0, 1, 2$) with distances of D_{sc} ($k = -2, 0, 2$) or $\sqrt{2}D_{sc}$ ($k = -1, 1$), where d_{pre} is the previous tree expansion direction. To reach the target with the shortest path, the exploration direction with the smallest cost is selected for tree expanding, which is defined by

$$\mathbf{v}_i.cost = Inf \cdot Collision(\mathbf{v}_i, \mathcal{E}) + \sqrt{(\mathbf{v}_i.x - x_g)^2 + (\mathbf{v}_i.y - y_g)^2} \quad (13)$$

where Inf denotes an infinite number, and $Collision(\mathbf{v}_i, \mathcal{E})$ returns 0 if \mathbf{v}_i dose not collide with obstacles or else it returns 1. $\mathbf{v}_i.x$ and x_g are the x coordinates of \mathbf{v}_i and the target, respectively. If all the exploration directions have costs of Inf or their

costs are larger than that of \mathbf{v}_{lead} , the leading vertex will be randomly replaced by an existing vertex in the tree to continue the exploration process. The algorithm *RandExpand()* is responsible for this action. With each tree expanding, a new vertex \mathbf{v}_{i+1} is added to the tree. To maintain the shortest path from the start, along with the tree expanding, the *ChooseParent()* algorithm will calculate the parent vertex of \mathbf{v}_{i+1} by minimizing the cumulative path cost defined by

$$\mathbf{v}_{i+1}.\text{cucost} = \mathbf{v}_{(i+1)^{\text{pr}}}.\text{cucost} + \sqrt{(\mathbf{v}_{i+1}.\text{x} - \mathbf{v}_{(i+1)^{\text{pr}}.\text{x}})^2 + (\mathbf{v}_{i+1}.\text{y} - \mathbf{v}_{(i+1)^{\text{pr}}.\text{y}})^2} \quad (14)$$

where $\mathbf{v}_{(i+1)^{\text{pr}}}$ represents the parent vertex of \mathbf{v}_{i+1} , i.e., $(i+1)^{\text{pr}} = \mathbf{v}_{i+1}.\text{parent}$. The indexes of parent vertex candidates forms the set \mathbf{I}_{near} . These vertexes locate at the neighborhood of \mathbf{v}_i within a radius of R_{near} . \mathbf{I}_{near} is found by the algorithm *FindNeighbor()*. After connecting the best parent vertex $\mathbf{v}_{(i+1)^{\text{pr}}}$ to \mathbf{v}_{i+1} , the *ReWire()* algorithm checks if the newly added vertex \mathbf{v}_{i+1} is a better parent vertex for those vertexes in \mathbf{I}_{near} . When the final tree vertex \mathbf{v}_i reaches the target, the exploration is terminated with $\mathbf{v}_{\text{reach}} = \mathbf{v}_i$, and the feasible trajectory is obtained as $\mathbf{T}_{\text{trajectory}}$ by the algorithm *ExtractTra()*. As the swarm distribution occupies an area, there should be a safety distance D_{sf} between each trajectory point and the obstacles in the environment. To this end, the algorithm *SafeDist()* is designed to optimize the positions of the obtained trajectory points. Three criteria are considered for the optimization: 1) the distance between the trajectory point and the obstacles should be larger than D_{sf} ; 2) the distance sum from the trajectory point and its two adjacent trajectory points should be minimized; 3) The trajectory point should locate at the middle of its two adjacent waypoints to keep the trajectory points uniform. Details of all the algorithms are included in Supplementary Note 10. Robustness test results on different environment morphologies are illustrated in Supplementary Fig. 18.

392 Autonomous target selection and FOV control algorithms

393 Due to the contradiction between the imaging resolution and FOV size of the microscopic imaging system, the long-distance
394 navigation requires FOV adjustment, and the navigation target should be selected at each FOV. Regarding the fully autonomous
395 navigation (autonomy Level 4), these two tasks should be autonomously determined by the swarm, for which we designed
396 corresponding algorithms.

Autonomous target selection: The target point P_{gcurrent} in current FOV should be able to guide the swarm for task execution in unstructured environments. **Algorithm 2** starts to search P_{gcurrent} from the direction d_{ts} with a distance D_{ts} from the current swarm position. According to different tasks, the d_{ts} and D_{ts} can be flexibly determined, where d_{ts} is computed by

$$d_{\text{ts}} = \begin{cases} d_{\text{npre}} = \text{Atan2d}(y_s - y_{\text{gpre}}, x_s - x_{\text{gpre}}), & S = 1 \\ d_{\text{target}} = \text{Atan2d}(y_g - y_{\text{gpre}}, x_g - x_{\text{gpre}}), & S = 0 \end{cases} \quad (15)$$

397 where d_{npre} and d_{target} are the previous navigation direction and the direction to the target, respectively. P_{gpre} and P_g represent
398 the target in the previous FOV and the final target, respectively. For instance, if the task is to reach a region for targeted
399 delivery (the task in Fig. 6b), d_{ts} is then set as the direction from the current swarm position to the targeted position, i.e., $S = 0$.
400 Otherwise, if one want to let the swarm explore unknown environment (the task in Fig. 6a), d_{ts} can be set as the previous swarm
401 navigation direction, , i.e., $S = 1$. D_{ts} should be adjusted according to the environment. A smaller D_{ts} is desired for a higher
402 dynamic environment. For example, we set $D_{\text{ts}} = 3$ mm for the static channel environment in Fig. 6a, whereas we set $D_{\text{ts}} = 1$
403 mm for the dynamic environment in Fig. 6b. When this prior searching direction is blocked by obstacles, different directions
404 will be searched starting from d_{ts} , until the available target is found.

Autonomous FOV control: After the microrobot swarm reaches the target in the current FOV, a new FOV should be selected to continue the long-distance navigation. Our strategy assumes that the previous navigation direction is the best one for future navigation. Thus, the FOV is controlled to move the swarm in the direction opposite to the previous navigation direction,

Algorithm 2 The proposed autonomous target selection algorithm.

```
1: Input:  $P_s$ ,  $P_g$ ,  $P_{gpre}$ ,  $\mathcal{E}$ ,  $D_{ts}$ , and  $S$ .
2: if  $S = 0$  then
3:    $d_{ts} = d_{target} = \text{Atan2d}(y_g - y_{gpre}, x_g - x_{gpre})$ 
4: else
5:    $d_{ts} = d_{npre} = \text{Atan2d}(y_s - y_{gpre}, x_s - x_{gpre})$ 
6: end if
7: for  $\delta = 0^\circ$  to  $360^\circ$  do
8:    $x_{can} = x_s + D_{ts} \cdot \cos(d_{ts} + \delta)$ 
9:    $y_{can} = y_s + D_{ts} \cdot \sin(d_{ts} + \delta)$ 
10:  if  $\text{Collision}(P_{can}, \mathcal{E}) = 0$  then
11:     $x_{gcurrent} = x_{can}$ 
12:     $y_{gcurrent} = y_{can}$ 
13:    Break for
14:  end if
15: end for
16: Output:  $P_{gcurrent}$ 
```

computed by

$$d_{fov} = \text{Atan2d}(y_{gpre} - y_{gcurrent}, x_{gpre} - x_{gcurrent}) \quad (16)$$

405 Then, the xy stage is automatically controlled to realize such FOV adjustment via a Proportional controller. To prevent losing
406 tracking of the swarm, when the swarm reaches the border of the FOV, the FOV control is terminated.

407 **Data availability**

408 The dataset⁴⁹ (~ 20 GB) used for training the DNNs are available in figshare: <https://doi.org/10.6084/m9.figshare.19149779.v1>.

409 **Code availability**

410 All the control and planning algorithms used in this study are available within the Article. Original codes for training the DNNs,
411 sample codes for executing the swarm distribution planning in different environment morphologies, and sample codes⁵⁰ for exe-
412 cuting the trajectory planning in a channel environment are available on GitHub <https://github.com/lidongYang22/Autonomous->
413 [microrobot-swarm-navigation](#) and Zenodo <https://doi.org/10.5281/zenodo.6032452>. The integrated software used for experi-
414 mental validation is available from the corresponding author on reasonable request.

415 **References**

- 416 1. Floreano, D. & Lipson, H. From individual robots to robot societies. *Sci. Robot.* **6** (2021).
- 417 2. Rubenstein, M., Cornejo, A. & Nagpal, R. Programmable self-assembly in a thousand-robot swarm. *Science* **345**, 795–799
418 (2014).
- 419 3. Soria, E., Schiano, F. & Floreano, D. Predictive control of aerial swarms in cluttered environments. *Nat. Mach. Intell.* **3**,
420 545–554 (2021).
- 421 4. Li, S. *et al.* Particle robotics based on statistical mechanics of loosely coupled components. *Nature* **567**, 361–365 (2019).
- 422 5. Shahrokhi, S., Lin, L., Ertel, C., Wan, M. & Becker, A. T. Steering a swarm of particles using global inputs and swarm
423 statistics. *IEEE Trans. Robot.* **34**, 207–219 (2017).

- 424 6. Ozkan-Aydin, Y. & Goldman, D. I. Self-reconfigurable multilegged robot swarms collectively accomplish challenging
425 terrodynamic tasks. *Sci. Robot.* **6** (2021).
- 426 7. Dorigo, M., Theraulaz, G. & Trianni, V. Reflections on the future of swarm robotics. *Sci. Robot.* **5** (2020).
- 427 8. Ahmed, D. *et al.* Bioinspired acousto-magnetic microswarm robots with upstream motility. *Nat. Mach. Intell.* **3**, 116–124
428 (2021).
- 429 9. Wu, C. *et al.* Ion-exchange enabled synthetic swarm. *Nat. Nanotech.* **16**, 288–295 (2021).
- 430 10. Dong, X. & Sitti, M. Controlling two-dimensional collective formation and cooperative behavior of magnetic microrobot
431 swarms. *Int. J. Robot. Res.* **39**, 617–638 (2020).
- 432 11. Yang, L. *et al.* A survey on swarm microrobotics. *IEEE Trans. Robot.* (2021, doi: 10.1109/TRO.2021.3111788).
- 433 12. Felfoul, O. *et al.* Magneto-aerotactic bacteria deliver drug-containing nanoliposomes to tumour hypoxic regions. *Nat.*
434 *Nanotechnol.* **11**, 941–947 (2016).
- 435 13. Wu, Z. *et al.* A swarm of slippery micropropellers penetrates the vitreous body of the eye. *Sci. Adv.* **4**, eaat4388 (2018).
- 436 14. Martel, S. & Mohammadi, M. Using a swarm of self-propelled natural microrobots in the form of flagellated bacteria to
437 perform complex micro-assembly tasks. In *2010 IEEE Int. Conf. Robot. Autom.*, 500–505 (IEEE, 2010).
- 438 15. Dekanovsky, L. *et al.* Chemically programmable microrobots weaving a web from hormones. *Nat. Mach. Intell.* **2**, 711–718
439 (2020).
- 440 16. Li, J., de Ávila, B. E.-F., Gao, W., Zhang, L. & Wang, J. Micro/nanorobots for biomedicine: Delivery, surgery, sensing,
441 and detoxification. *Sci. Robot.* **2** (2017).
- 442 17. Servant, A., Qiu, F., Mazza, M., Kostarelos, K. & Nelson, B. J. Controlled in vivo swimming of a swarm of bacteria-like
443 microrobotic flagella. *Adv. Mater.* **27**, 2981–2988 (2015).
- 444 18. Hortelao, A. C. *et al.* Swarming behavior and in vivo monitoring of enzymatic nanomotors within the bladder. *Sci. Robot.*
445 **6** (2021).
- 446 19. Wang, Q. *et al.* Ultrasound doppler-guided real-time navigation of a magnetic microswarm for active endovascular delivery.
447 *Sci. Adv.* **7**, eabe5914 (2021).
- 448 20. Miskin, M. Z. *et al.* Electronically integrated, mass-manufactured, microscopic robots. *Nature* **584**, 557–561 (2020).
- 449 21. Cheng, R. *et al.* Acceleration of tissue plasminogen activator-mediated thrombolysis by magnetically powered nanomotors.
450 *ACS Nano* **8**, 7746–7754 (2014).
- 451 22. Hwang, G. *et al.* Catalytic antimicrobial robots for biofilm eradication. *Sci. Robot.* **4** (2019).
- 452 23. Melde, K., Mark, A. G., Qiu, T. & Fischer, P. Holograms for acoustics. *Nature* **537**, 518–522 (2016).
- 453 24. Zhou, Z., Hou, Z. & Pei, Y. Reconfigurable particle swarm robotics powered by acoustic vibration tweezer. *Soft Robot.*
454 (2020).
- 455 25. Yu, J., Yang, L. & Zhang, L. Pattern generation and motion control of a vortex-like paramagnetic nanoparticle swarm. *Int.*
456 *J. Robot. Res.* **37**, 912–930 (2018).
- 457 26. Xie, H. *et al.* Reconfigurable magnetic microrobot swarm: Multimode transformation, locomotion, and manipulation. *Sci.*
458 *Robot.* **4** (2019).
- 459 27. Kaiser, A., Snezhko, A. & Aranson, I. S. Flocking ferromagnetic colloids. *Sci. Adv.* **3**, e1601469 (2017).
- 460 28. Loghin, D., Tremblay, C., Mohammadi, M. & Martel, S. Exploiting the responses of magnetotactic bacteria robotic agents
461 to enhance displacement control and swarm formation for drug delivery platforms. *Int. J. Robot. Res.* **36**, 1195–1210
462 (2017).

- 463 29. Yu, J., Wang, B., Du, X., Wang, Q. & Zhang, L. Ultra-extensible ribbon-like magnetic microswarm. *Nat. Commun.* **9**, 1–9
464 (2018).
- 465 30. Kokot, G. & Snezhko, A. Manipulation of emergent vortices in swarms of magnetic rollers. *Nat. Commun.* **9**, 1–7 (2018).
- 466 31. Yigit, B., Alapan, Y. & Sitti, M. Programmable collective behavior in dynamically self-assembled mobile microrobotic
467 swarms. *Adv. Sci.* **6**, 1801837 (2019).
- 468 32. Liang, X. *et al.* Hierarchical microswarms with leader–follower-like structures: electrohydrodynamic self-organization and
469 multimode collective photoresponses. *Adv. Funct. Mater.* **30**, 1908602 (2020).
- 470 33. Hong, Y., Diaz, M., Córdova-Figueroa, U. M. & Sen, A. Light-driven titanium-dioxide-based reversible microfireworks
471 and micromotor/micropump systems. *Adv. Funct. Mater.* **20**, 1568–1576 (2010).
- 472 34. Yang, L., Yu, J. & Zhang, L. A mobile paramagnetic nanoparticle swarm with automatic shape deformation control. In
473 *2020 IEEE Int. Conf. Robot. Autom. (ICRA)*, 9230–9236 (IEEE, 2020).
- 474 35. Sitti, M. Physical intelligence as a new paradigm. *Extrem. Mech. Lett.* **46**, 101340 (2021).
- 475 36. Cichos, F., Gustavsson, K., Mehlig, B. & Volpe, G. Machine learning for active matter. *Nat. Mach. Intell.* **2**, 94–103
476 (2020).
- 477 37. Kaspar, C., Ravoo, B., van der Wiel, W., Wegner, S. & Pernice, W. The rise of intelligent matter. *Nature* **594**, 345–355
478 (2021).
- 479 38. Mamdani, E. H. & Assilian, S. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man-mach. Stud.*
480 **7**, 1–13 (1975).
- 481 39. Yang, L., Yu, J. & Zhang, L. Statistics-based automated control for a swarm of paramagnetic nanoparticles in 2-d space.
482 *IEEE Trans. Robot.* **36**, 254–270 (2019).
- 483 40. Muñoz-Landin, S., Fischer, A., Holubec, V. & Cichos, F. Reinforcement learning with artificial microswimmers. *Sci.
484 Robot.* **6** (2021).
- 485 41. Karaman, S. & Frazzoli, E. Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **30**, 846–894
486 (2011).
- 487 42. Yu, J., Xu, T., Lu, Z., Vong, C. I. & Zhang, L. On-demand disassembly of paramagnetic nanoparticle chains for microrobotic
488 cargo delivery. *IEEE Trans. Robot.* **33**, 1213–1225 (2017).
- 489 43. Pérez-Higueras, N., Jardón, A., Rodríguez, Á. & Balaguer, C. 3d exploration and navigation with optimal-rrt planners for
490 ground robots in indoor incidents. *Sensors* **20**, 220 (2020).
- 491 44. Singh, A. Deep learning 3d structures. *Nat. methods* **17**, 249–249 (2020).
- 492 45. Karimi, D. *et al.* Accurate and robust deep learning-based segmentation of the prostate clinical target volume in ultrasound
493 images. *Med. image analysis* **57**, 186–196 (2019).
- 494 46. Chen, C. *et al.* Deep learning for cardiac image segmentation: a review. *Front. Cardiovasc. Medicine* **7**, 25 (2020).
- 495 47. Dou, Q., Coelho de Castro, D., Kamnitsas, K. & Glocker, B. Domain generalization via model-agnostic learning of
496 semantic features. *Adv. Neural Inf. Process. Syst.* **32**, 6450–6461 (2019).
- 497 48. Julien, D. 2d minimal bounding box. <https://www.mathworks.com/matlabcentral/fileexchange/31126-2d-minimal-bounding-box>. Updated: 2011-05-06.
- 498 49. Yang, L. *et al.* Autonomous environment-adaptive microrobot swarm navigation enabled by deep learning-based real-time
499 distribution planning (dataset). *Figshare* <https://doi.org/10.6084/m9.figshare.19149779.v1> (2022).
- 500 50. Yang, L. *et al.* Autonomous environment-adaptive microrobot swarm navigation enabled by deep learning-based real-time
501 distribution planning (sample codes). *Zenodo* <https://doi.org/10.5281/zenodo.6032452> (2022).

503 **Acknowledgements**

504 We would like to thank Shihao Yang and Dongdong Jin for help with the ultrasound experiments and thank Terence Lam and
505 Kevin Lai for help with the x-ray fluoroscopy experiments and thank Kai-Fung Chan for applying the permission of the placenta
506 experiments. This project has received funding support from the ITF project with project No. MRP/036/18X (to L.Z.), the
507 Croucher Foundation Grant with Ref. No. CAS20403 (to L.Z.), the CUHK internal grants (to L.Z.), the Multi-Scale Medical
508 Robotics Center (MRC), InnoHK, at the Hong Kong Science Park (to L.Z.), the SIAT-CUHK Joint Laboratory of Robotics and
509 Intelligent Systems (to L.Z.), and the CUHK Shun Hing Institute of Advanced Engineering (MMT-p5-20) (to Q.D.).

510 **Author contributions statement**

511 L.Y. conceived the study. L.Y. and J.J. designed and implemented the system hardware and software, and performed the
512 experiments. J.J., X.G. and Q.D. designed and coded the deep learning algorithms. Q. W. conducted the x-ray fluoroscopy
513 experiment. L.Y. and J.J. wrote the manuscript with contributions from all authors. All authors contributed to the scientific
514 discussion. L.Z. supervised the project.

515 **Competing interests**

516 The authors declare no competing interests.

Supplementary Information for

Autonomous environment-adaptive microrobot swarm navigation enabled by deep learning-based real-time distribution planning

4 Lidong Yang^{1,†}, Jialin Jiang^{1,†}, Xiaojie Gao², Qinglong Wang¹, Qi Dou^{2,5,6}, and Li Zhang^{1,3,4,5,6,*}

⁵ Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong.; ² Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong.; ³ Department of Surgery, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong.; ⁴ Chow Yuk Ho Technology Centre for Innovative Medicine, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong.; ⁵ T-Stone Robotics Institute, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong.; ⁶ Multi-Scale Medical Robotics Center (MRC), InnoHK, Hong Kong Science Park.; [†] Lidong Yang and Jialin Jiang contributed equally to this work.; * corresponding author: Li Zhang (lizhang@mae.cuhk.edu.hk.)

10 This PDF file includes:

- 11 Supplementary Notes S1 to S10**
- 12 Supplementary Figures. S1 to S18**
- 13 Supplementary Table S1**
- 14 Legends for Supplementary Videos S1 to S12**
- 15 Supplementary References**

16 Other supplementary materials for this manuscript include the following:

- 17 Supplementary Videos S1 to S12**

18 **Supplementary Note S1: Traversal-based optimization for the optimal swarm distribution planning**

19 Since unstructured environments cannot be described by an explicit model, the distribution planning method should be
20 model-free to the working environment. Moreover, the two parameters, i.e., the swarm shape ratio R_s and swarm orientation
21 angle α_s , are coupled with each other to determine the optimal swarm distribution. As a result, analytical methods are not
22 applicable for solving Eq. (1) to obtain the optimal swarm distribution on the next trajectory point, and thus the traversal-based
23 optimization should be used for this purpose. In this approach, to solve Eq. (1), a traversal process is executed for all the
24 candidate combinations of R_s and α_s . There are five candidates for R_s : 2, 3, 4, 5, 6, and α_s has ninety-one candidates, ranging
25 from -45° to 45° with an increasing step of 1° . The average distance between the swarm and the obstacles is computed
26 by averaging the distances of 360 boundary points of the swarm to the obstacle region. After the traversal process with
27 $w_1 = w_2 = 0.5$, the optimal swarm distribution can be obtained for the next trajectory point. However, as shown in Fig. 2(d),
28 the time consumed by the traversal-based optimization is too long to be used for real-time planning. Therefore, our solution is
29 to train deep neural networks (DNNs) to imitate the optimal solutions of the traversal-based optimization, which would have
30 both the real-time feature and the optimal planning performance. After learning with extensive environment morphologies, the
31 DL-based method would have sufficient robustness to unknown environments, as validated in Supplementary Video 1.

32 Supplementary Note S2: DNN training and application processes

33 For the environment-adaptive microrobot swarm navigation, a large dataset is required for the DNN training, but there is
34 no public dataset for direct use. On the other hand, manual creation and acquisition of a large real environment dataset are
35 difficult and time-consuming, due to which the DNNs cannot be conveniently re-trained for new environments. Therefore, we
36 propose to tailor a simulation engine to generate a large dataset for DNN training. To make this DL-based method applicable
37 for real environments, the DNN training and application processes are designed as follows (illustrated in Supplementary Fig.
38 [S1](#)).

39 In the training process, basic features of the real environment are manually extracted, after which the simulation engine
40 generates a large dataset composed of these basic features and their combinations. In this work, the targeted environment is
41 the vessel-like channel, whose basic features are: channel environments with varying diameters, branched channel environments,
42 and open-sided environments with curved boundaries. Then, the training using 90% data of the large dataset makes the DNN
43 have robustness to the differences/randomization between the training navigation scenarios and the real environment.

44 In the application process, the feedback image is not directly used for planning. Before the feedback image is sent to the
45 DNN, the swarm tracking and environment identification procedures (in Fig. 1b) should clearly segment the swarm and
46 obstacle regions and remove noise so that the DNN will process the images similar to those in the training process. This image
47 processing process can reduce the gap between the images in the simulation and the images captured by the imaging system.

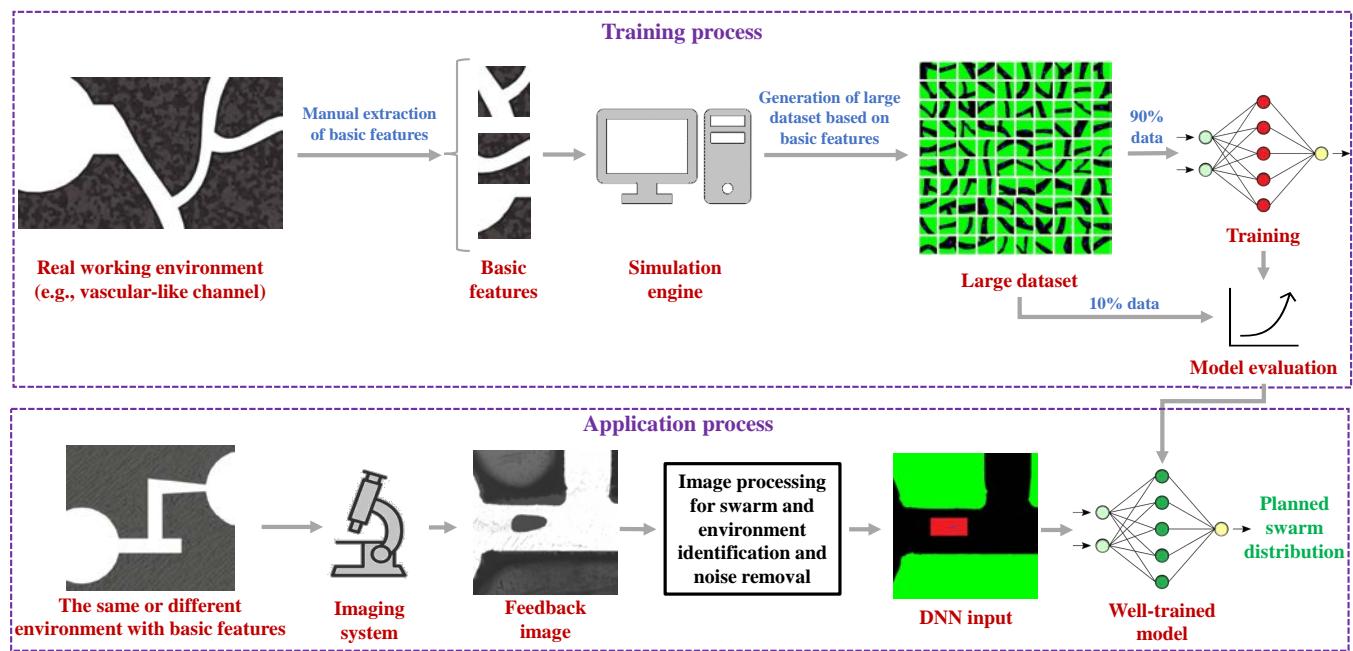


Fig. S1. Illustration of the training and application processes of the proposed DL-based swarm distribution planning method.

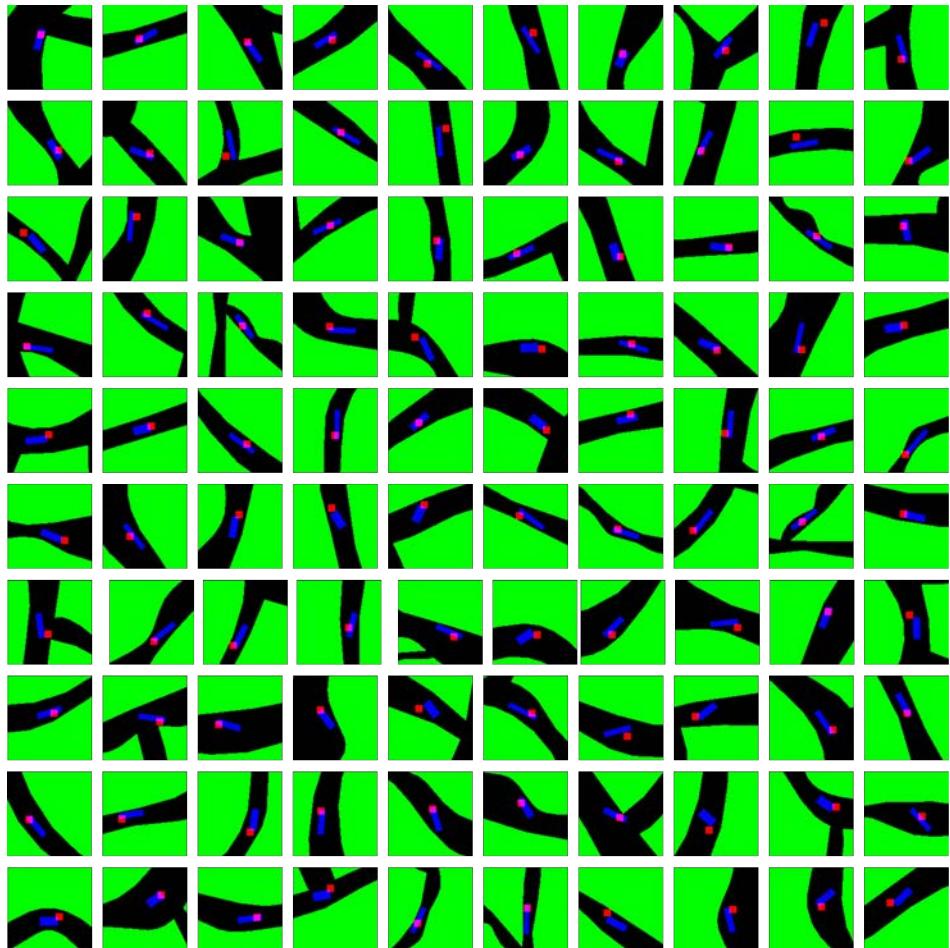


Fig. S2. Illustration of one hundred data scenarios used to train the DNN for swarm shape planing (SSP-DNN) and the DNN for swarm orientation planning (SOP-DNN). The black and green colors mark the available navigation space and the obstacle space, respectively. The blue and red rectangles are the current minimum swarm bounding box (MSBB) and the next trajectory point, respectively. The dataset for DNN training consists of 50 K such data with different working environment morphologies/swarm distribution states.

48 **Supplementary Note S3: Study on the influence of the dataset size on the planning accuracy**

49 We further conducted a comparison study, in which different dataset sizes were used to train the two DNNs. Six datasets
50 were formed, which contain 0%, 20%, 40%, 60%, 80%, and 100% of the total training dataset (90% of the 50 K training
51 scenarios). After ten training epochs, we then used the evaluation dataset (the remaining 10% of the 50 K training scenarios)
52 to test the accuracy of the two DNNs. The evaluation results are plotted in Supplementary Fig. S3. We can obtain that when
53 increasing the dataset size, the planning accuracy is also enhanced.

54 For the SSP-DNN, it converges faster than that of the SOP-DNN, and its accuracy increases from 85% to 90% when the
55 dataset proportion is extended from 20% to 80%. More training data could lead to saturation.

56 For the SOP-DNN, along with the increase of the dataset size, its planning accuracy increases. It is obtained that, the 80%
57 dataset proportion (36 K data) leads to an average angle error smaller than 5 degrees which is sufficient for the navigation
58 problem.

59 Therefore, 80% dataset proportion (36 K data) would result in sufficient planning accuracy for both the two DNNs.

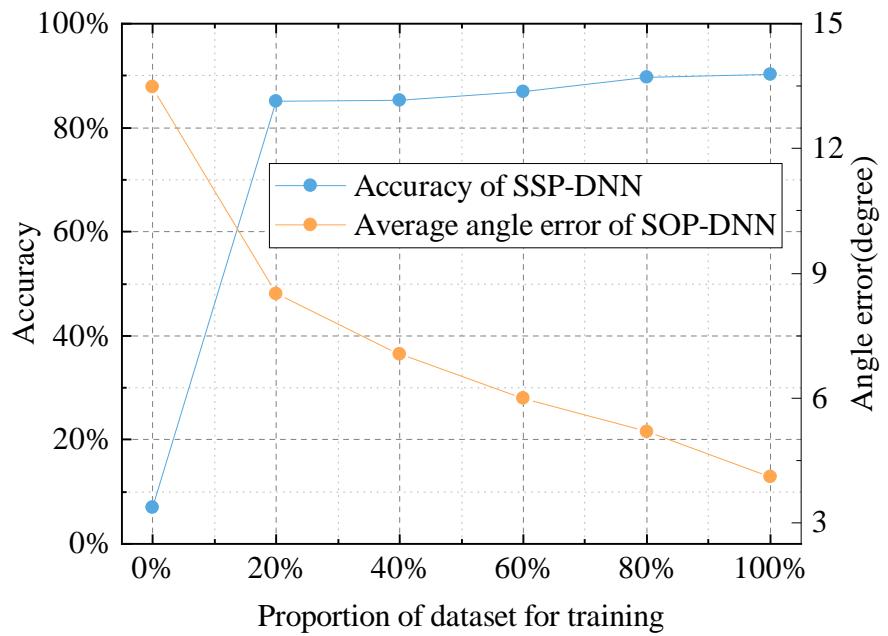


Fig. S3. Accuracy validation results when adopting different dataset sizes for training. The total training dataset contains 45 K navigation scenarios.

60 **Supplementary Note S4: Magnetic nanoparticles and experimental setup**

61 Fe₃O₄ nanoparticles with diameters around 400 nm are fabricated using the solvothermal method (1). A scanning electron
62 microscopy image is shown in Supplementary Fig. S4a. The 3D dynamic magnetic fields for actuation are provided by a 3-axis
63 Helmholtz electromagnetic coil setup, as illustrated in Supplementary Fig. S4b. The nanoparticle swarm and its working
64 environment are observed by an optical microscope and fed back to a computer via a high-speed (45 fps) camera mounted
65 on the microscope. Based on the proposed framework, control signals are generated by the programs coded by LabVIEW,
66 Matlab, and Python, which are then sent to the amplifiers (ADS 50/5 4-Q-DC, Maxon, Inc.) to generate desired currents
67 in coils. Computed by the field regulation principles in Methods, the oscillating field, rotating field, and 3D dynamic field
68 required for the ribbon-like swarm (RS), vortex-like swarm (VS), and spreading swarm (SS) are ready to be produced. For
69 experimental investigations, magnetic nanoparticle solutions (3 mg/mL) are added into an acrylic tank filled with deionized
70 (DI) water. For long-distance navigation, a home-designed 2D motorized stage (MTS25-Z8, Thorlabs, Inc.) is used to adjust
71 the field of view (FOV) of the microscope. Its control codes written by LabVIEW are also integrated into the overall program.

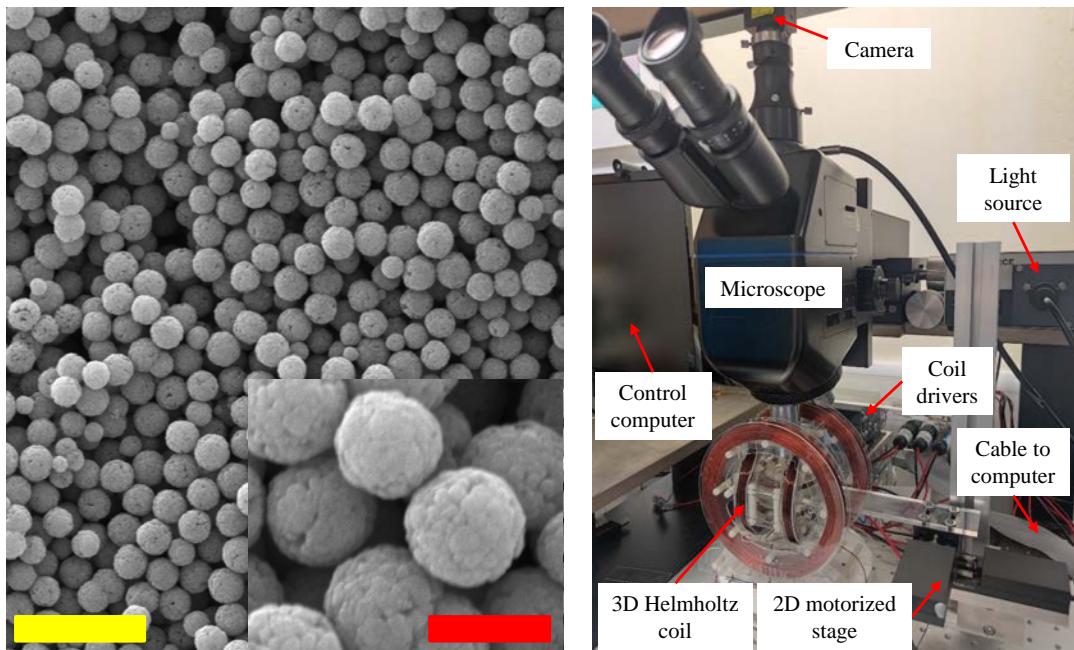


Fig. S4. a, The Fe₃O₄ nanoparticle with an average diameter of 400 nm. The yellow and red scale bars are 2 μm and 400 nm, respectively. **b,** The system setup used to generate 3D dynamic fields for actuation of the magnetic nanoparticle swarm.

72 **Supplementary Note S5: Unique challenges of autonomous navigation of magnetic nanoparticle swarms**

73 Compared with swarm navigation of traditional large-scale robots, the autonomous navigation of such microrobot swarms
74 has three unique challenges: (1) Unlike traditional robot swarms equipped with individual on-board sensors and actuators,
75 nanoparticles in the swarm cannot be individually controlled, resulting in the under-actuation nature and thus the limited
76 control degree-of-freedom. This fact brings challenges to the swarm distribution planning when navigating in unstructured
77 environments; (2) Unlike traditional robot swarms whose dynamics can be precisely modeled for motion control, the motion
78 of such microswarms is dominated by fluidic and particle-particle interaction forces that cannot be precisely modeled. This
79 property brings challenges to the automated swarm motion and reconfiguration control; (3) Unlike the working environments of
80 traditional robots that have public datasets for neural network training, there is no such dataset for microrobot swarms, which
81 makes it challenging to realize the robust DL-based planning for the environment-adaptive microrobot swarm navigation that
82 requires a large dataset.

Table S1. Pros and cons of the three configurations of the magnetic nanoparticle swarm

Configuration	Pros	Cons	Application scenario
Ribbon-like swarm (RS)	Fast generation process; Fast deformation process; Resilient to external disturbance.	Slower motion speed than VS.	Adaptive navigation; Cargo unloading utilizing its cyclic fluid flow.
Vortex-like swarm (VS)	Faster motion speed than RS.	Slow deformation process; Fragile to external disturbance.	Cargo loading and transport utilizing its inward fluidic force.
Spreading swarm (SS)	Controllable spreading area	No gathering force for collective navigation	Realizing the desired particle distribution area/density in targeted delivery/therapy

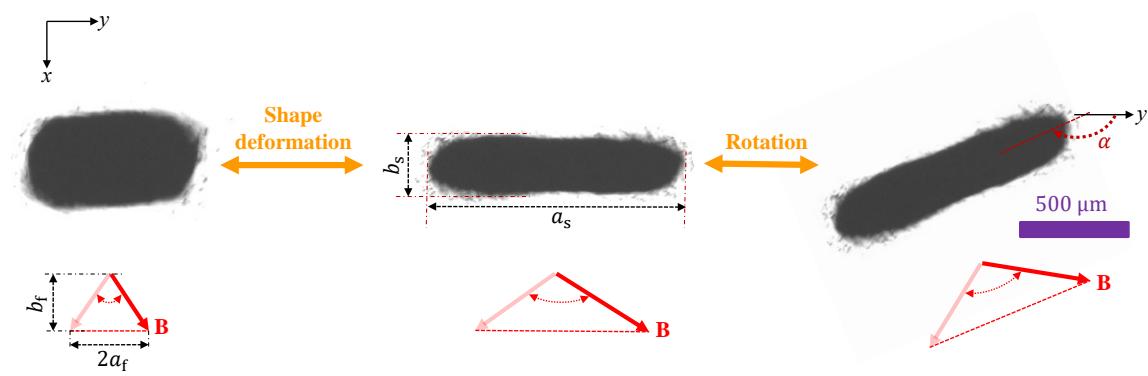


Fig. S5. Experimental demonstration of the swarm pattern deformation and rotation of the ribbon-like swarm (RS). Swarm shape ratio R_s and field ratio R_f are defined as $R_s = \frac{a_s}{b_s}$ and $R_f = \frac{a_f}{b_f}$, respectively. Swarm orientation angle α represents the angle between the long-axis of the swarm and y axis of the global coordinate frame. As shown in the figure, R_s increases when R_f is tuned larger. The swarm rotation is controlled by rotating the oscillating direction of the actuation magnetic field \mathbf{B} .

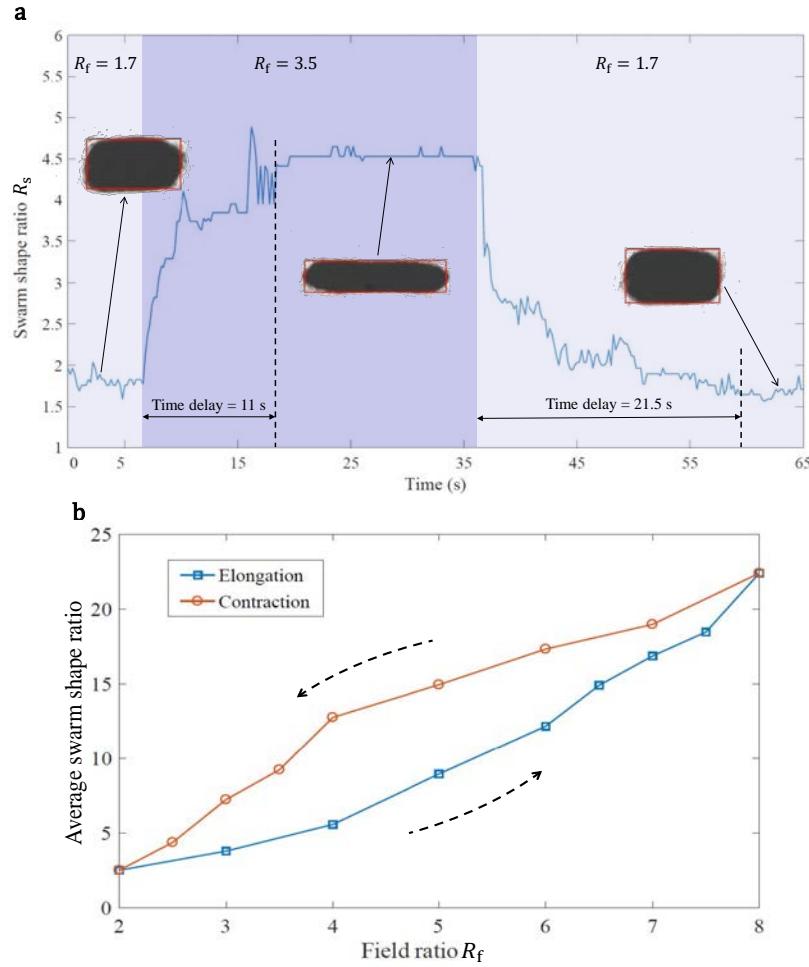


Fig. S6. Experimental characterization of the deformation properties for the ribbon-like swarm (RS). **a**, Characterization of the transient deformation, including the elongation and contraction processes. At first, the RS has a stable shape under actuation with a field ratio of 1.7. Then, the field ratio R_f is changed to 3.5 to trigger the swarm elongation. After the elongation process is finished, R_f is changed back to 1.7 to trigger the swarm contraction process. The characterization result shows that the deformation process has long time delay (> 10 s), and the elongation and contraction processes have different time delays. Different swarm sizes and field inputs will also have varying time delays. **b**, Characterization of the steady-state relationship between the input field ratio and the output swarm shape ratio. At first, the RS has a stable shape under field ratio $R_f = 2$. Then, the field ratio is increased by a step of 1 or 0.5 after the swarm reaches a steady state. The elongation process is finished till $R_f = 8$, after which R_f is decreased to characterize the steady-state contraction process. Interestingly, the elongation and contraction deformation processes form a hysteresis-like loop. Every average swarm shape ratio in the figure is calculated from five experiments.

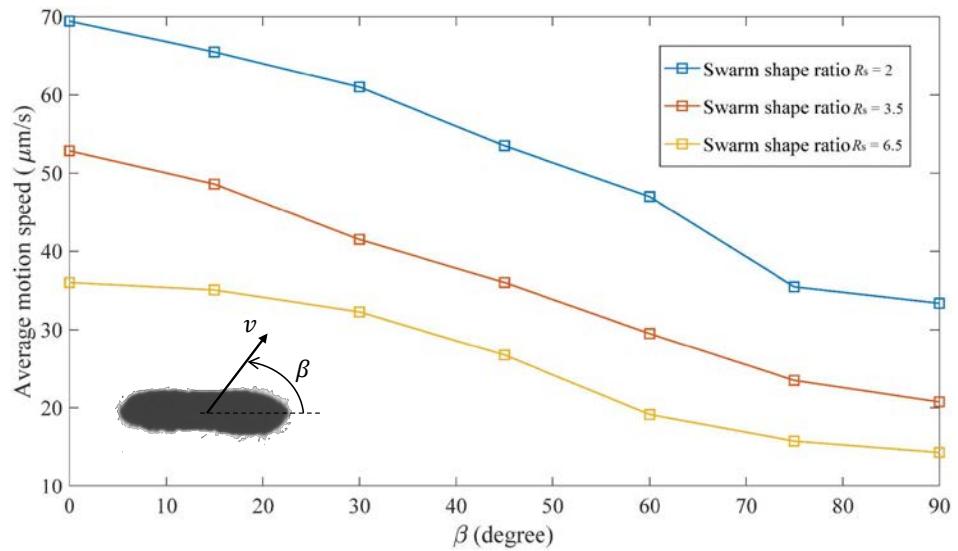


Fig. S7. Experimental characterization of the motion properties for the ribbon-like swarm (RS). β represents the intersection angle between the swarm orientation and the translational motion direction. v is the translational motion speed. The motion speed is positively correlated to the field pitch angle, and the field pitch angle is fixed at 3.5° for this experiment. Results show that swarms with larger shape ratios would have slower motion speed with the same β , and swarms with the same shape ratios would have slower motion speed if with a larger β .

83 **Supplementary Note S6: Common failure cases in manual navigation**

84 As the magnetic nanoparticle swarm contains millions of nanoparticles that loosely interact with each other, there would be
85 navigation failure, especially in complex environments, if inappropriate manual control actions happen. Herein, we illustrate
86 five common failure cases in manual navigation, as shown in Fig. S8.

87 In the first case (Fig. S8(a)), the RS has a relatively large shape ratio around 6. Meanwhile, the field orientation angle
88 α_f is suddenly changed by a large value of 90° . Because the nanoparticle reorganization in the swarm cannot finish during
89 such a sudden and large rotation motion, the swarm splits to several small swarms. As a result, the swarm stability, thus the
90 collective navigation, is disrupted.

91 In the second case (Fig. S8(b)) for the RS, wrong rotation direction control happens, and the swarm get contact with the
92 obstacles. Owing to the strong interaction force between the swarm and the channel wall, the swarm is stuck on the contact
93 position, causing failure of the navigation. Such undesired collisions with obstacles could also happen if the operator performs
94 wrong translational motion control.

95 In the third case (Fig. S8(c)) for the RS, the field pitch angle γ_f is suddenly set a too large value (e.g., 10°), so that the
96 swarm pattern becomes unstable. Besides, the low-frequency human control makes the correct motion control difficult. The
97 two factors cause the swarm to lose particles, and the swarm collides with the channel wall.

98 In the fourth case (Fig. S8(d)) for the RS, the swarm shape is not appropriately controlled. To pass through the narrow
99 channel, the swarm should elongate its shape until there is no collision with the channel wall. Without doing so, the swarm
100 with a too small shape ratio is stuck in the channel, and a large portion of nanoparticles is lost.

101 In the fifth case (Fig. S8(e)) for the VS, the swarm shape is also not appropriately controlled. To pass through the narrow
102 channel, the swarm should elongate its shape until there is no collision with the channel wall. Because the VS relies on the
103 vortex fluidic force to assemble, it is very fragile to such collision. As shown, the VS splits to several small groups, which fails
104 the collective navigation.

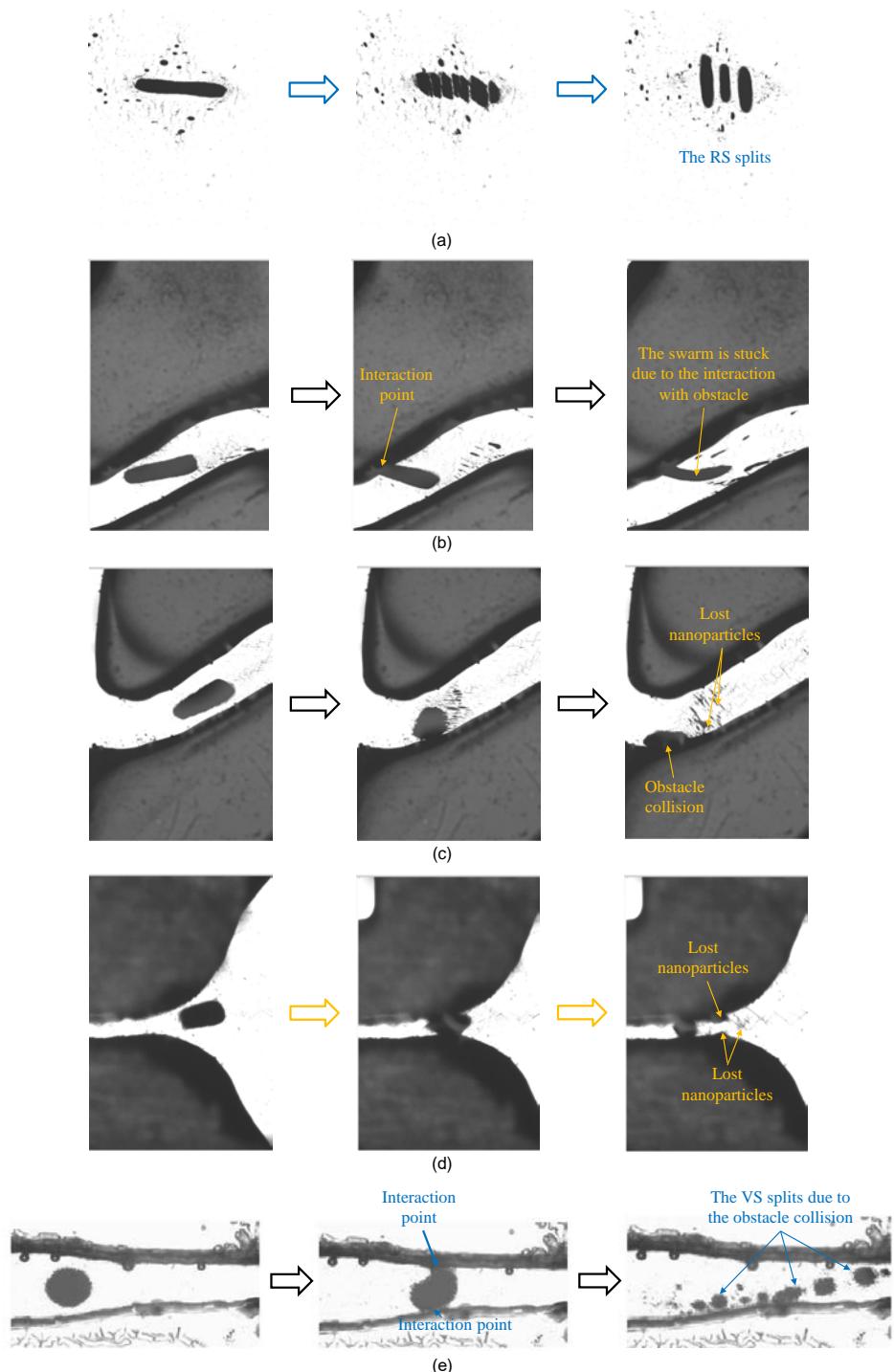


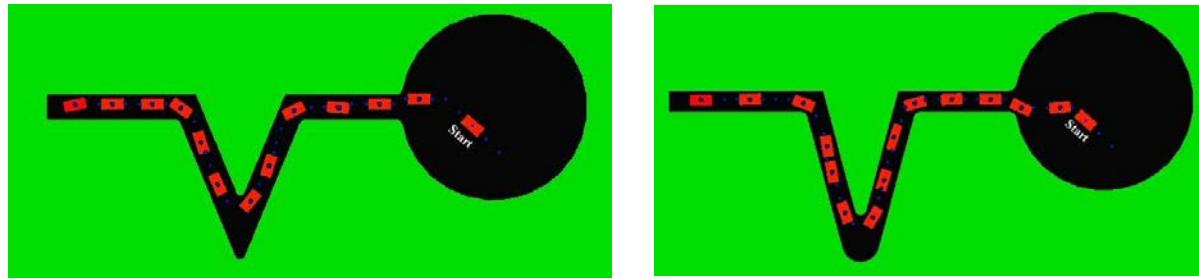
Fig. S8. Common failure cases in manual navigation due to (a) excessive rotation control; (b) wrong rotation direction control; (c) wrong field pitch angle control and low-frequency motion direction control; and (d) wrong swarm shape control; (e) wrong swarm shape control for the VS.

105 **Supplementary Note S7: Navigation in curved environments with sharp turn and narrow channel using
106 autonomy Level 3**

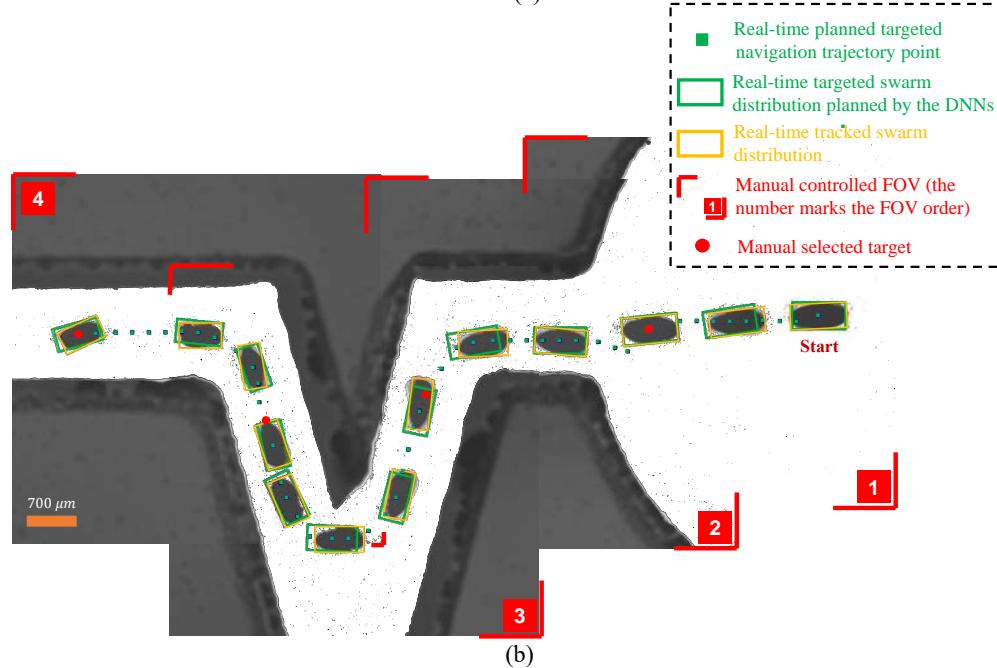
107 Here, we validate the navigation in curved environments with sharp turns and curved narrow channel environments. Since
108 such environments are still composed of these features in the training dataset, i.e., channel environments with varying diameters,
109 branched channel environments, open-sided environments with curved boundaries, the trained DNN would also work.

110 For evaluation, we further conducted navigation simulations and experiments in highly curved space containing sharp
111 turns. As illustrated in Supplementary Fig. S9(a), we first test the planning performance with 135° and 150° sharp turns via
112 simulation, and the results indicate that the trained model works well. For experimental validation, we fabricated the curved
113 environment with a 150° sharp turn via laser cutting. The navigation experiments are then conducted using autonomy Level 3.
114 To validate the robustness of the trained model to different nanoparticle amounts in the swarm, we conducted two experiments
115 with $0.75 \mu\text{g}$ and $2.0 \mu\text{g}$ nanoparticles. The experimental results are shown in Supplementary Fig. S9(b) and (c), respectively,
116 indicating that the swarm distributions can be correctly planned during navigation, and the swarms successfully reached the
117 manually given targeted positions. The navigation processes are included in Supplementary Video 7. Moreover, the results also
118 show that the trained model has good robustness to different swarm sizes for navigation in such highly curved environments.

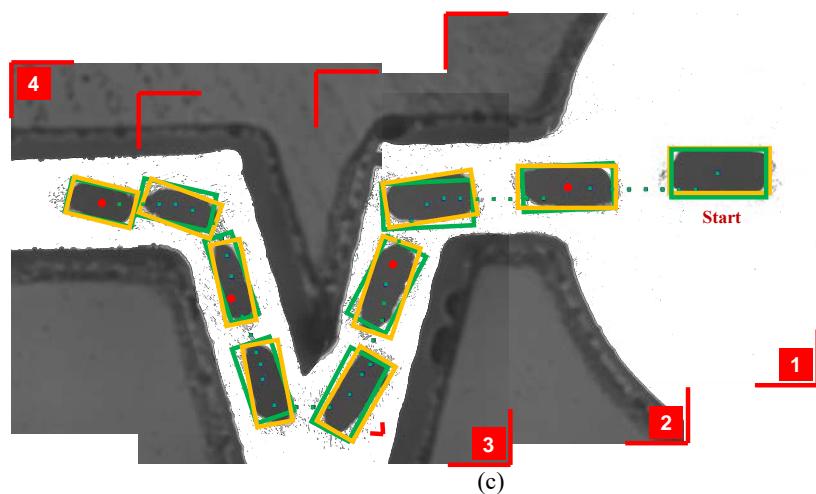
119 We also conducted navigation simulations and experiments in curved narrow channel environments, whose results are
120 illustrated in Supplementary Fig. S10. In this navigation environment, to pass the curved narrow channel, the swarm shape
121 should be dramatically controlled together with the swarm direction. The navigation results indicates that the proposed
122 DL-based swarm distribution planning method can correctly output swarm distributions in real time for such a challenging
123 case. The navigation processes in simulations and experiments are included in Supplementary Video 7.



(a)

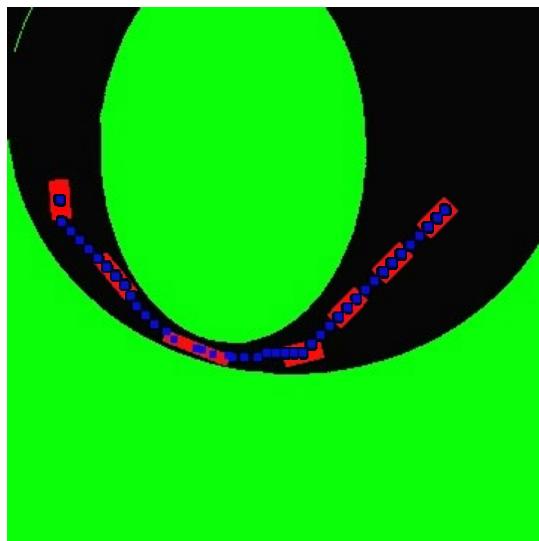


(b)

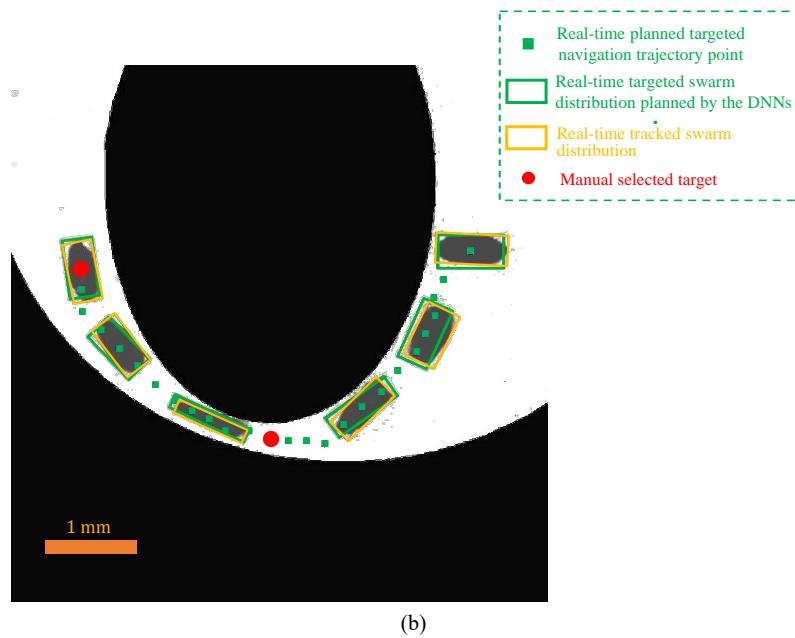


(c)

Fig. S9. Demonstration of Autonomy Level 3 in highly curved space with sharp turns. (a) Simulation results for environments with 135° and 150° sharp turns. (b) The experimental navigation result for an RS containing 0.75 μg nanoparticles. (c) The experimental navigation result for an RS containing 2.0 μg nanoparticles.



(a)



(b)

Fig. S10. Demonstration of Autonomy Level 3 in curved narrow channel environment. (a) Simulation results. (b) The experimental navigation result for an RS containing 0.75 μg nanoparticles.

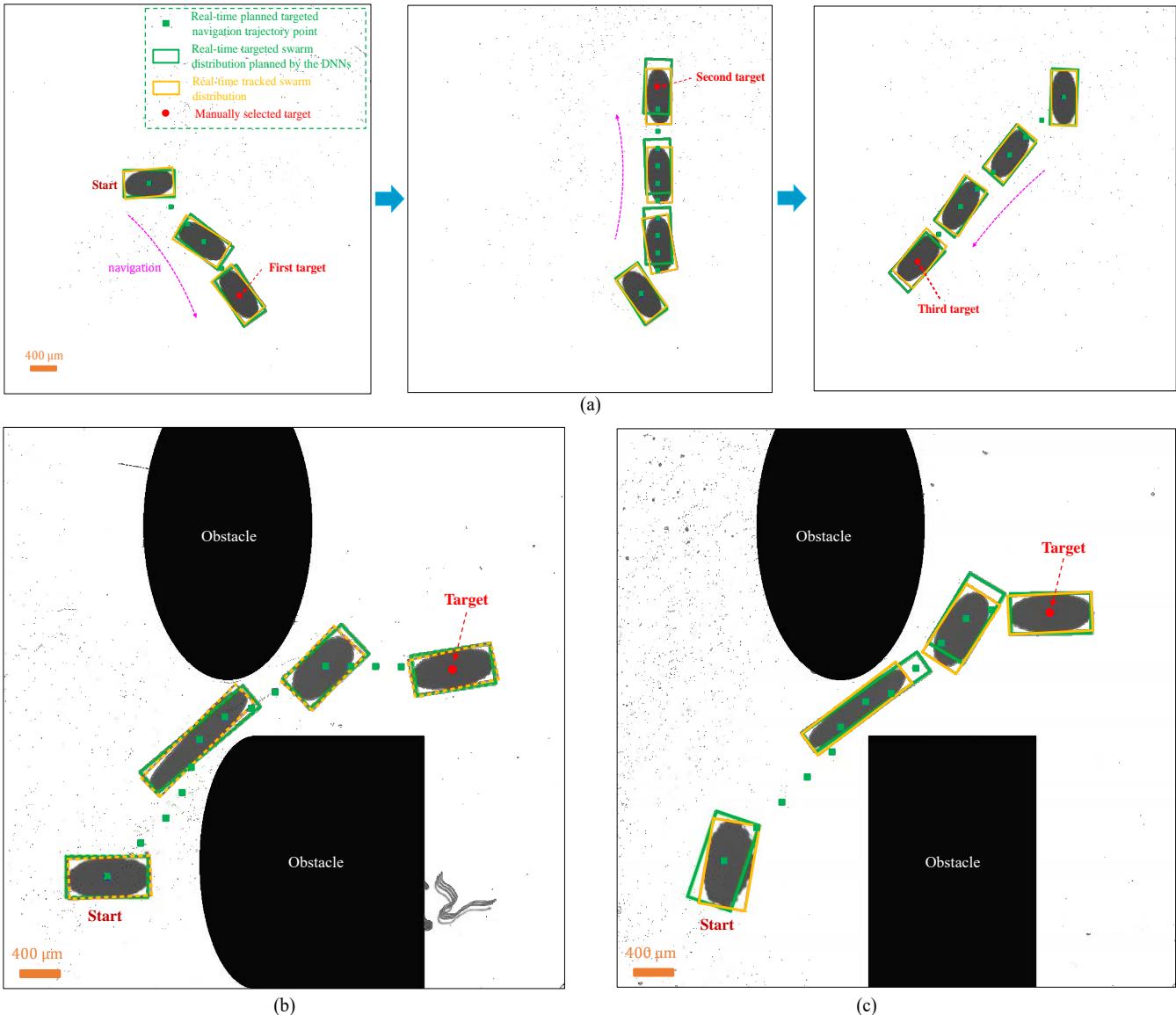


Fig. S11. Transfer of the methods to the elliptical vortex-like nanoparticle swarm. Three experiments using autonomy Level 3 are conducted to test the navigation performances in different environments, including (a) the blank space, (b) an environment with round-corner obstacles, and (c) an environment with sharp-corner obstacles. Navigation processes are included in Supplementary Video 11. The results show that, although the vortex-like swarm has slower deformation speed than that of the ribbon-like swarm, it can still autonomously and correctly adjust its motion and distribution according to environments, by which the manually given targets were successfully reached.

124 **Supplementary Note S8: Method validation under ultrasound imaging and x-ray fluoroscopy**

125 For applications using medical imaging modalities, the quality of the feedback information would be much worse. In order to
126 show the performance of the DL-based method in such cases, we further implemented ultrasound imaging and x-ray fluoroscopy
127 for validation purposes.

128 Regarding the ultrasound imaging, the developed system setup is shown in Supplementary Fig. S12(a). An EcoFlex tank
129 is fabricated for containing the magnetic nanoparticle swarm and water. An ultrasound probe (Model: 16HL7) is used for
130 imaging the swarm and the tank with a depth of 40 mm. An ultrasound processing system (Model:uSmart 3300 NexGen,
131 Terason Inc.) then computes and generates the ultrasound images. We can see that there is much noise in the raw feedback
132 image, which should be removed by the image processing procedure as shown in Supplementary Fig. S12(b). Also, the swarm
133 and obstacle regions are segmented in this procedure. After that, the processed image is used for the trajectory planning and
134 DL-based swarm distribution planning to make the swarm reach two targets, whose results are illustrated in Supplementary
135 Video 12 and Supplementary Fig. S12(c). The results indicate that, although the segmented obstacles have irregular shapes
136 due to ultrasound imaging, the DL-based method still works well thanks to the robustness obtained in the training process.

137 Regarding the x-ray fluoroscopy, we chose a human placenta for the navigation environment because the placenta contains
138 abundant blood vessels with branches that are suitable for justification of the DL-based method. The placenta was obtained
139 under the permission of The Joint Chinese University of Hong Kong-New Territories East Cluster Clinical Research Ethics
140 Committee (CREC Ref. No. 2020.384). The guideline of the placenta collection under the ethics approval is shown in the end
141 of this Note. In this experiment, at first, a swarm of nanoparticles is injected into a vessel of a human placenta and gathered
142 by a permanent magnet (Fig. S13(a)). Then, an x-ray fluoroscope (Model: Aritis Zeego, Siemens Inc) shown in Fig. S13(b) is
143 adopted. The imaging result in Fig. S13(c) shows that the swarm and the vessels can be observed. After the same swarm and
144 environment identification procedure for the ultrasound imaging, the processed image is used for validation of the trajectory
145 and swarm distribution planning method. Results in Supplementary Video 12 and Fig. S13(e)(f) indicate that the DL-based
146 method can correctly lead the swarm to navigate in the vessel to reach two targets.

147 The two experiments preliminarily validate that the DL-based method is applicable for ultrasound imaging and x-ray
148 fluoroscopy. Currently, since we cannot transmit the ultrasound image to the control system in real time and do not have the
149 permission to integrate the entire system to the x-ray fluoroscope, the autonomous adaptive navigation guided by ultrasound or
150 x-ray remains an important future work. Moreover, to deal with more complex images obtained by medical imaging modalities,
151 DL-based segmentation methods (2, 3) should also be further studied for accurate and reliable environment segmentation.

152 **Guideline of the placenta collection under ethics approval**

153 Before recruiting tissue donors and collecting the placenta, ethics approval from The Joint Chinese University of Hong Kong
154 – New Territories East Cluster Clinical Research Ethics Committee (The Joint CUHK-NTEC CREC) is required. The Joint
155 CUHK-NTEC CREC was established by The Chinese University of Hong Kong (CUHK) and Hong Kong Hospital Authority
156 New Territories East Cluster (NTEC) in accordance with its terms of reference for overseeing research involving human subjects
157 undertaken by and/or conducted in the premises owned, managed and/or controlled by CUHK and/or NTEC, and/or involving
158 patients and/or staff thereof as human subjects in such clinical studies. The research project and study protocol are reviewed
159 by the Joint CUHK-NTEC CREC before approval.

160 In our project, we obtained the ethics approval from The Joint CUHK-NTEC CREC (Ref. No. 2020.384). The details of
161 the placenta collection are as follows:

162 Women fulfilling the inclusion and exclusion criteria (listed below) were recruited from the Department of Obstetrics and
163 Gynaecology, Prince of Wales Hospital. All potential subjects were given a detailed explanation and their permission was
164 obtained before they were recruited into the study. A written consent form was signed by the participants. The participants
165 can withdraw from the research without any repercussions. The placenta was collected after donor's delivery only with their
166 signed informed consent form. On the day of labour, the placenta was collected by the doctors and shipped on dry ice to the
167 laboratory for experiments immediately after birth.

168 **Inclusion Criteria**

169 i Healthy pregnant women at 20-45 years of age of any ethnic origin, giving childbirth with natural delivery or Caesarean

- 170 sections after 37-42 weeks of gestation;
- 171 ii Singleton pregnancy;
- 172 iii Healthy as determined by laboratory results, physical exam and medical history;
- 173 iv Participant able to give voluntary, written, informed consent to participate in the study.

174 Exclusion Criteria

- 175 i Abnormal prenatal development (e.g. intrauterine growth restriction);
- 176 ii Early preterm birth < 37 weeks;
- 177 iii Verbal confirmation of hypercholesterolemia;
- 178 iv Family history of stroke or vascular disease;
- 179 v Type I or Type II diabetes and gestational diabetes;
- 180 vi Cancer, except skin cancers completely excised with no chemotherapy or radiation with a follow up that is negative;
- 181 vii Clinically significant abnormal laboratory results at screening;
- 182 viii Any other active or unstable medical condition;
- 183 ix History of liver disease;
- 184 x History of hypertension (including pre-clampsia).

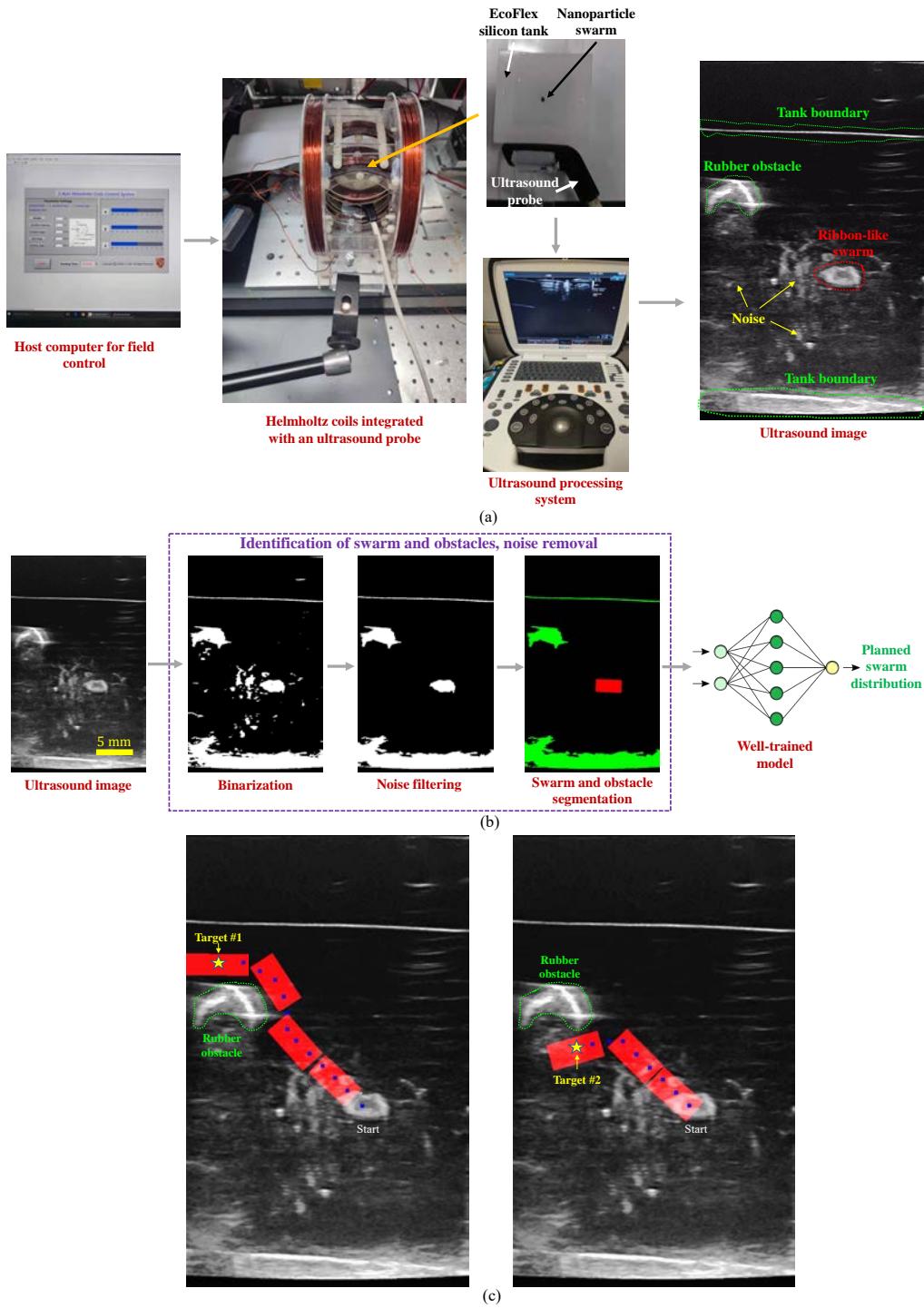


Fig. S12. Validation of the DL-based swarm distribution planning method under ultrasound imaging. (a) The system setup, in which an ultrasound processing system (Model:uSmart 3300 NexGen, Terason Inc.) is adopted. (b) The swarm and environment identification procedures before sending the image to the DNN. (c) The trajectory planning and swarm distribution planning results for reaching Target #1 and Target #2.

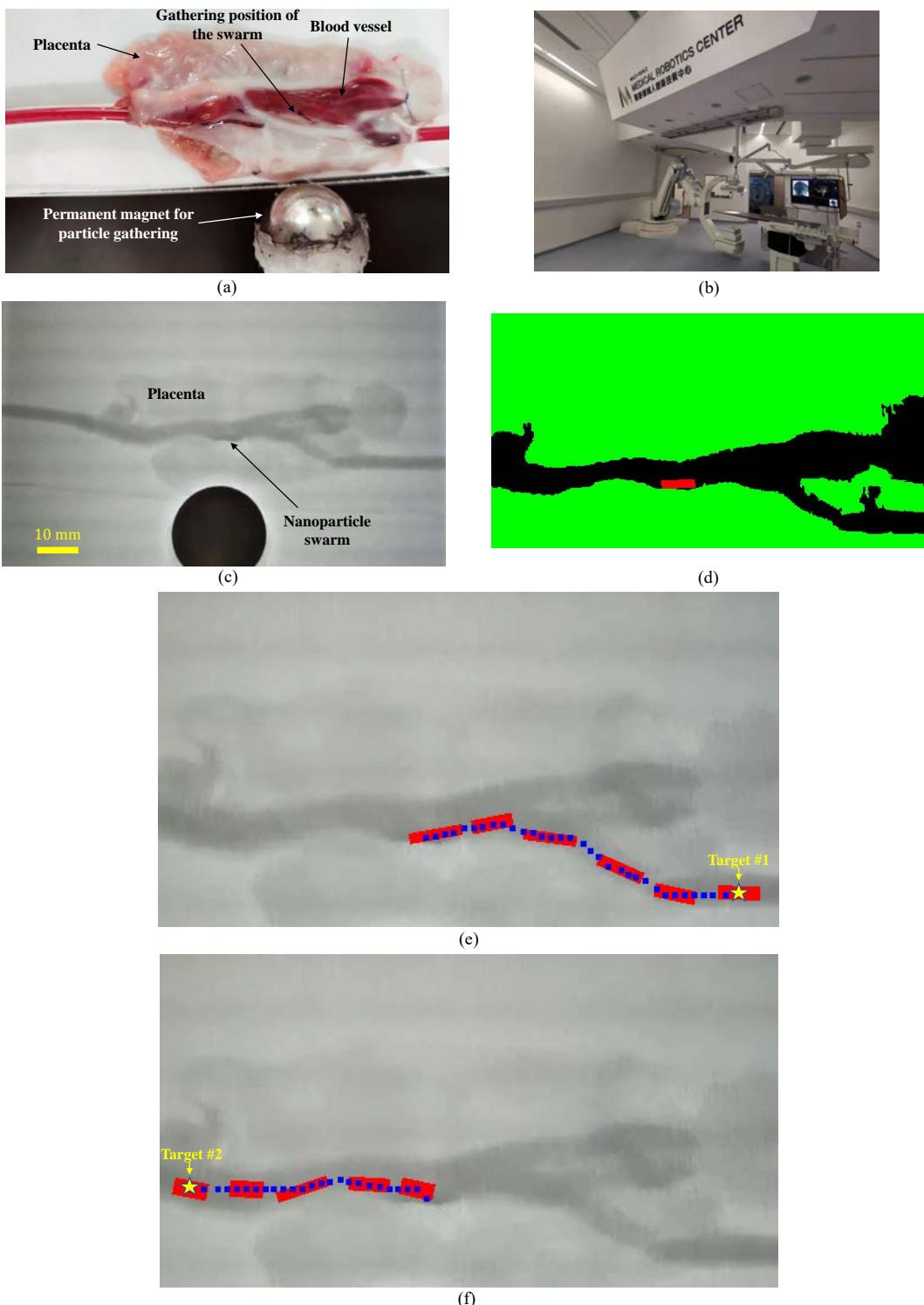


Fig. S13. Validation of the DL-based swarm distribution planning method under x-ray fluoroscopy. (a) The system setup. (b) The x-ray fluoroscope (Model: Aritis Zeego, Siemens Inc) adopted in this work. (c) The image captured by the x-ray fluoroscope. (d) The swarm and environment identification results. (e)(f) The trajectory planning and swarm distribution planning results for reaching Target #1 and Target #2, respectively.

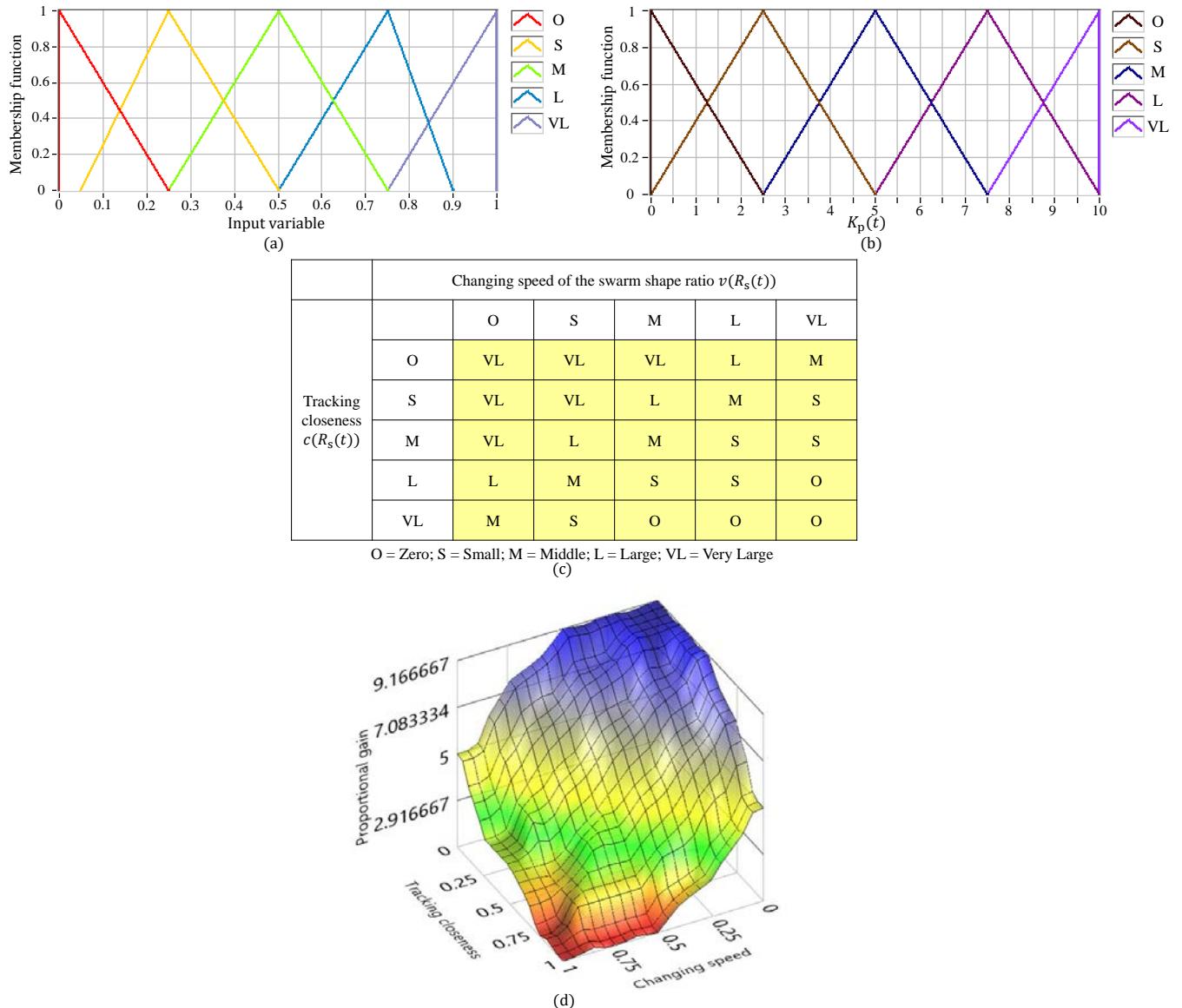


Fig. S14. The fuzzy logic controller for tuning $K_p(t)$ in the swarm shape control. **a**, The membership function for input variables $v(R_s(t))$ and $c(R_s(t))$. **b**, The membership function for the output $K_p(t)$. **c**, The fuzzy logic rules for the controller. **d**, The resulting control relationship for tuning $K_p(t)$ based on the current swarm states $v(R_s(t))$ and $c(R_s(t))$.

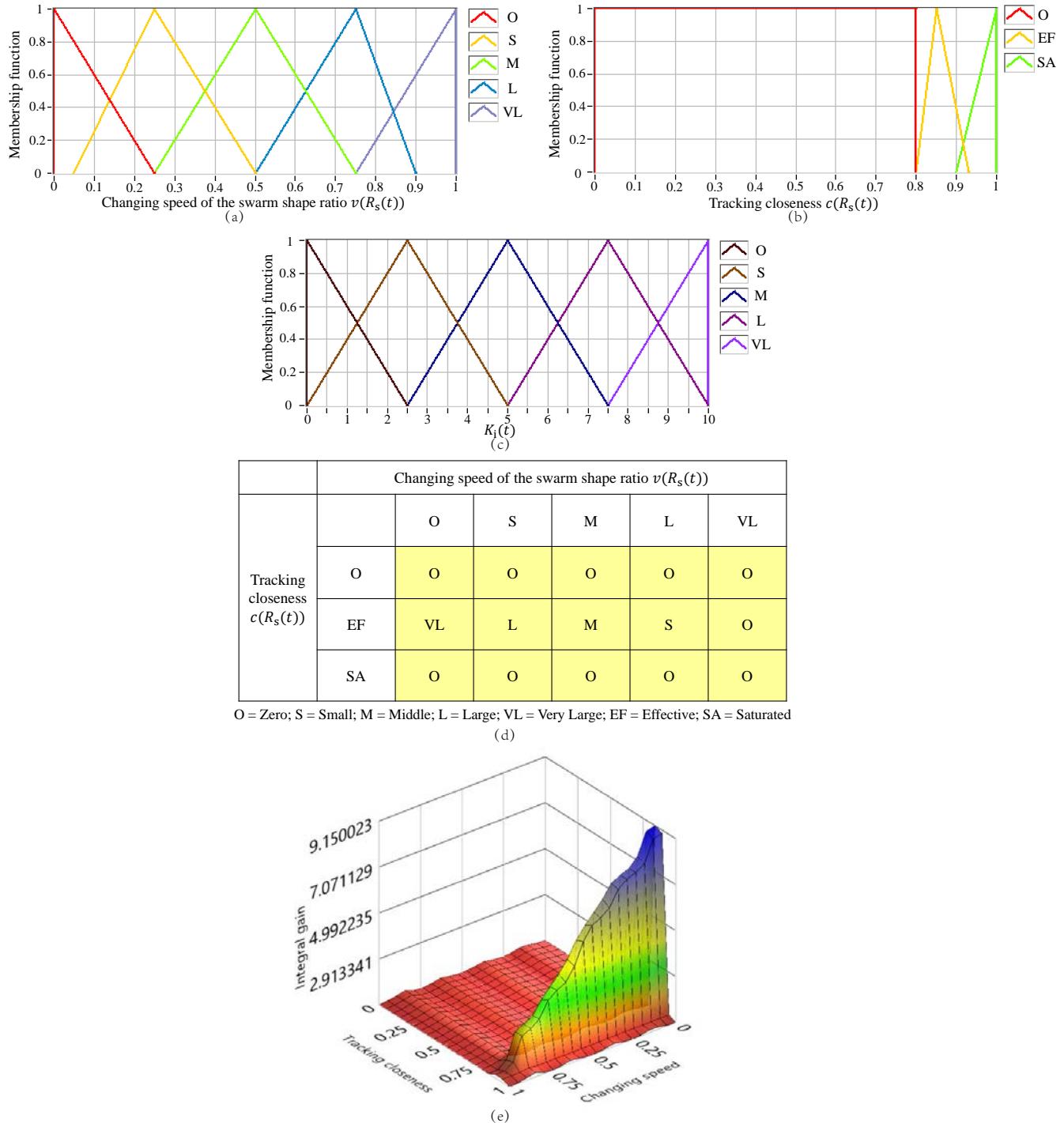


Fig. S15. The fuzzy logic controller for tuning $K_i(t)$ in the swarm shape control. **a**, The membership function for input variable $v(R_s(t))$. **b**, The membership function for input variable $c(R_s(t))$. **c**, The membership function for the output $K_i(t)$. **d**, The fuzzy logic rules for the controller. **e**, The resulting control relationship for tuning $K_i(t)$ based on the current swarm states $v(R_s(t))$ and $c(R_s(t))$.

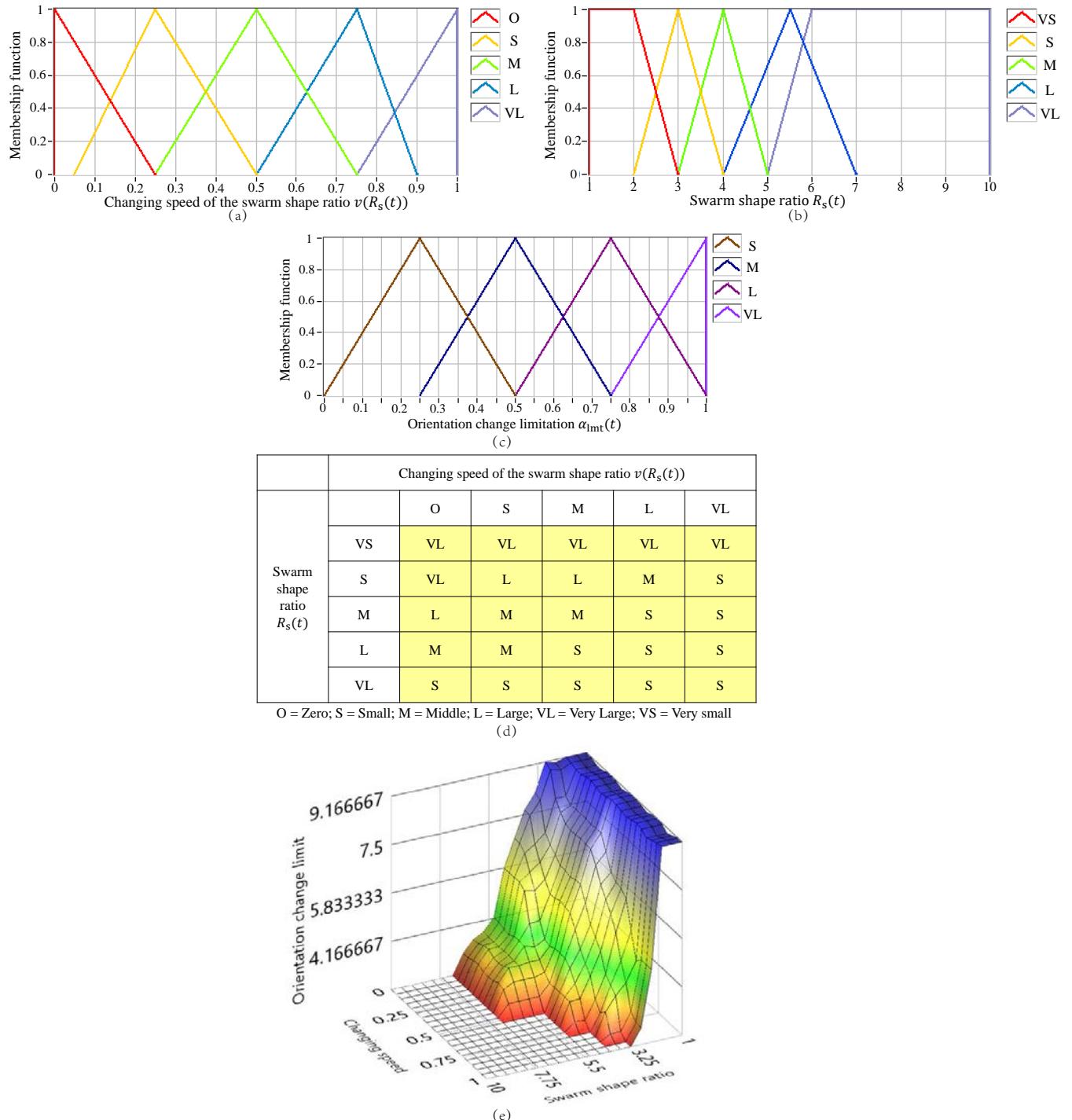


Fig. S16. The fuzzy logic controller for tuning $\alpha_{lmt}(t)$ in the swarm orientation control. **a**, The membership function for input variable $v(R_s(t))$. **b**, The membership function for input variable $R_s(t)$. **c**, The membership function for the output $\alpha_{lmt}(t)$. **d**, The fuzzy logic rules for the controller. **e**, The resulting control relationship for tuning $\alpha_{lmt}(t)$ based on the current swarm states $v(R_s(t))$ and $R_s(t)$.

185 **Supplementary Note S9: Performance comparisons of trajectory planning algorithms**

186 For collective microrobot swarm navigation, there are four requirements on the trajectory planning method: (1) the planning
187 method should be applicable for unstructured environments; (2) the planned trajectory points should have a near-uniform
188 distribution, because a constant distance between the swarm and the next trajectory point is used in the DNN training; (3)
189 a distance between the obstacles and the swarm trajectory points should be ensured to let the swarm pass through; (4) the
190 planning time should be sufficiently short for real-time use.

191 To check if traditional methods fulfill the specific requirements for the microrobot swarm navigation, we further conducted a
192 comparison study, implementing the traditional A*, traditional RRT*, and the method designed in this work. The comparison
193 results are illustrated in Fig. S17, which show that the A* algorithm has too long execution time (3.5 s) to be used in real
194 time, because it explores all the neighbor positions. In addition, as A* does not consider a distance to the obstacle, it does
195 not fulfill the third requirement either. Regarding the RRT*, it explores the environment with arbitrary exploration steps to
196 ensure a fast speed for real-time use. However, in turn, the obtained trajectory distribution is highly nonuniform, making it
197 against the second requirement. Besides, it does not consider the distance to the obstacle either. By contrast, the planning
198 method designed in this work satisfies all the four requirements.

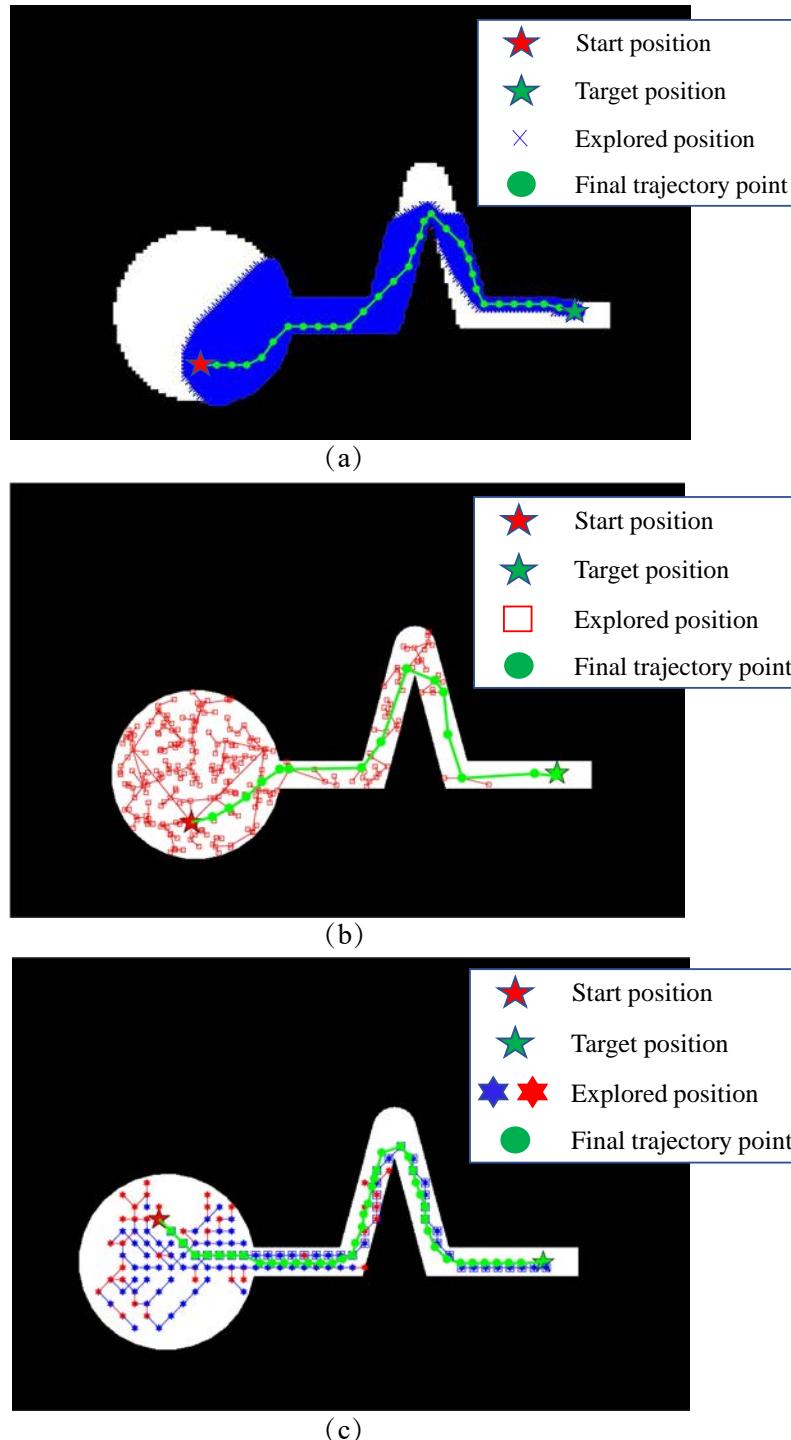


Fig. S17. Comparisons of three trajectory planning methods. (a) Traditional A* method, which takes 3.5 s. (b) Traditional RRT* method, which takes 0.55 s. (c) The designed method in this work, which takes 0.36 s.

199 **Supplementary Note S10: Detailed algorithms for the autonomous trajectory planning**

200 This part gives the algorithms used in **Methods: Autonomous trajectory planning**, including $UniExpand(\mathbf{T}, \mathcal{E}, P_g, \mathbf{v}_{\text{lead}}, D_{\text{sc}})$,
 201 $RandExpand(\mathbf{T}, \mathcal{E}, P_g, D_{\text{sc}})$, $FindNeighbor(\mathbf{v}_i, \mathbf{T}, R_{\text{near}})$, $ChooseParent(\mathbf{v}_i, \mathbf{T}, \mathbf{I}_{\text{near}})$, $ReWire(\mathbf{v}_i, \mathbf{T}, \mathbf{I}_{\text{near}}(k))$, $ExtractTra(\mathbf{T}, \mathbf{v}_{\text{reach}}, P_g)$,
 202 and $SafeDist(\mathbf{T}_{\text{trajecotry}}, \mathcal{E}, D_{\text{sf}})$.

203 To fulfill the four specific requirements for microrobot swarm navigation, we made three major modifications: (1) to make the
 204 planned trajectory points have a near-uniform distribution, we designed a tree expanding algorithm- $UniExpand(\mathbf{T}, \mathcal{E}, P_g, \mathbf{v}_{\text{lead}}, D_{\text{sc}})$
 205 that can expanding the tree uniformly; (2) to ensure a distance between the obstacles and the trajectory points, we designed
 206 the algorithm $SafeDist(\mathbf{T}_{\text{trajecotry}}, \mathcal{E}, D_{\text{sf}})$ to optimize the positions of the preliminarily obtained trajectory points; (e) to
 207 make the planning time sufficiently short for real-time use, unlike the RRT* that has a random exploration direction, in our
 208 algorithm, the exploration direction with the smallest cost defined by Equation (13) is selected for tree expanding.

Algorithm 1 The $UniExpand(\mathbf{T}, \mathcal{E}, P_g, \mathbf{v}_{\text{lead}}, D_{\text{sc}})$ algorithm. $Collision(\mathbf{a}, \mathcal{E})$ returns 0 if there is no collision between point \mathbf{a} and the environment \mathcal{E} . $Mod(a, b)$ returns the remainder after division of a by b . $Length(\mathbf{a})$ returns the element number of \mathbf{a} .

```

1: Input:  $\mathbf{T}, \mathcal{E}, P_g, \mathbf{v}_{\text{lead}}, D_{\text{sc}}$ .
2:  $\mathbf{V}_{\text{temp}} = []$ ;  $\mathbf{Flag} = []$ 
3:  $d_{\text{pre}} = Atan2d(\mathbf{v}_{\text{lead}}.y - \mathbf{T}(\text{lead} - 1).y, \mathbf{v}_{\text{lead}}.x - \mathbf{T}(\text{lead} - 1).x) / 45$ 
4: for  $k = (d_{\text{pre}} - 2)$  to  $(d_{\text{pre}} + 2)$  do
5:   if  $Mod(k, 2) \neq 0$  then
6:      $\mathbf{v}_{\text{can}}.x = \mathbf{v}_{\text{lead}}.x + \sqrt{2} \cdot D_{\text{sc}} \cdot \cos(k \cdot 45)$ 
7:      $\mathbf{v}_{\text{can}}.y = \mathbf{v}_{\text{lead}}.y + \sqrt{2} \cdot D_{\text{sc}} \cdot \sin(k \cdot 45)$ 
8:   else
9:      $\mathbf{v}_{\text{can}}.x = \mathbf{v}_{\text{lead}}.x + D_{\text{sc}} \cdot \cos(k \cdot 45)$ 
10:     $\mathbf{v}_{\text{can}}.y = \mathbf{v}_{\text{lead}}.y + D_{\text{sc}} \cdot \sin(k \cdot 45)$ 
11:   end if
12:   if  $Collision(\mathbf{v}_{\text{can}}, \mathcal{E}) == 0$  then
13:      $\mathbf{v}_{\text{can}}.cost = \sqrt{(\mathbf{v}_{\text{can}}.x - x_g)^2 + (\mathbf{v}_{\text{can}}.y - y_g)^2}$ 
14:     if  $\mathbf{v}_{\text{can}}.cost < \mathbf{v}_{\text{lead}}.cost$  then
15:        $\mathbf{v}_{\text{lead}}.cost = \mathbf{v}_{\text{can}}.cost$ 
16:        $\mathbf{V}_{\text{temp}} = [\mathbf{V}_{\text{temp}}, \mathbf{v}_{\text{can}}]$ ;
17:       for  $n = 1$  to  $Length(\mathbf{T}) - 1$  do
18:         if  $\sqrt{(\mathbf{v}_{\text{can}}.x - \mathbf{T}(n).x)^2 + (\mathbf{v}_{\text{can}}.y - \mathbf{T}(n).y)^2} < D_{\text{sc}}$  then
19:            $\mathbf{Flag}(Length(\mathbf{V}_{\text{temp}})) = 0$ 
20:           Break for
21:         else
22:            $\mathbf{Flag}(Length(\mathbf{V}_{\text{temp}})) = 1$ 
23:         end if
24:       end for
25:     end if
26:   end if
27: end for
28: if  $Length(\mathbf{Flag}) == 0$  then
29:   return  $\mathbf{v}_{\text{can}} = void$ 
30: else
31:   for  $m = 1$  to  $Length(\mathbf{Flag})$  do
32:      $j = Length(\mathbf{Flag}) - m + 1$ 
33:     if  $\mathbf{Flag}(j) == 1$  then
34:        $\mathbf{v}_{\text{can}}.x = \mathbf{V}_{\text{temp}}(j).x; \mathbf{v}_{\text{can}}.y = \mathbf{V}_{\text{temp}}(j).y; \mathbf{v}_{\text{can}}.cost = \mathbf{V}_{\text{temp}}(j).cost;$ 
35:       Break for
36:     end if
37:   end for
38: end if
39: Output:  $\mathbf{v}_{\text{can}}.x, \mathbf{v}_{\text{can}}.y, \mathbf{v}_{\text{can}}.cost = 0$ 
```

Algorithm 2 The $\text{RandExpand}(\mathbf{T}, \mathcal{E}, P_g, D_{sc})$ algorithm. $\text{Collision}(\mathbf{a}, \mathcal{E})$ returns 0 if there is no collision between point \mathbf{a} and the environment \mathcal{E} . $\text{Rand}(a)$ returns a random number between 0 and a . $\text{MinIndex}(\mathbf{a})$ returns the index of the minimum element in \mathbf{a} . $\text{Round}(a)$ returns the nearest integer of a .

```

1: Input:  $\mathbf{T}, \mathcal{E}, P_g, D_{sc}$ .
2: while  $t < Inf$  do
3:    $\mathbf{D}_{\text{temp}} = []$ 
4:    $\mathbf{p}_{\text{rand}}.x = \text{Rand}(1) \cdot E_x$            /*  $E_x$  is the width of  $\mathcal{E}$  in  $x$  direction */
5:    $\mathbf{p}_{\text{rand}}.y = \text{Rand}(1) \cdot E_y$            /*  $E_y$  is the width of  $\mathcal{E}$  in  $y$  direction */
6:   for  $i = 1$  to  $\text{Length}(\mathbf{T})$  do
7:      $\mathbf{D}_{\text{temp}}(i) = \sqrt{(\mathbf{p}_{\text{rand}}.x - \mathbf{T}(i).x)^2 + (\mathbf{p}_{\text{rand}}.y - \mathbf{T}(i).y)^2}$ 
8:   end for
9:    $l = \text{MinIndex}(\mathbf{D}_{\text{temp}})$ 
10:   $d_{\text{pre}} = \text{Atan2d}(\mathbf{p}_{\text{rand}}.y - \mathbf{T}(l).y, \mathbf{p}_{\text{rand}}.x - \mathbf{T}(l).x) / 45$ 
11:   $d_{\text{pre}} = \text{Round}(d_{\text{pre}})$ 
12:  if  $\text{Mod}(d_{\text{pre}}, 2) \neq 0$  then
13:     $\mathbf{v}_{\text{can}}.x = \mathbf{T}(l).x + \sqrt{2} \cdot D_{sc} \cdot \cos(d_{\text{pre}} \cdot 45)$ 
14:     $\mathbf{v}_{\text{can}}.y = \mathbf{T}(l).y + \sqrt{2} \cdot D_{sc} \cdot \sin(d_{\text{pre}} \cdot 45)$ 
15:  else
16:     $\mathbf{v}_{\text{can}}.x = \mathbf{T}(l).x + D_{sc} \cdot \cos(d_{\text{pre}} \cdot 45)$ 
17:     $\mathbf{v}_{\text{can}}.y = \mathbf{T}(l).y + D_{sc} \cdot \sin(d_{\text{pre}} \cdot 45)$ 
18:  end if
19:  if  $\text{Collision}(\mathbf{v}_{\text{can}}, \mathcal{E}) == 0$  then
20:     $\mathbf{v}_{\text{can}}.\text{cost} = \sqrt{(\mathbf{v}_{\text{can}}.x - x_g)^2 + (\mathbf{v}_{\text{can}}.y - y_g)^2}$ 
21:    Break while
22:  end if
23: end while
24: Output:  $\mathbf{v}_{\text{can}}.x, \mathbf{v}_{\text{can}}.y, \mathbf{v}_{\text{can}}.\text{cost} = 0$ 

```

Algorithm 3 The $FindNeighbor(\mathbf{v}_i, \mathbf{T}, R_{near})$ algorithm.

```
1: Input:  $\mathbf{v}_i, \mathbf{T}, R_{near}$ .
2:  $\mathbf{I}_{near} = []$ ;  $k = 1$ 
3: for  $j = 1$  to  $i - 1$  do
4:    $d_{test} = \sqrt{(\mathbf{v}_i.x - \mathbf{T}(j).x)^2 + (\mathbf{v}_i.y - \mathbf{T}(j).y)^2}$ 
5:   if  $d_{test} \leq R_{near}$  then
6:      $\mathbf{I}_{near}(k) = j$ 
7:      $k = k + 1$ 
8:   end if
9: end for
10: Output:  $\mathbf{I}_{near} = 0$ 
```

Algorithm 4 The $ChooseParent(\mathbf{v}_i, \mathbf{T}, \mathbf{I}_{\text{near}})$ algorithm.

```
1: Input:  $\mathbf{v}_i$ ,  $\mathbf{T}$ ,  $\mathbf{I}_{\text{near}}$ .
2:  $D_{\text{temp}} = 0$ 
3: for  $j = 1$  to  $Length(\mathbf{I}_{\text{near}})$  do
4:    $d_{\text{cu}} = \sqrt{(\mathbf{v}_i.x - \mathbf{T}(j).x)^2 + (\mathbf{v}_i.y - \mathbf{T}(j).y)^2} + \mathbf{T}(j).cucost$ 
5:   if  $d_{\text{cu}} \leq D_{\text{temp}}$  then
6:      $D_{\text{temp}} = d_{\text{cu}}$ 
7:      $\mathbf{v}_i.parent = j$ 
8:      $\mathbf{v}_i.cucost = d_{\text{cu}}$ 
9:   end if
10: end for
11: Output:  $\mathbf{v}_i.parent$ ,  $\mathbf{v}_i.cucost = 0$ 
```

Algorithm 5 The $ReWire(\mathbf{v}_i, \mathbf{T}, \mathbf{I}_{\text{near}}(k))$ algorithm.

```
1: Input:  $\mathbf{v}_i, \mathbf{T}, \mathbf{I}_{\text{near}}(k)$ .
2:  $d_{\text{cu}} = \sqrt{(\mathbf{v}_i.x - \mathbf{T}(\mathbf{I}_{\text{near}}(k)).x)^2 + (\mathbf{v}_i.y - \mathbf{T}(\mathbf{I}_{\text{near}}(k)).y)^2} + \mathbf{v}_i.cucost$ 
3: if  $d_{\text{cu}} \leq \mathbf{T}(\mathbf{I}_{\text{near}}(k)).cucost$  then
4:    $\mathbf{T}(\mathbf{I}_{\text{near}}(k)).parent = i$ 
5:    $\mathbf{T}(\mathbf{I}_{\text{near}}(k)).cucost = d_{\text{cu}}$ 
6: end if
7: Output:  $\mathbf{T}(\mathbf{I}_{\text{near}}(k)).parent, \mathbf{T}(\mathbf{I}_{\text{near}}(k)).cucost = 0$ 
```

Algorithm 6 The $\text{ExtractTra}(\mathbf{T}, \mathbf{v}_i, P_g)$ algorithm. $\text{Flip}(\mathbf{a})$ reverses the order of the elements in \mathbf{a} .

```
1: Input:  $\mathbf{T}, \mathbf{v}_i, P_g$ 
2:  $\mathbf{T}_{\text{trajecotry}}(1).x = x_g; \mathbf{T}_{\text{trajecotry}}(1).y = y_g; \mathbf{T}_{\text{trajecotry}}(1).parent = i$ 
3:  $w = 2$ 
4: while  $\mathbf{T}(i).parent \neq 1$  do
5:    $\mathbf{T}_{\text{trajecotry}}(w) = \mathbf{T}(\mathbf{T}(i).parent)$ 
6:    $w = w + 1$ 
7:    $i = \mathbf{T}(i).parent$ 
8: end while
9:  $\mathbf{T}_{\text{trajecotry}}(w).x = x_g; \mathbf{T}_{\text{trajecotry}}(w).y = y_g$ 
10:  $\mathbf{T}_{\text{trajecotry}} = \text{Flip}(\mathbf{T}_{\text{trajecotry}})$ 
11: Output:  $\mathbf{T}_{\text{trajecotry}} = 0$ 
```

Algorithm 7 The $\text{SafeDist}(\mathbf{T}_{\text{trajecotry}}, \mathcal{E}, D_{\text{sf}})$ algorithm. $\text{MinIndex}(\mathbf{a})$ returns the index of the minimum element in \mathbf{a} .

```

1: Input:  $\mathbf{T}_{\text{trajecotry}}$ ,  $\mathcal{E}$ ,  $D_{\text{sf}}$ 
2: for  $i = 2$  to  $\text{Length}(\mathbf{T}_{\text{trajecotry}}) - 1$  do
3:   for  $r = 1$  to  $5$  do
4:     for  $\eta = 1$  to  $18$  do
5:        $x_{\text{ser}} = \mathbf{T}_{\text{trajecotry}}(i).x + (r - 1) \cdot 0.3 \cdot D_{\text{sf}} \cdot \cos(20 \cdot \eta)$ 
6:        $y_{\text{ser}} = \mathbf{T}_{\text{trajecotry}}(i).y + (r - 1) \cdot 0.3 \cdot D_{\text{sf}} \cdot \sin(20 \cdot \eta)$ 
7:        $D_1 = \sqrt{(x_{\text{ser}} - \mathbf{T}_{\text{trajecotry}}(i - 1).x)^2 + (y_{\text{ser}} - \mathbf{T}_{\text{trajecotry}}(i - 1).y)^2}$ 
8:        $D_2 = \sqrt{(x_{\text{ser}} - \mathbf{T}_{\text{trajecotry}}(i + 1).x)^2 + (y_{\text{ser}} - \mathbf{T}_{\text{trajecotry}}(i + 1).y)^2}$ 
9:        $D_{\text{ser}}(\eta) = D_1 + D_2$ 
10:       $n_{\text{collision}} = 0$ 
11:      for  $k = 1$  to  $18$  do
12:         $x_{\text{eva}} = x_{\text{ser}}.x + D_{\text{sf}} \cdot \cos(20 \cdot k)$ 
13:         $y_{\text{eva}} = x_{\text{ser}}.y + D_{\text{sf}} \cdot \sin(20 \cdot k)$ 
14:        if  $\text{Collision}(\mathbf{a}, \mathcal{E}) \neq 0$  then
15:           $n_{\text{collision}} = n_{\text{collision}} + 1$ 
16:        end if
17:      end for
18:       $\mathbf{N}_{\text{collision}}(\eta) = n_{\text{collision}}$ 
19:       $\mathbf{Cost}(\eta) = 3000 \cdot \mathbf{N}_{\text{collision}}(\eta) + 2 \cdot D_{\text{ser}}(\eta) + 5 \cdot \text{Abs}(D_1 - D_2)$            /*the cost function to optimize the position of
   the updated trajectory point*/
20:    end for
21:     $l = \text{MinIndex}(\mathbf{Cost})$ 
22:     $\mathbf{L}(r) = l$ 
23:     $\mathbf{C}(r) = \mathbf{Cost}(l)$ 
24:  end for
25:   $r_{\text{global}} = \text{MinIndex}(\mathbf{C})$ 
26:   $\eta_{\text{global}} = \mathbf{L}(r_{\text{global}})$ 
27:   $\mathbf{T}_{\text{trajecotry}}(i).x = \mathbf{T}_{\text{trajecotry}}(i).x + (r_{\text{global}} - 1) \cdot 0.3 \cdot D_{\text{sf}} \cdot \cos(20 \cdot \eta_{\text{global}})$ 
28:   $\mathbf{T}_{\text{trajecotry}}(i).y = \mathbf{T}_{\text{trajecotry}}(i).y + (r_{\text{global}} - 1) \cdot 0.3 \cdot D_{\text{sf}} \cdot \sin(20 \cdot \eta_{\text{global}})$ 
29: end for
30: Output:  $\mathbf{T}_{\text{trajecotry}} = 0$ 

```

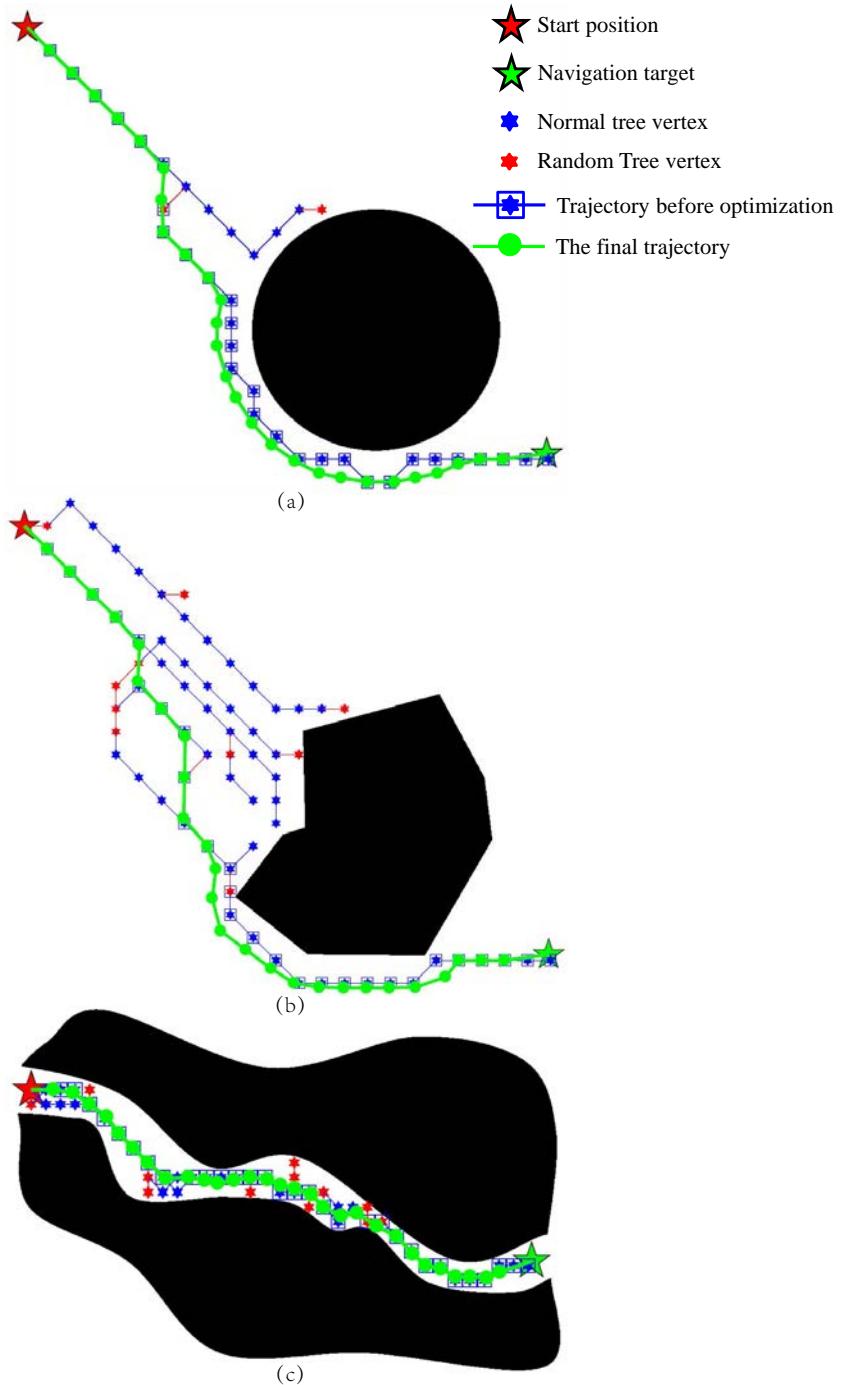


Fig. S18. Illustration of the autonomous trajectory planning results for different environment morphologies. **a**, A circular obstacle with a trajectory planning time of 0.1 s. **b**, An irregular obstacle with a trajectory planning time of 0.15 s. **c**, An environment with the channel morphology with a trajectory planning time of 0.13 s. It is validated that the method works appropriately for different environment morphologies. The short computation time fulfills the requirement for real-time swarm navigation.

- 209 **Movie S1.** The robustness validation of the DNN-based swarm distribution planning method for different
210 environment morphologies, and the comparison of computation time between this method and the traversal-
211 based optimization method.
- 212 **Movie S2.** Experimental demonstration of the three swarm configurations and the transformations among
213 them.
- 214 **Movie S3.** Experimental demonstration of the translational motion and rotational motion of the ribbon-like
215 swarm (RS).
- 216 **Movie S4.** Experimental comparisons between manual navigation (autonomy Level 0) and automated control
217 (autonomy Level 1).
- 218 **Movie S5.** Experimental demonstration of autonomy Level 2, where two sets of experiments with different
219 swarm sizes were conducted to assess the intelligence of the DNN-based swarm distribution planning method.
- 220 **Movie S6.** Experimental demonstration of autonomy Level 3, where a swarm of magnetic nanorobots accom-
221 plished the delivery task to a targeted region in a channel environment.
- 222 **Movie S7.** Experimental demonstration of autonomy Level 3, where the reconfigurable magnetic nanorobot
223 swarm navigated in highly curved space with a 150° sharp turn and in a curved narrow channel environment.
- 224 **Movie S8.** Experimental demonstration of autonomy Level 3, where the reconfigurable magnetic nanorobot
225 swarm executed the cooperative micromanipulation task in confined space.
- 226 **Movie S9.** Experimental demonstration of the fully autonomous environment exploration using the magnetic
227 nanorobot swarm in an unknown channel environment (autonomy Level 4).
- 228 **Movie S10.** Experimental demonstration of the fully autonomous targeted delivery to a region under dynamic
229 obstacles (autonomy Level 4).
- 230 **Movie S11.** Transfer the autonomy framework to the elliptical vortex-like magnetic nanoparticle swarm.
- 231 **Movie S12.** Method Validation under ultrasound imaging and x-ray fluoroscopy.

232 **References**

- 233 1. H Deng, et al., Monodisperse magnetic single-crystal ferrite microspheres. *Angew. Chem.* **117**, 2842–2845 (2005).
234 2. D Karimi, et al., Accurate and robust deep learning-based segmentation of the prostate clinical target volume in ultrasound
235 images. *Med. image analysis* **57**, 186–196 (2019).
236 3. C Chen, et al., Deep learning for cardiac image segmentation: a review. *Front. Cardiovasc. Medicine* **7**, 25 (2020).