ENGINEERING

CeyeHao: AI-driven microfluidic flow programming with hierarchically assembled obstacles and receptive field–augmented neural network

Zhenyu Yang[1,2]†, Zhongning Jiang[3]†, Haisong Lin[4,5], Xiaoxue Fan[2], Changjin Wu[2], Edmund Y. Lam[6]*, Hayden K. H. So[6]*, Ho Cheung Shum[1,2,3,7]*

Microfluidic fabrication technologies are increasingly used to produce functional anisotropic microstructures for broad applications. However, the limited flow manipulation methods hinder the production of intricate microstructure morphologies. In this work, we introduce CeyeHao, an artificial intelligence–driven flow programming methodology for designing microchannels to perform unprecedented flow manipulations. In CeyeHao, microchannels containing hierarchically assembled obstacles are constructed, offering more than double flow transformation modes and immense configurability compared to state-of-the-art methods. An AI model, CEyeNet, predicts the transformed flow profiles, reducing computation time by up to 2700 folds and achieving up to 97 and 90% accuracy with simulated and experiment results. CeyeHao facilitates microchannel design in both human-guided and automatic modes, enabling creation of flow morphologies with highly regulated geometries and elaborate artistic patterns, along with precise topology manipulation of multiple streams. The superior flow manipulation capability of CeyeHao can facilitate broad applications from complex microstructure fabrication to precise reaction control.

INTRODUCTION

The ability to fabricate microstructures with tailored morphologies and heterogeneous material compositions is crucial for achieving their functionality and expanding their utility in complex environments (*1*, *2*). For instance, helical structure enables magnetic microswimmers to navigate physiological fluids under external magnetic fields (*3*), while hollow threaded stud structures can carry stem cells for programmable delivery (*4*). Recently, microfluidic fabrication methods have presented a compelling approach for producing complex microstructures due to their ability to simultaneously shape and integrate multicomponent materials (*5*). This technique has been used to fabricate diverse microstructures, such as microrobots (*6*–*8*) and fibrous tissue constructs (*9*–*11*). Microfluidic fabrication also offers the advantage of high throughput (*12*), when compared to the other microfabrication methods such as two-photon polymerization (*13*). Despite these advances, the full potential of microfluidic fabrication is constrained by the simple and limited producible cross-sectional flow profiles in microchannels.

Flow programming provides a versatile solution for flow manipulation in microfluidic fabrication, wherein the fluid materials are shaped step by step through parametrical microchannel modules (*14*). This method allows the design of microchannels under a universal framework to produce diverse flow profiles from dumbbells (*15*) to alphabetic letter shapes (*16*). With flow programming methods, a wide range of microstructures can be produced, such as microswimmers (*17*), cell carriers (*18*), and monodisperse-droplet templates (*19*). To enhance the functionalities of microstructures, it is imperative to generate more complex flow profiles with rich and intricate features.

However, current flow programming methods are limited by constrains in both microfluidic system construction and microchannel design tools. The primitive shapes of microchannels used in these methods can barely produce the delicate secondary flows essential for shaping more complex flow profiles (*20*, *21*). Moreover, the finite and limited variations in the structural parameters of these microchannels prevent the precise tuning of the secondary flows within a continuous design space (*22*). Consequently, current flow programming methods can only produce meaningful flow profiles with simple morphologies. In terms of design toolkits, previous flow programming methods rely on computational fluid dynamics (CFD) or physical experiments to investigate the transformed flow profiles. These approaches require prohibitive amount of time and labor to screen the numerous microchannel designs involved in the design process. As a result, designing microchannels for specific flow manipulations becomes costly and time-consuming, further narrowing the design space of producible flow profiles. To expedite the design process, numerical toolbox of the flow-transforming modules (*23*) and automated optimization algorithms for the modular sequences (*24*–*26*) have been developed. However, these methods still fall short in addressing the flow investigation challenges and remain constrained by the limited microchannel variations. Therefore, without a revolutionary solution to these limitations, substantial breakthroughs in flow programming are unattainable.

To overcome these challenges, we introduce CEyeNet-enabled design of hierarchically assembled obstacles (HAOs), termed CeyeHao, to facilitate the design of microchannels that manipulate flow with unparalleled complexity and precision. CeyeHao uses HAOs (Fig. 1A) to induce more intricate flow profile transformations with significantly higher configurability than the state-of-the-art methods.

[1]Advanced Biomedical Instrumentation Centre, Hong Kong Science Park, Shatin, New Territories, Hong Kong, China. [2]Department of Mechanical Engineering, The University of Hong Kong, Hong Kong, China. [3]Department of Biomedical Engineering, City University of Hong Kong, Hong Kong, China. [4]School of Engineering, Westlake University, Hangzhou, China. [5]Research Center for Industries of the Future, Westlake University, Hangzhou, China. [6]Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, China. [7]Department of Chemistry, City University of Hong Kong, Hong Kong, China.
*Corresponding author. Email: elam@eee.hku.hk (E.Y.L.); hso@eee.hku.hk (H.K.H.S.); ashum@cityu.edu.hk (H.C.S.)
†These authors contributed equally to this work.

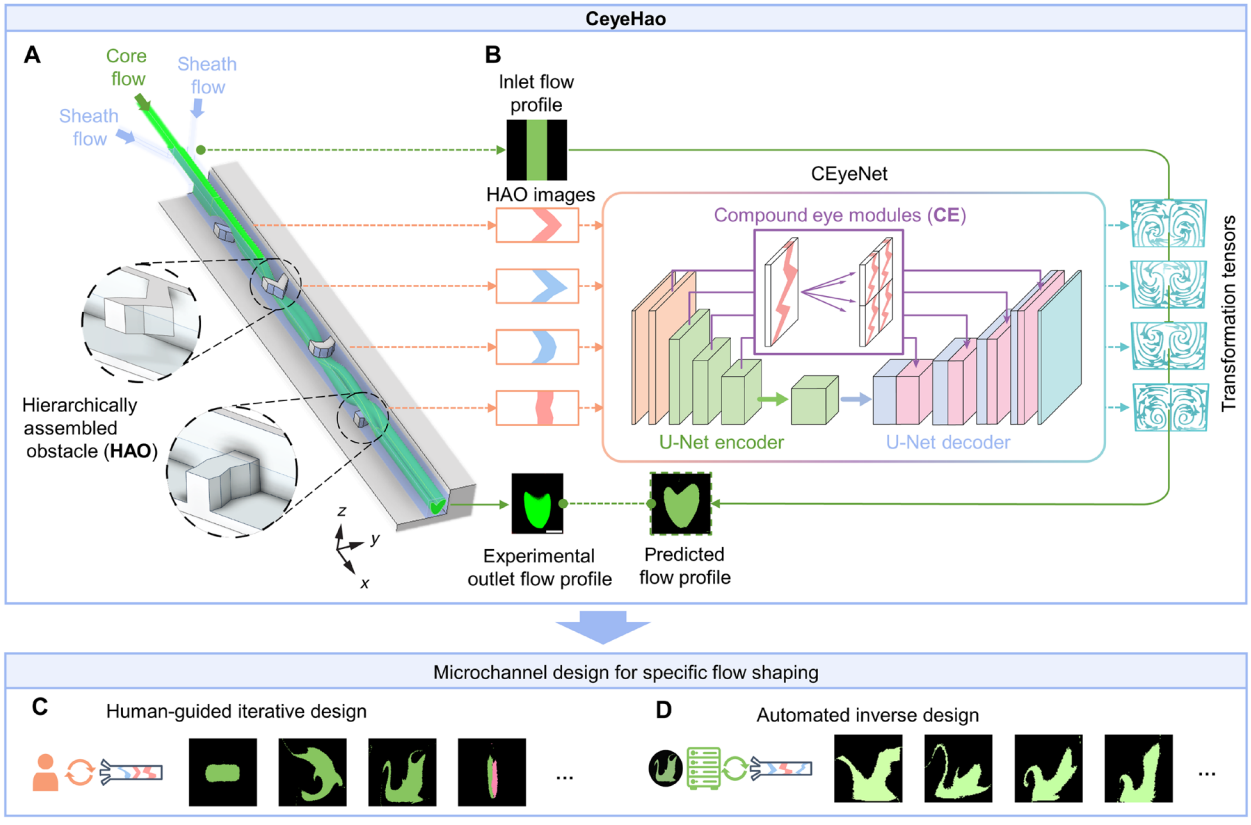

Fig. 1. Overview of CeyeHao. CeyeHao is an artificial intelligence (AI)–driven flow programming methodology developed to facilitate the design of microchannels that shape fluid materials into specific morphologies and combinations. (**A**) The microfluidic system used in CeyeHao. The fluid materials to be shaped are infused as core flow, accompanied by sheath flow. The microchannel contains sequentially arranged HAOs to induce complex flow transformation. (**B**) The flow profile prediction system of CeyeHao. A bespoke neural network model CEyeNet featuring a receptive field–augmenting module compound eye (CE) predicts the transformed flow profiles in the HAO-embedded microchannel. (**C** and **D**) CeyeHao facilitates both human-guided and automated inverse microchannel design tasks, enabling unprecedented flow programming capabilities.

Moreover, CeyeHao features an artificial intelligence (AI)–enabled prediction system for the transformed flow profiles (Fig. 1B). At the core of the system is a bespoke neural network model, CEyeNet, which predicts the flow transformation tensors in real time to compute the transformed flow profiles. Specifically, CEyeNet is equipped with a lightweight receptive field–augmenting module, termed compound eye (CE), to effectively learn the long-range dependencies within the dataset. As a result, CeyeHao enables seamless access to an expansive, complex, and continuous space of flow transformations, significantly enriching the toolset for flow manipulation. With CeyeHao, we are able to iteratively design microchannels to produce previously unattainable flow profiles (Fig. 1C). Additionally, we show that the CeyeHao can produce flow profiles with significantly improved fidelity in the automatic inverse design tasks (Fig. 1D). Our powerful results show that CeyeHao not only enhances the efficiency and feasibility of microstructure design but also provides a practical tool for the fine-scale manipulation of microfluidic flows, facilitating the exploration of advanced applications.

RESULTS

Microchannel framework of CeyeHao

We design the HAOs embedded in the microchannel to generate secondary flows from boundary-defined asymmetric pressure fields, enabling flow transformations in the Stokes flow regime [Reynolds number (*Re*) smaller than one]. We use vertical oblique walls that partially obstruct the microchannel as the basic building blocks of HAO, because these are the elemental microchannel boundary features that can induce net flow transformations in the inertialess regime (*27*). The oblique wall generates a rotary net secondary flow on the cross-sectional plane of the microchannel, which manifests as a vortex in the flow transformation pattern (Fig. 2A). The vortex exhibits substantial amplitude near the obstacle and attenuates in the direction away from the obstacle. The overall intensity of the vortex (i.e., the averaged net displacement magnitude of the flow elements on the microchannel cross section) is proportional to the oblique angle of the wall (fig. S1A), offering flexible tuning of flow transformation magnitudes. To create more complex obstacles with minimized feature redundancy, the oblique walls are assembled analogously to circuit connections: first, two oblique walls can be connected in series to form a ridge (Fig. 2B), which introduces an asymmetric vortex similar to those induced by individual oblique walls, with an intensity equal to the sum of the vortices of the composing oblique walls (fig. S1A). Additionally, the oblique walls can be connected in parallel to form a poly-segment wall, where each segment induces an individual vortex in its proximal region (Fig. 2C). The range and amplitude of each vortex depend on the widths and angles of the composing oblique walls (fig. S1B). In a

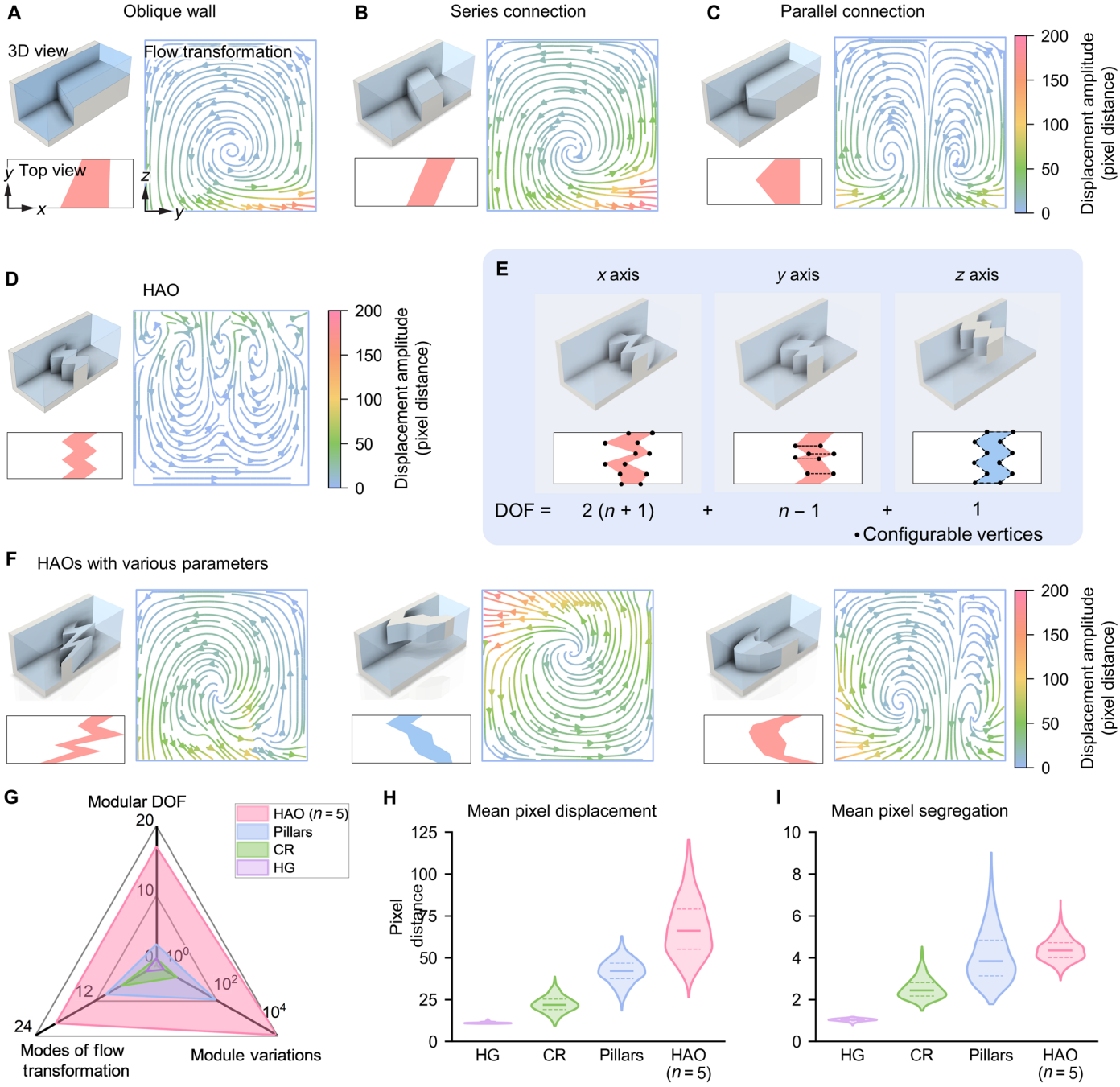

Fig. 2. Construction of HAOs and benchmarking with the previous microchannel frameworks. (**A**) The composing elements and assembly rules of HAO. Elementary oblique walls of HAO. Top left: Three-dimensional (3D) schematic of the obstacle. Bottom left: Top view of the obstacle. Bottom right: Net flow transformation caused by the obstacle, visualized as streamline plot of the lateral flow element displacements. The streamline colors indicate the local displacement magnitude measured in pixels. For reference, the microchannel cross-sectional dimension is 200 pixels by 200 pixels. (**B**) Ridges formed by serially connected oblique walls. (**C**) Poly-segment wall formed by parallelly connected oblique walls. (**D**) HAO assembled by recursively connecting the oblique walls in series and in parallel. (**E**) The DOFs of HAOs in the *x*, *y*, and *z* axes for adjusting their shapes and the induced flow transformations. The dots in the top-view images of the HAOs indicate the vertices that can move along the corresponding axes. The dashed lines connect the vertices that share the DOF in that axis. (**F**) Examples of HAOs with various parameters and their distinct flow transformations. In the top-view images, blue HAOs indicate placement on the top microchannel wall, and orange HAOs indicate the opposite. (**G**) Comparison between HAO and the pervious flow programming microchannel frameworks from configurability perspective. CR, chevron ridges; HG, herringbone grooves. (**H**) Violin plots of distributions of mean pixel displacements (MPDs) of the randomly sampled microchannels under each microchannel framework. The center lines of violins denote the median; dashed lines denote the quartile range; tips of violins denote 1.5× interquartile range; areas denote the distribution from kernel density estimation with parameter $n = 1000$. CR and HG are illustrated in the caption of (G). (**I**) The mean pixel segregation (MPS) of the microchannels as in (H).

special case where the two oblique walls are symmetric, as shown in Fig. 2C, two mirrored vortices of the same size and amplitude are formed. The parallel connection of oblique walls fundamentally alters the distribution and direction of the flow transformations (referred to as a distinct flow transformation mode), significantly enhancing the programmability and heterogeneity of the flow profiles inside the microchannel.

By recursively assembling the obstacle features in series and parallel, HAOs are constructed. As a result, multiple vortices equivalent in number to the ridges are simultaneously induced (Fig. 2D). The vertices of the HAO can be displaced along different axis without disrupting the assembly relations, creating vast tunability of the flow transformation (fig. S1C). For a given microchannel width, an HAO, consisting of n pairs of serially connected oblique walls, has $3n + 1$ configurable coordinates, each providing a degree of freedom (DOF) to tune the flow transformation (Fig. 2E). Furthermore, the HAO can be deployed on either the top or bottom wall of the microchannel, offering one more DOF. Therefore, an HAO provides $3n + 2$ DOF to tune its morphology and, consequently, the emergence, range, and amplitudes of the vortices. This substantial configurability allows HAOs to generate a vast diversity of flow transformations with distinct local features (Fig. 2F). Moreover, HAOs have multiple folds of DOF and more than double the modes of flow transformations (fig. S2) compared to the existing microchannel frameworks, such as herringbone grooves (*28*), chevron ridges (*23*), and pillars (*14*) (Fig. 2G). Furthermore, CeyeHao allows for an infinite number of HAO configurations, in contrast to the finite variations available in the previous flow programming methods (Fig. 2G). These results underscore the evolutional configurability of flow transformation afforded by HAO.

We further evaluate the flow transformation capability of microchannels embedded with multiple HAOs numerically (Materials and Methods). To simplify the microchannel fabrication process and ensure easily observable flow profile transformations, the parameter n of every HAO is set to five (fig. S1D). In this context, fewer than eight HAOs are able to displace the flow elements to virtually arbitrary positions on the cross-sectional plane, regardless of the initial positions of the flow elements (fig. S1E). Moreover, to characterize the flow transformations, we construct two flow transformation metrics: mean pixel displacement (MPD) and mean pixel segregation (MPS) (Materials and Methods). MPD measures the mean net displacements of all the flow elements on the microchannel cross section, reflecting the amplitude of flow transformation. MPS measures the mean difference of the net displacements of a flow element from its adjacent flow elements, reflecting the level of topology changes and chaoticity of the transformed flow profiles. We evaluate the MPS and MPD of 1000 microchannels constructed with HAOs and the previous microchannel frameworks, respectively, with each microchannel containing eight modules. The distributions of MPD for HAOs are substantially higher than the previous frameworks (Fig. 2H), demonstrating the superior ability of HAOs to produce substantial flow transformations. Meanwhile, the modest level of MPS for HAOs (Fig. 2I) suggests the moderate chaoticity level even during large flow transformations. Moreover, we conduct a comparative analysis on the practicality of the HAO structure against state-of-the-art pillar-based flow transformation methods. The results show that HAOs, operating at much lower *Re*, exhibit over 10-fold lower pressure drops compared to pillars, thus making the HAOs easier to implement and more suitable for pressure-sensitive applications (fig. S3A). Meanwhile, HAOs achieve fully developed flow within one microchannel width, unlike pillars that can require three to five folds longer distances depending on the *Re* (fig. S3B). This result, in combination with the observation that fewer HAOs are usually required than pillars to produce a specific flow profile (fig. S10), indicates that HAO has a significantly more compact footprint than the state-of-the-art method.

A notable distinct characteristic of HAOs from previous inertial-based flow programming methods is that they operate within the Stokes flow regime ($Re < 1$); hence, flow deformations in HAO-embedded microchannel are independent of inertial forces, allowing consistent reproduction of flow profiles across different flow velocities, viscosities, and microchannel dimensions, as demonstrated in experiments where robust reproduction of the consistent flow profile is achieved even after eight HAOs (fig. S4A). However, the Peclet number (*Pe*) critically influences the sharpness of the flow profile boundary; *Pe* values below 10^4 can result in notable edge blurring and complete mixing (fig. S4B). The experiment results suggest maintaining *Pe* above 10^4 is essential for producing distinguishable flow profile edges.

Flow profile prediction system of CeyeHao

Predicting the flow transformations with CEyeNet

To train CEyeNet, we numerically collect a dataset consisting of HAO top-view images and the corresponding flow transformation tensors (Fig. 3A and Materials and Methods). The inspection of the dataset reveals that even trivial feature differences in the HAO images can cause notable differences to the global transformation tensors patterns (Fig. 3A and fig. S5B). This indicates the existence of global dependencies between the model's input and output data, which are difficult to capture using conventional flow transforming networks with local operators such as convolutional kernels (*29*). Although neural networks with global operators, such as transformers (*30*), are proficient at capturing these global dependencies, their computational cost is significantly higher, adversely affecting design efficiency. Consequently, the design of HAO-embedded microchannel necessitates a bespoke neural network model that combines the high efficiency of local operators with the broad receptive field of global operators.

To this end, we constructed CEyeNet (Fig. 3B) by inserting CE modules into the skip pathways of the original U-Net (*31*). CE is a lightweight neural network block that grants a global-level receptive field to the subsequent convolutional kernels. In the CE module, the input feature map is pooled and tiled into a thumbnail array at the same spatial dimensions (Fig. 3C and Materials and Methods). By adjusting the ratio of pooling and tiling within the CE module, the subsequent convolutional operations can process a sufficiently large portion of the thumbnail, capturing the essential long-range information embedded in it. The design of the CE module is reminiscent of the CEs of insects, which inspired names of the module and the neural network.

To demonstrate the effectiveness of CEyeNet and its advantages over state-of-the-art neural network architectures, we compare CEyeNet against two types of baseline models: U-Net++ (*32*), which enhances its receptive field by stacking local operators in the skip pathways of U-Net, and Global Voxel Transformer Network (GVTNet) (*29*), which achieves global receptive fields through attention-based (*30*) global operators (Materials and Methods). All the models are trained using the same protocol (Materials and Methods). We evaluate the test-set accuracy of the trained models

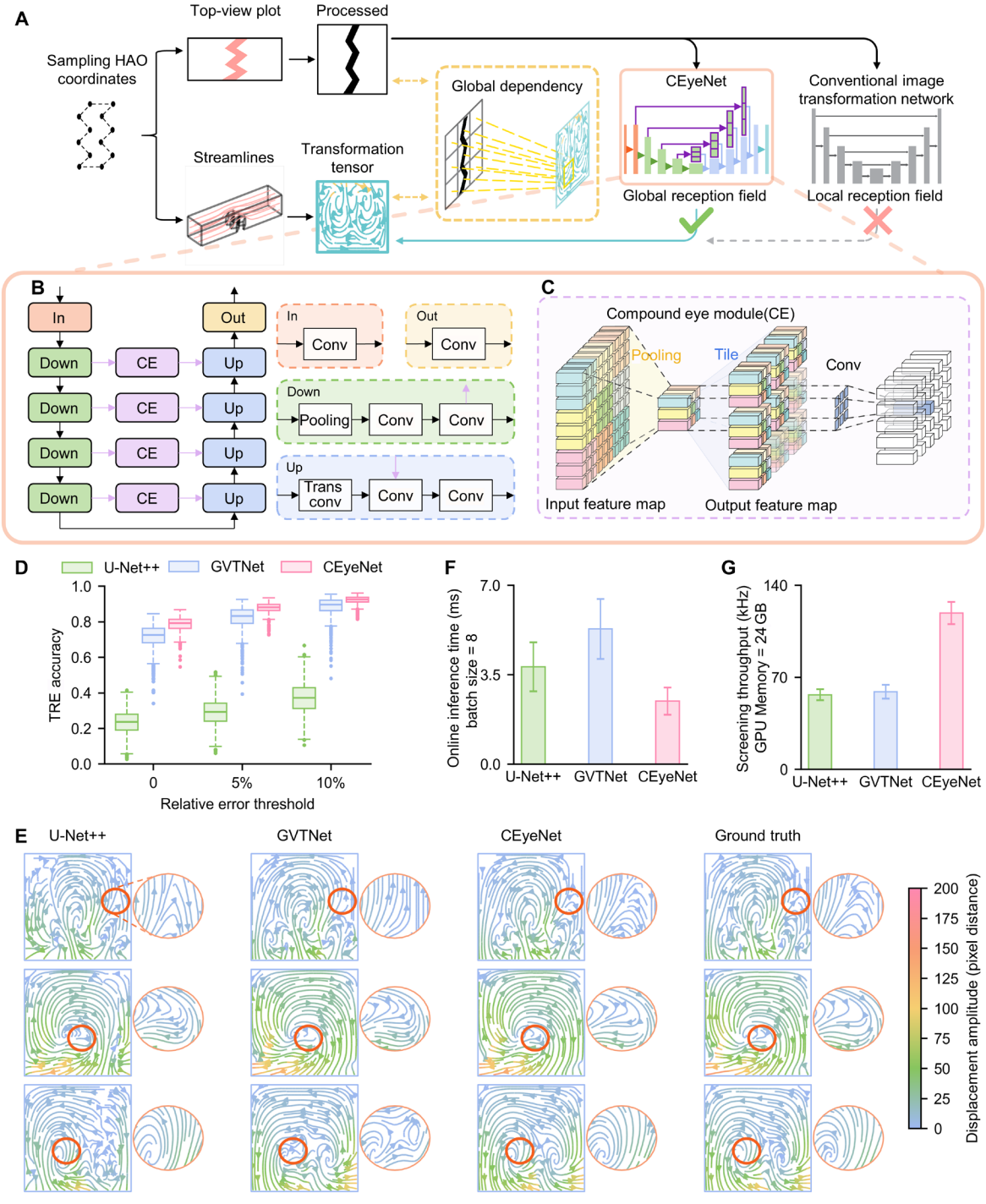

Fig. 3. Development and evaluation of CEyeNet. (**A**) Schematic of data collection process. The top-view image of HAO and the corresponding transformation tensor are generated through programming and finite element simulations. The analysis of the data shows that global-range dependencies exist, which cannot be effectively captured by conventional image transformation neural networks, prompting the development of CEyeNet. (**B**) Schematic of CEyeNet. Left: The high-level architecture of CEyeNet. Right: The compositional neural network layers in the modules of CEyeNet. (**C**) Schematic of the CE module. The feature map input into CE module is first pooled into a single thumbnail feature map and is then tiled into grids to output a thumbnail array of the same spatial dimension as the input feature map. The convolution kernel following the CE module can thus obtain the global information of the input feature map of CE module, as indicated by the dash lines. (**D**) The TRE accuracy of the test set flow transformation tensors predicted by U-Net++, GVTNet, and CEyeNet, respectively, at various relative error thresholds. The box center line denotes the median; box limits denote upper and lower quartiles; whiskers denote the 1.5× interquartile range; circles denote outliners. (**E**) Examples of predicted transformation tensors of CEyeNet and the baseline models. The color of the streamlines denotes the local magnitude of pixel displacement. Specific local feature differences across the predictions of different models are highlighted in red circles, with close-ups plotted right next to the corresponding flow profiles. (**F**) The inferences efficiency of CEyeNet and the baseline models in online inference mode and (**G**) in large-batch screening mode. Bar denotes the mean; error bars denote SDs.

using thresholded relative error (TRE) metric (Materials and Methods). The results demonstrate that CEyeNet consistently outperforms the state-of-the-art neural networks with a mean accuracy of more than 50% higher and an SD of 40% lower (Fig. 3D). Sample-wise comparisons show that CEyeNet is exceptionally accurate in determining both the global transformation trends and the details in the sharp transition zones (Fig. 3E).

Moreover, CEyeNet has a very compact architecture and requires significantly fewer floating-point operations than the baseline models, offering the high efficiency that is critical for the microchannel design tasks. In human-guided iterative design scenarios, the real-time knowledge of the flow transformations is essential for the users to evaluate and optimize the current microchannel design. Running on a NVIDIA GeForce RTX 4090 graphics processing unit (GPU), CEyeNet takes ~3 ms to predict the flow transformations in a microchannel containing eight HAOs, which is up to 44% less compared to the baseline models (Fig. 3F). For automatic inverse design tasks where high-throughput screening of a vast number of HAO designs is needed, prediction batch size is adjusted to fully use the GPU memory to maximize the screening efficiency. In this scenario, CEyeNet can predict around 120,000 flow transformation tensors per second using 24 GB of GPU memory, doubling the throughput of the baseline models (Fig. 3G).

The exceptional accuracy and efficiency of CEyeNet render it the optimal neural network model for the flow transformation prediction task. Additionally, the CE module demonstrates unique advantages in effectively capturing the long-range dependency within the scope of lightweight neural network models (note S1).

Predicting flow profiles with CEyeNet

The flow profiles in HAO-embedded microchannels can be predicted by numerically transforming the input flow profiles step by step using the model-predicted transformation tensors. The accuracy of predicted flow profile is measured by the intersection-over-union (IoU; see Materials and Methods) score between the predicted flow profiles and the ground-truth flow profiles, which are obtained by transforming the same input flow profiles using the ground-truth transformation tensors of the same HAOs.

We evaluate the accuracy of predicted flow profiles for 100 HAO-embedded microchannels, each containing 20 HAOs randomly sampled from the test dataset (Fig. 4A). The CEyeNet-predicted flow transformation tensors result in the mean flow-profile IoU of 0.907 after 10 transformations (Fig. 4B), substantially higher than the baseline models. Although the mean IoU of predicted flow profiles decreases as the number of HAOs increases, the mean IoU of CEyeNet-predicted flow profiles remains above 0.8 after 20 HAOs, which is more than sufficient for engineering complex flow shapes.

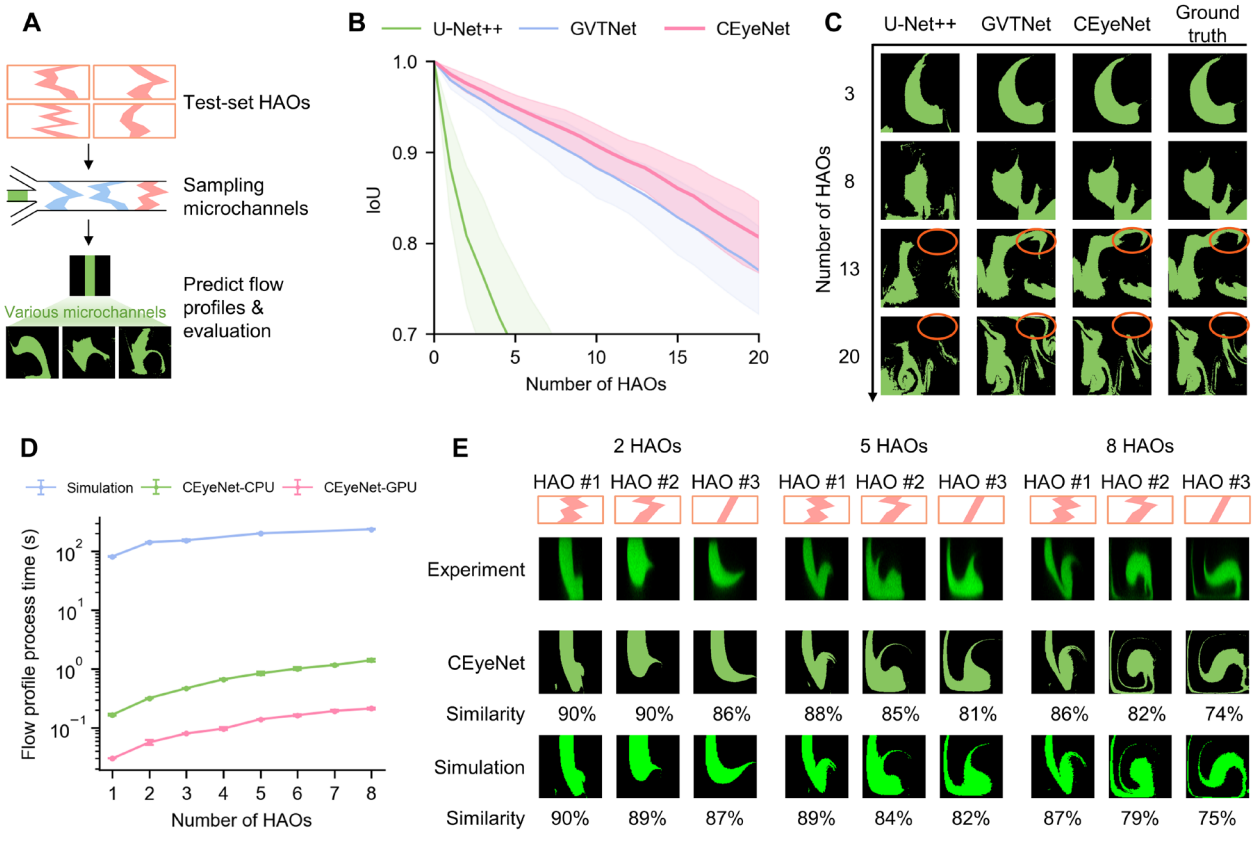

Fig. 4. Evaluation of predicted flow profiles. (**A**) The evaluation process of the flow profile prediction accuracy of CEyeNet and the baseline models. (**B**) The intersection-over-union (IoU) scores of predicted flow profiles that are transformed by different number of HAOs. The lines denote the mean; filled area denotes the SD range. (**C**) Examples of predicted flow profiles that are transformed by various number of HAO. Specific feature differences are highlighted with orange circles. (**D**) The flow profile processing time of CEyeNet using a CPU (Intel Core i9-13900) and GPU (NVIDIA GeForce RTX4090), in comparison to the finite element simulation. (**E**) Examples of flow profiles transformed by different numbers (2, 5, 8) of identical obstacles (HAO #1 to #3), acquired through confocal microscopy in experiment, CEyeNet prediction, and finite element simulation, respectively. The similarity of the numerically acquired flow profiles with respect to the experimental flow profiles is shown beneath them.

Compared to the ground-truth flow profiles, CEyeNet-predicted flow profiles show no obvious variance as the number of transformations increases (Fig. 4C). In contrast, errors in U-Net++– and GVTNet-predicted flow profiles gradually accumulate to a nonnegligible level, as highlighted by red circles in Fig. 4C. Additionally, the results suggest that the morphology of HAOs can affect the accuracy of flow profile prediction. Generally, HAOs with milder morphology transitions yield higher profile accuracies (note S2).

Now, investigating the flow transformations of unseen microchannel structures relies on CFD methods. To demonstrate the advantage of CEyeNet, we compared the accuracy and efficiency of CEyeNet with finite element simulations in predicting the transformed flow profiles with reference to the physical experiment results (Materials and Methods). CEyeNet requires only $^1/_{160}$ to $^1/_{470}$ of the time needed for simulation when running on a central processing unit (CPU) and $^1/_{1100}$ to $^1/_{2700}$ of the simulation time when running on a GPU (Fig. 4D). Furthermore, the flow profiles predicted by CEyeNet have comparable similarity as the simulation results when referenced against experimentally imaged flow profiles, showing no notable feature variance (Fig. 4E). In certain cases, CEyeNet even produces noticeably more accurate predictions than the simulation, with observably higher feature fidelity (Fig. 4E, 8 HAOs, HAO #2), demonstrating the enhanced flow profile prediction ability of CEyeNet. The accuracy of the experimentally reproduced flow profiles can be influenced by the fabrication errors of the HAOs and the microchannels. We model and test three fabrication variations: fillet ratio (fr) of HAO vertices, aspect ratio (ar) of microchannel cross sections, and misalignment ratio (mr) between the top and bottom halves of the microchannel (fig. S7A). Results show that the flow profile is robust to fr and ar but highly sensitive to mr (fig. S7, B to F), underscoring the importance for accurate microchannel alignments during the fabrication process (Materials and Methods). With common microfluidic fabrication equipment and techniques, the HAOs and the microchannel can be precisely fabricated with microchannel cross-sectional dimension down to 100 μm by 100 μm (fig. S8) to accurately reproduce the flow profiles.

Flow manipulations enabled by CeyeHao

Combining the HAO framework with the CEyeNet-enabled prediction, CeyeHao enables real-time access to a significantly more complex and expansive flow profile design space than previously possible. We demonstrate that CeyeHao facilitates powerful flow manipulations through human-guided iterative designs and automated searches.

To facilitate the use of CeyeHao by individuals without expertise in flow dynamics or programming, we developed a flow programming design software with a graphical user interface (GUI) (fig. S9). The GUI allows users to easily configure the input flow profiles and the HAO morphologies and get the feedbacks on the output flow profiles in real time. Therefore, the designers can first conceptualize the desired flow transformation patterns, roughly estimate the corresponding angles and lengths of the oblique walls of the HAO, add the HAO to the microchannel, and check the resulting flow profile. They can then fine-tune the flow profile topology by adjusting the parameters of the HAO with real-time feedbacks (Fig. 5A). For complicated flow transformation patterns that are not achievable with a single HAO, more HAOs can be added. The designers are also able to adjust the parameters of the previous HAOs to further fine-tune the flow shape. As a result, diverse flow profiles and flow manipulations can be achieved through human guided iterative design.

To demonstrate the precise feature-shaping capability of CeyeHao, we use the GUI to design flow profiles with highly regulated geometries (Fig. 5B). Taking the shaping of a rectangle in Fig. 5B as an example, the initial two HAOs push the ends of the stripe-shaped flow profile away from the microchannel walls, inevitably rounding the edges into curves. The subsequent two HAOs further rotate the top and bottom of the flow profile from the lateral sides to flatten the tips and form the four vertices of the rectangle. The final two HAOs delicately transform the curved edges to fine tune the curvature and length, thereby finalizing the shape of the rectangle. By tuning the edges with various lengths and curvatures, square and circle flow profiles can also be achieved (Fig. 5B). It is worth noting that, as the transformation of flow profile goes from coarse to delicate, more complex obstacle shapes are required. The numerous continuous DOF and modes of flow transformation provided by HAO uniquely enable this delicate regulation of flow profiles. Moreover, the described flow transformation process closely reflects the pottery-analog design logic of flow profiles, which is sequentially creating new features from the initial flow profile shape while gradually refining these features into their final states. CeyeHao allows for the creation of sophisticated and precise flow shapes in an accurate and efficient manner.

To demonstrate the capability of CeyeHao in programming flow profiles with intricate and meaningful features, we design microchannels to draw artistic animals on flow profiles, such as the dolphin and the swan (Fig. 5C). These drawings encompass numerous fine details and features, such as the body, fins, and the tails of the dolphin, as well as the head, neck, and the wings of the swan. Producing these drawings inside a microchannel requires simultaneous achievement of precise control over the dimensions and curvatures of the local features, as well as integrating these features without inducing undesired artifact features. Moreover, the complex flow profiles designed numerically are successfully reproduced in experiments with ease (Fig. 5C), as the HAOs can be readily fabricated using basic soft lithography technologies (Materials and Methods) (*33*). This extent of flow control is unprecedented in microfluidics, showcasing the superior flow programming capability of CeyeHao and its substantial potential for enabling new microfluidic functions.

Moreover, the advanced flow manipulation ability of CeyeHao extends to multiple flow streams. To demonstrate this, we design microchannels to split streams precisely along an axis and to merge streams while maintaining their initial morphology (Fig. 5D). The capability of CeyeHao to simultaneously manipulate multiple streams and control their shapes highlights its potential for fluidic assembly of multiple materials with tailored morphologies, which can further facilitate microfluidic fabrication of multifunctional microstructures with complex shapes.

In practice, automatic inverse design of microchannels for target flow manipulation is highly desirable to reduce the burden on human designers. Nevertheless, automatic inverse design for producing complex flow profiles has not yet been demonstrated. Using the high-throughput screening capability of CeyeHao, we first approach the automatic inverse design tasks for complex flow profiles using random stochastic search (Fig. 5E and Materials and Methods). As a proof of concept, we performed automatic inverse design tasks to produce the specified swan-shaped flow profile using CeyeHao and another state-of-the-art automatic design tool, FlowSculpt (*16*), which uses genetic optimization to sequence the pillar obstacles

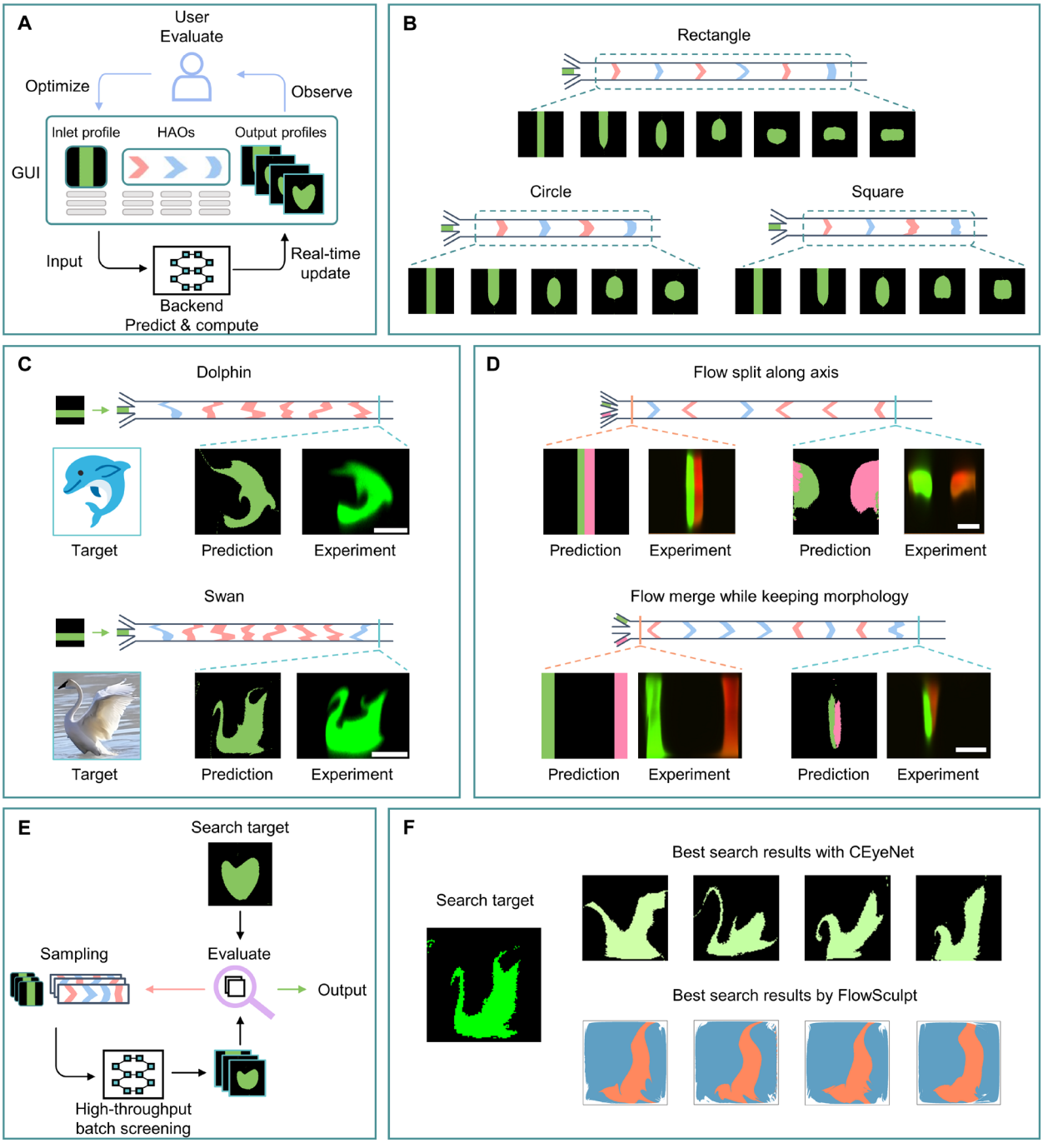

Fig. 5. Demonstration of advanced flow programming enabled by CeyeHao. (**A**) The diagram of human-guided iterative design process for microchannels that perform a specific flow manipulation. (**B**) CeyeHao-designed microchannels that can produce flow profiles of highly regulated geometries. (**C**) CeyeHao-designed microchannels that produces artistic drawings with rich meaningful features. Scale bars, 100 μm. (**D**) CeyeHao-designed microchannels for split or merge of multiple streams with precise morphology controls. Scale bars, 100 μm. (**E**) The diagram of automatic inverse design process via CeyeHao-enabled stochastic search. (**F**) The best search results for a swan flow profile by CEyeNet and FlowSculpt, respectively.

from a finite toolset. CeyeHao successfully finds microchannels that integrate all the essential features of a swan profile, such as the head, neck, wings, and main body (Fig. 5F). Although the output flow profiles exhibit noticeable quantitative deviations from the design target, they remain qualitatively interpretable as a swan. In comparison, FlowSculpt fails to find flow profiles that include swan heads and necks and produces flow profiles with redundant features such as cracks and spikes. Moreover, to facilitate inverse design scenarios where quantitative similarity is preferred, we develop a genetic optimization algorithm-based inverse design program for CeyeHao. We systematically compare its performance with FlowSculpt on the design performance toward a diverse target shape collection (fig. S10).

In most cases, CeyeHao-achieved flow profiles are more quantitatively similar to the target profiles and requires fewer microstructures (fig. S11, A and B). However, the search process with CeyeHao takes ~25 min, which over 10 times longer than FlowSculpt (fig. S11C). This increased computation time is attributed to less optimized code efficiency and the additional computational demands of the neural network. Despite the longer computation time, the design process using CeyeHao constitutes only a minor portion of the overall flow programming workflow, which includes hours to days of microfluidic device fabrication and experimentation. It should be noted that FlowSculpt offers a user-friendly interface and additional functionalities such as multi-steam flow profile searching, which are now unavailable in CeyeHao's inverse design program. The designers can, therefore, select the appropriate design programs and methods according to their specific requirements.

DISCUSSION

In this work, we present CeyeHao, a comprehensive flow programming methodology for designing microchannels that enables complex and delicate flow manipulation. CeyeHao leverages a microchannel framework, HAO, which provides multi-folded flow profile configurability and complexity. The transformed flow profiles in HAO-embedded microchannels are predicted by a bespoke neural network model, CEyeNet, which demonstrates with state-of-the-art accuracy and efficiency, surpassing the baseline neural network models and the finite element simulation method. As a result, CEyeNet exclusively enables the seamless employment of the entire flow transformation space of HAOs. To showcase the capabilities of CeyeHao, we apply it to design microchannels that produce feature-rich semantic drawings with highly regulated morphologies and to precisely manipulate the topology of two streams, reflecting its superior global and local flow manipulation ability. Moreover, CeyeHao enables integrating all essential features of complex flow profiles in automatic inverse design tasks. CeyeHao is poised to significantly enhance microfluidic fabrication technologies by enabling the production of intricate morphologies and material compositions. Additionally, it holds great potential for a wide range of microfluidic applications where precise control over spatial flow distribution is required. Specifically, we envision the use of CeyeHao in extrusion-based bioprinting for its simple and compact microchannel structure, which can be easily integrated into printing nozzles for filament or sheet-based bioprinting (*34*). The system's loose requirement of *Re* makes it compatible with highly viscous bioprinting materials. By tailoring HAOs within the printing nozzle, CeyeHao could enable multimaterial extrusion bioprinting with biomimetic intertissue structures at the microscale (*35*), facilitating the creation of complex vascularized and innervated tissue structures (*36*). Moreover, CeyeHao offers powerful approaches for engineering the microfluidic environments in bioreactors. HAOs can precisely manipulate three-dimensional (3D) shear, pressure, and concentration gradients within microfluidic channels, thus creating finely tuned microenvironments for cells (*37*) and enhancing substance exchange and production efficiency (*38*). The passive flow regulation of CeyeHao further simplifies its adaptation for bioreactors.

A key component contributing to CeyeHao's advanced flow programming ability is its high-DOF microchannel module framework, which fundamentally improves diversity and complexity of the flow transformations. Previous research has focused significantly on optimizing the obstacle sequences to expand the design capabilities, while few substantial improvements have been observed. Our study introduces an alternative paradigm in flow programming researches by fundamentally broadening the diversity and complexity of accessible flow transformations, instead of only optimizing the modular sequence with limited flow transformation variations.

Additionally, our work underscores the critical role of AI in advancing flow programming. The design process of the flow programming inherently involves numerous trials, whether automatic or human guided. However, conventional flow investigation methods, including simulation, significantly limit the trial efficiency, thus becoming a major bottleneck in expanding the toolset of flow programming methods. Even if a more advanced microchannel scheme is proposed, without efficient flow investigation tool, its expansive flow transformation space cannot be effectively explored, leading to significantly restricted improvements. As demonstrated in this work, deep learning holds tremendous potential in addressing the design efficiency issue, making us anticipate that deep learning will become the mainstream flow investigation tool for flow programming.

While CeyeHao offers unparalleled flow programming capability, it does not require demanding computational resources or specialized fabrication equipment. The HAOs can be easily fabricated via soft lithography, one of the most prevalent and fundamental microchannel fabrication techniques. Additionally, the transformation of flow profiles occurs passively, circumventing the high demand for equipment and expertise by the active methods, such as acoustic (*39*) and magnetic (*40*) flow excitations. Furthermore, because the flow operates within the inertialess flow regime, it does not require stringent velocity conditions like those needed in inertia-based methods. The AI-empowered numerical design tools are lightweight and can be executed on mainstream computers or laptops, offering magnitudes higher efficiency compared to traditional simulations. The GUI also enables users without a background in deep learning or flow dynamics to effectively use the system. We anticipate that CeyeHao will enable broad communities to explore previously unknown functionalities and applications with the aid of advanced microfluidic flow manipulations.

CeyeHao could also contribute to the development of automated intelligent flow programming systems. While human-guided iterative design offers substantial flexibility for diverse task goals, it becomes increasingly challenging to make contributive design decisions as the complexity of the design target increases. Meanwhile, the increase in target complexity significantly expands the microchannel configuration space, rendering the stochastic searching methods including the genetic algorithms increasingly inefficient. Therefore, producing arbitrary complex flow profiles is more likely to be accomplished through highly intelligent design methods. Modern large-scale neural network architectures, such as diffusion models (*41*), have shown potential in solving complex inverse problems (*42*) and could provide a promising solution for the inverse design of flow programming processes. The training of such models requires a substantial dataset, which could be generated with CeyeHao in high-throughput for this future development.

One limitation of the CeyeHao system is its susceptibility to diffusion-induced blurring of flow profiles, attributed to its operation at $Re < 1$ and consequently lower Peclet numbers ($Pe = Re \times \nu / D$, where ν is the kinematic viscosity and D is the diffusivity). While

inertial-based methods typically operate at higher *Re* (*10*–*40*) with sharper flow edges, CeyeHao requires working fluids with sufficiently high viscosity to achieve comparable Peclet numbers and minimize blurring. For fluids with very low viscosity, such as deionized water, achieving the necessary $Pe > 10^4$ while maintaining $Re < 1$ is challenging. In such scenarios, increasing the fluid viscosity by adding compatible chemicals, such as glycerol, can help mitigate this issue. The influence of these flow conditions on profile quality is essential to consider for optimal application of CeyeHao. Additionally, it should be noted that CeyeHao is developed and evaluated on microchannels with square cross sections. Therefore, the conclusions of the performance comparisons are specifically applicable to square microchannels. Other flow programming methods, such as the pillar-based techniques that are originally developed in microchannels with high width-to-height ratio, may have better performance in microchannels with other aspect ratios. Nevertheless, CeyeHao can be developed for microchannels with various aspect ratios or microchannel cross-sectional shapes by following the same workflow described in this study. We anticipate that diverse and substantial flow transformations, similar to that observed in square microchannels, could be achieved with these adaptations of CeyeHao.

MATERIALS AND METHODS

Digitization of microfluidic flow system

In the present study, the flows in the microchannels are assumed Newtonian Stokes flow, and the flow at the inlets, outlet, and between the obstacles all reaches fully developed states. Therefore, the transformed flow profiles only depend on the inlet flow profile, and the morphologies of microchannel obstacles.

The flow profiles are represented by tensors of shape (*W*, *W*, *C*), where *W* is the number of elements along the *y* and *z* axes of microchannel, as indicated in Fig. 1A, and *C* is the number of traced flow streams

$$\mathbf{P} = \left(P_{ijk}\right), i, j \in \{1, 2, \ldots, W\}, k \in \{1, 2, \ldots, C\}$$

The flow profile tensor is analogous to a multichannel image, where each pixel is physically equivalent to a flow element with finite size.

Correspondingly, the flow transformation between the input flow profile $\mathbf{P}_{\text{in}}$ and the output flow profile $\mathbf{P}_{\text{out}}$ becomes the relocation of the pixels of $\mathbf{P}_{\text{in}}$. Therefore, the flow transformation is represented by a tensor of shape (*W*, *W*, *2*)

$$\mathbf{T} = \left(T_{ijk}\right), i, j \in \{1, 2, \ldots, W\}, k \in \{1, 2\}$$

$$T_{ij1} = i' - i$$

$$T_{ij2} = j' - j$$

$$\left(i, j\right) = \left(i' - T_{ij1}, j' - T_{ij2}\right)$$

where i and i' are the *y*-axis positions of the flow element after and before relocation, respectively, and j and j' are the *z*-axis positions of the flow elements after and before relocation, respectively.

$\mathbf{T}$ is intrinsically a mapping function for elements in $\mathbf{P}_{\text{in}}$ and $\mathbf{P}_{\text{out}}$

$$\mathbf{P}_{\text{out}}\left(i, j\right) = \mathbf{P}_{\text{in}}\left(i + T_{ij1}, j + T_{ij2}\right)$$

Additionally, $\mathbf{T}$ is an array of 2D vectors and, thus, can be visualized as streamline plots as shown in Figs. 1 to 3 or 2D vector fields as shown in fig. S1, S2, and S5.

The obstacles are represented by their top-view images in the *xy* plane, as they are 2.5D structures extruded from *xy* plane with the same dimension along the *z* axis. Additionally, obstacles are painted in blue if they are placed on the top wall and in orange if they are placed on the bottom wall.

Given an input flow profile tensor and the flow transformation tensors of the obstacles in a microchannel, the output flow profile tensor can be computed by iteratively remapping the input flow profile tensor with the flow transformation tensors of the obstacles in the sequence of their presence in the microchannel

$$\mathbf{P}_{\text{out}} = \mathbf{P}_{\text{in}} \circ \mathbf{T}^{(1)} \circ \mathbf{T}^{(2)} \circ \cdots \circ \mathbf{T}^{(N)}$$

where N is the number of obstacles in the microchannel, $\mathbf{T}^{(i)}$ is the transformation tensor of the ith obstacle in the microchannel, and $\circ$ denotes applying the flow transformation tensor to the flow profile tensor.

MPDs and MPS

We refer the detailed implementation of calculating MPD and MPS to the code provided. Below is a high-level summary of the calculation process. Given the flow transformation tensors of the N modules of a microchannel, the total flow transformation tensor for the whole microchannel T' is computed

$$\mathbf{T}' = \mathbf{T}^{(0)} \circ \mathbf{T}^{(1)} \circ \mathbf{T}^{(2)} \circ \cdots \circ \mathbf{T}^{(N)} - \mathbf{T}^{(0)}$$

where

$$\mathbf{T}^{(0)}_{ij} = \left(i, j\right)$$

The pixel displacement magnitude tensor $\mathbf{A}$ is given by

$$A_{ij} = \sqrt{\left(T'_{ij1}\right)^2 + \left(T'_{ij2}\right)^2}$$

The mean of total displacement amplitude of all pixels in the resultant flow transformation tensor is defined as the MPD

$$\text{MPD} = \frac{1}{W^2} \sum_{i=1}^{W} \sum_{j=1}^{W} A_{ij}$$

Pixel segregation ΔA is measured by calculating the differences in displacement magnitudes between each pixel and its adjacent pixels along the *y* and *z* axes and then averaging the sum of these differences

$$\Delta A_{ij} = \frac{1}{2}\left(\left\|T'_{ij} - T'_{i+1,j}\right\| + \left\|T'_{ij} - T'_{i,j+1}\right\|\right)$$

The mean of pixel segregation of all pixels in the resultant transformation tensor is defined as MPS

$$\mathrm{MPS} = \frac{1}{W^2}\sum_{i=1}^{W-1}\sum_{j=1}^{W-1}\Delta A_{ij}$$

Finite element simulation of HAO-embedded microchannels

The finite element simulation was performed using COMSOL 5.0 (COMSOL AB, Stockholm, Sweden). Microchannels containing several HAOs are modeled. The width and height of the microchannel cross section are both 300 μm. The distances between adjacent HAOs and between HAOs and inlets/outlets were set sufficiently long to ensure that the flow is fully developed around every HAO. The fluid material is mixture 80 wt % glycerol and 20 wt % water, of which the density and viscosities are calculated using this web calculator: www.met.reading.ac.uk/~sws04cdw/viscosity_calc.html. As this study assumes $Re < 1$ and the inertia force is negligible, the creeping flow module was used. The inlet flow was set at a fully developed state at steady flow rates satisfying $Re < 1$, and the outlet was defined with zero pressure. The pressure and velocity field were discretized in first order and solved. Detailed configurations and schematics of the simulation can be found in note S3 and fig. S12.

Computation of flow transformation tensors of individual obstacles

The flow in a microchannel containing a single HAO or other microchannel features was simulated as described above. The streamlines between the inlet and the outlet were computed. Notably, the streamlines were seeded by the endpoints on the outlet, which are aligned to the centers of the pixels of the flow profile and traced backward to the inlet cross section of the microchannel (fig. S12C). The net displacements of the flow elements along each streamline were computed by subtracting the endpoint coordinates from the start point coordinates. The displacements of all the streamlines form the flow transformation tensor.

Dataset generation

To ensure that the randomly sampled HAOs are physically valid and fabricable with basic soft lithography technology (e.g., no intersecting oblique walls or tiny features), we put the following constrains on the HAO parameter space

$$n = 5$$

$$-175\ \mu\mathrm{m} \le x_{i0} \le 175\ \mu\mathrm{m}$$

$$90\ \mu\mathrm{m} \le x_{i1} - x_{i0} \le 210\ \mu\mathrm{m}$$

$$30\ \mu\mathrm{m} \le y_i - y_{i-1} \le 108\ \mu\mathrm{m}$$

To ensure an unbiased representation of the regulated HAO parameter space, we quasi-randomly sample the coordinates of HAO using sobol generator provided in PyTorch (Linux Foundation, USA). Note that only bottom obstacles are included in the dataset, as the top obstacles sharing the same top-view shape have symmetric flow transformation to the bottom obstacles.

The top-view image of the HAO is drawn according to the sampled parameters using Matplotlib (The Matplotlib development team). The top-view image is then resized, converted to grayscale, and autocontrasted before being input into the neural network. The flow transformation tensors of the HAOs were computed following the methods in the previous section.

It should be noted that, while the elements of flow transformation tensor represent the pixel displacement and thus should have integer values, the elements in the transformation tensors used for training have floating point values in a physical continuous space, i.e., the displacement of the flow elements in the microchannel cross-sectional plane. The continuous-space transformation tensor is directly feed into the model to learn a finer mapping of the underlying physical principles. Before using the transformation tensor for prediction and evaluation, the element values of the transformation tensor are rounded to get integers.

To automate the large number of simulations involved in data generation, a batch simulation program was developed using the Application Builder Module in COMSOL, which takes in the coordinate lists of the obstacles, automatically setting up and running the simulation cases for each obstacle and exporting the streamline data.

The dataset is further augmented through randomly inverting the obstacle images and randomly translating the obstacle images in the x-axis direction while keeping the original flow transformation tensors.

Architectures of CEyeNet and the benchmarked models

The architecture of the proposed CEyeNet and the benchmarked models in this work, including U-Net++, GVTNet, and U-Nets with enhanced receptive fields, are briefly overviewed here. Detailed implementations can be found in the provided codes.

CEyeNet is constructed by inserting CE modules into the skip pathways of the original U-Net (*31*), respectively. In each CE, the input feature map of shape B, C, H, W is first average-pooled at ratio of t, resulting in a thumbnail of the input feature map of shape $B, \frac{C}{t}, \frac{H}{t}, W$. Then, the thumbnail is replicated t^2 times and tiled into an t-row–by–t-column array of shape $B \times C \times H \times W$. The encoder/encoder in the CEyeNet consists of four downsampling/upsampling blocks, each consisting of sequentially connected one max-pooling/transposed convolution layer and two convolutional layers. Additionally, we appended a convolutional layer to the neural network to adjust the shape of the output feature map to be the same as the flow transformation tensor.

U-Net++ is an augmented version of U-Net with nested and dense skip connection paths, which facilitates the generation of features at multiple scales. In this work, the original implementation of the U-Net++ is used for benchmarking.

We take GVTNet as a representative of models with global receptive fields, as it is a U-Net–based model architecture and has a relatively lightweight implementation compared with the other transformer-type models. GVTNet replaced the up- and downsampling layers with attention-based global voxel transformer operators. Additionally, the stacked conv layers are replaced with more modern Resnet blocks (*43*). In this work, we build a GVTNet network following the original implementation with GVTO v1.

For all the models, the latent dimensions of the feature map after the first convolution layer are 64, which is doubled/halved at each downsampling/upsampling depth. All the models are implemented using PyTorch framework (*44*).

Model training

All the models studied in this work are trained following the same protocol, with key points highlighted as follows. Full implementation details about model training can be found in the provided codes (*45*).

Before the obstacle images are input into the neural networks, the images are first binarized and resized to the same spatial resolution as the output flow transformation tensors. Our initial model training experiments shows that the training batch size of eight yields the optimum accuracy, and this batch size is used for all models.

The models are trained in an end-to-end manner. The element values in individual transformation tensors generally follows a Poisson distribution, with majority of them being close to 0 and a few of them being two magnitudes higher (fig. S5A). Despite the scarcity of the large-value elements, they are equally important as the low-value elements. Moreover, the transformation tensor should have smooth value transitions across the major area of itself, as a result of the physical continuity of fluid flow, and will be iteratively applied to the input profiles. Therefore, minimizing the random directional noise is important for flow profile prediction quality. As a result, we used a combination of mean absolute error (MAE) loss and mean absolute loss of gradients (MAGE)

$$\mathrm{Loss}_{\mathrm{total}}\left(\mathbf{Y},\hat{\mathbf{Y}}\right)=\mathrm{Loss}_{\mathrm{MAE}}\left(\mathbf{Y},\hat{\mathbf{Y}}\right)+\mathrm{Loss}_{\mathrm{MAGE}}\left(\mathbf{Y},\hat{\mathbf{Y}}\right)$$

$$\mathrm{Loss}_{\mathrm{MAE}}\left(\mathbf{Y},\hat{\mathbf{Y}}\right)=\frac{1}{N}\sum_{i=1}^{N}\left|\mathbf{Y}_i-\hat{\mathbf{Y}}_{\mathrm{i}}\right|$$

$$\mathrm{Loss}_{\mathrm{MAGE}}\left(\mathbf{Y},\hat{\mathbf{Y}}\right)=\mathrm{Loss}_{\mathrm{MAE}}\left(\frac{\delta\mathbf{Y}}{\delta x},\frac{\delta\hat{\mathbf{Y}}}{\delta x}\right)+\mathrm{Loss}_{\mathrm{MAE}}\left(\frac{\delta\mathbf{Y}}{\delta y},\frac{\delta\hat{\mathbf{Y}}}{\delta y}\right)$$

where $\mathbf{Y}$ is the ground truth of the flow transformation tensor, $\hat{\mathbf{Y}}$ is the predicted flow transformation tensor, and N is the total number of elements in the flow transformation tensor.

Stochastic gradient descendent method is used to update the model parameters. The learning rate is updated per training epoch using a cosine learning rate decay function with warmup (*46*, *47*). The device used for training equips an Intel i9-13900k CPU, NVIDIA RTX4090 GPU, and Ubuntu 22.04.2 LTS system. Automated mixed precision training is applied to accelerate the training for all the models present in this work except GVTN, which is incompatible possibly due to gradient underflowing.

Evaluation of predicted flow transformation tensors

We designed a TRE metric to specifically evaluate the accuracy of the predicted flow transformation tensor with reference to the ground truth. TRE computes the percentage of elements that have relative errors below the specified threshold

$$\mathrm{TRE}=\frac{1}{\mathrm{n}}\sum_{i=1}^{C}\sum_{j=1}^{H}\sum_{k=1}^{W}I\left(\left|\frac{Y_{ijk}-\hat{Y}_{ijk}}{\hat{Y}_{ijk}}\right|\leq\mathrm{RET}\right)$$

where n is the number of elements; I is an indicator function that returns 1 if the condition inside the brackets is true, and 0 otherwise; and RET is the relative error threshold. This metric ensures uniform significance of elements throughout the spatial domain and demonstrates robustness against substantial absolute errors in elements with values approaching zero.

Evaluation of predicted flow profiles

To evaluate the accuracy of the predicted flow profiles versus the ground-truth flow profiles, we computed the IoU score

$$\mathrm{IoU}=\frac{|\hat{\mathbf{Y}}\cap\mathbf{Y}|}{|\hat{\mathbf{Y}}\cup\mathbf{Y}|}$$

where $\hat{\mathbf{Y}}$ and $\mathbf{Y}$ are the predicted and ground-truth flow profiles, respectively.

To evaluate the accuracy of the numerical flow profiles with respect to the experimental ones and the designed flow profiles with respect to the inverse design target, we used a perceptual similarity metric inspired from the previous work (*48*), as feature-level resemblance is prioritized over pixelwise exact matching in this scenario. The sample flow profiles are inputted into a VGG16 model (*49*) pretrained with dataset ImageNet-1K (*50*). The feature maps are compared at the neural network layers no. 4, 9, 16, and 23 using TRE metric, resulting in a percentage-based similarity score at each layer, respectively. The final similarity score is given by the weighted-sum of these feature-level similarity values

$$SI(y,\hat{y})=\sum_{l}w_l\odot\mathrm{TRE}(y_l,\hat{y}_l)$$

where l is the layer of the VGG16 model where the feature map is extracted for evaluation, w_l is the weight for similarity score at layer l, and y_l and $\hat{y}_l$ are the feature maps of the prediction and the ground truth at layer l, respectively. *SI* presents aligned evaluation results to human perception.

Additionally, before computing the SI with reference to experimentally imaged flow profiles, the experimental flow profile photos were first cropped to contain only the channel cross sections and then resized to the same resolution as the numerical flow profiles. To filter our imaging noise, the experimental flow profiles were converted to binary images by applying Gaussian blur and adaptive thresholding. These image preprocesses used OpenCV functions, with detailed implementation available in the provided codes.

Inverse design of HAO-embedded microchannels toward producing target flow profiles

The inverse design tasks are approached by random stochastic searching and genetic algorithm, respectively. We refer the implementation details to the code provided and provide the high-level overview here.

The random stochastic searching starts with specifying the target flow profile and the similarity metric used to measure the quality of the output flow profile, and the similarity threshold to determine if the flow profile has satisfactory quality. In a single search, the flow rate ratio and position of the input flow profile and the parameters of the HAOs in the microchannel are sampled. The image of the HAOs is then generated accordingly, from which the transformation tensors and the output flow profiles are predicted. If the output flow profiles are evaluated to be satisfactory, then the configurations of

the input flow profile and the HAOs in the microchannel, along with the output flow profiles are recorded. The search is repeated until the specific number of search or satisfactory flow profiles is reached. The search is run in batch and in parallel to optimize the computation efficiency.

In the genetic algorithm–based inverse design, each chromosome consists of the parameters of a single inlet flow profile and HAOs of a single microchannel. In each generation of search, the raw fitness score of each chromosome is computed the same as in the random stochastic search, and the ranked fitness of the chromosomes is computed. Elitism, Laplace crossover (*51*), and Gaussian distribution–sampled mutation are used to update the chromosomes.

Design and fabrication of microfluidic device

Each HAO was situated within an adequately long, empty segment of the microchannel to ensure fully developed flow at both ends of the microchannel module. The height of the HAO structure was precisely half that of the microchannel, resulting in a dual-layer microchannel architecture. Consequently, the microchannel was bisected horizontally, creating two distinct components that could be easily fabricated using fundamental soft lithography techniques (fig. S13A). Initially, molds for the top and bottom microchannel components (fig. S13B) were created via lithography using SU-8 photoresist (2075, Kakayu Advanced Materials, USA) and 4-inch (10.16-cm) silicon wafers (N100, University, USA) with a maskless lithography machine (MicroWriter ML3 Baby Plus, Durham Magneto Optics, UK). To facilitate the release of polydimethylsiloxane (PDMS) devices, the molds were placed in a vacuum chamber along with a petri dish containing 20 μl of trichloro(1*H*,1*H*,2*H*,2*H*-perfluorooctyl) silane (Sigma-Aldrich, USA) for 30 min.

The PDMS precursor and curing agent (Sylgard 184 Elastomer Kit, Dow Corning Corporation) were mixed at a weight ratio of 10:1. The top microchannel device was cast into a 3-mm-thick PDMS layer, while the bottom microchannel device involved spin-coating PDMS onto the mold at 500 rpm for 30 s, resulting in a thin PDMS film of ~250 μm thick. The PDMS polymers were subsequently cured on hot plates at 75°C for 1.5 hours. After casting, 1-mm-radius outlet holes were punched into the top microchannel component. The bottom microchannel component was inverted, with microchannel side facing up and the back side attached to a 0.17-mm-thick cover glass for structural support. The use of cover glass rather than glass slides also reduced imaging aberrations in confocal microscopy.

The top and bottom microchannel parts were aligned using a digital microscope. During alignment, the bottom part is fixed to the microscope platform, while the top part was manually positioned above it. The top microchannel part was delicately placed onto the bottom part, leading to a natural and reversible bonding. This alignment process may require repeated detachment and realignment to achieve optimal alignment.

To ensure adequate sealing capable of withstanding microfluidic flow pressure, customized clamping plates (fig. S13B) were used. The top clamping plate featured through-holes corresponding to the microchannel's inlet and outlet positions, permitting tube connections to the microchannel. To avoid optical interference during confocal microscopy, the bottom clamping plate was cut out in the area corresponding to the main microchannel path. These clamping plates were laser-cut from 2-mm-thick polymethyl methacrylate sheets and secured with binder clips.

Confocal imaging of flow profiles

Before the experiment, the microchannel walls were dyed by infusing saturated rhodamine B water solution into the microchannel for overnight. Ethanol was infused into the microchannels to wash off the dye solution and purge the air bubbles trapped in the microchannel. A mixture of 80 wt % glycerol and 20 wt % deionized water was used as the working fluid. The flow streams of interest were labeled with fluorescein sodium salt (Sigma-Aldrich, USA) at a concentration of 1 mM. The core flow and the sheath flow were infused into the microchannel from different inlets using syringe pumps.

The flow profiles were captured using a confocal microscope (AX R, Nikon Instruments Inc., Japan). After the flow becomes steady, *z*-stack images of the microchannel modules were captured using 10× air objectives (Nikon, numerical aperture of 0.45). During the image acquisition, the confocal microscope used a resonant scanner, and four times signal averaging was applied. The flow profile images were cross-sectioned from the *z*-stack images. Height correction was applied using ImageJ (National Institutes of Health, USA) to the flow profile images to account for the compression in the *z* dimension caused by the mismatch of refraction index of the media between the working fluid and the objective immersion. The height correction factor *c* is approximated by

$$c = \frac{\mathrm{RI}_{\mathrm{PDMS}}}{\mathrm{RI}_{\mathrm{air}}} \approx \frac{1.4}{1} = 1.4$$

where $\mathrm{RI}_{\mathrm{PDMS}}$ and $\mathrm{RI}_{\mathrm{air}}$ are the refractive indices of PDMS and air, respectively.

Supplementary Materials

This PDF file includes:

Supplementary Notes S1 to S3
Figs. S1 to S13

REFERENCES AND NOTES

1. H. Ceylan, I. C. Yasa, U. Kilic, W. Hu, M. Sitti, Translational prospects of untethered medical microrobots. *Prog. Biomed. Eng.* **1**, 012002 (2019).
2. G.-Z. Yang, J. Bellingham, P. E. Dupont, P. Fischer, L. Floridi, R. Full, N. Jacobstein, V. Kumar, M. McNutt, R. Merrifield, B. J. Nelson, B. Scassellati, M. Taddeo, R. Taylor, M. Veloso, Z. L. Wang, R. Wood, The grand challenges of *Science Robotics*. *Sci. Robot.* **3**, eaar7650 (2018).
3. A. Servant, F. Qiu, M. Mazza, K. Kostarelos, B. J. Nelson, Controlled in vivo swimming of a swarm of bacteria-like microrobotic flagella. *Adv. Mater.* **27**, 2981–2988 (2015).
4. I. C. Yasa, A. F. Tabak, O. Yasa, H. Ceylan, M. Sitti, 3D-printed microrobotic transporters with recapitulated stem cell niche for programmable and active cell delivery. *Adv. Funct. Mater.* **29**, 1808992 (2019).
5. C. Yang, X. Liu, X. Song, L. Zhang, Design and batch fabrication of anisotropic microparticles toward small-scale robots using microfluidics: Recent advances. *Lab Chip* **24**, 4514–4535 (2024).
6. Y. Yu, L. Shang, W. Gao, Z. Zhao, H. Wang, Y. Zhao, Microfluidic lithography of bioinspired helical micromotors. *Angew. Chem. Int. Ed.* **56**, 12127–12131 (2017).
7. G. Tang, L. Chen, L. Lian, F. Li, H. Ravanbakhsh, M. Wang, Y. S. Zhang, C. Huang, Designable dual-power micromotors fabricated from a biocompatible gas-shearing strategy. *Chem. Eng. J.* **407**, 127187 (2021).
8. L. Cai, Z. Luo, H. Chen, Y. Zhao, Lithographic microneedle-motors from multimodal microfluidics for cargo delivery. *Small* **19**, e2206108 (2023).
9. Y. Cheng, Y. Yu, F. Fu, J. Wang, L. Shang, Z. Gu, Y. Zhao, Controlled fabrication of bioactive microfibers for creating tissue constructs using microfluidic techniques. *ACS Appl. Mater. Interfaces* **8**, 1080–1086 (2016).
10. Y. Zuo, X. He, Y. Yang, D. Wei, J. Sun, M. Zhong, R. Xie, H. Fan, X. Zhang, Microfluidic-based generation of functional microfibers for biomimetic complex tissue construction. *Acta Biomater.* **38**, 153–162 (2016).

11. M. Samandari, F. Alipanah, K. Majidzadeh-A, M. M. Alvarez, G. Trujillo-de Santiago, A. Tamayol, Controlling cellular organization in bioprinting through designed 3D microcompartmentalization. *Appl. Phys. Rev.* **8**, 021404 (2021).
12. D. Dendukuri, D. C. Pregibon, J. Collins, T. A. Hatton, P. S. Doyle, Continuous-flow lithography for high-throughput microparticle synthesis. *Nat. Mater.* **5**, 365–369 (2006).
13. C. C. J. Alcântara, S. Kim, S. Lee, B. Jang, P. Thakolkaran, J.-Y. Kim, H. Choi, B. J. Nelson, S. Pané, 3D fabrication of fully iron magnetic microrobots. *Small* **15**, e1805006 (2019).
14. H. Amini, E. Sollier, M. Masaeli, Y. Xie, B. Ganapathysubramanian, H. A. Stone, D. Di Carlo, Engineering fluid flow using sequenced microstructures. *Nat. Commun.* **4**, 1826 (2013).
15. A. L. Thangawng, P. B. Howell Jr., J. J. Richards, J. S. Erickson, F. S. Ligler, A simple sheath-flow microfluidic device for micro/nanomanufacturing: Fabrication of hydrodynamically shaped polymer fibers. *Lab Chip* **9**, 3126–3130 (2009).
16. D. Stoecklein, M. Davies, J. M. De Rutte, C.-Y. Wu, D. Di Carlo, B. Ganapathysubramanian, FlowSculpt: Software for efficient design of inertial flow sculpting devices. *Lab Chip* **19**, 3277–3291 (2019).
17. C.-Y. Wu, K. Owsley, D. Di Carlo, Rapid software-based design and optical transient liquid molding of microparticles. *Adv. Mater.* **27**, 7970–7978 (2015).
18. C.-Y. Wu, D. Stoecklein, A. Kommajosula, J. Lin, K. Owsley, B. Ganapathysubramanian, D. Di Carlo, Shaped 3D microcarriers for adherent cell culture and analysis. *Microsyst. Nanoeng.* **4**, 21 (2018).
19. C.-Y. Wu, M. Ouyang, B. Wang, J. de Rutte, A. Joo, M. Jacobs, K. Ha, A. L. Bertozzi, D. Di Carlo, Monodisperse drops templated by 3D-structured microparticles. *Sci. Adv.* **6**, eabb9023 (2020).
20. P. B. Howell Jr., J. P. Golden, L. R. Hilliard, J. S. Erickson, D. R. Mott, F. S. Ligler, Two simple and rugged designs for creating microfluidic sheath flow. *Lab Chip* **8**, 1097–1103 (2008).
21. D. R. Mott, P. B. Howell, K. S. Obenschain, E. S. Oran, The numerical toolbox: An approach for modeling and optimizing microfluidic components. *Mech. Res. Commun.* **36**, 104–109 (2009).
22. D. Stoecklein, C.-Y. Wu, K. Owsley, Y. Xie, D. Di Carlo, B. Ganapathysubramanian, Micropillar sequence designs for fundamental inertial flow transformations. *Lab Chip* **14**, 4197–4204 (2014).
23. D. R. Mott, J. P. B. Howell, J. P. Golden, C. R. Kaplan, F. S. Ligler, E. S. Oran, Toolbox for the design of optimized microfluidic components. *Lab Chip* **6**, 540–549 (2006).
24. D. Stoecklein, C.-Y. Wu, D. Kim, D. Di Carlo, B. Ganapathysubramanian, Optimization of micropillar sequences for fluid flow sculpting. *Phys. Fluids* **28**, 012003 (2016).
25. D. Stoecklein, M. Davies, N. Wubshet, J. Le, B. Ganapathysubramanian, Automated design for microfluid flow sculpting: Multiresolution approaches, efficient encoding, and CUDA implementation. *J. Fluids Eng.* **139**, 031402 (2017).
26. K. G. Lore, D. Stoecklein, M. Davies, B. Ganapathysubramanian, S. Sarkar, A deep learning framework for causal shape transformation. *Neural Netw.* **98**, 305–317 (2018).
27. Z. Yang, L. Nan, H. C. Shum, A versatile flow-profile engineering method in the stokes flow regime for complex-shaped flows. *Engineering* **7**, 655–662 (2021).
28. A. D. Stroock, S. K. Dertinger, A. Ajdari, I. Mezic, H. A. Stone, G. M. Whitesides, Chaotic mixer for microchannels. *Science* **295**, 647–651 (2002).
29. Z. Wang, Y. Xie, S. Ji, Global voxel transformer networks for augmented microscopy. *Nat. Mach. Intell.* **3**, 161–171 (2021).
30. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30**, 5998–6008 (2017).
31. O. Ronneberger, P. Fischer, T. Brox, in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: U-Net: Convolutional Networks for Biomedical Image Segmentation, Munich, Germany, October 5-9, 2015, Proceedings, Part III* (Springer International Publishing, 2015), vol. 9351, pp. 234–241.
32. Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, J. Liang, in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4* (Springer, 2018), pp. 3–11.
33. D. Qin, Y. Xia, G. M. Whitesides, Soft lithography for micro- and nanoscale patterning. *Nat. Protoc.* **5**, 491–502 (2010).
34. R. Hooper, C. Cummings, A. Beck, J. Vazquez-Armendariz, C. Rodriguez, D. Dean, Sheet-based extrusion bioprinting: A new multi-material paradigm providing mid-extrusion micropatterning control for microvascular applications. *Biofabrication* **16**, 025032 (2024).
35. Z. Wu, R. Liu, N. Shao, Y. Zhao, Developing 3D bioprinting for organs-on-chips. *Lab Chip* **25**, 1081–1096 (2025).
36. J. Gilbert-Honick, W. Grayson, Vascularized and innervated skeletal muscle tissue engineering. *Adv. Healthc. Mater.* **9**, e1900626 (2020).
37. H. Chen, Y. Fang, Z. Luo, Y. Wang, W. Sun, Y. Zhao, Emerging microfluidics for bioreactor. *Mater. Today* **86**, 356–392 (2025).
38. J. Huang, H. Chen, N. Li, L. Shang, Y. Bi, Y. Zhao, Harvesting stem cell exosomes from herringbone microfluidic bioreactor for wound healing. *Nano Today* **52**, 101983 (2023).
39. P. Zhang, H. Bachman, A. Ozcelik, T. J. Huang, Acoustic microfluidics. *Annu. Rev. Anal. Chem.* **13**, 17–43 (2020).
40. P. Su, C. Ren, Y. Fu, J. Guo, J. Guo, Q. Yuan, Magnetophoresis in microfluidic lab: Recent advance. *Sens. Actuators A: Phys.* **332**, 113180 (2021).
41. R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, "High resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2022), pp. 10684–10695.
42. J.-H. Bastek, D. M. Kochmann, Inverse design of nonlinear mechanical metamaterials via video denoising diffusion models. *Nat. Mach. Intell.* **5**, 1466–1475 (2023).
43. K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition* (IEEE, 2016), pp. 770–778.
44. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems* (Curran Associates Inc., 2019), vol. 32, pp. 8024–8035.
45. Z. Yang, zhyyang-hi/CeyeHao: Software for CeyeHao, version 1.0.0, Zenodo (2025); https://doi.org/10.5281/zenodo.15447573.
46. P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, K. He, Accurate, large minibatch sgd: Training imagenet in 1 hour. arXiv:1706.02677 (2017).
47. I. Loshchilov, F. Hutter, Sgdr: Stochastic gradient descent with warm restarts. arXiv:1608.03983 (2016).
48. R. Zhang, P. Isola, A. A. Efros, E. Shechtman, O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition* (IEEE, 2018), pp. 586–595.
49. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (2014).
50. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**, 211–252 (2015).
51. K. Deep, M. Thakur, A new crossover operator for real coded genetic algorithms. *Appl. Math Comput.* **188**, 895–911 (2007).

Acknowledgments

Funding: This work was supported by Research Grants Council Research Impact Fund, RIF 7003-21 (E.Y.L. and H.K.H.S.); Research Grants Council General Research Fund, nos. 17303123 (H.C.S.) and 17306820 (H.C.S.); Research Grants Council Collaborative Research Fund, C7165-20GF (H.C.S.); Research Grants Council Senior Research Fellow, SRFS2425-7S04 (H.C.S.); Croucher Foundation Croucher Senior Research Fellowship (H.C.S.); and Innovation and Technology Commission of the Hong Kong Special Administrative Region Government InnoHK initiative (Z.Y. and H.C.S.). **Author contributions:** Conceptualization: Z.Y., E.Y.L., H.K.H.S, and H.C.S. Methodology: Z.Y., Z.J., and H.L. Investigation: Z.Y. and Z.J. Visualization: Z.Y., H.L., X.F., and C.W. Supervision: E.Y.L., H.K.H.S., and H.C.S. Writing—original draft: Z.Y. Writing—review and editing: Z.Y., Z.J., H.L., E.Y.L., H.K.H.S., H.C.S., X.F., and C.W. **Competing interests:** H.C.S. is a scientific advisor of EN Technology Limited, MicroDiagnostics Limited, PharmaEase Tech Limited, and Upgrade Biopolymers Limited, in which he owns some equity, and also a founding director and co-director of the research center, namely, Advanced Biomedical Instrumentation Center Limited. The works in the paper are, however, not directly related to the works of these entities. The remaining authors declare that they have no competing interests. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are present in the paper and/or the Supplementary Materials and the following repositories. The code used to train the models and run the design tools are available at https://doi.org/10.5281/zenodo.15447573 and https://github.com/zhyyang-hi/CeyeHao. The training and validation dataset for all the models present here and the checkpoint for pretrained CEyeNet are available on a public online repository at https://doi.org/10.5281/zenodo.13363708.

Submitted 7 March 2025
Accepted 2 July 2025
Published 30 July 2025
10.1126/sciadv.adx2826