Supplementary Information for

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

Lidong Yang[1,†], Jialin Jiang[1,†], Xiaojie Gao[2], Qinglong Wang[1], Qi Dou[2,5,6], and Li Zhang[1,3,4,5,6,*]

[1]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.)

This PDF file includes:

Other supplementary materials for this manuscript include the following:

Supplementary Videos S1 to S12

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

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

Supplementary Note S2: DNN training and application processes

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

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

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

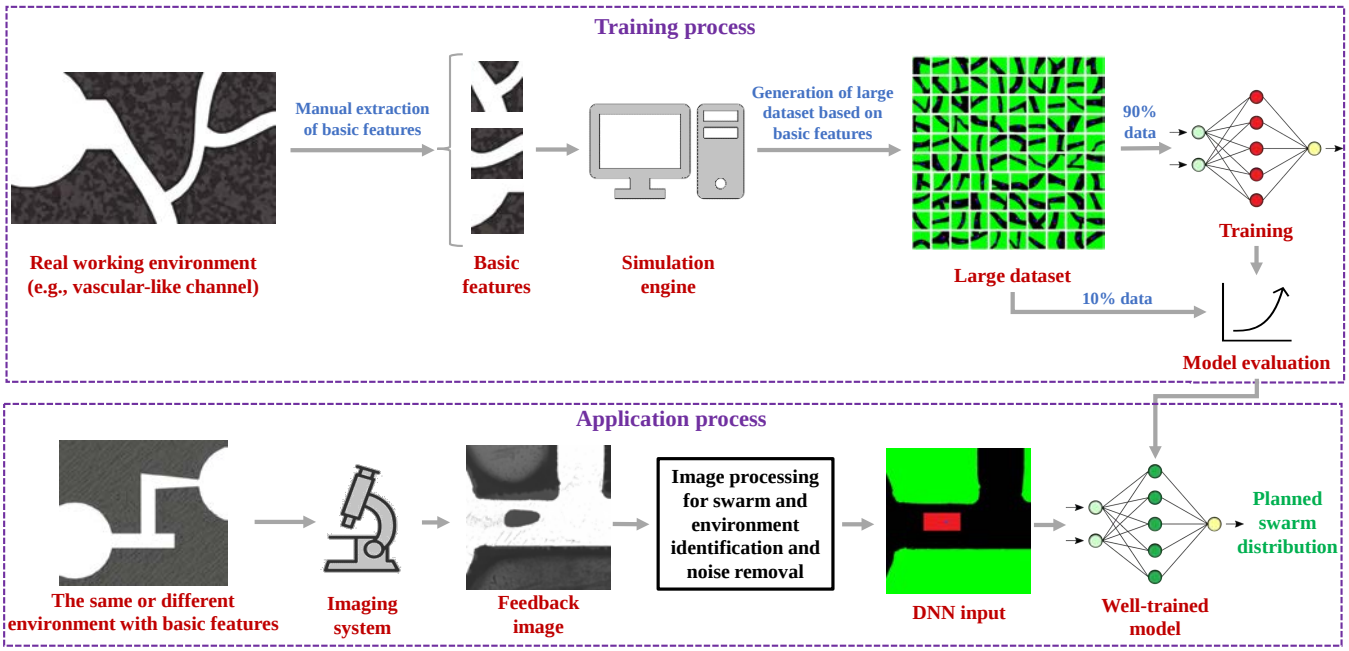


Fig. S1. Illustration of the training and application processes of the proposed DL-based swarm distribution planning method.

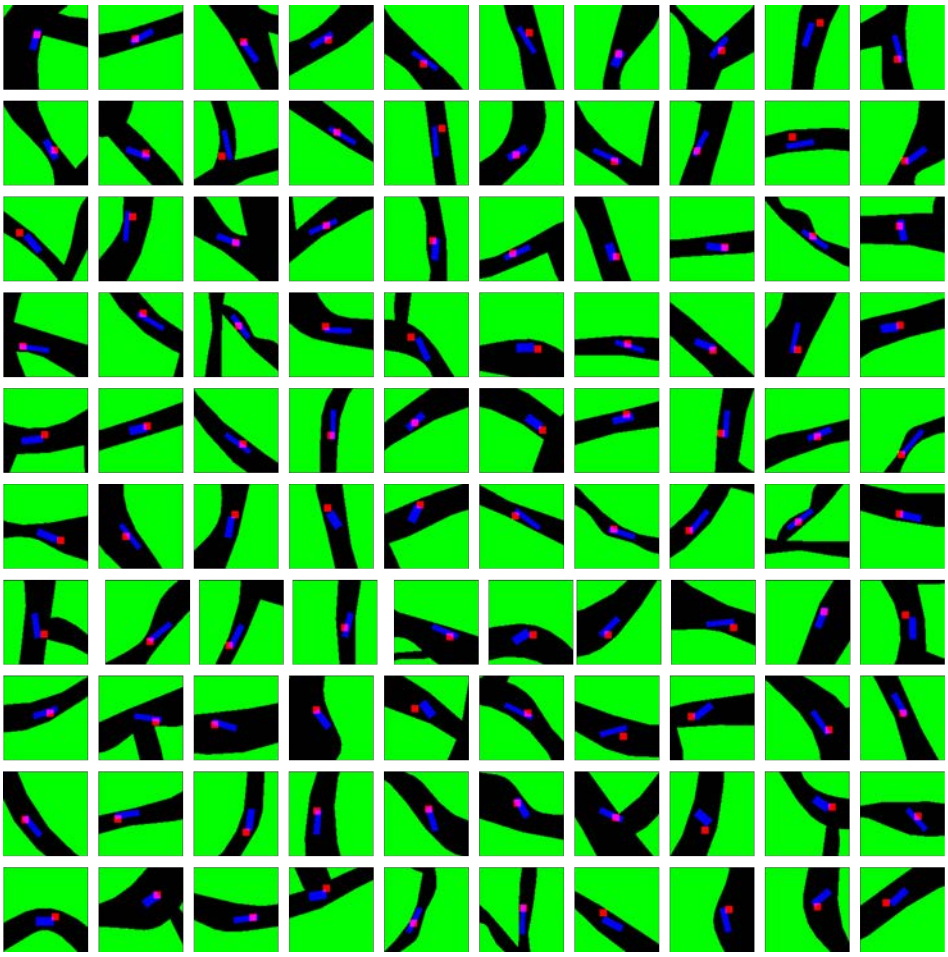


Fig. S2. Illustration of one hundred data scenarios used to train the DNN for swarm shape planing (SSP-DNN) and the DNN for swarm orientation planning (SOP-DNN). The black and green colors mark the available navigation space and the obstacle space, respectively. The blue and red rectangles are the current 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.

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

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

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

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

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

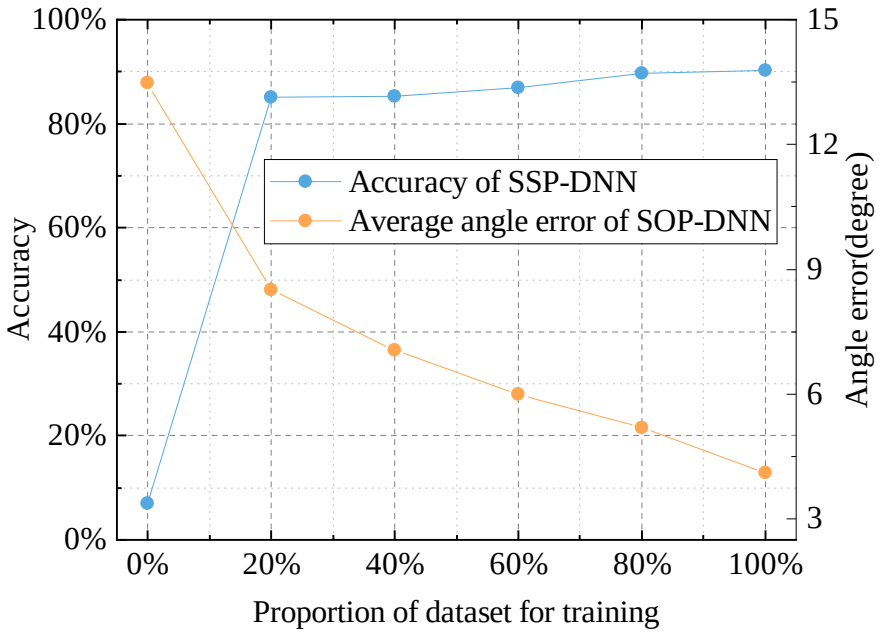


Fig. S3. Accuracy validation results when adopting different dataset sizes for training. The total training dataset contains 45 K navigation scenarios.

Supplementary Note S4: Magnetic nanoparticles and experimental setup

Fe_3O_4 nanoparticles with diameters around 400 nm are fabricated using the solvothermal method (1). A scanning electron microscopy image is shown in Supplementary Fig. S4a. The 3D dynamic magnetic fields for actuation are provided by a 3-axis Helmholtz electromagnetic coil setup, as illustrated in Supplementary Fig. S4b. The nanoparticle swarm and its working environment are observed by an optical microscope and fed back to a computer via a high-speed (45 fps) camera mounted on the microscope. Based on the proposed framework, control signals are generated by the programs coded by LabVIEW, Matlab, and Python, which are then sent to the amplifiers (ADS 50/5 4-Q-DC, Maxon, Inc.) to generate desired currents in coils. Computed by the field regulation principles in Methods, the oscillating field, rotating field, and 3D dynamic field required for the ribbon-like swarm (RS), vortex-like swarm (VS), and spreading swarm (SS) are ready to be produced. For experimental investigations, magnetic nanoparticle solutions (3 mg/mL) are added into an acrylic tank filled with deionized (DI) water. For long-distance navigation, a home-designed 2D motorized stage (MTS25-Z8, Thorlabs, Inc.) is used to adjust the field of view (FOV) of the microscope. Its control codes written by LabVIEW are also integrated into the overall program.

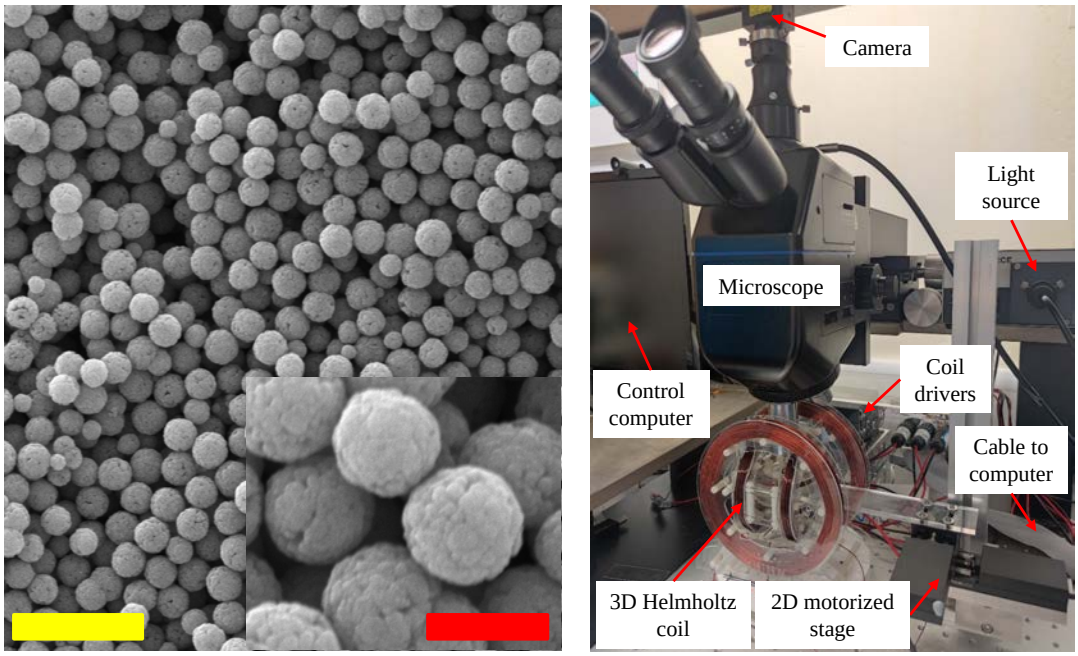


Fig. S4. a, The Fe_3O_4 nanoparticle with an average diameter of 400 nm. The yellow and red scale bars are 2 μm and 400 nm, respectively. **b**, The system setup used to generate 3D dynamic fields for actuation of the magnetic nanoparticle swarm.

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

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

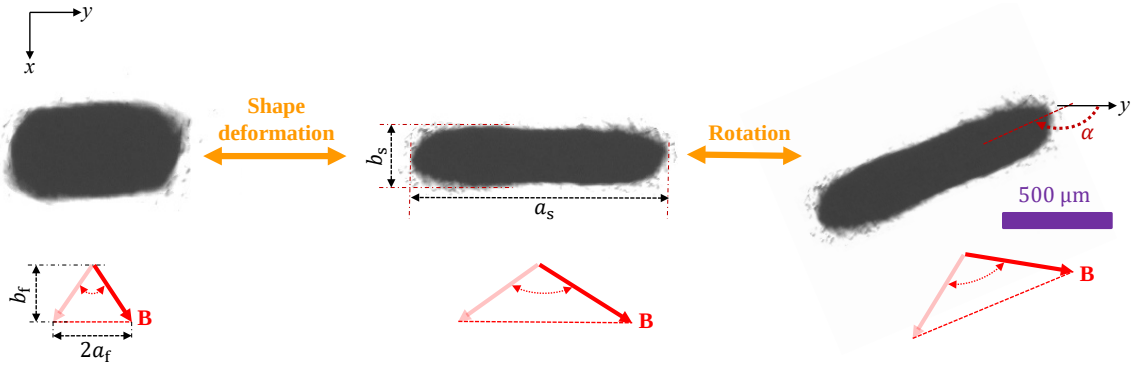


Fig. S5. Experimental demonstration of the swarm pattern deformation and rotation of the ribbon-like swarm (RS). Swarm shape ratio R_s and field ratio R_f are defined as $R_s = \frac{a_s}{b_s}$ and $R_f = \frac{a_f}{b_f}$, respectively. Swarm orientation angle α represents the angle between the long-axis 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 **B**.

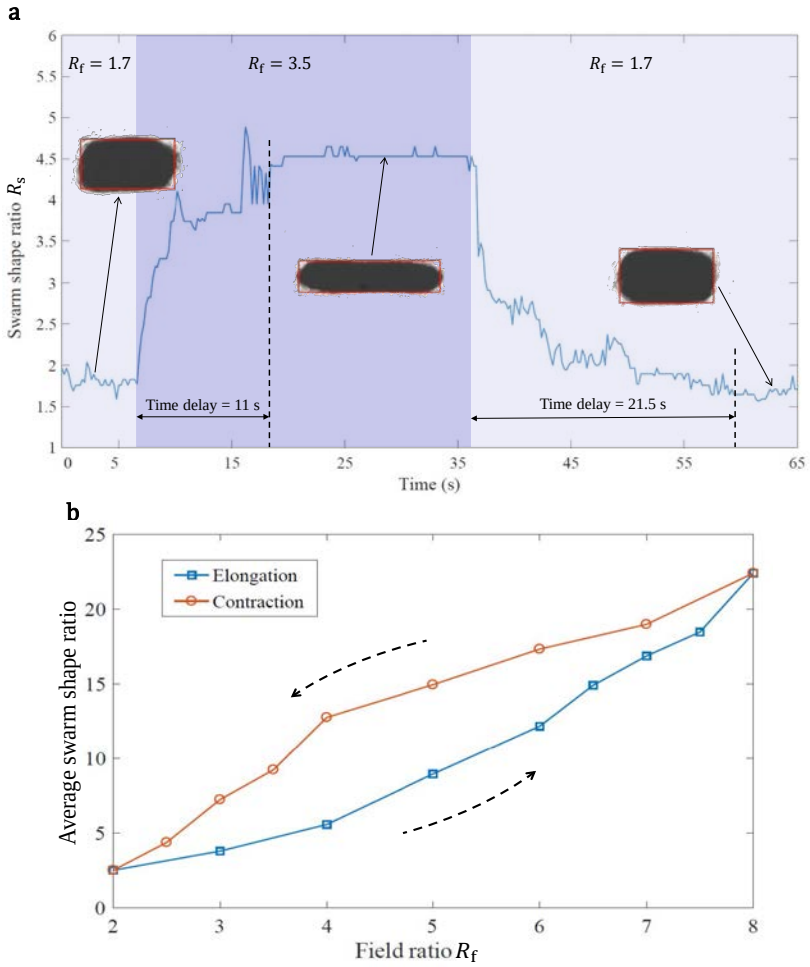


Fig. S6. Experimental characterization of the deformation properties for the ribbon-like swarm (RS). **a**, Characterization of the transient deformation, including the elongation and contraction processes. At first, the RS has a stable shape under actuation with a field ratio of 1.7. Then, the field 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.

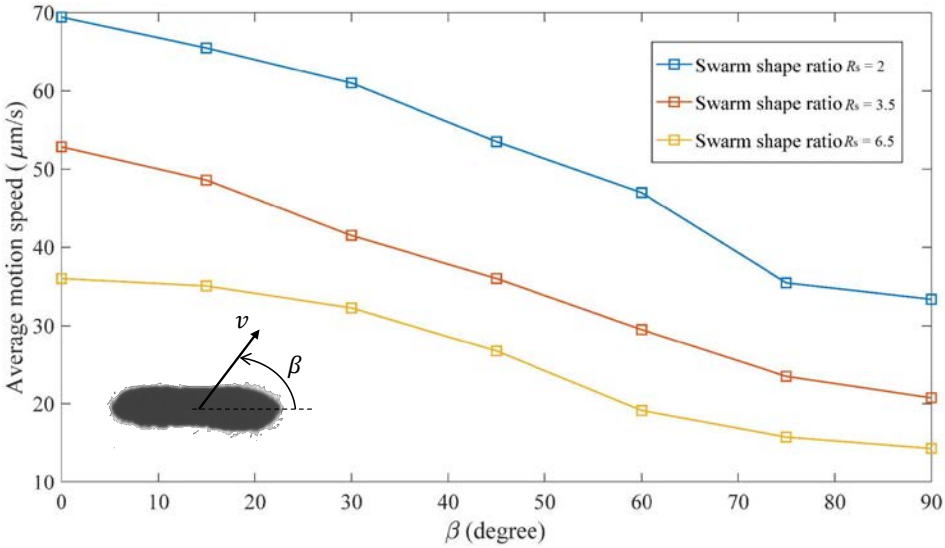


Fig. S7. Experimental characterization of the motion properties for the ribbon-like swarm (RS). β represents the intersection angle between the swarm orientation and the translational motion direction. v is the translational motion speed. The motion speed is positively correlated to the field pitch angle, 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.

Supplementary Note S6: Common failure cases in manual navigation

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

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

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

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 swarm pattern becomes unstable. Besides, the low-frequency human control makes the correct motion control difficult. The two factors cause the swarm to lose particles, and the swarm collides with the channel wall.

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

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

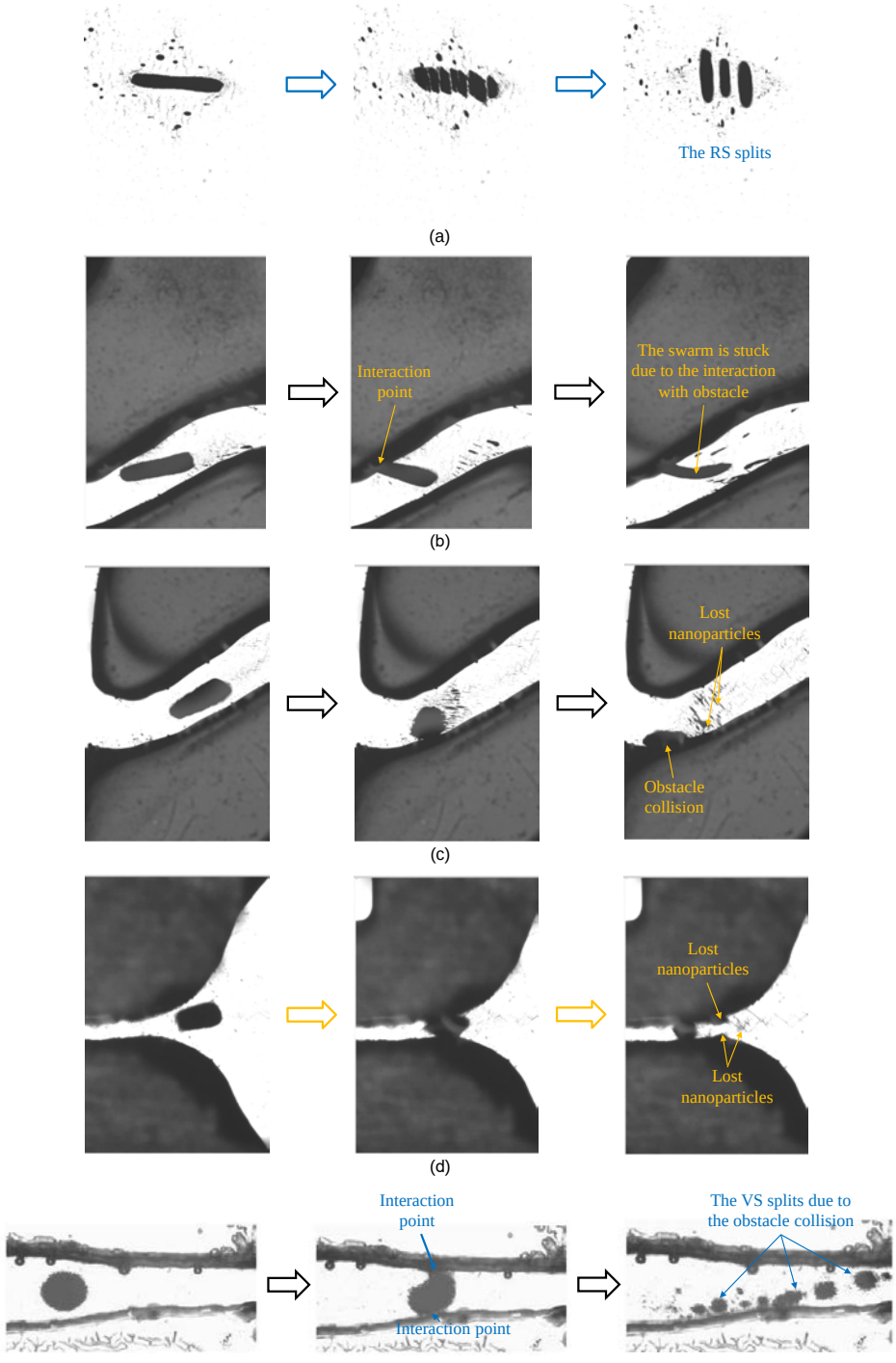


Fig. S8. Common failure cases in manual navigation due to (a) excessive rotation control; (b) wrong rotation direction control; (c) wrong field pitch angle control and low-frequency motion direction control; and (d) wrong swarm shape control; (e) wrong swarm shape control for the VS.

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

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

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

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

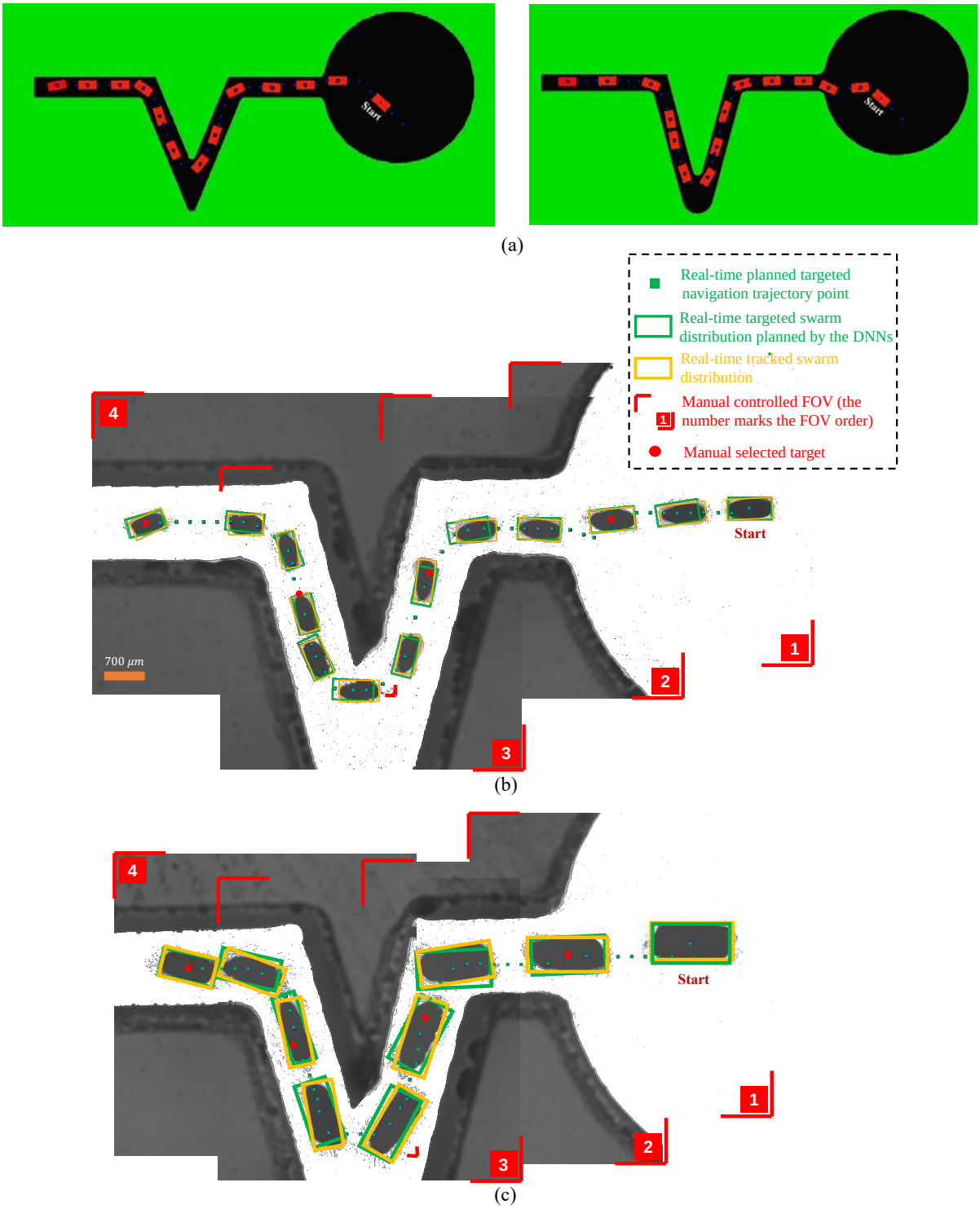
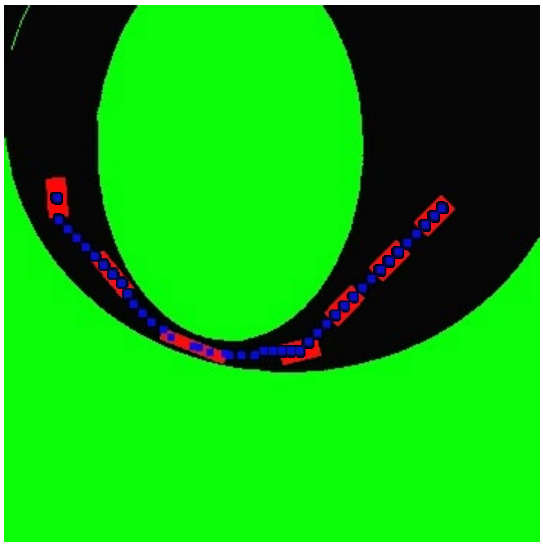
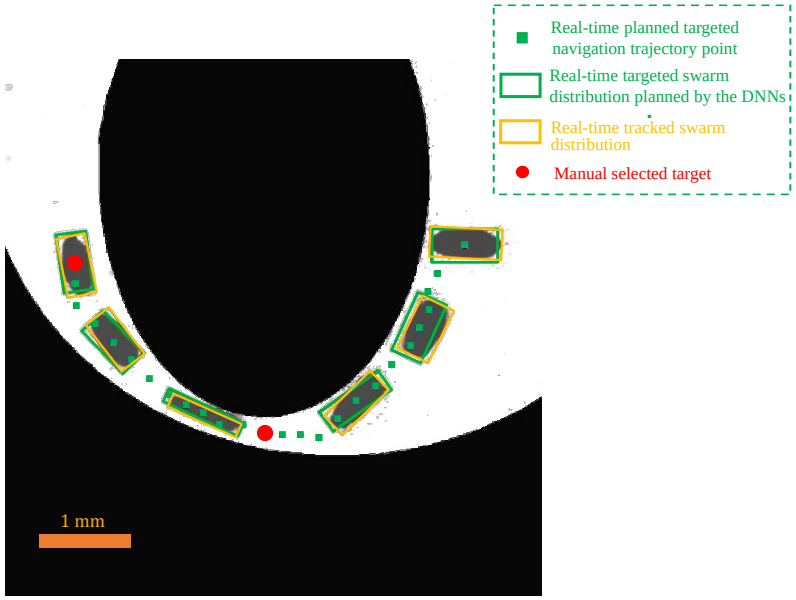


Fig. S9. Demonstration of Autonomy Level 3 in highly curved space with sharp turns. (a) Simulation results for environments with 135° and 150° sharp turns. (b) The experimental navigation result for an RS containing 0.75 μg nanoparticles. (c) The experimental navigation result for an RS containing 2.0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

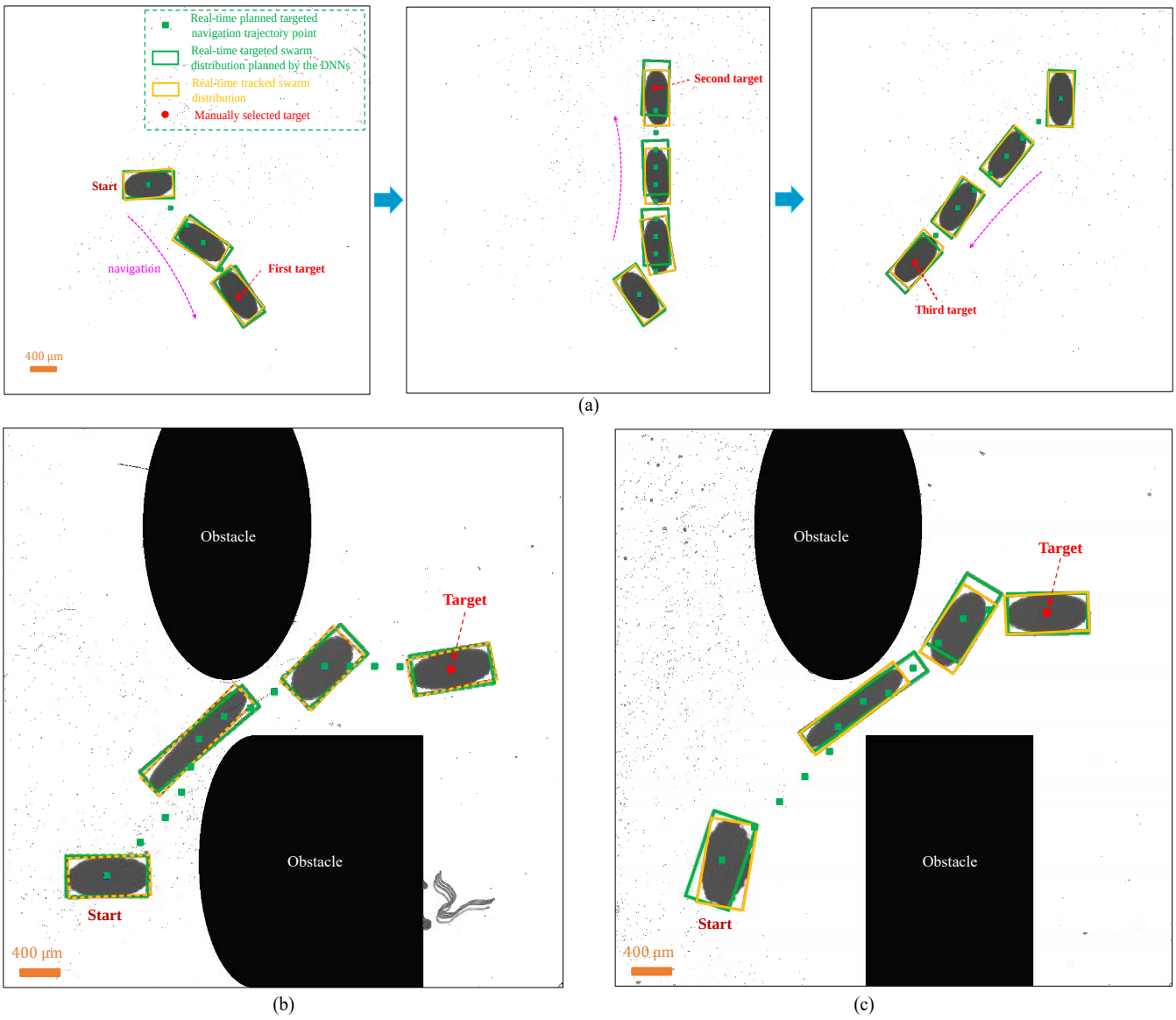


Fig. S11. Transfer of the methods to the elliptical vortex-like nanoparticle swarm. Three experiments using autonomy Level 3 are conduced to test the navigation performances in different environments, including (a) the blank space, (b) an environment with round-corner obstacles, and (c) an environment 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.

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

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

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

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

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

Guideline of the placenta collection under ethics approval

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

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

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

Inclusion Criteria

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

sections after 37-42 weeks of gestation;

ii Singleton pregnancy;

iii Healthy as determined by laboratory results, physical exam and medical history;

iv Participant able to give voluntary, written, informed consent to participate in the study.

Exclusion Criteria

i Abnormal prenatal development (e.g. intrauterine growth restriction);

ii Early preterm birth < 37 weeks;

iii Verbal confirmation of hypercholesterolemia;

iv Family history of stroke or vascular disease;

v Type I or Type II diabetes and gestational diabetes;

vi Cancer, except skin cancers completely excised with no chemotherapy or radiation with a follow up that is negative;

vii Clinically significant abnormal laboratory results at screening;

viii Any other active or unstable medical condition;

ix History of liver disease;

x History of hypertension (including pre-clampsia).

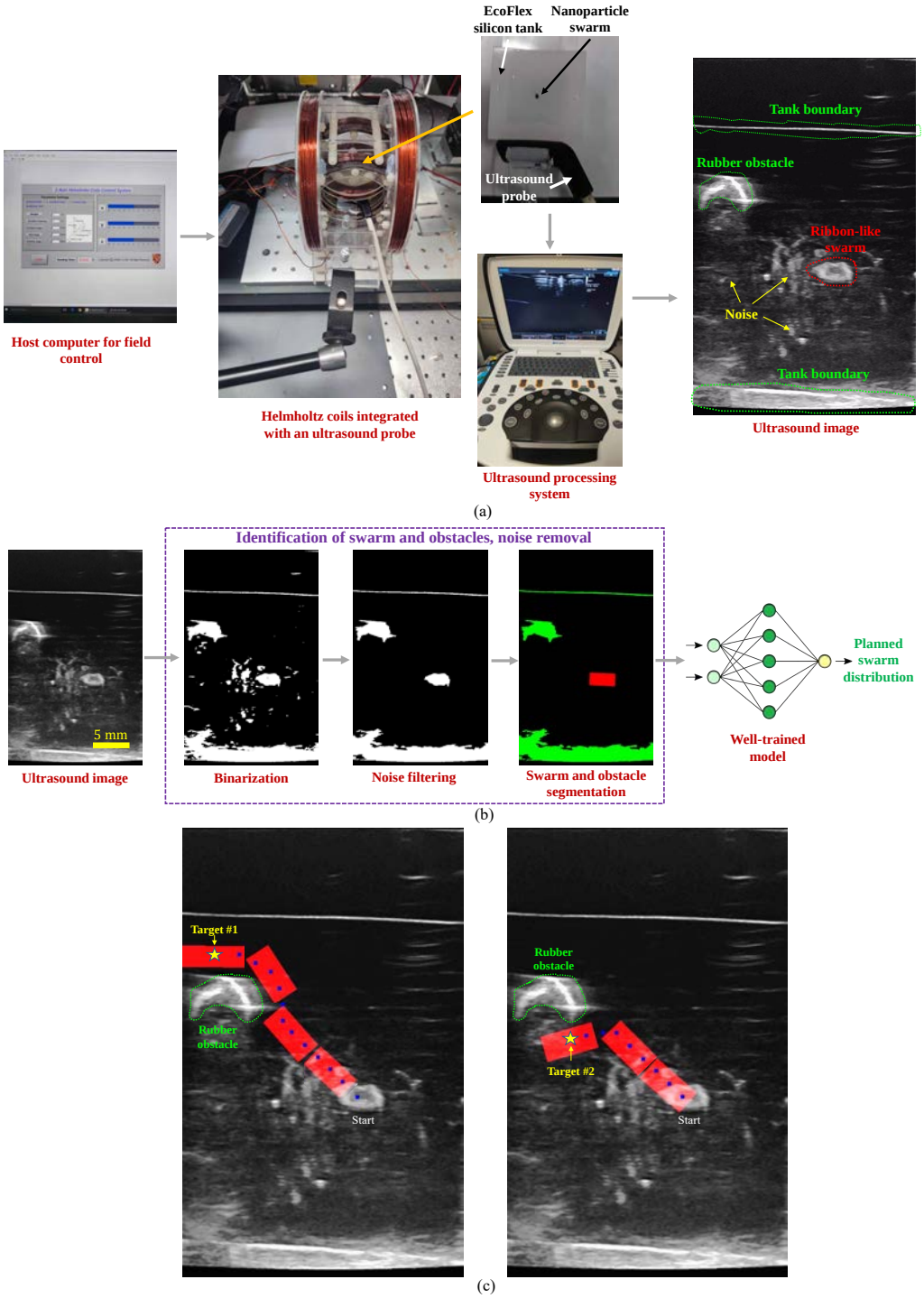
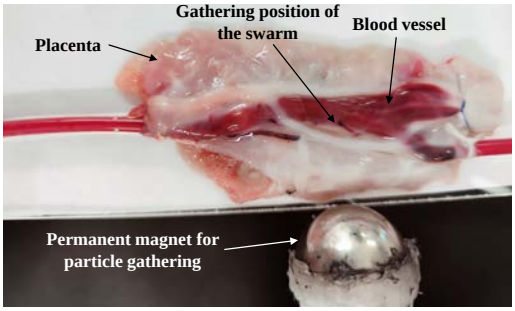
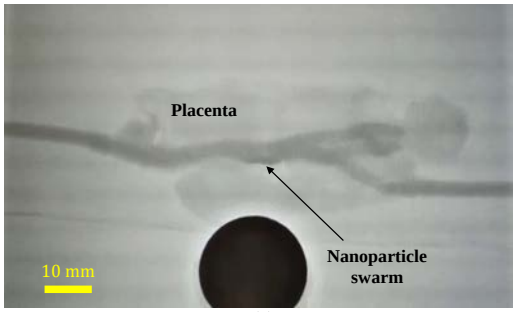


Fig. S12. Validation of the DL-based swarm distribution planning method under ultrasound imaging. (a) The system setup, in which an ultrasound processing system (Model:uSmart 3300 NexGen, Terason Inc.) is adopted. (b) The swarm and environment identification procedures before sending the image to the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

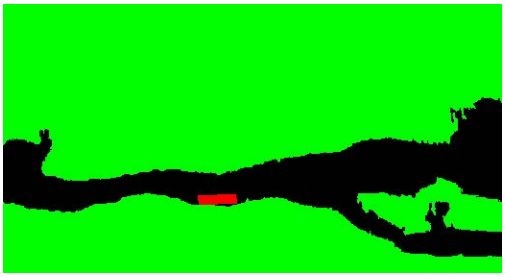


(a)

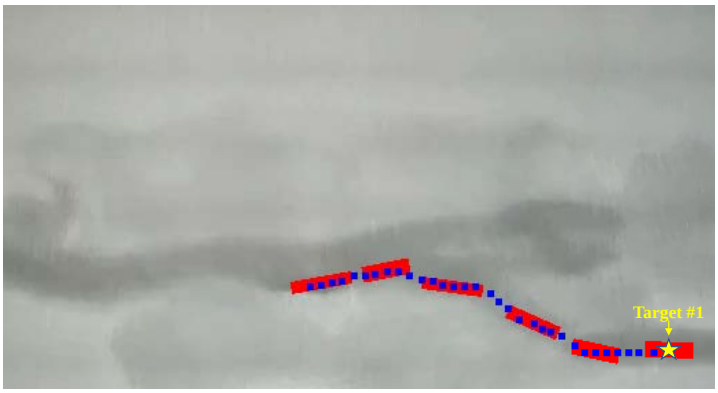
(b)



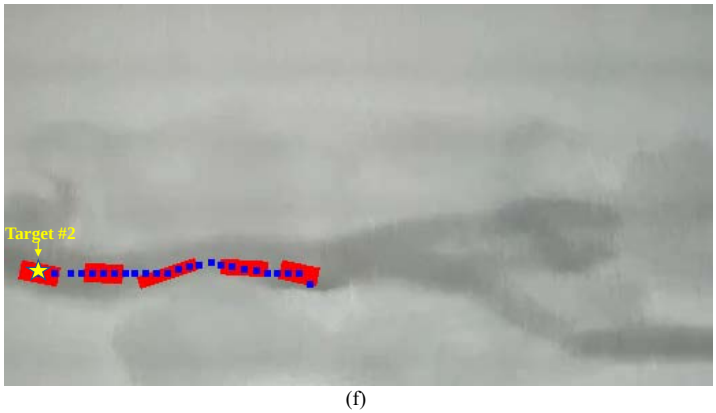
(c)



(d)

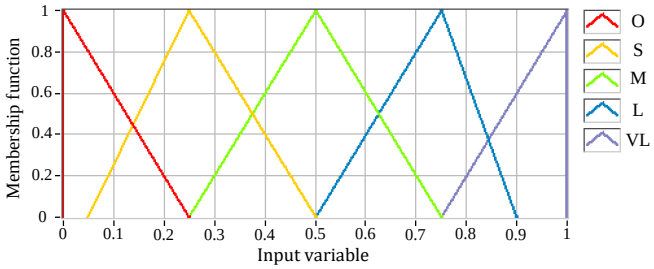


(e)

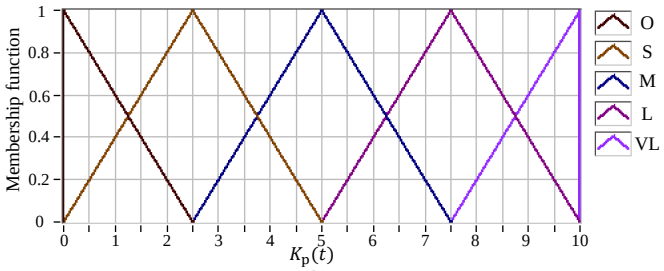


(f)

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.



(a)

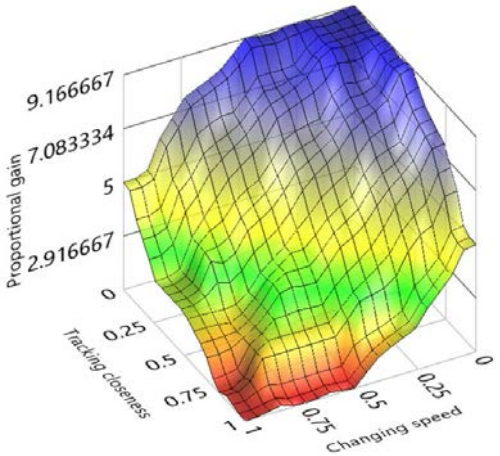


(b)

	Changing speed of the swarm shape ratio $v(R_s(t))$					
Tracking closeness $c(R_s(t))$		O	S	M	L	VL
	O	VL	VL	VL	L	M
	S	VL	VL	L	M	S
	M	VL	L	M	S	S
	L	L	M	S	S	O
	VL	M	S	O	O	O

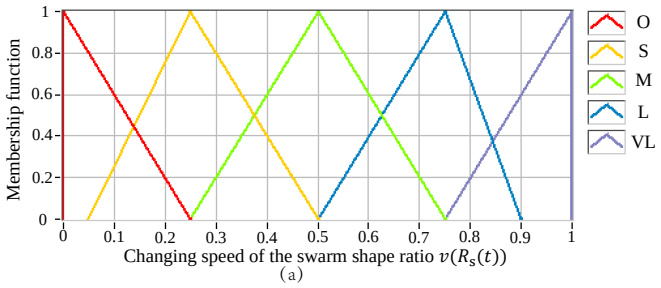
O = Zero; S = Small; M = Middle; L = Large; VL = Very Large

(c)

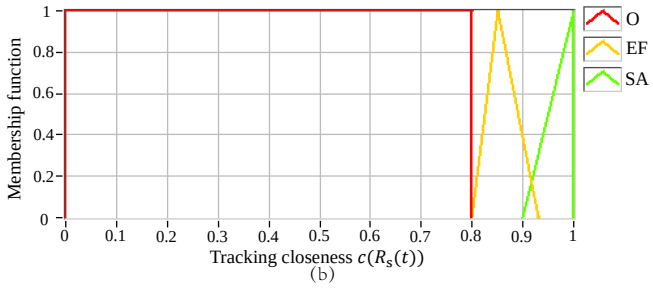


(d)

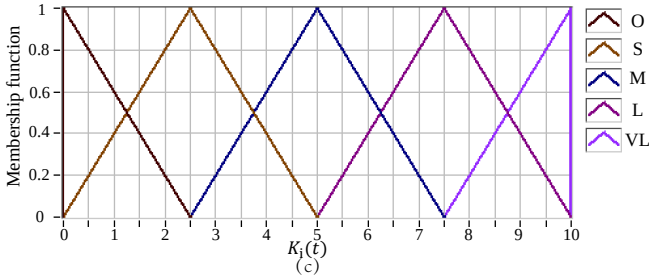
Fig. S14. The fuzzy logic controller for tuning $K_p(t)$ in the swarm shape control. **a**, The membership function for input variables $v(R_s(t))$ and $c(R_s(t))$. **b**, The membership function for the output $K_p(t)$. **c**, The fuzzy logic rules for the controller. **d**, The resulting control relationship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



(a)



(b)

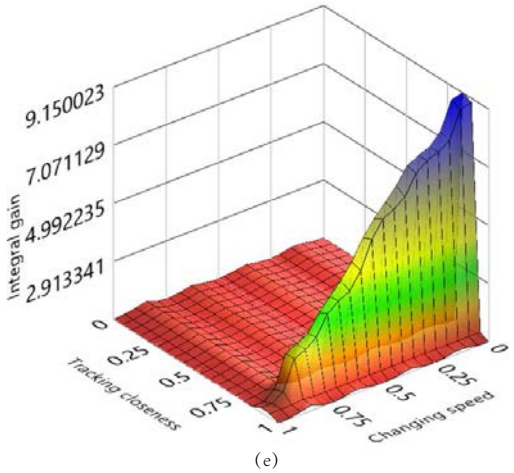


(c)

		Changing speed of the swarm shape ratio $v(R_s(t))$				
		O	S	M	L	VL
Tracking closeness $c(R_s(t))$	O	O	O	O	O	O
	EF	VL	L	M	S	O
	SA	O	O	O	O	O

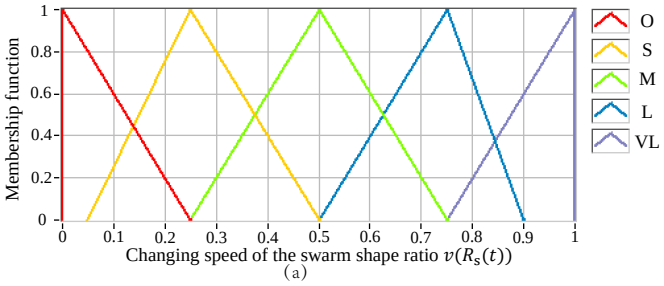
O = Zero; S = Small; M = Middle; L = Large; VL = Very Large; EF = Effective; SA = Saturated

(d)

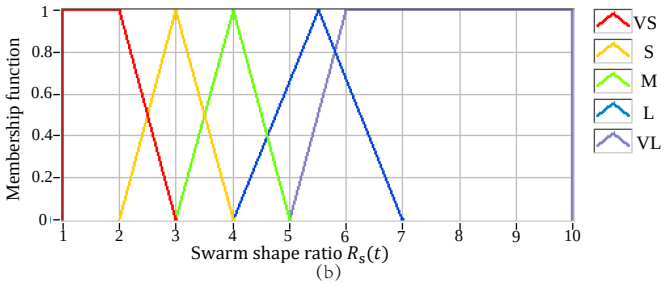


(e)

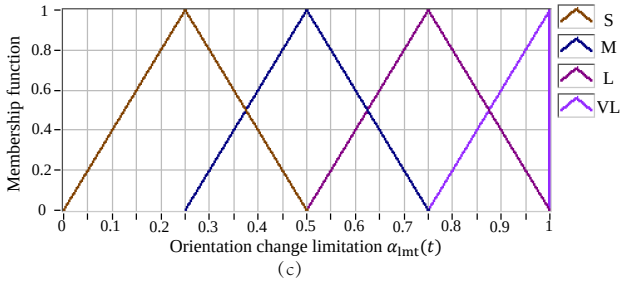
Fig. S15. The fuzzy logic controller for tuning $K_i(t)$ in the swarm shape control. **a**, The membership function for input variable $v(R_s(t))$. **b**, The membership function for input variable $c(R_s(t))$. **c**, The membership function for the output $K_i(t)$. **d**, The fuzzy logic rules for the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



(a)



(b)

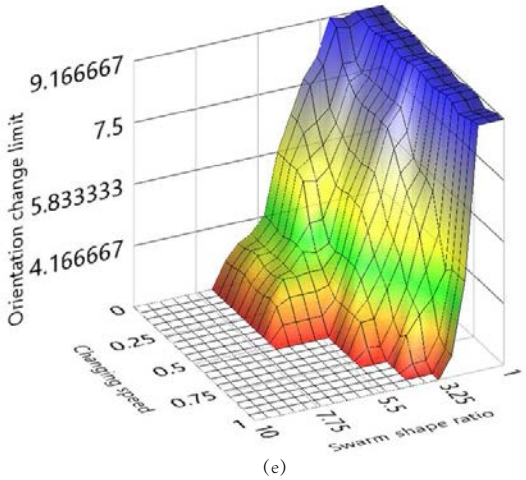


(c)

	Changing speed of the swarm shape ratio $v(R_s(t))$				
Swarm shape ratio $R_s(t)$	O	S	M	L	VL
VS	VL	VL	VL	VL	VL
S	VL	L	L	M	S
M	L	M	M	S	S
L	M	M	S	S	S
VL	S	S	S	S	S

O = Zero; S = Small; M = Middle; L = Large; VL = Very Large; VS = Very small

(d)



(e)

Fig. S16. The fuzzy logic controller for tuning $\alpha_{\mathrm{lmt}}(t)$ 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 $\alpha_{\mathrm{lmt}}(t)$. **d**, The fuzzy logic rules for the controller. **e**, The resulting control relationship for tuning $\alpha_{\mathrm{lmt}}(t)$ based on the current swarm states $v(R_s(t))$ and $R_s(t)$.

Supplementary Note S9: Performance comparisons of trajectory planning algorithms

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

To check if traditional methods fulfill the specific requirements for the microrobot swarm navigation, we further conducted a comparison study, implementing the traditional A*, traditional RRT*, and the method designed in this work. The comparison 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 time, because it explores all the neighbor positions. In addition, as A* does not consider a distance to the obstacle, it does not fulfill the third requirement either. Regarding the RRT*, it explores the environment with arbitrary exploration steps to ensure a fast speed for real-time use. However, in turn, the obtained trajectory distribution is highly nonuniform, making it against the second requirement. Besides, it does not consider the distance to the obstacle either. By contrast, the planning method designed in this work satisfies all the four requirements.

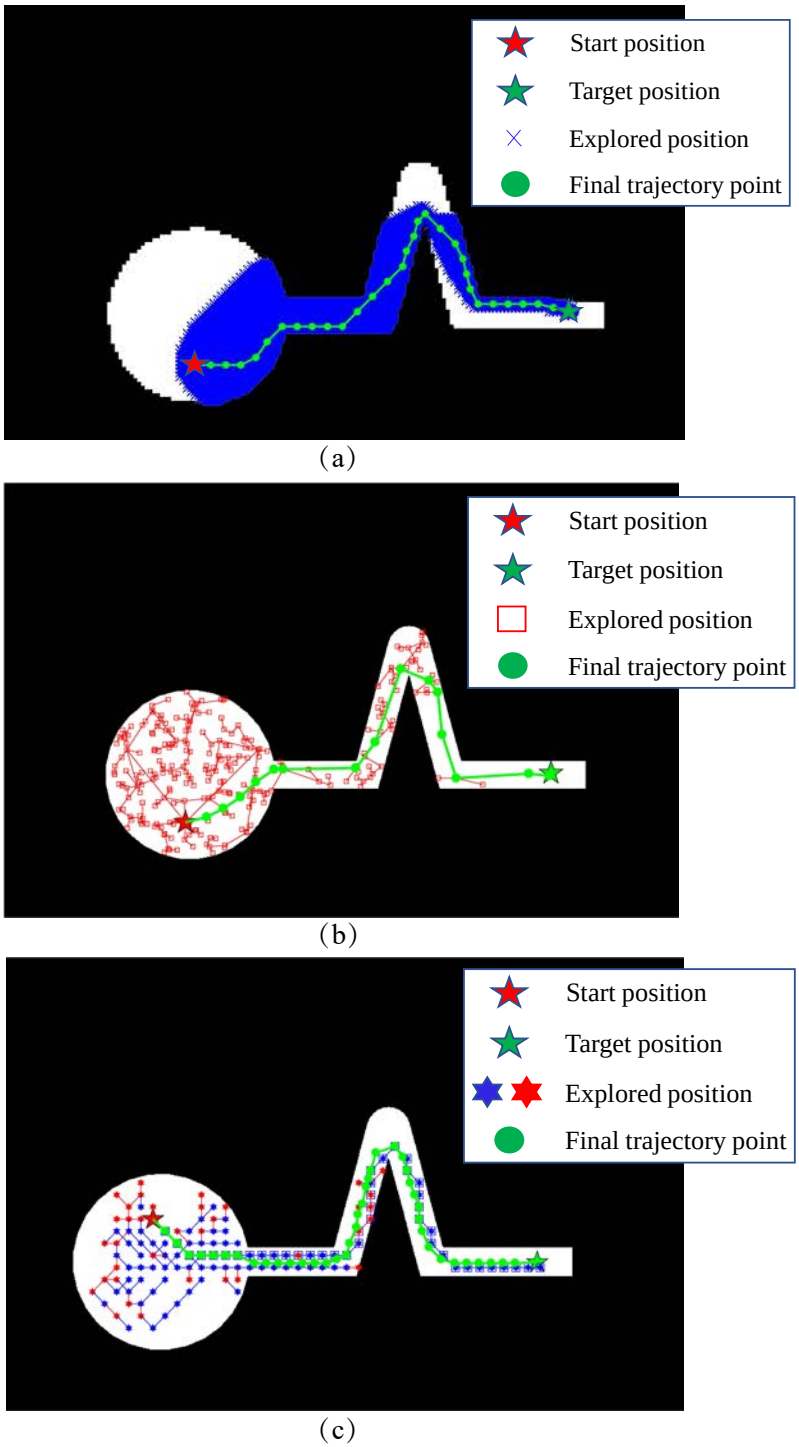


Fig. S17. Comparisons of three trajectory planning methods. (a) Traditional A* method, which takes 3.5 s. (b) Traditional RRT* method, which takes 0.55 s. (c) The designed method in this work, which takes 0.36 s.

Supplementary Note S10: Detailed algorithms for the autonomous trajectory planning

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

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

Algorithm 1 The $UniExpand(\mathbf{T}, \mathcal{E}, P_{\mathrm{g}}, \mathbf{v}_{\mathrm{lead}}, D_{\mathrm{sc}})$ 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 $\mathbf{a}$.

Input: $\mathbf{T}, \mathcal{E}, P_{\mathrm{g}}, \mathbf{v}_{\mathrm{lead}}, D_{\mathrm{sc}}$.
$\mathbf{V}_{\mathrm{temp}} = []$; **Flag** $= []$
$d_{\mathrm{pre}} = Atan2d(\mathbf{v}_{\mathrm{lead}}.y - \mathbf{T}(\mathrm{lead}-1).y, \mathbf{v}_{\mathrm{lead}}.x - \mathbf{T}(\mathrm{lead}-1).x)/45$
for $k = (d_{\mathrm{pre}} - 2)$ to $(d_{\mathrm{pre}} + 2)$ **do**
  if $Mod(k, 2) \neq 0$ **then**
    $\mathbf{v}_{\mathrm{can}}.x = \mathbf{v}_{\mathrm{lead}}.x + \sqrt{2} \cdot D_{\mathrm{sc}} \cdot \cos(k \cdot 45)$
    $\mathbf{v}_{\mathrm{can}}.y = \mathbf{v}_{\mathrm{lead}}.y + \sqrt{2} \cdot D_{\mathrm{sc}} \cdot \sin(k \cdot 45)$
  else
    $\mathbf{v}_{\mathrm{can}}.x = \mathbf{v}_{\mathrm{lead}}.x + D_{\mathrm{sc}} \cdot \cos(k \cdot 45)$
    $\mathbf{v}_{\mathrm{can}}.y = \mathbf{v}_{\mathrm{lead}}.y + D_{\mathrm{sc}} \cdot \sin(k \cdot 45)$
  end if
  if $Collision(\mathbf{v}_{\mathrm{can}}, \mathcal{E}) == 0$ **then**
    $\mathbf{v}_{\mathrm{can}}.cost = \sqrt{(\mathbf{v}_{\mathrm{can}}.x - x_{\mathrm{g}})^2 + (\mathbf{v}_{\mathrm{can}}.y - y_{\mathrm{g}})^2}$
    if $\mathbf{v}_{\mathrm{can}}.cost < \mathbf{v}_{\mathrm{lead}}.cost$ **then**
      $\mathbf{v}_{\mathrm{lead}}.cost = \mathbf{v}_{\mathrm{can}}.cost$
      $\mathbf{V}_{\mathrm{temp}} = [\mathbf{V}_{\mathrm{temp}}, \mathbf{v}_{\mathrm{can}}]$;
      for $n = 1$ to $Length(\mathbf{T}) - 1$ **do**
        if $\sqrt{(\mathbf{v}_{\mathrm{can}}.x - \mathbf{T}(n).x)^2 + (\mathbf{v}_{\mathrm{can}}.y - \mathbf{T}(n).y)^2} < D_{\mathrm{sc}}$ **then**
          Flag$(Length(\mathbf{V}_{\mathrm{temp}})) = 0$
          Break for
        else
          Flag$(Length(\mathbf{V}_{\mathrm{temp}})) = 1$
        end if
      end for
    end if
  end if
end for
if $Length(\mathbf{Flag}) == 0$ **then**
  return $\mathbf{v}_{\mathrm{can}} = void$
else
  for $m = 1$ to $Length(\mathbf{Flag})$ **do**
    $j = Length(\mathbf{Flag}) - m + 1$
    if $\mathbf{Flag}(j) == 1$ **then**
      $\mathbf{v}_{\mathrm{can}}.x = \mathbf{V}_{\mathrm{temp}}(j).x$; $\mathbf{v}_{\mathrm{can}}.y = \mathbf{V}_{\mathrm{temp}}(j).y$; $\mathbf{v}_{\mathrm{can}}.cost = \mathbf{V}_{\mathrm{temp}}(j).cost$;
      Break for
    end if
  end for
end if
Output: $\mathbf{v}_{\mathrm{can}}.x, \mathbf{v}_{\mathrm{can}}.y, \mathbf{v}_{\mathrm{can}}.cost = 0$

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

Input: $\mathbf{T}, \mathcal{E}, P_{\mathrm{g}}, D_{\mathrm{sc}}$.
while $t < Inf$ **do**
  $\mathbf{D}_{\mathrm{temp}} = []$
  $\mathbf{p}_{\mathrm{rand}}.x = Rand(1) \cdot E_x$ /* E_x is the width of $\mathcal{E}$ in x direction */
  $\mathbf{p}_{\mathrm{rand}}.y = Rand(1) \cdot E_y$ /* E_y is the width of $\mathcal{E}$ in y direction */
  for $i = 1$ to $Length(\mathbf{T})$ **do**
    $\mathbf{D}_{\mathrm{temp}}(i) = \sqrt{(\mathbf{p}_{\mathrm{rand}}.x - \mathbf{T}(i).x)^2 + (\mathbf{p}_{\mathrm{rand}}.y - \mathbf{T}(i).y)^2}$
  end for
  $l = MinIndex(\mathbf{D}_{\mathrm{temp}})$
  $d_{\mathrm{pre}} = Atan2d(\mathbf{p}_{\mathrm{rand}}.y - \mathbf{T}(l).y, \mathbf{p}_{\mathrm{rand}}.x - \mathbf{T}(l).x)/45$
  $d_{\mathrm{pre}} = Round(d_{\mathrm{pre}})$
  if $Mod(d_{\mathrm{pre}}, 2) \neq 0$ **then**
    $\mathbf{v}_{\mathrm{can}}.x = \mathbf{T}(l).x + \sqrt{2} \cdot D_{\mathrm{sc}} \cdot \cos(d_{\mathrm{pre}} \cdot 45)$
    $\mathbf{v}_{\mathrm{can}}.y = \mathbf{T}(l).y + \sqrt{2} \cdot D_{\mathrm{sc}} \cdot \sin(d_{\mathrm{pre}} \cdot 45)$
  else
    $\mathbf{v}_{\mathrm{can}}.x = \mathbf{T}(l).x + D_{\mathrm{sc}} \cdot \cos(d_{\mathrm{pre}} \cdot 45)$
    $\mathbf{v}_{\mathrm{can}}.y = \mathbf{T}(l).y + D_{\mathrm{sc}} \cdot \sin(d_{\mathrm{pre}} \cdot 45)$
  end if
  if $Collision(\mathbf{v}_{\mathrm{can}}, \mathcal{E}) == 0$ **then**
    $\mathbf{v}_{\mathrm{can}}.cost = \sqrt{(\mathbf{v}_{\mathrm{can}}.x - x_{\mathrm{g}})^2 + (\mathbf{v}_{\mathrm{can}}.y - y_{\mathrm{g}})^2}$
    Break while
  end if
end while
Output: $\mathbf{v}_{\mathrm{can}}.x, \mathbf{v}_{\mathrm{can}}.y, \mathbf{v}_{\mathrm{can}}.cost$ =0

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

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

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

Input: $\mathbf{v}_i$, $\mathbf{T}, \mathbf{I}_{\text{near}}$.
$D_{\text{temp}} = 0$
for $j = 1$ to $Length(\mathbf{I}_{\text{near}})$ **do**
  $d_{\text{cu}} = \sqrt{(\mathbf{v}_i.x - \mathbf{T}(j).x)^2 + (\mathbf{v}_i.y - \mathbf{T}(j).y)^2} + \mathbf{T}(j).cucost$
  if $d_{\text{cu}} \leq D_{\text{temp}}$ **then**
    $D_{\text{temp}} = d_{\text{cu}}$
    $\mathbf{v}_i.parent = j$
    $\mathbf{v}_i.cucost = d_{\text{cu}}$
  end if
end for
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.

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

Algorithm 6 The $ExtractTra(\mathbf{T}, \mathbf{v}_\mathrm{i}, P_\mathrm{g})$ algorithm. $Flip(\mathbf{a})$ reverses the order of the elements in $\mathbf{a}$.

Input: $\mathbf{T}$, $\mathbf{v}_i$, P_g
$\mathbf{T}_{\mathrm{trajecotry}}(1).x = x_\mathrm{g}$; $\mathbf{T}_{\mathrm{trajecotry}}(1).y = y_\mathrm{g}$; $\mathbf{T}_{\mathrm{trajecotry}}(1).parent = i$
$w = 2$
while $\mathbf{T}(i).parent \neq 1$ **do**
  $\mathbf{T}_{\mathrm{trajecotry}}(w) = \mathbf{T}(\mathbf{T}(i).parent)$
  $w = w + 1$
  $i = \mathbf{T}_(i).parent$
end while
$\mathbf{T}_{\mathrm{trajecotry}}(w).x = x_\mathrm{g}$; $\mathbf{T}_{\mathrm{trajecotry}}(w).y = y_\mathrm{g}$
$\mathbf{T}_{\mathrm{trajecotry}} = Flip(\mathbf{T}_{\mathrm{trajecotry}})$
Output: $\mathbf{T}_{\mathrm{trajecotry}}$ =0

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

Input: $\mathbf{T}_{\text{trajecotry}}$, $\mathcal{E}$, D_{sf}
for $i = 2$ to $Length(\mathbf{T}_{\text{trajecotry}}) - 1$ **do**
  for $r = 1$ to 5 **do**
    for $\eta = 1$ to 18 **do**
      $x_{\text{ser}} = \mathbf{T}_{\text{trajecotry}}(i).x + (r-1) \cdot 0.3 \cdot D_{\text{sf}} \cdot \cos(20 \cdot \eta)$
      $y_{\text{ser}} = \mathbf{T}_{\text{trajecotry}}(i).y + (r-1) \cdot 0.3 \cdot D_{\text{sf}} \cdot \sin(20 \cdot \eta)$
      $D_1 = \sqrt{(x_{\text{ser}} - \mathbf{T}_{\text{trajecotry}}(i-1).x)^2 + (y_{\text{ser}} - \mathbf{T}_{\text{trajecotry}}(i-1).y)^2}$
      $D_2 = \sqrt{(x_{\text{ser}} - \mathbf{T}_{\text{trajecotry}}(i+1).x)^2 + (y_{\text{ser}} - \mathbf{T}_{\text{trajecotry}}(i+1).y)^2}$
      $D_{\text{ser}}(\eta) = D_1 + D_2$
      $n_{\text{collision}} = 0$
      for $k = 1$ to 18 **do**
        $x_{\text{eva}} = x_{\text{ser}}.x + D_{\text{sf}} \cdot \cos(20 \cdot k)$
        $y_{\text{eva}} = x_{\text{ser}}.y + D_{\text{sf}} \cdot \sin(20 \cdot k)$
        if $Collision(\mathbf{a}, \mathcal{E}) \neq 0$ **then**
          $n_{\text{collision}} = n_{\text{collision}} + 1$
        end if
      end for
      $\mathbf{N}_{\text{collision}}(\eta) = n_{\text{collision}}$
      $\mathbf{Cost}(\eta) = 3000 \cdot \mathbf{N}_{\text{collision}}(\eta) + 2 \cdot D_{\text{ser}}(\eta) + 5 \cdot Abs(D_1 - D_2)$ /*the cost function to optimize the position of the updated trajectory point*/
    end for
    $l = MinIndex(\mathbf{Cost})$
    $\mathbf{L}(r) = l$
    $\mathbf{C}(r) = \mathbf{Cost}(l)$
  end for
  $r_{\text{global}} = MinIndex(\mathbf{C})$
  $\eta_{\text{global}} = \mathbf{L}(r_{\text{global}})$
  $\mathbf{T}_{\text{trajecotry}}(i).x = \mathbf{T}_{\text{trajecotry}}(i).x + (r_{\text{global}} - 1) \cdot 0.3 \cdot D_{\text{sf}} \cdot \cos(20 \cdot \eta_{\text{global}})$
  $\mathbf{T}_{\text{trajecotry}}(i).y = \mathbf{T}_{\text{trajecotry}}(i).y + (r_{\text{global}} - 1) \cdot 0.3 \cdot D_{\text{sf}} \cdot \sin(20 \cdot \eta_{\text{global}})$
end for
Output: $\mathbf{T}_{\text{trajecotry}}$ =0

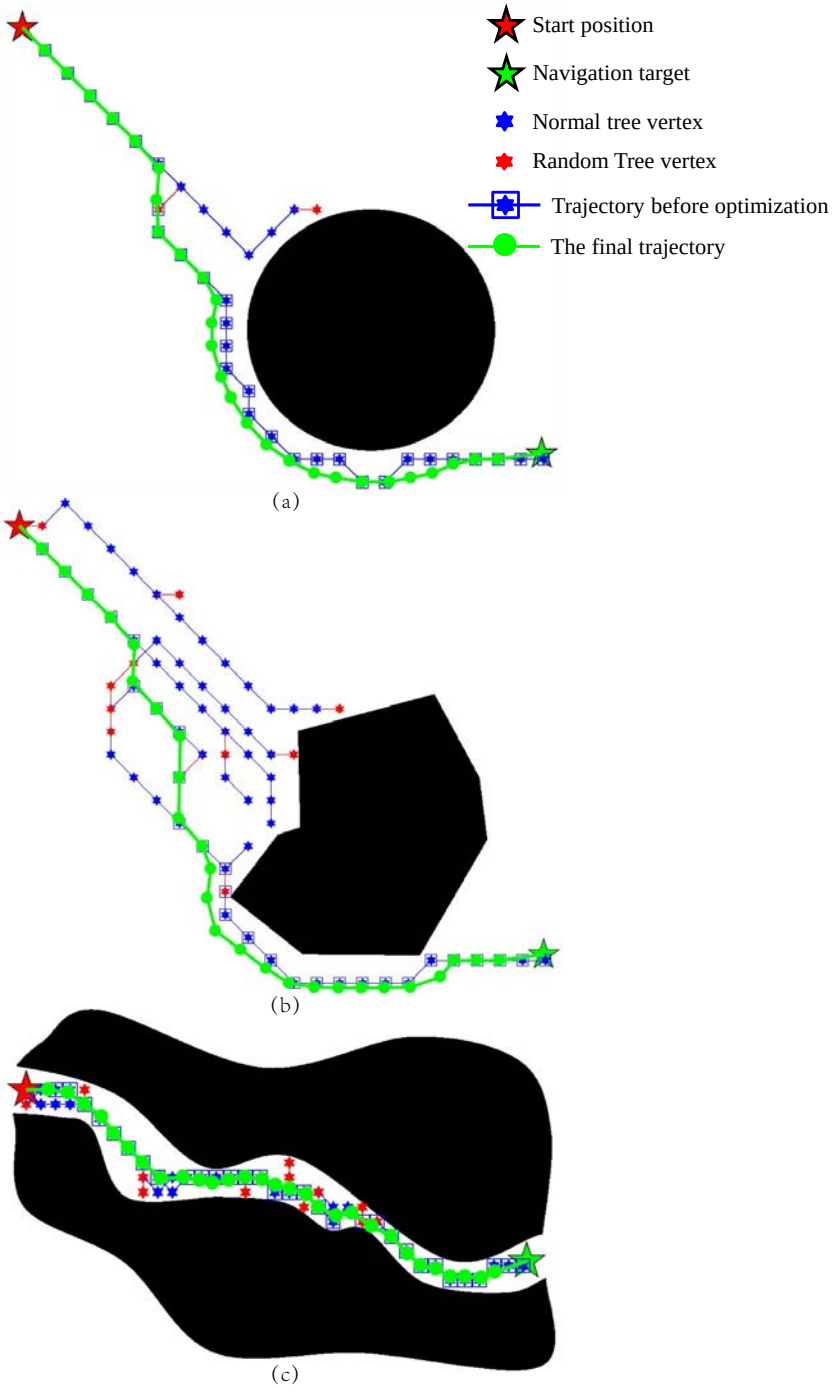


Fig. S18. Illustration of the autonomous trajectory planning results for different environment morphologies. **a**, A circular obstacle with a trajectory planning time of 0.1 s. **b**, An irregular obstacle with a trajectory planning time of 0.15 s. **c**, An environment with the channel morphology with 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.

Movie S1. The robustness validation of the DNN-based swarm distribution planning method for different environment morphologies, and the comparison of computation time between this method and the traversal-based optimization method.

Movie S2. Experimental demonstration of the three swarm configurations and the transformations among them.

Movie S3. Experimental demonstration of the translational motion and rotational motion of the ribbon-like swarm (RS).

Movie S4. Experimental comparisons between manual navigation (autonomy Level 0) and automated control (autonomy Level 1).

Movie S5. Experimental demonstration of autonomy Level 2, where two sets of experiments with different swarm sizes were conducted to assess the intelligence of the DNN-based swarm distribution planning method.

Movie S6. Experimental demonstration of autonomy Level 3, where a swarm of magnetic nanorobots accomplished the delivery task to a targeted region in a channel environment.

Movie S7. Experimental demonstration of autonomy Level 3, where the reconfigurable magnetic nanorobot swarm navigated in highly curved space with a 150° sharp turn and in a curved narrow channel environment.

Movie S8. Experimental demonstration of autonomy Level 3, where the reconfigurable magnetic nanorobot swarm executed the cooperative micromanipulation task in confined space.

Movie S9. Experimental demonstration of the fully autonomous environment exploration using the magnetic nanorobot swarm in an unknown channel environment (autonomy Level 4).

Movie S10. Experimental demonstration of the fully autonomous targeted delivery to a region under dynamic obstacles (autonomy Level 4).

Movie S11. Transfer the autonomy framework to the elliptical vortex-like magnetic nanoparticle swarm.

Movie S12. Method Validation under ultrasound imaging and x-ray fluoroscopy.

References

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