Fluid mixing is crucial in various industrial processes. In this study, focusing on the characteristics that reinforcement learning (RL) is suitable for global-in-time optimization, we propose utilizing RL for fluid mixing optimization of passive scalar fields. For the two-dimensional fluid mixing problem described by the advection–diffusion equations, a trained mixer realizes an exponentially fast mixing without any prior knowledge. The stretching and folding by the trained mixer around stagnation points are essential in the optimal mixing process. Furthermore, this study introduces a physically reasonable transfer learning method of the trained mixer: reusing a mixer trained at a certain Péclet number to the mixing problem at another Péclet number. Based on the optimization results of the laminar mixing, we discuss applications of the proposed method to industrial mixing problems, including turbulent mixing.

Fluid mixing plays a fundamental role in various industrial processes. However, most mixing processes are empirically designed by using trial-and-error methods through physical experiments, rather than mathematical optimization. Although turbulence is an “effective mixer” in, in some cases (e.g., a bioreactor or a mixer in food industry processes), turbulent mixing is not appropriate because strong shear flows damage the materials to be mixed. Moreover, sustaining turbulent flows in micro-mixers is difficult due to low Reynolds numbers; this necessitates enhanced mixing by laminar flows. Therefore, mixing optimization by laminar flows is crucial. Several analytical studies have evaluated the efficiency of laminar mixing protocols, e.g., proving the exponential bounds of the mixing speed; however, the research on constructive optimization methods remains limited.

This study proposes a mixing optimization based on reinforcement learning (RL) as a constructive method. To illustrate the effectiveness of the RL algorithm for fluid mixing optimization, we first summarize its mathematical framework. The RL algorithm is formulated in terms of the Markov decision process (MDP) \(M = (S, A, p_0, P, R)\), where \(S\) denotes the set of states, \(S = \{s_1, \ldots, s_s\}\); \(A\) denotes the set of actions, \(A = \{a_1, \ldots, a_A\}\); \(p_0\) denotes the probability distribution of the initial state, \(p_0 : S \rightarrow [0, 1]\); \(P\) denotes the transition probability, \(P : S \times S \times A \rightarrow [0, 1]\); and \(R\) denotes the reward function, \(R : S \times A \rightarrow \mathbb{R}\). The initial state, \(s_0\), is determined by \(p_0(\cdot)\), and in the next step, the state is determined by the transition probability, \(P(\cdot|s_0, a_0)\), which requires the action, \(a_0\). The action is determined by the policy, \(\pi : S \rightarrow A\), as \(a = \pi(s)\). The RL algorithm is implemented to determine the optimal policy, \(\pi^*\), for the given MDP, which maximizes the expectation of the cumulative reward, \(\sum_{t=0}^{\infty} \gamma^t R_{t+1}\). Here, \(\gamma \in (0, 1)\) denotes the discount factor and \(R_{t+1} := R(s_t, a_t)\).

The RL algorithm maximizes the cumulative reward (i.e., global-in-time) rather than the instantaneous reward, \(R_t\) (i.e., local-in-time). Therefore, it is suitable for global-in-time optimization problems. Designing efficient mixing protocols is one of the global-in-time optimization problems, as the final scalar field depends on the temporal order of actions in the entire mixing process, which includes stretching and folding by fluid flows and its coupling with molecular diffusion. An illustrative example was presented in History matters of Villermaux. Despite the effectiveness of the RL algorithms in solving a diverse range of problems in fluid mechanics, including nuclear fusion and turbulence modelling, the fluid mixing problem remains unexplored.

The RL algorithm is suitable for the global-in-time optimization problems, but not for problems with a high-dimensional state space in general, which is known as the curse of dimensionality. Particularly, the high dimensionality of the state space for fluid mixing renders the implementation of the RL algorithm challenging. This study investigates an optimization problem formulated by Mathew et al., in which the velocity field is given by the superposition of the prescribed fields. This reduces the dimension of the state space for the fluid motion to one; a single parameter, denoted by \(\theta\) later, determines the state of the fluid motion. This optimization problem was based on a physical experiment using the electromagnetically driven flow. The conjugate gradient descent method was introduced as a prototype of the fluid mixing optimization. To ensure that the RL algorithm can handle the flow field with a reduced degree of freedom, we focus on the same optimization problem.

1Graduate School of Engineering Science, Osaka University, Osaka 560-8531, Japan. 2Department of Applied Mathematics, Tokyo University of Science, Tokyo 162-8601, Japan. 3These authors contributed equally: Mikito Konishi and Susumu Goto. Email: inubushi@rs.tus.ac.jp
In this paper, we demonstrate for the first time that the RL algorithm is suitable for fluid mixing optimization. This algorithm identifies an effective flow control, which results in exponentially fast mixing without prior knowledge. The mechanisms behind efficient mixing are uncovered by focusing on the flow around the fixed points from the viewpoint of dynamical system theory\textsuperscript{15,16}. This study also proposes an effective transfer learning method for the trained mixer by considering the diffusion effect on mixing. Based on the optimization results of the laminar mixing, we discuss applications of the proposed method to industrial mixing problems, including turbulent mixing, in the “Conclusion and discussion” section.

Methods

Fluid mixing optimization. We consider the following optimization problem formulated by Mathew et al.\textsuperscript{2} as the benchmark problem, in which the velocity field, $u(x, y, t) = u_1(t)u_1(x, y) + u_2(t)u_2(x, y)$, is used. Here, $u_1(x, y) = (-\sin(2\pi x) \cos(2\pi y), \cos(2\pi x) \sin(2\pi y))$ and $u_2(x, y) = u_1(x - 0.25, y - 0.25)$ (see Fig. 1a). The time evolution of the passive scalar, $c(x, y, t)$, is described by the advection–diffusion equations on the two-dimensional torus, $\mathbb{T}^2$ (the periodic boundary condition):

\[
\partial_t c + u \cdot \nabla c = \frac{1}{\text{Pe}} \Delta c, \quad c(x, y, 0) = c_0(x, y), \quad (x, y, t) \in \mathbb{T}^2 \times [0, 1],
\]

where $\text{Pe} \in (0, \infty)$ represents the Péclet number. As a constraint on the flow control, the time integral of the kinetic energy, $\frac{1}{2} \int_0^1 \int_{\mathbb{T}^2} u^2 \, dx \, dt = \frac{1}{2} \int_0^1 a_i(t)R_{ij}(t) \, dt =: \delta'$, is fixed, where $R_{ij} := \frac{1}{2} \int_{\mathbb{T}^2} u_i \cdot u_j \, dx \ (i = 1, 2, j = 1, 2)$. We set $a(t) = 2\sqrt{\delta'} \ (\cos(\theta(t)), \sin(\theta(t)))$, by which the constraint is always satisfied. We also set $\delta' = 1.25$ as in Mathew et al.\textsuperscript{2}. In this problem, the velocity field, $u(x, y, t)$, is determined by a single parameter, $\theta(t)$, referred to as the flow parameter.

The variance of the scalar field is often used to measure the degree of mixedness. However, as it is a conserved quantity in the absence of diffusion (i.e., $\partial_t \int_{\mathbb{T}^2} \phi \, dx = 0 \ (\forall \phi \in \mathcal{H})$), it is unsuitable as a measure of the mixing process. Instead, we employ the mix-variance defined by $\Phi(c) = \|c\|_{H^{-1/2}}^2 := \sum_k \frac{1}{\sqrt{1+(2\pi|k|)^2}} |c_k|^2$, where $c_k$ denotes the Fourier coefficient of the scalar field\textsuperscript{17}. The mix-variance is equivalent to the Mix-Norm that was originally introduced to characterize the multiscale property of the mixed scalar field\textsuperscript{17}. Furthermore, Mathew et al.\textsuperscript{17} showed the equivalence between the decay of $\Phi(c)$, the weak convergence in $L^2$, and the mixing of ergodic dynamical systems in Theorem 3.2 (see also Lin et al.\textsuperscript{17} for the extension of the theorem). To summarize the optimization problem, we use the RL algorithm to determine the function, $\theta : [0, 1] \rightarrow \mathbb{R}$, that minimizes the mix-variance at the end of the mixing process, $\Phi(c, \cdot; t = 1)$.

We conduct a numerical simulation of the advection–diffusion equations (Eq. 1) using the fourth-order Runge–Kutta scheme for the temporal integration with $\Delta t = 0.001$ and the Fourier spectral method for spatial discretization with a grid of 250 $\times$ 250, which is the same as the one used in Mathew et al.\textsuperscript{2}.

Optimization of mixing with RL. Here, we consider the optimization of the action-value function (Q function) $Q^\pi(s, a) := \mathbb{E}[\sum_{t=0}^\infty \gamma^t R_{t+1}|S_0 = s, A_0 = a]$ instead of the policy $\pi$, and obtain the optimal Q-func-
tion, $Q^*: S \times A \rightarrow \mathbb{R}$. The Banach fixed-point theorem mathematically ensures that such an optimal $Q$ function exists as a fixed point of the Bellman operator \(^6\). We obtain the optimal policy as $\pi^*(s) = \text{argmax}_{a \in A} Q^*(s, a)$.

As a standard implementation of the RL algorithm, we employ the deep Q network \(^{18}\), which approximates the $Q$ function by using the deep neural network denoted by $Q^* : \mathbb{R}^{N_s} \times A \rightarrow \mathbb{R}$. Here, $N_s$ and $w$ denote the dimension of the state space and the connection weights in the neural network, respectively. The inputs to the network are the scalar field, $(x, y, t)$, and the velocity field, $u(x, y, t)$. The values of these fields on $\mathbb{T}^2$ are observed on the square grid $83 \times 83$, and the state, $s_i$, of the MDP is defined as the observed values of the velocity field, $u(x_i, y_i, t)$ ($i = 1, \ldots, N_0$), where $N_0 = 83 \times 83$, and those of the scalar fields over the last five steps; that is, $s_i := (u(x_i, y_i, t), [c(x_i, y_i, \tau)]_{\tau = t, t-\Delta t_0, \ldots, t-4\Delta t_0}) \in \mathbb{R}^{N_s}$, and $\Delta t_0$ denotes the time interval of the successive observations. Therefore, the dimension of the state space is $N_s = 7 \times N_0$. The network consists of four hidden layers, and each activation function is ReLU as Mnih et al. \(^{18}\). The discount factor is $\gamma = 0.99$. The more details of the deep Q network structure and its implementations are described in the "Supplementary Information". The initial distribution, $p_0$, is given by the delta function such that $\theta(0) = 0$ and $c(x,y,0) = \sin(2\pi y)$.

The time interval of the successive observations is $\Delta t_0 = 0.004$, which is the same value used in the benchmark problem \(^2\), and $\Delta t_0 = 5 \Delta t_0$, where $\Delta t_0$ denotes the time interval of the successive updates of the Q function. Hence, for each period of $\Delta t_0$, the RL algorithm observes the scalar fields determined by the advection–diffusion equations (Eq. 1) with the fixed velocity field. Then, the Q function, i.e., the weights in the neural network, is updated. A single unit of episode corresponds to a single mixing process, i.e., solving the initial value problem of the advection–diffusion equations (Eq. 1) for $0 \leq t \leq 1$. The total number, $N_e$, of episodes for the training is $N_e = 4000$. The results with the larger number of episodes, $N_e = 5000$, are qualitatively identical to those with $N_e = 4000$.

As the action, $A$, of the MDP, the RL algorithm can change the value of the flow parameter, $\theta(t)$ ($0 \leq t \leq 1$). The velocity field, $u(x, y, t)$, is determined by the single-parameter $\theta(t)$, and the flow control is realized by changing $\theta(t)$. The discretization of the temporal change of the flow parameter is $\theta(t + \Delta t_0) = \theta(t) + \omega \Delta t_0$ with $\omega \in [0, \omega_0, \ldots, \{=\ A\}, \omega_0 = \pi/(4 \Delta t_0)$ and $\omega = -\pi/(4 \Delta t_0)$. The action, $\omega$, is selected following the $\epsilon$-greedy method \(^{6,18}\), which changes the value of $\epsilon$ linearly from 1 to 0.001.

The reward function, $R$, is defined by using the mix-variance, $\Phi$, which is set to be a monotonically decreasing function of $\Phi$ to ensure that the smaller value of $\Phi$ represents a better mixed scalar field:

$$R(\Phi) = \frac{\Phi - \Phi_0}{\xi}, \quad \xi = \begin{cases} \Phi_0 - \Phi & (\Phi \geq \Phi_0), \\ \Phi - \Phi_T & (\Phi < \Phi_0), \end{cases}$$

where $\Phi, \Phi_0, \Phi_T$ denote a threshold, an initial value, and a target value of the mix-variance, respectively. By definition, $R = -1$ initially, and $R = +1$ if the mix-variance, $\Phi$, attains the target value. The values of $\Phi$ and $\Phi_T$ are set based on the Péclet number: $(\Phi, \Phi_T) = (1 \times 10^{-3}, 4 \times 10^{-3})$ for $Pe = \infty$ and $(\Phi, \Phi_T) = (5 \times 10^{-3}, 1 \times 10^{-4})$ for $Pe = 100$.

Results

RL-based optimization. The optimization results are presented in the absence of diffusion ($Pe = \infty$). The optimal policy, $\pi^* : \mathbb{R}^{N_s} \rightarrow A$, approximated by the deep Q network, is obtained from the RL-based optimization. Thereafter, the state vector, $s_i \in \mathbb{R}^{N_s}$, determines the optimal action through $\theta_0 = \pi^*(s_i)$. This determines the velocity field during the next interval, $\Delta t_0$, which advects the scalar field, and the process carries on to the next observation. This flow controller based on the optimal policy, $\pi^*$, is referred to as a trained mixer. Figure 1b shows the left to right panels that the trained mixer makes the scalar field, $c(x, t)$, evolve in time. Here, the black and white colors correspond to the high and low values of the scalar field, respectively. The trained mixer produces a complex layered structure of the scalar field. The following subsection presents a detailed description of the successive stretching and folding of the interface between the two colors.

The mix-variance, $\Phi_n(t)$ ($n = 1, \ldots, 4000$), is shown in Fig. 1c. During the initial stage of the training, (i.e., in the first half of the total episodes such as $n = 1, 800$, and 1600), the RL algorithm with the $\epsilon$-greedy method chooses actions randomly. Although this “random mixer” can decrease the mix-variance, such mixing is inefficient, as explained below.

Mathew et al. \(^2\) reported that the proposed conjugate gradient descent method resulted in a $\Phi(t = 1) \simeq 6 \times 10^{-3}$; this value of the mix-variance is used for the comparison as a reference. In the first half of the total episodes, the mix-variance at the end of the mixing process, $\Phi_n(t = 1)$, is larger than the reference value; that is, the insufficient training of the mixer results in inefficient mixing. Conversely, $\Phi_n(t = 1)$ is reduced in the latter half of the total episodes, $n = 2400, 3200$, and 4000. Particularly, $3 \times 10^{-3} < \Phi_n(t = 1) < 4 \times 10^{-3}$ for $n = 4000$, which are almost identical to (slightly smaller than) the reference value. Interestingly, the mix-variance decreases exponentially fast for $0.3 \leq t \leq 1$ for the latter episodes such as $n = 3200$ and $n = 4000$. While here we focus on the qualitative comparison using the mix-variance, there are some qualitative differences between the method by Mathew et al. \(^2\) and our RL-based method. In the "Conclusion and discussion" section, we illustrate significant advantages of the RL-based method.

Figure 1d presents the mix-variance at the end of each mixing process, $\Phi_n(t = 1)$, which fluctuates due to the $\epsilon$-greedy methods and the fact that the policy, $Q^*$, is not converged. However, the fluctuation decreases as the episode progresses; see also Figs. S1 and S2 in the "Supplementary Information". The RL algorithm significantly decreases the mix-variance, $\Phi_n(t = 1)$; that is, the RL-based optimization effectively enhances the mixing.

Characteristics of the trained mixer. The flow parameter in episode $n$ is denoted by $\theta_0(n)$. In the first half of the training, $n < 2000$, the flow parameter, $\theta_0(n)$, evolves randomly in time due to the $\epsilon$-greedy methods.
and the fact that the policy is not converged. However, as the episode progresses, $\theta_n(t)$ converges to a single function, $\theta^*(t)$, except for the final stage of the process, as shown in Fig. 2a. The time series of $\theta_n(t)$ consists of square waves, as the velocity field (i.e., $\theta_n(t)$) is fixed in each interval, $\Delta t/\pi$. The optimal mixing process by the trained mixer corresponding to $\theta^*(t)$ is divided into the following three stages:

1. Initial stage ($0 < t \leq 0.3$): the flow parameter is a constant; $\theta^*(t) = \pi/4$, indicating the steady flow, $u(x, y) = -c \sin 2\pi(x + y)$, $v(x, y) = c \sin 2\pi(x + y)$ ($c$ : const.). The velocity vector is parallel to the diagonal line; for example, the flow along the line, $x + y = 1/4$, traverses the domain, $\Pi^2$, with the velocity vector $(u, v) = (-c, c)$.

2. Middle stage ($0.3 < t \leq 0.7$): the flow parameter changes linearly; $\theta^*(t) = \omega^* t$ ($\omega^* \simeq 16$), indicating the temporally periodic flow with a constant angular frequency.

3. Final stage ($0.7 < t \leq 1$): there are no common characteristics of the time evolution of the flow parameter.

The different time evolutions of the flow parameter, $\theta(t)$, in the final stage result in the almost identical value of the mix-variances, $\Phi(t = 1)$, at the end of the process. Therefore, the essential process in the mixing is not the final stage but the initial and middle stages. Numerical experiments supporting this point are presented in the next subsection.

Figure 2b presents the time evolution of the velocity fields corresponding to the flow parameter obtained in the final episode, i.e., $\theta_n(t)$ ($n = 4000$). The blue line in each panel indicates the material line advected by the flow, which is initially placed along with the line $y = 0.5$. The upper panels in Fig. 2b depict the flow in the initial stage, where the material line is elongated to have the diagonal length of the domain. Subsequently, the
flow is temporally periodic in the middle stages, which are shown in the lower panels in Fig. 2b. Each velocity field has eight fixed (stagnation) points, \( n_1 \) and \( n_2 \). Half of them are elliptic; that is, the Jacobian matrix has purely imaginary eigenvalues. The other half are saddle points; that is, the Jacobian matrix has real eigenvalues. We focus on them at \( (x, y) = (0.5, 0.5) \), which is depicted by the red point in each panel of Fig. 2b as a reference. The material line around the fixed point is stretched along the unstable eigendirections when the fixed point is a saddle, whereas it is folded (approximately \( \pi/2 \) rotation) when the fixed point is elliptic. The local stretching and folding around the eight fixed points occur simultaneously, resulting in efficient mixing. The use of the specific protocol by the trained mixer with the constant angular frequency, \( \theta(t) = \omega^* t \), is explained in the section of “Conclusion and discussion” section.

Remarkably, the period of the flow in the middle stage, \( 2\pi/\omega^* \), which determines the period of the successive switching of the saddle and elliptic types of the fixed points, is optimal in the following sense. Apart from the RL algorithm, we conduct numerical simulations of the scalar field advected by the flow determined by \( \theta(t) = \omega t \) with a constant angular frequency, \( \omega \), throughout the mixing process, \( 0 \leq t \leq 1 \). The inset of Fig. 2a shows \( \Phi(t = 1) \) evaluated for \( \omega \in [0, 30] \). The minimum of \( \Phi(t = 1) \) in this setting is obtained at \( \omega \simeq \omega^* \). This implies that the RL algorithm determines the optimal angular frequency, \( \omega^* \), without any prior knowledge, and the trained mixer uses the temporally periodic flow with the optimal period in the middle stage of the process.

Comparison with randomized mixers. To characterize the flow by the trained mixer in the initial and middle stages, we introduce three different mixing processes, called randomized mixers:

- Completely randomized mixer: It uses the random controller that takes one of the three actions, \( \omega \in \{ 0, \omega_+, \omega_- \} \), independently, with the same probabilities for all the stages (0 ≤ \( t \) ≤ 1).
- Partially randomized mixer I: It uses the trained mixer for the initial stage (0 ≤ \( t \) < 0.3), and then switches to use the random controller for 0.3 ≤ \( t \) ≤ 1.
- Partially randomized mixer II: It uses the trained mixer for the initial and middle stages (0 ≤ \( t \) < 0.7), and then switches to use the random controller for 0.7 ≤ \( t \) ≤ 1.

Numerical simulations are conducted 200 times independently for each control. Fig. 2c presents the probability density functions (PDFs) of the mix-variance, \( \Phi(t = 1) \), at the end of the mixing process. The gray solid line indicates the value of the mix-variance by the trained mixer, \( \Phi_{tr}(t = 1) \) (\( n = 4000 \)) (see Fig. S1 in the “Supplementary Information” for the related PDF of the trained mixer).

The top panel of Fig. 2c depicts the PDF in the case of the completely randomized mixer, where the mix-variances are larger than the reference value of the trained mixer. The left and right panels of Fig. 2d represent the final state of the scalar field, \( e(x, t = 1) \), produced by the trained mixer and one completely randomized mixer that presents the mix-variance, \( \Phi(t = 1) \), close to the median value of the PDF. Videos 1 and 2 in the “Supplementary Information” correspond to the scalar fields mixed by the trained mixer and the completely randomized mixer, respectively. Large unmixed blobs remain in the scalar field produced by the completely randomized mixer. That is, the training mixer with the RL algorithm is effective. The second panel of Fig. 2c depicts the PDF in the case of the partially randomized mixer I, which is more effective than the completely randomized mixer. However, a substantial gap exists between the results of the partially randomized mixer I and that of the trained mixer. This indicates that the mixing process during the middle stage is also crucial. Finally, the third panel of Fig. 2c depicts the PDF produced by the partially randomized mixer II. The results are almost identical to those obtained using the trained mixer. Therefore, the effectiveness of the partially randomized mixer II is the same as that of the trained mixer. These observations demonstrate that the mixing process during the initial and middle stages is essential for the mixing efficiency, whereas the mixing process during the final stage is not.

Transferring the trained mixer. This subsection considers the diffusion effect on the RL-optimization of the mixing described by the advection–diffusion equations (Eq. 1) with finite Péclet numbers. The details of the problem settings are identical to those in the previous sections, except for the values of the Péclet numbers. The RL-based optimization is applied to the mixing problem for the case of \( Pe = 10^5 \), 10^4, and 10^3, which are as effective as for the case of \( Pe = \infty \), regardless of the Péclet numbers. For instance, at \( Pe = 100 \), the mix-variance, \( \Phi_n(t) \), decreases faster for the later episodes, as shown in the inset of Fig. 3b, where \( n = 1, 600, 1200, 1800, 2400, \) and 3000 and lighter (thicker) curves correspond to larger \( n \). We note that the curves of \( \Phi_n(t) \) for \( n \geq 1200 \) are almost the same, implying that the RL algorithm converges to find the optimal policy at \( n = 1200 \). Interestingly, this convergence is faster than the case of \( Pe = \infty \) (Fig. 1c). The number of episodes required for convergence is \( n \simeq 3000 \) at \( Pe = \infty \); however, \( n \geq 1200 \) seems to be sufficient for convergence around \( Pe = 100 \).

The diffusion effect appears in the flow controls at the later stages. If the mixer successfully generates fine layered structures in an early stage, the flow control becomes less important in the later stages of mixing due to the diffusion effect. In other words, at a low Péclet number, once the RL algorithm finds the optimal mixing control in an early stage of mixing, nothing is to be learned as the diffusion reduces the mix-variance rapidly, irrespective of the control by the mixer. This may result in the faster convergence observed above. The implications of the fast convergence at the low Péclet numbers to the training mixer are given in the “Conclusion and discussion” section.

This diffusion effect implies the asymmetric transferability of a trained mixer; that is, a mixer trained at a high Péclet number can be used for mixing at a lower Péclet number, whereas the converse does not hold true. Let \( Pe_T \) be the Péclet number where the mixer is trained, and the asymmetric transferability is then rephrased as follows: the trained mixer can be reused for the same mixing process for the range of \( (0, Pe_T) \). Figure 3a presents the mix-variance, \( \Phi(t) \), for 0 ≤ \( t \) ≤ 1 at \( Pe = \infty \). The thick blue lines indicate the results for the case of \( Pe_T = \infty \),
and the thin red lines indicate the results for the case of $\text{Pe}_T = 100$. In Fig. 3a, b, the solid, dashed, and dashed-dotted lines indicate the results with different random numbers for learning. The mixers that trained at $\text{Pe}_T = \infty$ realize the exponentially fast mixing for the entire process when we use it for $\text{Pe} = \infty$. On the other hand, the mixers that trained at $\text{Pe}_T = 100$ realize the exponentially fast mixing only for the first half of the process, but fail to mix for the latter half.

Fig. 3b presents the mix-variance, $\Phi(t)$, for $0 \leq t \leq 1$ at $\text{Pe} = 100$. Similar to Fig. 3a, the thick blue lines represent the results for the case of $\text{Pe}_T = \infty$, and the thin red lines represent the results for the case of $\text{Pe}_T = 100$. Unlike the case of $\text{Pe} = \infty$, no significant difference exists between the results for the cases of $\text{Pe}_T = 100$ and $\text{Pe}_T = \infty$, and both cases realize the exponentially fast mixing. In summary, the mixers of $\text{Pe}_T = \infty$ can be used for the mixing at $\text{Pe} = 100$, whereas the converse does not hold true. Therefore, a mixer trained at a higher Péclet number can be used for the mixing process for a broader range of $\text{Pe}$.

Conclusion and discussion

By illustrating why the RL algorithm is suitable for fluid mixing optimization, we demonstrated, as a proof-of-concept, that the mixer trained by using the RL algorithm is effective for the two-dimensional fluid mixing problem (Fig. 1), which paves the way for the development of RL-based training of mixers. The proposed method was quantitatively evaluated by focusing on the benchmark problem of the mixing optimization studied in the pioneer work. In addition to the comparison of mix-variance values, we note that our RL-based method solves the optimization problem in more restrictive conditions compared to the method proposed by Mathew et al. For example, in our setting, the number of the states of the velocity field is restricted to eight, $\theta = 0, \pi/4, \ldots, 7\pi/4$. Furthermore, the proposed method is more flexible; that is, it uses only the scalar and velocity field as the input to the neural network. Provided that these fields can be observed, physical implementations are possible in principle, even if the evolution equations of these fields are unknown. For instance, mixing problems of granular or viscoelastic fluids are essential; however, the evolution equation of such a complex material has not necessarily been established, and therefore, the conjugate gradient descent method cannot be applied to these industrially fundamental problems. On the other hand, the RL-based method is equation-free, hence it is applicable if the sensory data of the mixture states are available as the input to the neural network.

The optimized mixing process was divided into three distinct stages. It is particularly interesting to note that, in the middle stage, the optimized flow is temporally periodic with the constant angular frequency. Here, we discuss why the RL algorithm makes the angular frequency constant. The fixed points in both velocity fields, $u_1$ and $u_2$, are located at the same position and are homogeneously placed in the domain, $\mathbb{T}^2$. If the angular frequency is not constant, the switching period between the saddle and elliptic types of the fixed point may differ at each location. This spatial difference makes the scalar field inhomogeneous. The inhomogeneity increases the amplitude of the Fourier coefficient of the small wavenumber, thereby increasing the mix-variance. Consequently, the time variation of the angular frequency results in the larger value of the mix-variance. The RL algorithm employs the constant angular frequency to avoid this undesired effect. The detailed justification of the aforementioned interpretation is one of the future works.

Another related future work is to understand the optimal mixing in more detail. For instance, we claim that the random variation of the flow parameter in the final stage ($t > 0.7$) is not essential for optimal mixing, in the sense that the results of the partially randomized mixer II (Fig. 2c) and the trained mixer (Fig. S1 in the “Supplementary Information”) are almost identical. However, there is a small difference between these PDFs, which suggests that the randomization of the actions in the final stage may eliminate some actions, which the RL algorithm regards to be essential, in the optimized mixing process.

For practical application, reduction of learning costs is crucial. Despite the effectiveness of transfer learning in reducing learning cost, its application to the problems in fluid mechanics remains limited. In this regard,
this study has introduced the physically-reasonable notion of the asymmetric transferability of the trained mixer. The demonstration in this study (Fig. 3) indicates that, in terms of transfer learning, the Péclet number of the source domain \( P_e \) should be as high as possible, if the trained mixer is required to reuse for the broader range. If the mixer is trained at a high Péclet number, it can learn how to mix the scalar field to create the fine striped structures. If the trained mixer is transferred to a lower Péclet number, it makes the fine structures at the beginning of the mixing process. Then, smoothing such structures by diffusion reduces the mix-variance, irrespective of the actions of the trained mixer in the later stage. Therefore, the transfer of the trained mixer from a high Péclet number to a lower one is effective.

Concerning another aspect of learning costs, we have found that the learning of mixing at a lower Péclet number converges faster (Inset of Fig. 3b). Therefore, if fast learning at a Péclet number is required, the Péclet number of the source domain \( P_e \) should be as low as possible. Considering together with the discussion in the previous paragraph, the above discussions suggest a trade-off between broad transferability and fast learning; in other words, there is an optimal Péclet number of the source domain that balances these two advantages in each application. Although this study is restricted to the transfer of the trained mixer over the different Péclet numbers, future developments of transfer learning methods of trained mixers may be significant.

Large gaps exist between the mathematical toy problem discussed in this study and the existing mixing problems in the industrial processes. However, the results of this study indicate some directions overcoming these gaps. First, we discuss the implications of this study to turbulent mixing. Turbulence comprises multi-scale counter-rotating pairs of coherent vortices\(^1\), and strong turbulent mixing stems from the effective mixing around such pairs of vortices at each scale\(^1\). As observed in the transfer learning method, scalar mixing occurs from these gaps. First, we discuss the implications of this study to turbulent mixing. Turbulence comprises multi-scale problems in the industrial processes. However, the results of this study indicate some directions overcoming numbers, future developments of transfer learning methods of trained mixers may be significant.

In addition, in industry, the multiphase and/or thermal flows with chemical reactions may have to be considered, which increases the complexity of the flow dynamics. In such cases, embedding prior knowledge, such as the evolution equations or some physical constraints into the RL-based optimization may be effective, as discussed in Brunton\(^1\). As another future task for the RL-based optimization in industrial mixing problems, it will be important to study the robustness of the mixing control with the obtained policy with respect to changes in the initial scalar field. Furthermore, whereas the deep Q network is employed as the first step in this study, a more specific and state-of-the-art implementation of the RL algorithm would be necessary for such complex flows. Extending the proposed method to incorporate knowledge on fluid mechanics and suitable RL implementation techniques can further enhance mixing even in industrial processes with laminar and turbulent flows.

**Data availability**
The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Received: 25 March 2022; Accepted: 4 August 2022
Published online: 22 August 2022

**References**

1. Goto, S. & Kida, S. Reynolds-number dependence of line and surface stretching in turbulence: Folding effects. *J. Fluid Mech.* **586**, 59–81 (2007).
2. Mathew, G., Mezic, I., Grivopoulos, S., Vaidya, U. & Pettold, L. Optimal control of mixing in stokes fluid flows. *J. Fluid Mech.* **580**, 261–281 (2007).
3. Lin, Z., Thiffeault, J.-L. & Doering, C. R. Optimal stirring strategies for passive scalar mixing. *J. Fluid Mech.* **675**, 465–476 (2011).
4. Seis, C. Maximal mixing by incompressible fluid flows. *Nonlinearity* **26**, 3279–3289 (2013).
5. Alberti, G., Crippa, G. & Mazzucato, A. L. Exponential self-similar mixing by incompressible flows. *J. Am. Math. Soc.* **32**, 445–490 (2019).
6. Szepesvári, C. *Algorithms for Reinforcement Learning*. (Synthesis Lectures on Artificial Intelligence and Machine Learning Morgan and Claypool Publishers, 2010)
7. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction* (MIT Press, 2018).
8. Villenmaux, E. Mixing versus stirring. *Ann. Rev. Fluid Mech.* **51**, 245–273 (2019).
9. Garnier, P. et al. A review on deep reinforcement learning for fluid mechanics. *Comput. Fluids* **225**, 104973–104996 (2021).
10. Brunton, S. L., Noack, B. R. & Koumoutsakos, P. Machine learning for fluid mechanics. *Ann. Rev. Fluid Mech.* **52**, 477–508 (2020).
11. Brunton, S. L. Applying machine learning to study fluid mechanics. *Acta Mechanica Sinica* **37**, 1718–1726. https://doi.org/10.1007/s10409-021-01143-6 (2021).
12. Degrave, J. et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature* **602**, 414–419 (2022).
13. Novati, G., de Laroussilhe, H. L. & Koumoutsakos, P. Automating turbulence modelling by multi-agent reinforcement learning. *Nat. Mach. Intell.* **3**, 87–96 (2021).
14. Rothstein, D., Henry, E. & Gollub, J. Persistent patterns in transient chaotic fluid mixing. *Nature* **401**, 770–772 (1999).
15. Wiggins, S. & Ottino, J. M. Foundations of chaotic mixing. *Philos. Trans. R. Soc. Lond. Series A Math. Phys. Eng. Sci.* **362**, 937–970 (2004).
16. Aref, H. et al. Frontiers of chaotic advection. *Rev. Modern Phys.* **89**, 025007 (2017).
17. Mathew, G., Mezic, I. & Pettold, L. A multiscale measure for mixing. *Physica D Nonlinear Phenomena* **211**, 23–46 (2005).
18. Mihc, V. et al. Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
19. Inubushi, M. & Goto, S. Transfer learning for nonlinear dynamics and its application to fluid turbulence. *Phys. Rev. E.* **102**, 043301(R) (2020).
20. Goto, S., Saito, Y. & Kawahara, G. Hierarchy of antiparallel vortex tubes in spatially periodic turbulence at high Reynolds numbers. *Phys. Rev. Fluids* **2**, 064603 (2017).
Acknowledgements
This work was partially supported by JSPS Grant-in-Aid for Early-Career Scientists No. 19K14591 and JSPS Grants-in-Aid for Scientific Research Nos. 19KK0067, 20H02068, 20K20973, and 22K03420.

Author contributions
M.K. and M.I. conceived and conducted the numerical experiments; M.K., M.I, and S.G. analyzed the results. All authors reviewed the manuscript.

Competing interests
The authors declare no competing interests.

Additional information
Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-022-18037-7.

Correspondence and requests for materials should be addressed to M.I.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

© The Author(s) 2022