Supplementary Information (SI) for Lab on a Chip.

# Supporting Information for

## Deep learning-driven microfluidic chip architecture design for intelligent particle motion control

Hongxia Li[a,*,#], Xuhui Chen[a,#], Du Qiao[a,#], Xue Zhang[a], Jiang Zhang[a], Jianan Zou[a], Danyang Zhao[a], Xuhong Qian[b], Honglin Li[b,*]

**a** State Key Laboratory of High-Performance Precision Manufacturing, Dalian University of Technology, Dalian 116023, China
**b** Innovation Center for AI and Drug Discovery, School of Pharmacy, East China Normal University, Shanghai 200062, China
# These authors contributed equally to this work.

The PDF file includes:

- Supplementary note 1 to 4
- Supplementary figures 1 to 10
- Supplementary tables 1 to 5
- Supplementary videos 1 to 4

# Supplementary Note 1

To validate the effectiveness of the proposed numerical model, a physical experiment was conducted. A representative set of model parameters was randomly selected (excluding the inlet particle ratio $I_R$) for experimental verification, as listed in Supplementary Tab. 2.

Based on the parameters in Supplementary Tab. 2, a photomask was fabricated and used to produce a PDMS microfluidic chip. A 5 mL suspension of 1 µm polystyrene (PS) microspheres was prepared and loaded into a 10 mL syringe. The inlet ports were connected to syringe pumps according to the selected flow parameters, with prescribed inlet flow rates. The PS suspension syringe was connected to inlet port 1 via tubing, and the flow was initiated at a fixed rate. Particle motion was recorded using a high-speed camera and saved for analysis. Under identical parameter settings, a numerical simulation model was constructed, with all configurations kept constant except for the number of input particles, which was set to 200 to better visualize particle distribution under different inlet ratios. A comparison between experimental results and simulation trajectories (Supplementary Fig. 2) demonstrates strong agreement, confirming the reliability of the numerical model for generating training data for trajectory prediction.

# Supplementary Note 2

To simplify the sequential data of continuous particle trajectories (i.e., two-dimensional spatial coordinates over time), only the particle position at the module exit—defined as the cross-section location at a fixed distance downstream of the channel—is retained for each module. This position is represented as a one-dimensional scalar along the cross-sectional width. Additionally, the particle outlet at each branching junction is treated as a categorical label to avoid unphysical predictions beyond the channel boundaries. The model also predicts the instantaneous velocity of the particle at the cross-section, along with the transit time across the module. A schematic of the predicted parameters is shown in Supplementary Fig. 3.

Here, $O_L$ denotes the outlet label, $O_L \in \{2,3,4\}$, $D$ is the channel width at the outlet, and $d$ is the vertical distance from the particle to the left channel wall at the module exit. The normalized outlet particle position is defined as $O_R = d / D$. Similarly, the inlet ratio $I_R$ is defined as $I_R = d_i / D_i$, where $d_i$ is the vertical distance from the particle to the left wall at the module entry, and $D_i$ is the corresponding inlet channel width for module/node $i$. $V_X$ and $V_Y$ denote the particle velocity components in the $x$ and $y$ directions at the outlet, respectively, and $T$ is the time taken for the particle to traverse from the module inlet to the exit location.

## Supplementary Note 3

Deep learning was implemented in MATLAB using a multi-layer residual backpropagation neural network to predict particle trajectories. The input layer receives 20 features capturing the influence of flow conditions, channel geometry, and particle properties on the trajectory. The output corresponds to a single node encoding multiple outcomes, including the outlet label, particle position along the cross-section, local flow velocity at the outlet, and transit time through the module.

For outlet label classification, categorical labels were transformed into one-hot encoded vectors, mapping each class to an independent binary representation. This avoids introducing numerical bias and facilitates the network's handling of discrete categorical features. The dataset was subsequently normalized, randomly shuffled, and split into training and testing subsets in an 8:2 ratio, enabling later evaluation of classification accuracy on the test set. The network architecture comprises an input layer with 20 neurons (encoding position ratios, flow conditions, channel structures, and particle attributes), followed by 20 hidden layers constructed using residual blocks. Each residual block contains two fully connected sublayers—the first with 20 neurons and the second with 25 neurons—and incorporates skip connections to mitigate the degradation problem associated with deep architectures. ReLU activation is applied to all hidden layers. The output layer contains three neurons (corresponding to the three-bit one-hot encoding of outlet labels), equipped with a softmax activation function to yield class probabilities. Cross-entropy loss was employed as the objective function (Eq. 1), and model optimization was performed using the NADAM optimizer with adaptive learning rate scheduling.

$$CrossEntropy = -\frac{1}{n}\sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i) \qquad (1)$$

Here, $y_i$ denote the ground truth of the $i$-th training instance and $\hat{y}_i$ is the corresponding prediction, where $n$ is the total number of samples. The NADAMs optimizer, a refined variant combining the Adam and Nesterov Accelerated Gradient (NAG) methods, enhances gradient updates by incorporating Nesterov momentum. This allows the optimizer to anticipate the trajectory of parameter updates early during training, enabling faster convergence and improved training stability. To further accelerate convergence and improve computational efficiency, mini-batch training with an appropriate batch size is employed. Supplementary Fig. 8 illustrates the decrease in cross-entropy loss on the training set and the corresponding accuracy trend on the test set during classification.

For the prediction of normalized position, outlet velocity, and particle arrival time—formulated as regression tasks—the same normalization, shuffling, and train-test split procedures are applied as in the classification task. The regression networks adopt a similar architecture: 20 input neurons encoding features related to position ratio, flow conditions, channel geometry, and particle properties. The networks predicting position and arrival time comprise 20 residual layers, while the velocity prediction network uses 10 residual layers. Each residual block contains two fully connected layers with 50 and 100 neurons respectively, equipped with ReLU activations. The output layers consist of a single neuron for position and time prediction, and two neurons for velocity prediction (representing x- and y-direction velocities). The models are trained using the L1 loss, minimized by the NADAMs optimizer:

$$L1=\frac{1}{N}\sum_{i=1}^{N}|y_i-\hat{y}_i| \tag{2}$$

where $N$ is the total number of samples, $y_i$ denotes the ground truth, and $\hat{y}_i$ is the model prediction. Supplementary Fig. 9 shows the decrease in L1 loss during training and the corresponding accuracy trend on the test set.

## Supplementary Note 4

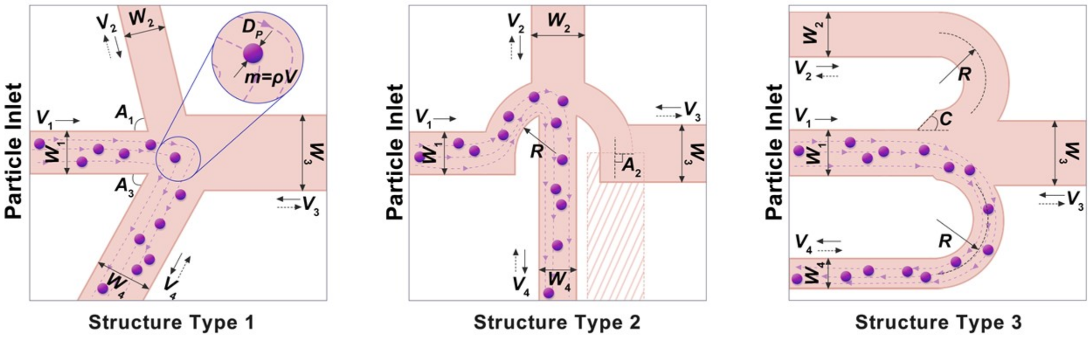
The microfluidic channel layout was retroactively designed based on predefined particle trajectories or designated outlet positions. To reconstruct device structures from the predicted sequence of trajectory nodes or the identified terminal node index, the following assumptions were made:

1. **Modules are assumed to be seamlessly connected**, or any additional interconnection length is negligibly small and thus treated as infinitesimal in the analysis.
2. **The inter-module connection segments are assumed to exert negligible influence** on the information received ($I_R$) by the downstream module, which is computed via a logical $O_R$ operation based on the upstream state.
3. **Inlet flow conditions are presumed to remain invariant**, such that structural modifications within the device affect only the local flow characteristics within individual modules.
4. **The geometric configuration of the channels is assumed not to perturb the flow state at the interface between adjacent modules**.

These assumptions ensure that:

1. Assumption 1 allows for consistent fluid velocity across connecting segments, enabling the approximation of flow continuity between modules.
2. Assumption 2 safeguards the validity of spatial parameter transmission across modules by isolating it from transitional segment effects.
3. Assumption 3 guarantees that particle trajectories are modulated solely by internal geometric changes, independent of inlet disturbances.
4. Assumption 4 preserves the modular independence of local flow fields, preventing unintended cross-influence via interface geometry.

# Supplementary Figures

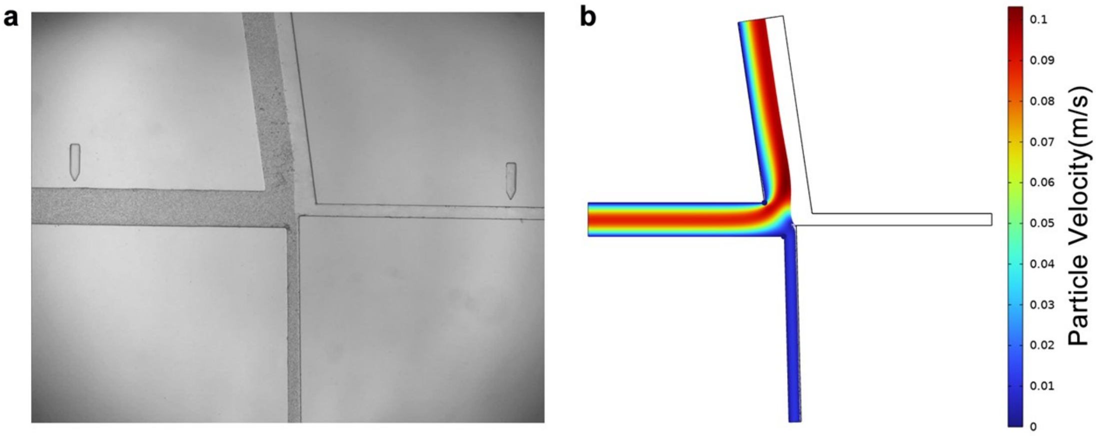


Structure Type 1

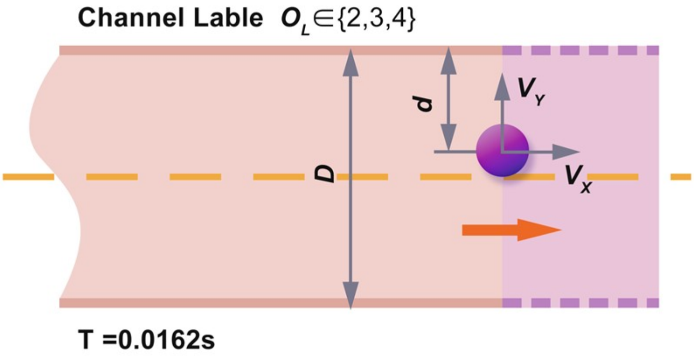
Structure Type 2

Structure Type 3

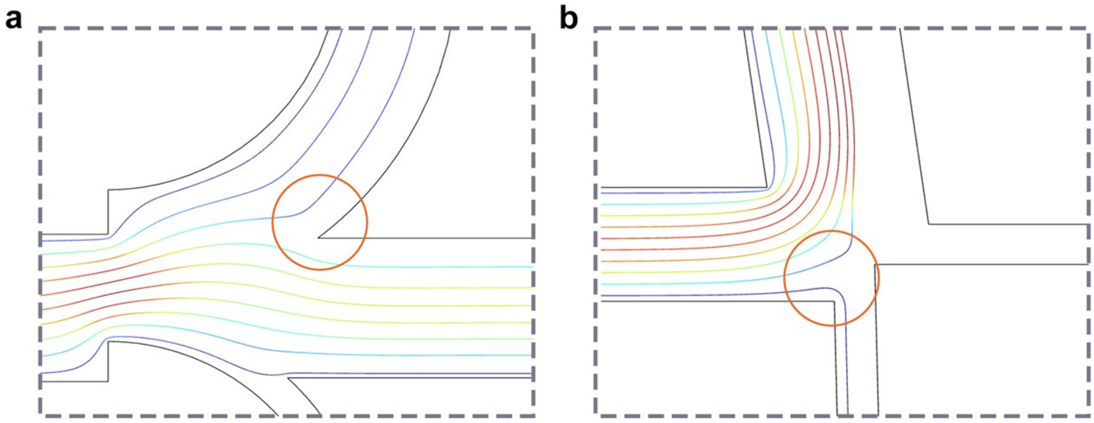
**Supplementary Fig. 1. Illustration of flow, channel and particle design parameters.**



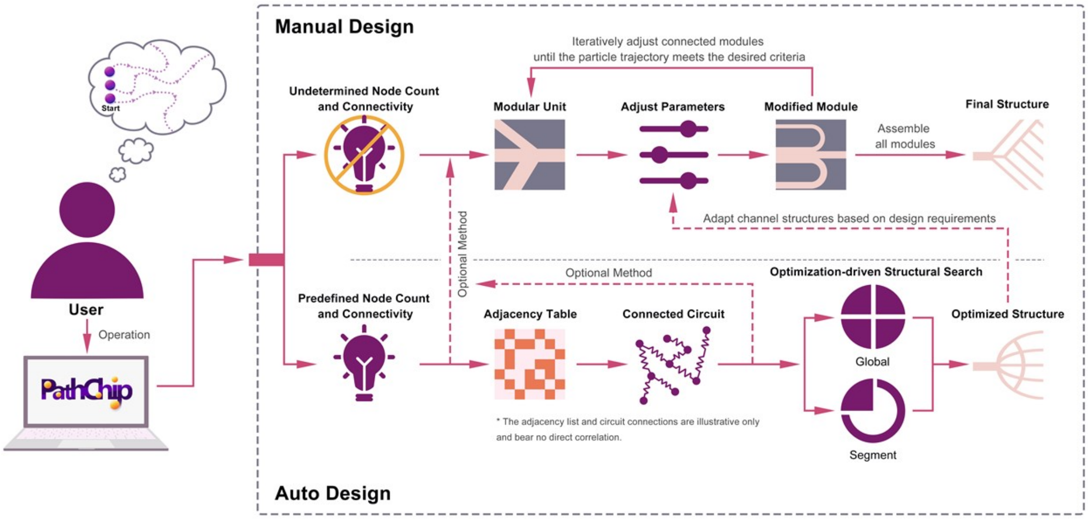
**Supplementary Fig. 2. Comparison between experimental and simulation results.**
**a** Physical experiment. **b** Numerical simulation.



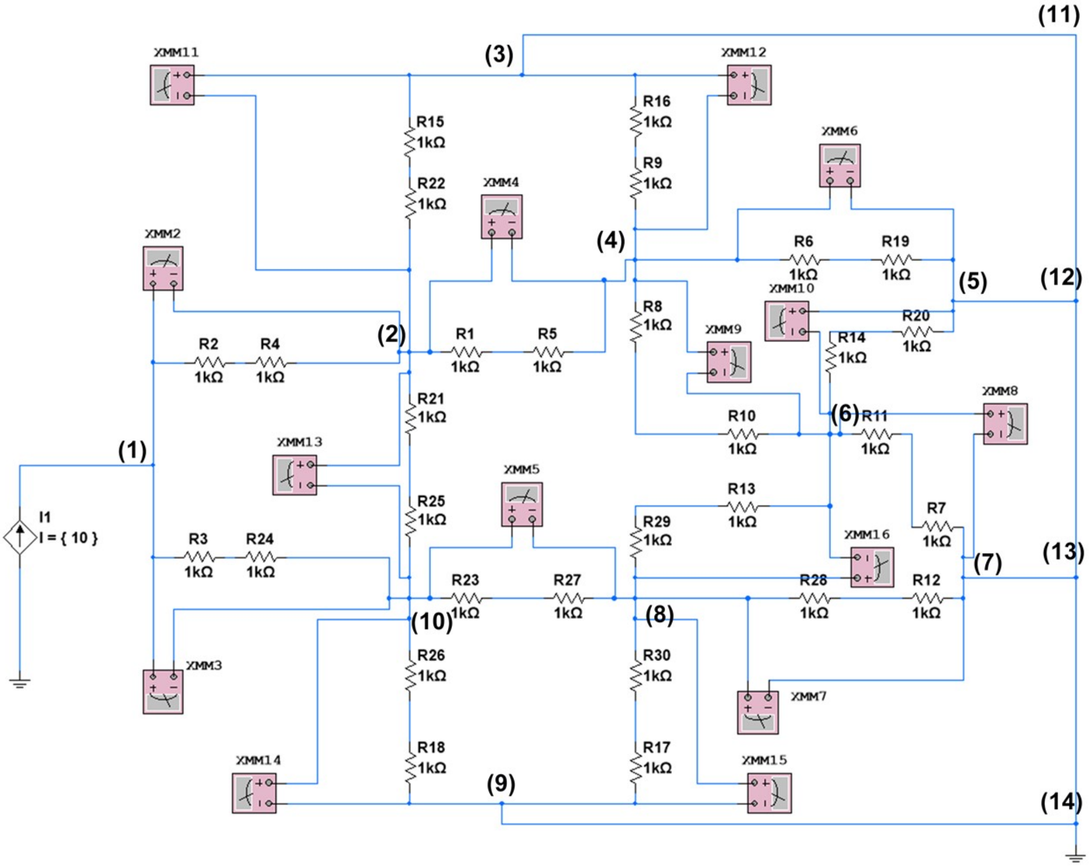
**Supplementary Fig. 3. Schematic illustration of output parameters.**



**Supplementary Fig. 4. Ambiguous regions in trajectory classification.**
Orange circles indicate regions prone to classification ambiguity, primarily arising from: **a** particles entering adjacent branches within spatially close bifurcations, or **b** narrow high-curvature regions near multiple outlet junctions that induce trajectory divergence.

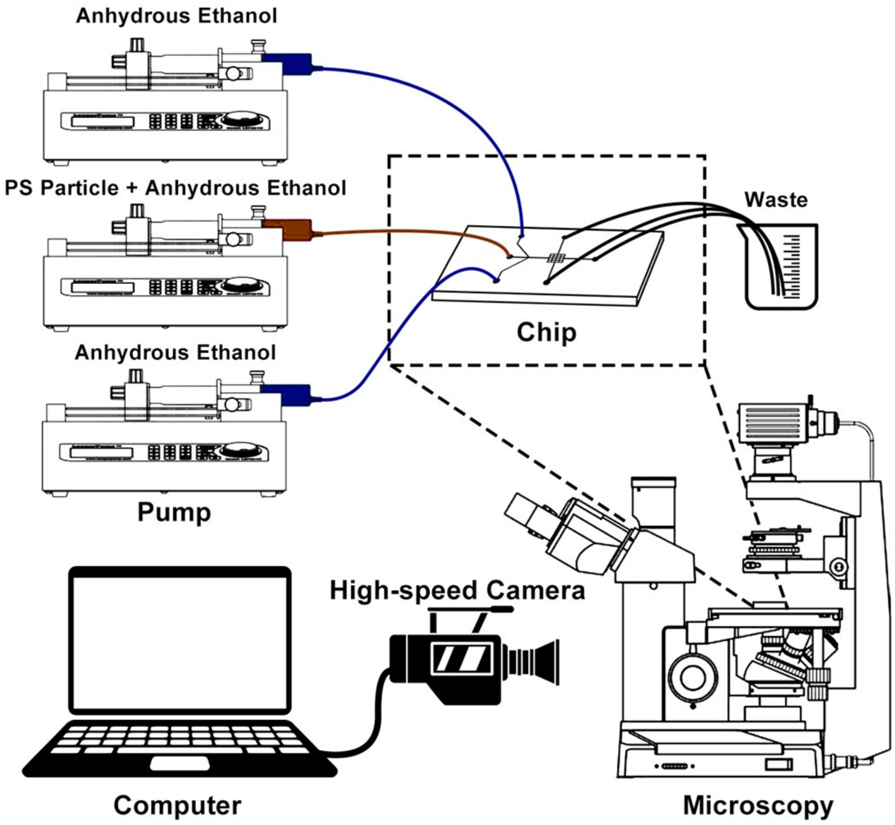


**Supplementary Fig. 5. Workflow of manual and automated channel design.**
Users can interactively assemble modular microchannel geometries in PathChip by combining three base channel types, allowing them to generate trajectory-guided structures that meet functional specifications under predefined flow field conditions. Manual forward design is natively supported within PathChip without external dependencies. In contrast, automated inverse design represents an extended functionality requiring integration of adjacency matrices and circuit-based logic modules.



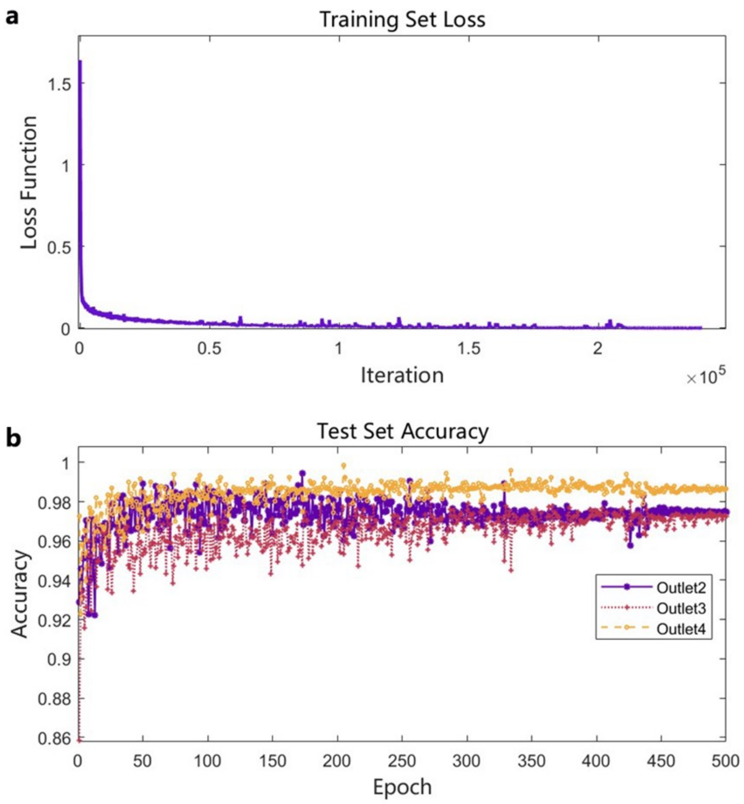
**Supplementary Fig. 6. Circuit configuration derived from adjacency matrix.**

Each labeled node corresponds to an index in the adjacency matrix. A fixed 10 A current is applied at the inlet, and voltmeters are used to measure voltage drops across resistors for current computation. The initial circuit design assumes uniform resistance $R$=1 k$\Omega$, disregarding length-dependent effects of channel segments.



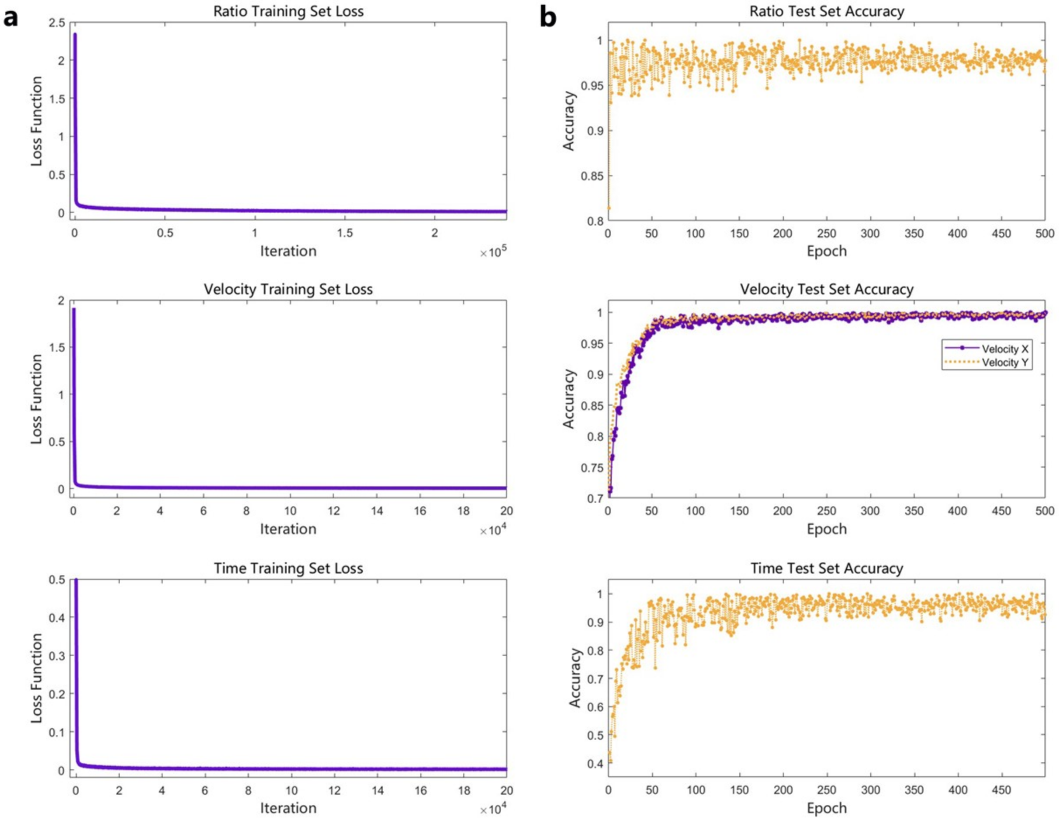
**Supplementary Fig. 7. Schematic of the experimental setup.**

A diagrammatic overview of the physical platform used for flow field generation, particle tracking, and validation.



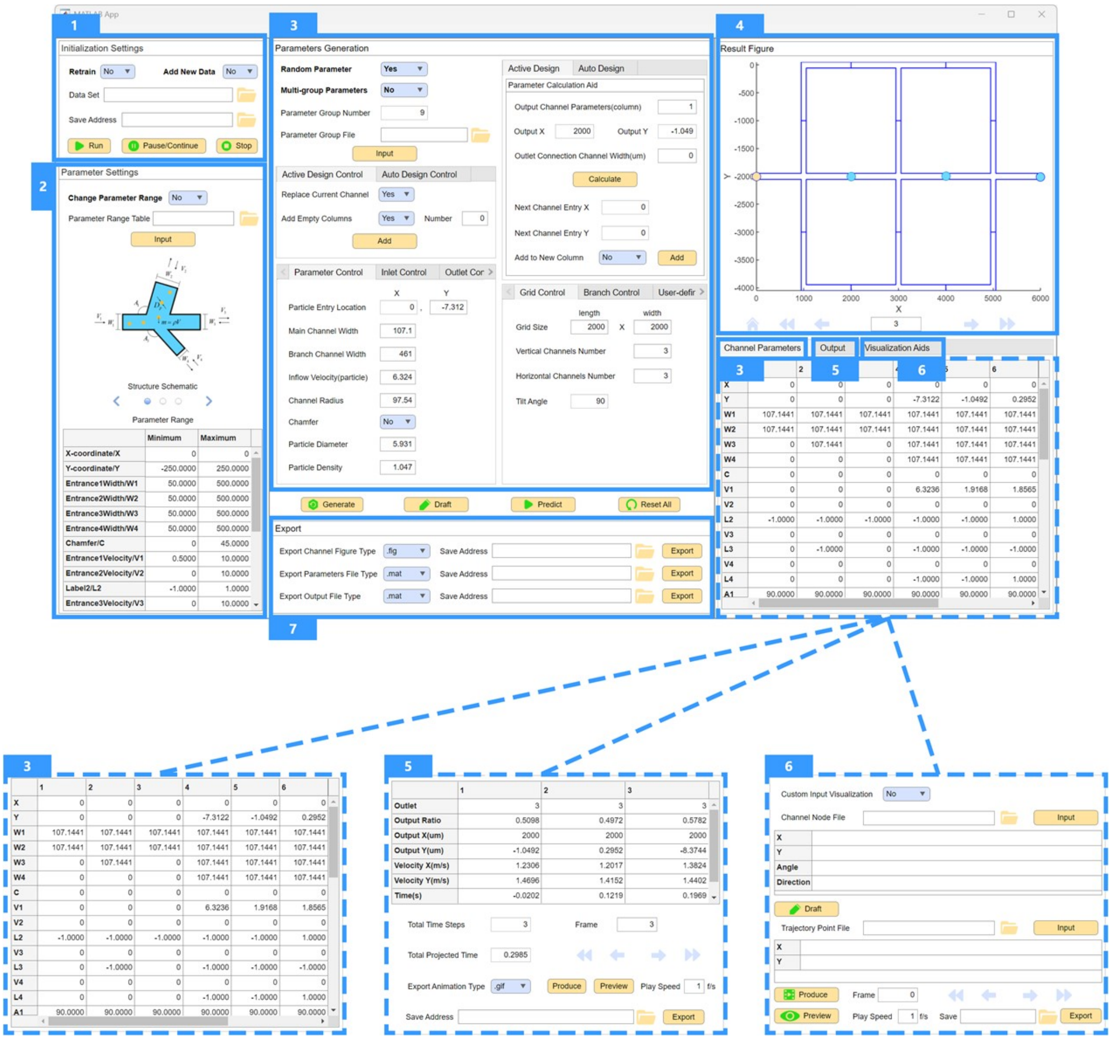
**Supplementary Fig. 8. Training performance of the classification neural network.**

**a** The optimized classification model exhibits a decreasing training loss over epochs. **b** Accuracy on the test set improves progressively during training.



**Supplementary Fig. 9. Training performance of the regression neural network.**

**a** The optimized regression model shows reduced training loss; this model is used to predict outlet position ratio, outlet velocity, and time to arrival. **b** Accuracy on the test set increases as training progresses.



Initialization Settings

Retrain: No
Add New Data: No
Data Set
Save Address

Parameter Settings

Change Parameter Range: No
Parameter Range Table

Parameter Range

| | Minimum | Maximum |
|---|---|---|
| X-coordinate/X | 0 | 0 |
| Y-coordinate/Y | -250.0000 | 250.0000 |
| Entrance1Width/W1 | 50.0000 | 500.0000 |
| Entrance2Width/W2 | 50.0000 | 500.0000 |
| Entrance3Width/W3 | 50.0000 | 500.0000 |
| Entrance4Width/W4 | 50.0000 | 500.0000 |
| Chamfer/C | 0 | 45.0000 |
| Entrance1Velocity/V1 | 0.5000 | 10.0000 |
| Entrance2Velocity/V2 | 0 | 10.0000 |
| Label2/L2 | -1.0000 | 1.0000 |
| Entrance3Velocity/V3 | 0 | 10.0000 |

Parameters Generation

Random Parameter: Yes
Multi-group Parameters: No
Parameter Group Number: 9
Parameter Group File

Active Design Control | Auto Design Control
Replace Current Channel: Yes
Add Empty Columns: Yes
Number: 0

Parameter Control | Inlet Control | Outlet Cor
Particle Entry Location: X 0, Y -7.312
Main Channel Width: 107.1
Branch Channel Width: 461
Inflow Velocity(particle): 6.324
Channel Radius: 97.54
Chamfer: No
Particle Diameter: 5.931
Particle Density: 1.047

Active Design | Auto Design
Parameter Calculation Aid
Output Channel Parameters(column): 1
Output X: 2000
Output Y: -1.049
Outlet Connection Channel Width(um): 0
Next Channel Entry X: 0
Next Channel Entry Y: 0
Add to New Column: No

Grid Control | Branch Control | User-defir
Grid Size: length 2000 X width 2000
Vertical Channels Number: 3
Horizontal Channels Number: 3
Tilt Angle: 90

Export

Export Channel Figure Type: .fig
Save Address
Export Parameters File Type: .mat
Save Address
Export Output File Type: .mat
Save Address

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| X | 0 | 0 | 0 | 0 | 0 | 0 |
| Y | 0 | 0 | 0 | -7.3122 | -1.0492 | 0.2952 |
| W1 | 107.1441 | 107.1441 | 107.1441 | 107.1441 | 107.1441 | 107.1441 |
| W2 | 107.1441 | 107.1441 | 107.1441 | 107.1441 | 107.1441 | 107.1441 |
| W3 | 0 | 107.1441 | 0 | 107.1441 | 107.1441 | 107.1441 |
| W4 | 0 | 0 | 0 | 107.1441 | 107.1441 | 107.1441 |
| C | 0 | 0 | 0 | 0 | 0 | 0 |
| V1 | 0 | 0 | 0 | 6.3236 | 1.9168 | 1.8565 |
| V2 | 0 | 0 | 0 | 0 | 0 | 0 |
| L2 | -1.0000 | -1.0000 | -1.0000 | -1.0000 | -1.0000 | 1.0000 |
| V3 | 0 | 0 | 0 | 0 | 0 | 0 |
| L3 | 0 | -1.0000 | 0 | -1.0000 | -1.0000 | -1.0000 |
| V4 | 0 | 0 | 0 | 0 | 0 | 0 |
| L4 | 0 | 0 | 0 | -1.0000 | -1.0000 | 1.0000 |
| A1 | 90.0000 | 90.0000 | 90.0000 | 90.0000 | 90.0000 | 90.0000 |

| | 1 | 2 | 3 |
|---|---|---|---|
| Outlet | 3 | 3 | 3 |
| Output Ratio | 0.5098 | 0.4972 | 0.5782 |
| Output X(um) | 2000 | 2000 | 2000 |
| Output Y(um) | -1.0492 | 0.2952 | -8.3744 |
| Velocity X(m/s) | 1.2306 | 1.2017 | 1.3824 |
| Velocity Y(m/s) | 1.4696 | 1.4152 | 1.4402 |
| Time(s) | -0.0202 | 0.1219 | 0.1969 |

Total Time Steps: 3
Frame: 3
Total Projected Time: 0.2985
Export Animation Type: .gif
Play Speed: 1 f/s
Save Address

Custom Input Visualization: No
Channel Node File

| | |
|---|---|
| X | |
| Y | |
| Angle | |
| Direction | |

Trajectory Point File

| | |
|---|---|
| X | |
| Y | |

Frame: 0
Play Speed: 1 f/s
Save

**Supplementary Fig. 10. Main interface and functional layout of the application.**

1. Initialization panel; 2. Parameter configuration panel; 3. Parameter generation panel; 4. Visualization window; 5. Trajectory output panel; 6. Visualization assistance panel; 7. Export panel.

# Supplementary Tables

**Supplementary Tab. 1. Parameter settings and corresponding ranges.**

| No. | Parameter Name | Symbol | Range | Unite | Explanation |
|---|---|---|---|---|---|
| 1 | Inlet Ratio | $I_R$ | 0~1 | N/A | The position of particles in the cross section at the inlet |
| 2 | Inlet 1 Width | $W_1$ | 50~500 | μm | Width of channel 1 |
| 3 | Inlet 2 Width | $W_2$ | 50~500 | μm | Width of channel 2 |
| 4 | Inlet 3 Width | $W_3$ | 50~500 | μm | Width of channel 3 |
| 5 | Inlet 4 Width | $W_4$ | 50~500 | μm | Width of channel 4 |
| 6 | Chamfer | $C$ | 0/45 | ° | Chamfer angle of the joint |
| 7 | Inlet 1 Velocity | $V_1$ | 0.5~10 | cm/s | Inlet flow velocity of channel 1 |
| 8 | Inlet 2 Velocity | $V_2$ | 0~10 | cm/s | Inlet flow velocity of channel 2 |
| 9 | Label 2 | $L_2$ | -1/0/1 | N/A | Mark channel 2 as outflow/none/inflow |
| 10 | Inlet 3 Velocity | $V_3$ | 0~10 | cm/s | Inlet flow velocity of channel 3 |
| 11 | Label 3 | $L_3$ | -1/0/1 | N/A | Mark channel 3 as outflow/none/inflow |
| 12 | Inlet 4 Velocity | $V_4$ | 0~10 | cm/s | Inlet flow velocity of channel 4 |
| 13 | Label 4 | $L_4$ | -1/0/1 | N/A | Mark channel 4 as outflow/none/inflow |
| 14 | Angle 1 | $A_1$ | 30~150 | ° | Angle between channel 1 and channel 2 |
| 15 | Angle 2 | $A_2$ | 0/90 | ° | Angle between channel 1 and channel 3 |
| 16 | Angle 3 | $A_3$ | 30~150 | ° | Angle between channel 1 and channel 4 |
| 17 | Structure Type | $K$ | 0/1/2 | N/A | Channel structure type No. |
| 18 | Radius | $R$ | 0~1000 | μm | Radius of curved channel |

| | | | | | |
|---|---|---|---|---|---|
| 19 | Particle Diameter | $D_P$ | 0.5~20 | μm | Diameter of the particles |
| 20 | Particle Density | $\rho$ | 0.5~1.5 | g/cm3 | Density of the particles |

*Note: The explicit formulation of $I_R$ (Inlet Ratio) is provided in Supplementary Note 2: Definition of particle position ratio at the outlet.*

**Supplementary Tab. 2. Sampled parameter set used for experimental validation.**

| No. | Param. | Val. | No. | Param. | Val. | No. | Param. | Val. |
|---|---|---|---|---|---|---|---|---|
| 1 | $W_1$ | 333 (μm) | 8 | $L_2$ | -1 | 15 | $A_3$ | 81(° ) |
| 2 | $W_2$ | 121 (μm) | 9 | $V_3$ | 9 (cm/s) | 16 | $K$ | 0 |
| 3 | $W_3$ | 117 (μm) | 10 | $L_3$ | 1 | 17 | $R$ | 0 (μm) |
| 4 | $W_4$ | 452 (μm) | 11 | $V_4$ | 0 (cm/s) | 18 | $D_P$ | 1 (μm) |
| 5 | $C$ | 0(° ) | 12 | $L_4$ | -1 | 19 | $\rho$ | 1.05 (g/cm$^3$) |
| 6 | $V_1$ | 6 (cm/s) | 13 | $A_1$ | 91(° ) | | | |
| 7 | $V_2$ | 0 (cm/s) | 14 | $A_2$ | 0(° ) | | | |

**Supplementary Tab. 3. Performance of PMPM regression for predicting particle motion**

| Parameter | Symbol | R$^2$ | RMSE | Unit | MAE | Unit |
|---|---|---|---|---|---|---|
| Outlet Position | $O_R$ | 0.825±0.002 | 0.114±0.006 | N/A | 0.022±0.007 | N/A |
| Velocity X | $V_X$ | 0.956±0.011 | 0.013±0.002 | m/s | 0.005±0.002 | m/s |
| Velocity Y | $V_Y$ | 0.946±0.039 | 0.007±0.001 | m/s | 0.003±0.002 | m/s |
| Passage Time | $T$ | 0.775±0.112 | 0.022±0.012 | s | 0.006±0.002 | s |

**Supplementary Tab. 4. Kirchhoff's current and voltage equations derived from the adjacency matrix in Supplementary Fig. 6.**

The number of equations must be greater than or equal to the number of unknown branch currents to ensure solvability. The listed equations are formulated into a linear system and solved in MATLAB to obtain the magnitude and direction of each branch current.

| No. | Kirchhoff's Current Equition | Kirchhoff's Voltage Equition |
|---|---|---|
| 1 | $I_{16}=I_1-I_{10}$ | $\left(I_{11}+I_{12}+I_{13}+I_{14}+I_{15}\right)R=0$ |

| | | |
|---|---|---|
| 2 | $I_{17}=I_3-I_2$ | $\left(I_1+I_{12}+I_{13}+I_{14}+I_{15}+I_{10}\right)R=0$ |
| 3 | $I_{18}=I_5-I_4$ | $\left(I_2+I_3+I_{13}+I_{14}+I_{15}+I_{11}\right)R=0$ |
| 4 | $I_{19}=I_7-I_6$ | $\left(I_4+I_5+I_{12}+I_{14}+I_{15}+I_{11}\right)R=0$ |
| 5 | $I_{20}=I_9-I_8$ | $\left(I_6+I_7+I_{12}+I_{13}+I_{15}+I_{11}\right)R=0$ |
| 6 | $I_1+I_{11}=I_2+I_{12}$ | $\left(I_8+I_9+I_{11}+I_{13}+I_{14}+I_{12}\right)R=0$ |
| 7 | $I_3+I_{12}=I_4+I_{13}$ | $\left(I_1+I_{12}+I_{13}+I_6+I_7+I_{15}+I_{10}\right)R=0$ |
| 8 | $I_{13}+I_5=I_6+I_{14}$ | $\left(I_2+I_3+I_{13}+I_6+I_7+I_{15}+I_{11}\right)R=0$ |
| 9 | $I_{14}+I_7=I_8+I_{15}$ | $\left(I_4+I_5+I_{14}+I_{15}+I_{10}+I_1+I_{12}\right)R=0$ |
| 10 | $I_{15}+I_9=I_{10}+I_{11}$ | $\left(I_8+I_9+I_{11}+I_2+I_3+I_{13}+I_{14}\right)R=0$ |

**Supplementary Tab. 5. The number of modules and the corresponding AID algorithm computation time**

| Modules Number | Average Computation Time (s) | Time Standard Deviation (s) |
|---|---|---|
| 10 | 0.710 | 0.04295 |
| 50 | 0.897 | 0.04448 |
| 100 | 1.025 | 0.04625 |
| 500 | 2.057 | 0.47286 |
| 1000 | 2.995 | 0.48312 |
| 5000 | 18.821 | 1.80655 |