micromachines

MDPI

Article

ANN-Based Instantaneous Simulation of Particle Trajectories in Microfluidics

Naiyin Zhang [1], Kaicong Liang [2], Zhenya Liu [2], Taotao Sun [2] and Junchao Wang [2,*]

[1] School of Automation, Hangzhou Dianzi University, Hangzhou 310018, China
[2] Key Laboratory of RF Circuits and Systems, Ministry of Education, Hangzhou Dianzi University, Hangzhou 310018, China
* Correspondence: junchao@hdu.edu.cn

Abstract: Microfluidics has shown great potential in cell analysis, where the flowing path in the microfluidic device is important for the final study results. However, the design process is time-consuming and labor-intensive. Therefore, we proposed an ANN method with three dense layers to analyze particle trajectories at the critical intersections and then put them together with the particle trajectories in straight channels. The results showed that the ANN prediction results are highly consistent with COMSOL simulation results, indicating the applicability of the proposed ANN method. In addition, this method not only shortened the simulation time but also lowered the computational expense, providing a useful tool for researchers who want to receive instant simulation results of particle trajectories.

Keywords: microfluidics; machine learning; particle trajectory; design automation; computer-aided design

check for updates

Citation: Zhang, N.; Liang K.; Liu, Z.; Sun, T.; Wang, J. ANN-Based Instantaneous Simulation of Particle Trajectories in Microfluidics. *Micromachines* **2022**, *13*, 2100. https://doi.org/10.3390/mi13122100

Academic Editor: Stéphane Colin

Received: 2 November 2022
Accepted: 27 November 2022
Published: 28 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

As an emerging technology in biological analysis, microfluidics has shown great potential in isolating cells and enriching rare cells in the past decades due to the advantages of the small quantity of sample consumption, ease of operation, and high accuracy [1]. Researchers have developed different types of microfluidic cell sorters for separating cells such as circulating tumor cells [2,3], platelets [4–6], T-cells [7,8], target single cells [9,10], and other biological markers from blood samples or cell suspensions extracted from interest biological areas [8,11,12]. The sorting methods are developed based on different properties of the cells such as size [13,14] and affinity [7,8], or different technologies such as dielectrophoresis [5], magnetic activation [15], fluorescence activation [16,17], image recognition [18,19], vortex [20], and acoustic waves [3,6,16].

The above-mentioned microfluidic cell sorters provide sufficient solutions to the target goals, however, it is undeniable that those devices are usually designed by hand based on the experience of researchers, which can take months or even years to design. For example, in a deterministic lateral displacement (DLD) sorter, the target cells can differ in size, morphology, and affinity to fluorescence dyes as well as fluorescence intensities, which makes it extremely important to design proper pillar arrays for cell detection and separation [21]. These design units include pillar shape, size, angle, the distance between each other, the length of the path, etc., and the whole process can be repeated multiple times from design, fabrication, test, and design revision to correct cell flow trajectory, which is time-consuming and labor-intensive. Therefore, an economic method to predict cell trajectories becomes an urgent need. Researchers have developed tools utilizing computer simulation such as finite element analysis (FEA) to test and optimize their designs before chip fabrication [22–24]. In simulations, the biological cells can be treated as particles with parameters mimicking reality, while the particle trajectory simulation result can be used to optimize the design in return until a qualified chip structure is obtained.

However, the FEA method for simulating particle behaviors in microfluidic devices is not only computationally expensive and labor-intensive but also hard for researchers to learn and use. Specifically, you need to use the Laminar Flow module to solve the velocity field, then use the Particle Tracing module to solve the cell trajectory, which has a steep learning curve for engineers, and takes a long time for researchers to fully master without considering lots of tricky configurations in COMSOL. A promising solution is applying machine learning techniques to accelerate the simulation process by replacing the FEA simulation kernel [25–27]. Among all machine learning techniques, artificial neural networks (ANNs) play an important role in highly complex particle trajectory analyses [28,29]. Briefly, an ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. The ANN receives signals (input data) and then processes (does a few calculation steps) to generate output signals as output data. The output data will have different physical meanings in the corresponding fields such as computer vision [30], natural language process [31], and pattern recognition [32].

To partially address the above issue, a simulation method was presented in our previous work that decomposed microfluidic devices into units including channels and intersections and constructed a database containing almost ten thousand pre-simulated channel intersections [33]. That work has accelerated the simulation process and lowered the need for advanced computational devices, but the limitations of FEA still exist. Therefore, a cloud database was constructed subsequently using an improved method inspired by dynamic programming to accelerate simulations of particle behaviors in the channels [34]. In this work, a further step was taken by increasing the computational rate and reducing the computational time by leveraging machine learning techniques. Specifically, an ANN was proposed to predict the instant particle trajectory at critical intersections in microfluidic devices.

2. Theory of the Design

In most microfluidic devices, the channels are classical "H" channels and can be decomposed into two types: straight channels and intersections (Figure 1A). In those straight channels, particles usually flow along the channel with their trajectories showing straight lines (Figure 1A (unit a, c, d, e, f, g, and i)), which is akin to the flow of electricity in a network of resistors and can be modeled using principles of electrical circuit analysis. Therefore, particle behaviors at critical intersections are crucial to be analyzed and their trajectories to be predicted (Figure 1A (unit b, h)). As a result, the complete particle trajectory can be computed by piecing the trajectories in the channels and at the intersections. In our previous work, the FEA method was used to predict the particle behavior at critical intersections but not the whole chip [33]. In this work, an ANN method was proposed to predict particle trajectory at critical intersections as shown in the red squares in Figure 1A (unit b, h), and the complete particle trajectory was computed by using the piecing method. The whole process using the ANN can greatly reduce computational time and have low or even zero cost in computation. The schematic of the critical intersection is defined in Figure 1B, which has four ports named the East, West, South, and North ports. For the West port, the fluid can only flow in, while for the other three ports, the fluid can both flow in and out. The fluid behavior at critical intersections can be illustrated in seven cases as shown in Figure 1C. For example, in case 1, the fluid can flow into the channels from the East port, West port, and South port, while flowing out from the North port. There are also seven cases to illustrate the particle behavior at critical intersections as shown in Figure 1D; however, the particle behavior is not consistent with the fluid behavior. For example, the particle behavior in case 1 can happen in the flow behaviors in cases 1, 4, 6, and 7. Consequently, it is important to analyze both the fluid behavior and particle behavior specifically.

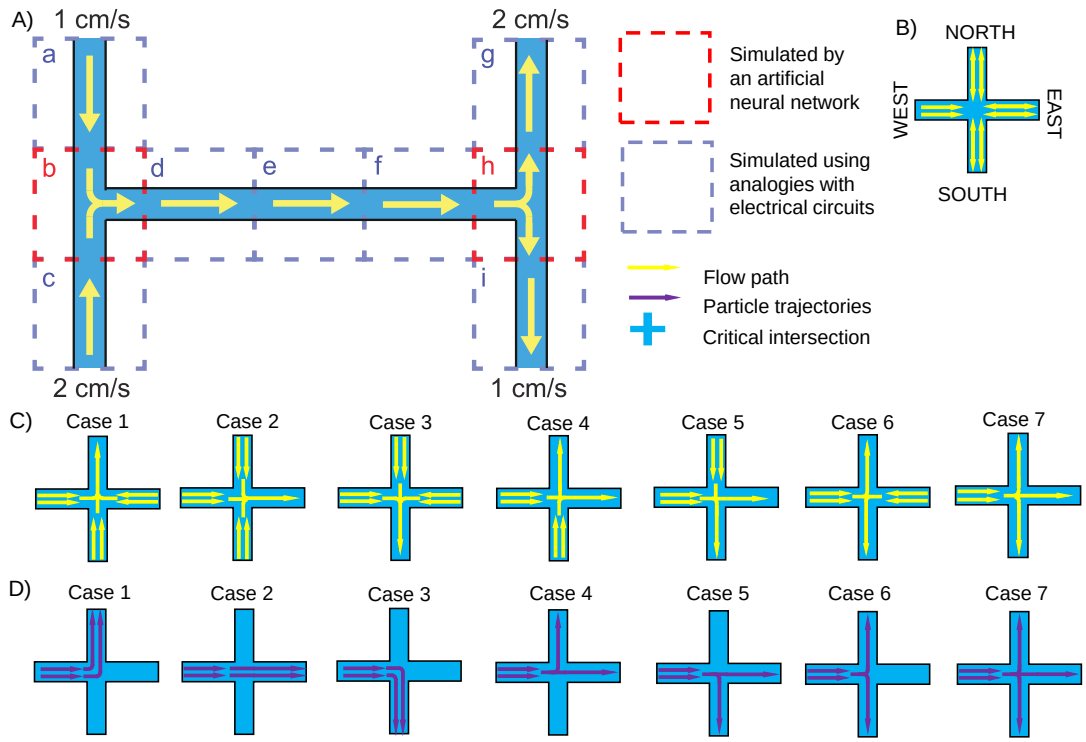


Figure 1. Theory of the design. (**A**) A basic "H" channel chip is decomposed into nine units (a to i) to demonstrate the straight channels and critical intersections. Units a, c, d, e, f, g, and i are straight channels, while units b and h are critical channel intersections. (**B**) Schematic of critical intersections. (**C**) Seven cases illustrating the fluid behavior at critical intersections. (**D**) Seven cases illustrating the particle behavior at critical intersections.

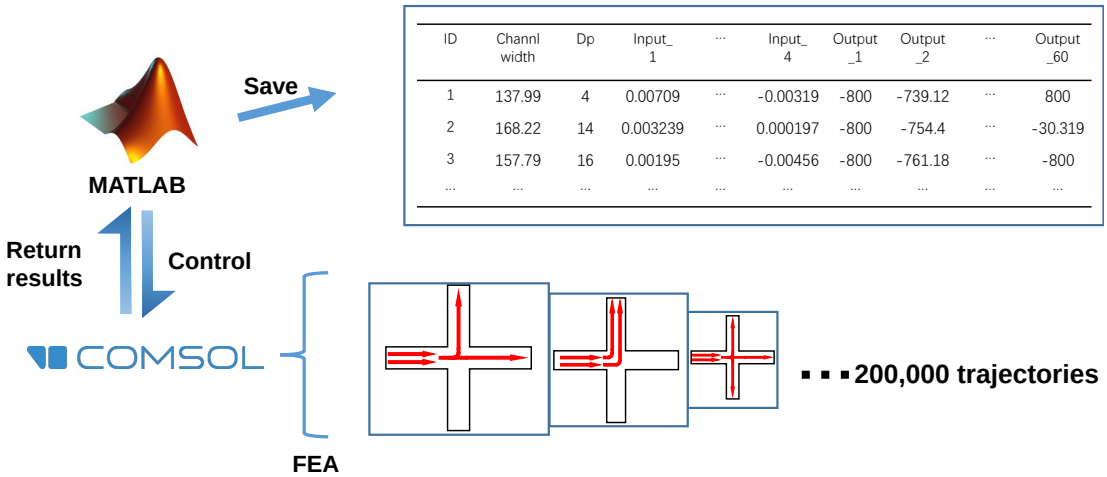
3. Materials and Methods

Supervised learning was applied to train the proposed ANN. In order to construct the training set as the very first step, MATLAB was used to control COMSOL to generate the database. As shown in Figure 2, the West port of the critical intersection shown in Figure 1 was set to be an inlet only, while the other three ports could be configured as both inlets or outlets in this simulation process. To generate random cases of the critical intersection, the inflow rates of the fluid were randomly varied from 0 to 2 cm/s, and for an intersection with outlet number of N = 1, 2, or 3, the outflow rates of N − 1 of the outlets were randomly varied from 0 to 2 cm/s. To guarantee mass conservation, the outflow rate of the remaining outlet was defined to be the total inflow rate minus the total outflow rate. Particle diameter was a variable ranging from 1 to 20 µm, set to be the input, and channel width was a variable ranging from 50 µm to 200 µm. Using this approach, 200,000 different particle trajectories at critical intersections were predicted.

Simulation of the 200,000 microfluidic intersections was conducted in COMSOL Multiphysics and the COMSOL LiveLink API for MATLAB was used to automate the process. The Laminar Flow physics module with a customized mesh of 1 µm maximum mesh size was used to solve the fluid velocity field at each intersection. The simulation results were confirmed to be not altered by finer meshes, which demonstrated mesh independence. The inlet and outlet boundary conditions were defined as introduced in the above paragraph. The other boundaries were defined as walls with no-slip boundary conditions, and the channels were filled with water under incompressible flow. A stationary solver was used to calculate the velocity field.

The path of particle flow through each intersection in our database was calculated using the Particle Tracing for Fluid Flow physics module in COMSOL Multiphysics. A "Drag Force" boundary condition was added to each channel, a particle "Inlet" boundary condition with "Uniform distribution" of initial positions was added to all inlets (10 particles per

release), and the rest of the channels were assigned "Outlet" boundary condition. The walls were assigned the "Freeze" boundary condition, meaning that particles in contact with the channel walls would stick there, which is a realistic assumption in microfluidic devices. The simulation process was repeated for each intersection using particles with diameters ranging from 1 µm to 20 µm. The resulting 200,000 simulated particle trajectories were stored in the simulation database. A time-dependent solver was used for the calculation of the particle trajectory.



ID	Channl width	Dp	Input_ 1	···	Input_ 4	Output _1	Output _2	···	Output _60
1	137.99	4	0.00709	···	-0.00319	-800	-739.12	···	800
2	168.22	14	0.003239	···	0.000197	-800	-754.4	···	-30.319
3	157.79	16	0.00195	···	-0.00456	-800	-761.18	···	-800
...	...	...	...	...	...	...	...	...	...

Figure 2. Data generation flow chart. MATLAB was used to automate the COMSOL simulation process.

The proposed ANN is physics-informed and the schematic of ANN training is shown in Figure 3. Three dense layers were used to train the target ANN and each dense layer was a fully connected ANN, predicting part of the particle trajectory. Specifically, the first dense Input 1 had 1 × 7 parameters, which were particle diameters (Dp), channel width, the initial coordinates x0 and y0 of the particle, and the inflow rate or outflow rate of the West, North, and East ports. Output 1 was the coordinates of the first ten points of the simulated entrance trajectory (1 × 20). Input 2 for dense 2 was composed of the 7 parameters of Input 1 and 20 outputs of Output 1, which was 27 parameters (1 × 27). There were also 20 outputs at Output 2, which were the coordinates of the first ten points of the simulated middle trajectory (1 × 20). Input 3 for dense 3 was composed of the 7 parameters of Input 1 and 20 outputs of Output 2, which was 27 parameters (1 × 27) as well. Similarly, there were also 20 outputs at Output 3, which were the coordinates of the first ten points of the simulated exit trajectory (1 × 20). The 1 × 20 parameters at Outputs 1, 2, and 3 provided the total 1 × 60 outputs. The three dense layers were three fully connected ANNs, and the complete particle trajectory was obtained by piecing the entrance, middle, and exit trajectories. A more detailed configuration of the dense network in the proposed ANN can be found in Table 1.

The training process was implemented in Python 3.10 with the help of PyTorch. The loss during the training process was calculated using mean squared error (MSE), and the accuracy rate was determined by using R square.

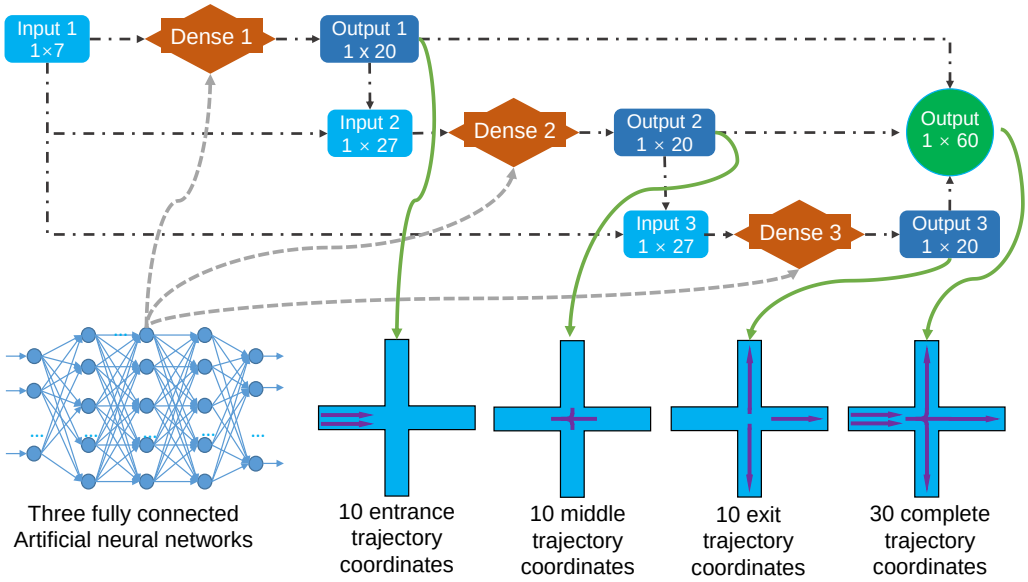


Figure 3. The structure of the proposed ANN.

Table 1. The configuration of the proposed dense network.

ID	Layer	Dense 1 (Neurons)	Dense 2 (Neurons)	Dense 3 (Neurons)	Activation
1	Input	7	27	27	Linear
2	Fully connected	400	400	400	Leaky ReLU
3	Fully connected	400	400	400	Leaky ReLU
4	Fully connected	400	400	400	Leaky ReLU
5	Fully connected	400	400	400	Leaky ReLU
6	Fully connected	400	400	400	Leaky ReLU
7	Output	20	20	20	Linear

4. Results and Discussion

The training process and the performance of the proposed ANN using four different activation functions are presented in Figure 4. In total, 70% of the dataset is used as the training set, and the other 30% is used as the test set. As it is shown in Figure 4A, using the Leaky ReLU as the activation function, the loss rate is 2.58×10^{-4}, the accuracy rate of the training set is 99.48%, and the accuracy rate of the test set is 98.88% after training for 50,000 epochs. Figure 4B shows the training result using the ReLU function, the loss rate is 5.00×10^{-4}, the accuracy rate of the training set is 98.98%, and the accuracy rate of the test set is 98.18% after training for 50,000 epochs. Figure 4C shows the training result using the Sigmoid function, the loss rate is 6.76×10^{-3}, the accuracy rate of the training set is 86.80%, and the accuracy rate of the test set is 82.84% after training for 50,000 epochs. Figure 4D shows the training result using the Tanh function, the loss rate is 1.44×10^{-3}, the accuracy rate of the training set is 96.96%, and the accuracy rate of the test set is 95.34% after training for 50,000 epochs. There are obvious differences when comparing the four groups of results. The loss rate, the accuracy rate of the training set, and the accuracy rate of the test set all decreased from the proposed ANN method using the Leaky ReLU as the activation function. The next best was ReLU, then Sigmoid, and then Tanh. In a word, the three parameters all achieved the highest in the ANN method using the Leaky ReLU function. Therefore, the Leaky ReLU function was chosen as the activation function in the proposed ANN.

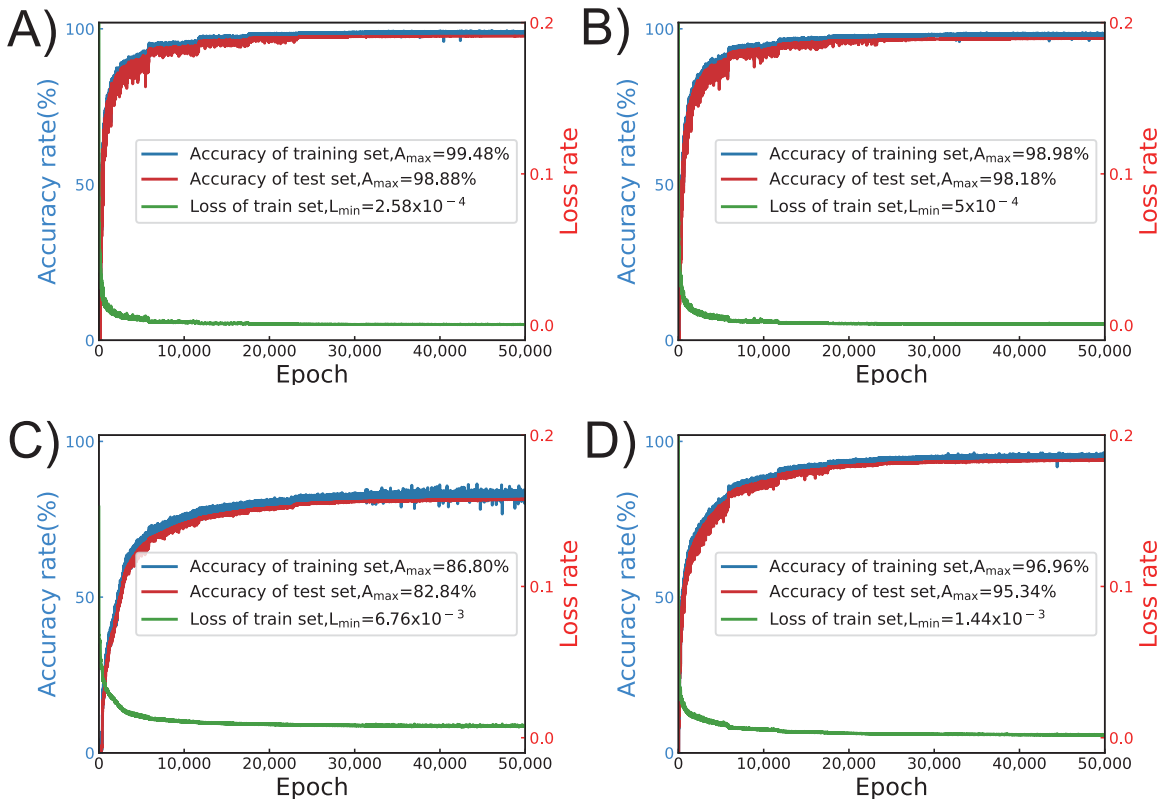


Figure 4. The training curve of the proposed ANN using four different activation functions. (**A**) Leaky ReLU function. (**B**) ReLU function. (**C**) Sigmoid function. (**D**) Tanh function.

Figure 5 demonstrates nine randomly selected cases simulated using our proposed ANN method compared with the COMSOL method. The detailed parameters of nine randomly selected cases are shown in Table 2. It can be seen from both Figure 5 and Table 2 that case A and case G all had particle diameters of 13 µm, and similar channel widths of 190.34 and 194.02 µm, respectively. Although the particle flowing path was input from the West and output from the North for case A, and input from the West and output from the South for case G, the particle trajectories predicted with the proposed ANN method were both highly consistent with the COMSOL method. In case H and case I, the input particle diameter was 11 µm, while the channel widths were 156.21 and 72.04 µm, respectively. In those two cases, the particle flowing path was closer to the wall in case H while farther from the wall in case I, which showed that the input position of the particle does not affect the ANN prediction results. In other cases, D, E, and F had three different channel widths and particle diameters, and as a result, the particle flow paths varied from the upper wall, to the middle channel, and to the lower wall, which showed that the particle diameter and channel width had no effect on the ANN prediction results. Cases B and C were consistent with the above phenomenon. The nine randomly selected cases showed that no matter what the value of particle diameter and the channel width is, the particle trajectories predicted by our ANN method are always highly consistent with COMSOL predictions.

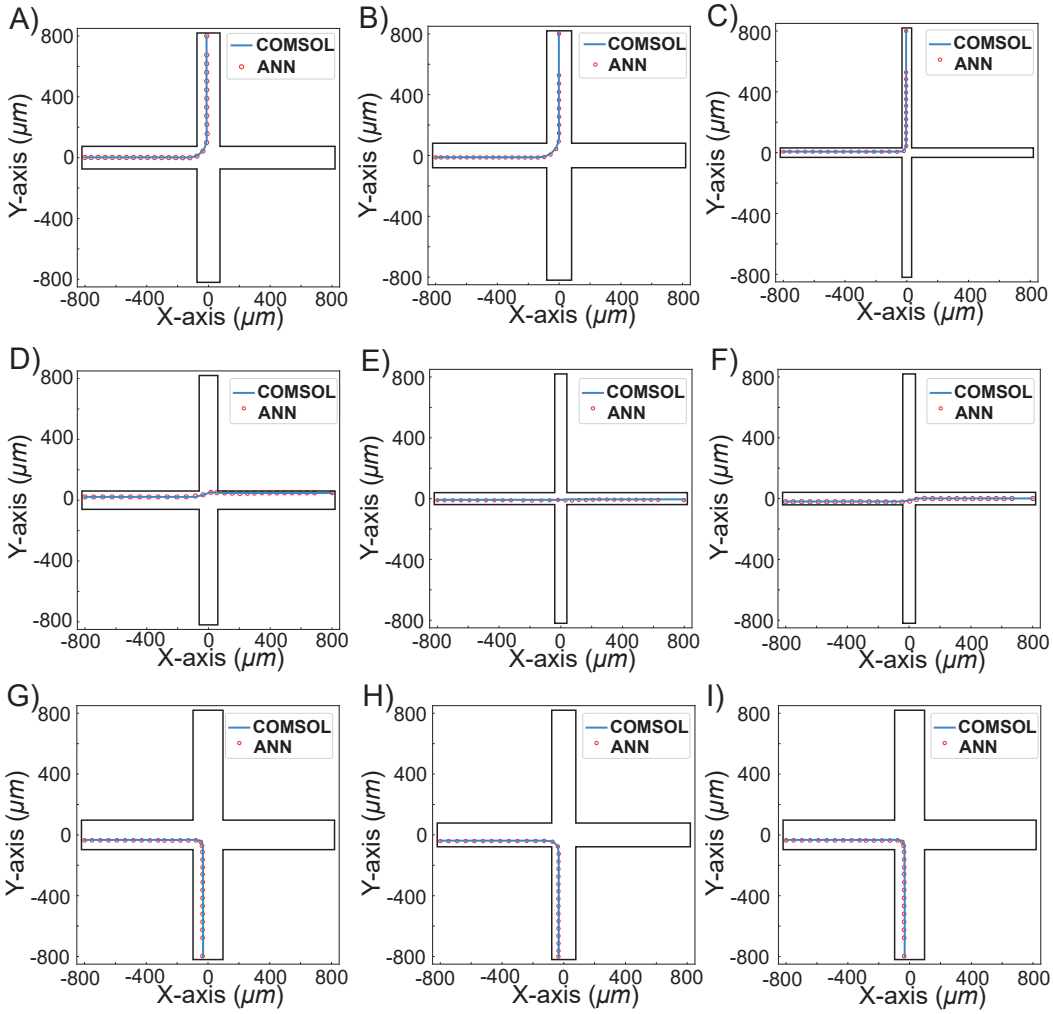


Figure 5. The nine randomly selected simulation results showing the comparison between ANN-predicted and COMSOL-predicted particle trajectories.

Table 2. The detailed parameters of nine randomly selected cases.

ID	Channel Width (μm)	Particle Diameter (μm)
A	190.34	13
B	161.23	11
C	62.36	10
D	120.63	17
E	78.65	10
F	82.36	18
G	194.02	13
H	156.21	11
I	72.94	11

Potential Limitations

The proposed ANN prediction method showed good results that match the COMSOL simulation method. However, there are still some potential limitations when using this ANN method. First of all, this ANN method is only adapted for rectangular intersections currently, other structures such as circular intersections or angles between channels less or more than 90 degrees are not applicable. Secondly, this method is only applied for low Re number laminar flow cases, while the turbulence flow cannot be predicted with this ANN method. The third limitation is that the resolution of the method is fixed, with only

60 points per trajectory, which might not be enough for some cases requiring more data points. Our future work will be dedicated to solving the above issues.

In addition, the overall simulation time in COMSOL is usually minutes or even hours depending on different models, while the ANN processing time is usually at the millisecond level no matter what kind of the model is. Therefore, it is not so meaningful to compare the specific computational time for these two methods. However, for most microfluidic chip simulation models, the simulation cost is reduced from several hours or minutes to several seconds or even the millisecond level.

5. Conclusions

In this work, an ANN method was proposed to facilitate the microfluidic device design process and lower the cost, and the results showed high consistency with the COMSOL prediction method. The ANN method with three dense layers was used to solve particle trajectories at the critical intersections, and in the future, it can be further applied to accelerate the simulation process of particle trajectory in the whole chip, by taking advantage of analogy circuits. With our current method, the simulation cost is reduced from several hours or minutes to several seconds, which is 100× or even 1000× faster. Although there are still some limitations using the current ANN method, we are confident to work on it and solve the issues, bringing a more advanced ANN tool to accelerate the design process for researchers to invent new cell sorting chips for many different applications.

Author Contributions: Conceptualization, J.W.; methodology, N.Z., K.L. and J.W.; investigation, N.Z., K.L., Z.L., T.S. and J.W.; writing—original draft preparation, N.Z. and J.W.; writing—N.Z. and J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Natural Science Foundation of China under grant number 62206081, the Zhejiang Provincial Natural Science Foundation of China under Grant number LQ20F040003, and the Fundamental Research Funds for the Provincial Universities of Zhejiang under Grant GK209907299001-028.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tang, W.; Jiang, D.; Li, Z.; Zhu, L.; Shi, J.; Yang, J.; Xiang, N. Recent advances in microfluidic cell sorting techniques based on both physical and biochemical principles. *Electrophoresis* **2019**, *40*, 930–954. [CrossRef] [PubMed]
2. Zhu, S.; Jiang, F.; Han, Y.; Xiang, N.; Ni, Z. Microfluidics for label-free sorting of rare circulating tumor cells. *Analyst* **2020**, *145*, 7103–7124. [CrossRef] [PubMed]
3. Hu, D.; Liu, H.; Tian, Y.; Li, Z.; Cui, X. Sorting technology for circulating tumor cells based on microfluidics. *ACS Comb. Sci.* **2020**, *22*, 701–711. [CrossRef] [PubMed]
4. Laxmi, V.; Joshi, S.S.; Agrawal, A. Effect of various parameters on the distribution and extraction of platelets in a microfluidic system. *Microfluid. Nanofluidics* **2021**, *25*, 1–12. [CrossRef]
5. Zhang, Y.; Chen, X. Dielectrophoretic microfluidic device for separation of red blood cells and platelets: A model-based study. *J. Braz. Soc. Mech. Sci. Eng.* **2020**, *42*, 1–11. [CrossRef]
6. Richard, C.; Fakhfouri, A.; Colditz, M.; Striggow, F.; Kronstein-Wiedemann, R.; Tonn, T.; Medina-Sánchez, M.; Schmidt, O.G.; Gemming, T.; Winkler, A. Blood platelet enrichment in mass-producible surface acoustic wave (SAW) driven microfluidic chips. *Lab Chip* **2019**, *19*, 4043–4051. [CrossRef]
7. Segaliny, A.I.; Li, G.; Kong, L.; Ren, C.; Chen, X.; Wang, J.K.; Baltimore, D.; Wu, G.; Zhao, W. Functional TCR T cell screening using single-cell droplet microfluidics. *Lab Chip* **2018**, *18*, 3733–3749. [CrossRef]
8. Wang, Y.; Jin, R.; Shen, B.; Li, N.; Zhou, H.; Wang, W.; Zhao, Y.; Huang, M.; Fang, P.; Wang, S.; et al. High-throughput functional screening for next-generation cancer immunotherapy using droplet-based microfluidics. *Sci. Adv.* **2021**, *7*, eabe3839. [CrossRef]

9. Buryk-Iggers, S.; Kieda, J.; Tsai, S.S. Diamagnetic droplet microfluidics applied to single-cell sorting. *AIP Adv.* **2019**, *9*, 075106. [CrossRef]
10. Sesen, M.; Whyte, G. Image-based single cell sorting automation in droplet microfluidics. *Sci. Rep.* **2020**, *10*, 1–14. [CrossRef]
11. Kim, G.Y.; Han, J.I.; Park, J.K. Inertial microfluidics-based cell sorting. *BioChip J.* **2018**, *12*, 257–267. [CrossRef]
12. Nakao, S.; Takeo, T.; Watanabe, H.; Kondoh, G.; Nakagata, N. Successful selection of mouse sperm with high viability and fertility using microfluidics chip cell sorter. *Sci. Rep.* **2020**, *10*, 1–9. [CrossRef] [PubMed]
13. Lyu, Y.; Yuan, X.; Glidle, A.; Fu, Y.; Furusho, H.; Yang, T.; Yin, H. Automated Raman based cell sorting with 3D microfluidics. *Lab Chip* **2020**, *20*, 4235–4245. [CrossRef] [PubMed]
14. Xiang, N.; Ni, Z. Electricity-free hand-held inertial microfluidic sorter for size-based cell sorting. *Talanta* **2021**, *235*, 122807. [CrossRef] [PubMed]
15. Descamps, L.; Audry, M.C.; Howard, J.; Mekkaoui, S.; Albin, C.; Barthelemy, D.; Payen, L.; Garcia, J.; Laurenceau, E.; Le Roy, D.; et al. Self-assembled permanent micro-magnets in a polymer-based microfluidic device for magnetic cell sorting. *Cells* **2021**, *10*, 1734. [CrossRef] [PubMed]
16. Mutafopulos, K.; Spink, P.; Lofstrom, C.; Lu, P.; Lu, H.; Sharpe, J.; Franke, T.; Weitz, D. Traveling surface acoustic wave (TSAW) microfluidic fluorescence activated cell sorter (μFACS). *Lab Chip* **2019**, *19*, 2435–2443. [CrossRef]
17. Cai, K.; Mankar, S.; Ajiri, T.; Shirai, K.; Yotoriyama, T. An integrated high-throughput microfluidic circulatory fluorescence-activated cell sorting system (μ-CFACS) for the enrichment of rare cells. *Lab Chip* **2021**, *21*, 3112–3127. [CrossRef]
18. LaBelle, C.A.; Massaro, A.; Cortés-Llanos, B.; Sims, C.E.; Allbritton, N.L. Image-based live cell sorting. *Trends Biotechnol.* **2021**, *39*, 613–623. [CrossRef]
19. Lee, K.; Kim, S.E.; Doh, J.; Kim, K.; Chung, W.K. User-friendly image-activated microfluidic cell sorting technique using an optimized, fast deep learning algorithm. *Lab Chip* **2021**, *21*, 1798–1810. [CrossRef] [PubMed]
20. Pritchard, R.H.; Zhukov, A.A.; Fullerton, J.N.; Want, A.J.; Hussain, F.; la Cour, M.F.; Bashtanov, M.E.; Gold, R.D.; Hailes, A.; Banham-Hall, E.; et al. Cell sorting actuated by a microfluidic inertial vortex. *Lab Chip* **2019**, *19*, 2456–2465. [CrossRef]
21. Turan, B.; Masuda, T.; Lei, W.; Noor, A.M.; Horio, K.; Saito, T.I.; Miyata, Y.; Arai, F. A pillar-based microfluidic chip for T-cells and B-cells isolation and detection with machine learning algorithm. *Robomech J.* **2018**, *5*, 1–9. [CrossRef]
22. Hung, S.; Hsu, C.H.; Chen, C. Cell sorting in microfluidic systems using dielectrophoresis. In Proceedings of the 2015 IEEE 15th International Conference on Nanotechnology (IEEE-NANO), Rome, Italy, 27–30 July 2015; IEEE: New York, NY, USA, 2015; pp. 872–875.
23. Lin, Y.T.; Huang, C.S.; Tseng, S.C. How to Control the Microfluidic Flow and Separate the Magnetic and Non-Magnetic Particles in the Runner of a Disc. *Micromachines* **2021**, *12*, 1335. [CrossRef]
24. Myklatun, A.; Cappetta, M.; Winklhofer, M.; Ntziachristos, V.; Westmeyer, G.G. Microfluidic sorting of intrinsically magnetic cells under visual control. *Sci. Rep.* **2017**, *7*, 1–8. [CrossRef] [PubMed]
25. McIntyre, D.; Lashkaripour, A.; Fordyce, P.; Densmore, D. Machine learning for microfluidic design and control. *Lab Chip* **2022**, *22*, 2925–2937. [CrossRef] [PubMed]
26. Wang, J.; Zhang, N.; Chen, J.; Su, G.; Yao, H.; Ho, T.Y.; Sun, L. Predicting the fluid behavior of random microfluidic mixers using convolutional neural networks. *Lab Chip* **2021**, *21*, 296–309. [CrossRef] [PubMed]
27. Zhang, N.; Liu, Z.; Wang, J. Machine-Learning-Enabled Design and Manipulation of a Microfluidic Concentration Gradient Generator. *Micromachines* **2022**, *13*, 1810. [CrossRef]
28. Ji, J.; Zhang, J.; Wang, J.; Huang, Q.; Jiang, X.; Zhang, W.; Sang, S.; Guo, X.; Li, S. Three-dimensional analyses of cells' positioning on the quadrupole-electrode microfluid chip considering the coupling effect of nDEP, ACEO, and ETF. *Biosens. Bioelectron.* **2020**, *165*, 112398. [CrossRef]
29. Ji, J.; Wang, J.; Wang, L.; Zhang, Q.; Duan, Q.; Sang, S.; Huang, Q.; Li, S.; Zhang, W.; Jiang, X. Dynamic-coupling analyses of cells localization by the negative dielectrophoresis. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2021**, *235*, 402–411. [CrossRef]
30. Du, C.J.; Sun, D.W. Learning techniques used in computer vision for food quality evaluation: A review. *J. Food Eng.* **2006**, *72*, 39–55. [CrossRef]
31. Nadkarni, P.M.; Ohno-Machado, L.; Chapman, W.W. Natural language processing: An introduction. *J. Am. Med. Inform. Assoc.* **2011**, *18*, 544–551. [CrossRef]
32. Almeida, L.G.; Backović, M.; Cliche, M.; Lee, S.J.; Perelstein, M. Playing tag with ANN: Boosted top identification with pattern recognition. *J. High Energy Phys.* **2015**, *2015*, 1–21. [CrossRef]
33. Wang, J.; Rodgers, V.G.; Brisk, P.; Grover, W.H. Instantaneous simulation of fluids and particles in complex microfluidic devices. *PLoS ONE* **2017**, *12*, e0189429. [CrossRef] [PubMed]
34. Wang, J.; Fu, L.; Yu, L.; Huang, X.; Brisk, P.; Grover, W.H. Accelerating Simulation of Particle Trajectories in Microfluidic Devices by Constructing a Cloud Database. In Proceedings of the 2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Hong Kong, China, 8–11 July 2018; IEEE: New York, NY, USA, 2018; pp. 666–671.