Physics of Fluids

RESEARCH ARTICLE | JANUARY 27 2025

Physics-informed reinforcement learning framework for kinematics optimization of self-propelled articulated swimmers

Chengyun Wang (汪乘雲) ; Haibo Huang (黄海波) ✉

Physics of Fluids 37, 011919 (2025)
https://doi.org/10.1063/5.0249580

AIP Publishing

Physics-informed reinforcement learning framework for kinematics optimization of self-propelled articulated swimmers

Cite as: Phys. Fluids **37**, 011919 (2025); doi: 10.1063/5.0249580
Submitted: 19 November 2024 · Accepted: 2 January 2025 ·
Published Online: 27 January 2025

View Online Export Citation CrossMark

Chengyun Wang (汪乘雲) and Haibo Huang (黄海波)[a)]

AFFILIATIONS

Department of Modern Mechanics, University of Science and Technology of China, Hefei, Anhui 230026, China

[a)]Author to whom correspondence should be addressed: huanghb@ustc.edu.cn

ABSTRACT

Our study presents a novel optimization framework dedicated to refining the swimming gaits of self-propelled articulated swimmers. The approach integrates a fluid–structure interaction solver for multibody systems with a single-step deep reinforcement learning optimization algorithm. To overcome the computational costs incurred by evaluations during parameter search, we introduced controlled transfer learning to improve performance and efficiency. By leveraging pre-trained policies on low-fidelity models and adapting them to high-fidelity environments, the learning procedure can be accelerated with significantly less high-fidelity evaluations. Moreover, the optimization algorithm is complemented by an intricate mapping procedure designed to enforce stringent constraints derived from prior knowledge within the expansive high-dimensional design space. Then, this framework is applied to investigate the influence of segment length and number on the optimal swimming kinematics of an articulated fish model. Findings reveal that the variable-length approach may yield more parsimonious yet comparable models with fewer segments compared to the equal-length approach. This study contributes valuable insight into the design and behavior of both natural and robotic swimmers, paving the way for future advancements in optimization algorithms and fish body models.

I. INTRODUCTION

Fish have undergone millions of years of natural selection, honing their shapes and movements to thrive in aquatic environments. The physical and biological mechanisms governing fish locomotion offer valuable insight for engineering applications,[1] hydrodynamically efficient designs,[2] autonomous underwater vehicles,[3,4] and energy-harnessing technologies.[5] The remarkable maneuverability and propulsion efficiency demonstrated by bio-inspired underwater robots have ignited theoretical research for over half a century. Lighthill's slender body theory (SBT)[6] reignited interest in unraveling the physical principles underlying fish locomotion. Ongoing refinements to Lighthill's slender fish model[7–9] have provided researchers with profound insight. However, direct application of these models in optimizing and controlling underwater robots still faces challenges due to the lack of high physical precision and sufficient datasets, hindering the training of intelligent behaviors in real environments that mimic or even surpass the capabilities of these bionic creatures in accomplishing complex tasks.

Alternatively, numerical simulation stands out as a valuable and essential tool for understanding optimal fish behaviors in diverse scenarios. Two commonly used simplified models for simulating fish locomotion are the continuum model and the articulated model. In the continuum model, fish motion is generated by providing an analytical time-varying expression for the lateral displacement of the midline. While the articulated model of an undulating fish is simplified to a multi-rigid-body system, which is linked by hinges with either prescribed or passive motions. In this study, we primarily concentrate on the articulated model, and specific justifications will be provided in Sec. II B Solid Solver. It is noteworthy that existing simulators still suffer from significant limitations, such as high inaccuracy,[10] specialization to specific agents or environments,[11,12] and prohibitively expensive costs for generating training data.[13] Presently, there is a shortage of simulation platforms capable of delivering modular, efficient, and precise results suitable for optimizing control policies for underwater robots.

In recent years, gradient-free optimization algorithms have exhibited remarkable efficacy in identifying optimal parameters for biological functions.[2,14–17] Deep reinforcement learning (DRL), a typical gradient-free algorithm, excels in scenarios characterized by multiple local minima or environmental uncertainty, leveraging unsupervised learning through iterative trials (episodes). Viquerat *et al.*[18] introduced the concept of single-step DRL, applying it to the direct shape optimization of airfoils, while Ghraieb *et al.*[19] extended its application to various open-loop control problems featuring expansive parameter spaces. Our prior work[20] further investigated and improved this approach by addressing the optimization problem of a splitter plate downstream in the cylinder wake. However, to increase the chance of acquiring the global optimum from the entire search space, DRL-based optimizers usually operate at the expense of a costly trial-and-error process.

Although the accumulated engineering experience and physical knowledge are beneficial in appropriate selection of initial configurations (baseline designs), it is difficult to intelligently guide the search of globally optimal solution by effectively embedding these priori knowledge into the gradient-free optimization algorithms. However, the recent developments in the DRL domain, especially in transfer learning (TL),[21] create the possibilities for the optimizer to imprint physical knowledge. Yan *et al.*[22] integrates DRL and TL for efficient aerodynamic shape optimization in missile control surfaces, where the optimization experience is extracted from a semi-empirical model. Bhola *et al.*[23] proposed a multi-fidelity framework using controlled transfer learning (CTL) for efficient airfoil shape optimization in high Reynolds numbers. They primarily focused on the generalization across multi-fidelity environments without explicating the underlying working mechanism. Anyway, such multi-model training scheme, which can also be referred to as physics-informed reinforcement learning, involves enhancing the learning process by incorporating physical information extracted from the surrogate model.

Therefore, we aim to develop a physics-informed reinforcement learning optimization framework integrated with an in-house fluid–structure interaction (FSI) solver. The fluid solver is based on the lattice Boltzmann method (LBM), with rigid bodies modeled using the immersed boundary method (IBM). The dynamics of multibody systems are addressed using rigid multibody dynamics[24] due to its compatibility and generalization. This framework aims to optimize the kinematic gaits of a self-propelled articulated swimmer while adhering to specific goals. To expedite policy convergence, we introduce two techniques that significantly boost the performance of our framework: (1) We extend the CTL to the swimming problem, rather than the extensively researched shape optimization problem; (2) For DRL-based algorithms originally suitable for box-constrained optimization, we add a mapping network layer to enforce hard linear inequality constraints based on the known relationships between the parameters. This adaptation helps prevent non-compliant evaluations, reducing redundant efforts and resource wastage.

We demonstrate the effectiveness of our framework by discovering the optimal joint motion that achieves a desired locomotion goal (i.e., joint motion leading to the fastest or most energy-efficient locomotion) for an articulated swimmer. Our results are compared with a prior experimental study, where segmented fish models are derived from actual fish data.[25] We elucidate how segment length and number influence actual performance, aspects not directly addressed by experimental data. The paper is organized as follows: In Sec. II, we provide detailed information on the numerical methods, along with the optimization algorithm. Section III introduces the problem formulation and algorithm setup. Section IV presents the optimized results and includes rational mechanism explanation. Finally, the paper is concluded in Sec. V.

II. METHODOLOGY

A. Fluid solver

The Navier–Stokes equations governing incompressible viscous flow with solid bodies can be expressed as follows:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\nabla p + \frac{1}{Re} \nabla^2 \boldsymbol{u} + \boldsymbol{F}, \quad \nabla \cdot \boldsymbol{u} = 0, \tag{1}$$

where $\boldsymbol{u}$ represents the velocity, p is the pressure, and $\boldsymbol{F}$ denotes the Eulerian momentum force on the surrounding fluid, subject to the no-slip boundary condition. According to lattice gas automata theory, Eq. (1) can be written in the form of discrete lattice Boltzmann equations:

$$f_\alpha(\boldsymbol{x} + \boldsymbol{e}_\alpha \delta t, t + \delta t) - f_\alpha(\boldsymbol{x}, t) = -\frac{1}{\tau}\left[f_\alpha(\boldsymbol{x}, t) - f_\alpha^{eq}(\boldsymbol{x}, t)\right] + \delta t F_\alpha, \tag{2}$$

where τ is the non-dimensional relaxation time related to the fluid viscosity and $f_\alpha(\boldsymbol{x}, t)$ is the distribution function associated with the α th discrete velocity $\boldsymbol{e}_\alpha$, where $\alpha = 0, \ldots, 8$ for D2Q9 velocity model. The equilibrium distribution function f_α^{eq} and the forcing term F_α are calculated as[26]

$$\begin{aligned} f_\alpha^{eq} &= \rho\omega_\alpha\left[1 + \frac{\boldsymbol{e}_\alpha \cdot \boldsymbol{u}}{c_s^2} + \frac{(\boldsymbol{e}_\alpha \cdot \boldsymbol{u})^2}{2c_s^4} - \frac{\boldsymbol{u}^2}{2c_s^2}\right], \\ F_\alpha &= \left(1 - \frac{1}{2\tau}\right)\omega_\alpha\left[\frac{\boldsymbol{e}_\alpha - \boldsymbol{u}}{c_s^2} + \frac{(\boldsymbol{e}_\alpha \cdot \boldsymbol{u})}{c_s^4}\boldsymbol{e}_\alpha\right] \cdot \boldsymbol{F}, \end{aligned} \tag{3}$$

where ω_α is the weighing factor and $c_s = \frac{c}{\sqrt{3}}$ is the lattice sound speed. The lattice speed c is given by $c = \frac{\Delta x}{\Delta t}$, where Δx is the lattice size, and Δt is the time step. The macrovariables such as density, velocity, and pressure can be obtained by

$$\rho = \sum_{\alpha=0}^{8} f_\alpha, \quad \rho\boldsymbol{u} = \sum_{\alpha=0}^{8} \boldsymbol{e}_\alpha f_\alpha + \frac{1}{2}\boldsymbol{F}\delta t, \quad p = c_s^2\rho. \tag{4}$$

The direct-forcing immersed boundary method is employed to handle fluid–structure interaction. While Eq. (2) are resolved on Eulerian grids ($\boldsymbol{x}_{i,j}$), the solid boundary is discretized into small elements by Lagrangian points ($\boldsymbol{x}_b$). The boundary force on a Lagrange point can be calculated by[26]

$$\boldsymbol{f}(\boldsymbol{x}_b, t) = 2\rho\frac{\boldsymbol{u}^d - \boldsymbol{u}^{noF}(\boldsymbol{x}_b, t + \Delta t)}{\Delta t}, \tag{5}$$

where the desired velocity $\boldsymbol{u}^d$ is obtained from the solid solver and the unforced velocity $\boldsymbol{u}^{noF}$ is calculated by $\mathbf{u}^{noF}(\mathbf{x}_b, t + \Delta t) = \sum_{\mathbf{x}_{i,j}} \mathbf{u}^{noF} D(\mathbf{x}_{i,j} - \mathbf{x}_b)\Delta x^2$. The force density $\boldsymbol{F}$ on the Eulerian mesh is calculated by spreading the force density on the Lagrangian points $\boldsymbol{f}(\boldsymbol{x}_b, t)$ to the surrounding area

$$\boldsymbol{F}(\boldsymbol{x}_{i,j}, t) = \sum_{\boldsymbol{x}_b} D(\boldsymbol{x}_{i,j} - \boldsymbol{x}_b)\boldsymbol{f}(\boldsymbol{x}_b, t)\Delta S_b, \tag{6}$$

where ΔS_b is the boundary element length including the Lagrange point $\boldsymbol{x}_b$. $D(\cdot)$ represents a discrete δ function,

$$D(\boldsymbol{x}_{i,j} - \boldsymbol{x}_b) = \frac{1}{(\delta x)^2} d_{\delta x}\left(\frac{x_i - x_b}{\delta x}\right) d_{\delta x}\left(\frac{y_i - y_b}{\delta x}\right), \tag{7}$$

with[27]

$$d_{\delta x}(r) = \begin{cases} \frac{1}{8}\left(3 - 2r + \sqrt{1 + 4r - 4r^2}\right), & 0 \le r < 1, \\ \frac{1}{8}\left(3 - 2r + \sqrt{1 + 4r - 4r^2}\right), & 0 \le r < 1, \\ 0, & r \ge 2. \end{cases} \tag{8}$$

B. Solid solver

As highlighted in the Introduction, prior research has predominantly focused on two simplified physical models (the continuum model and the articulated model) with a specific focus on the hydrodynamic mechanisms underlying fish locomotion. In both models, the fish's motion is characterized by finite parameters linked to the model's degrees of freedom, offering the potential to identify optimal locomotion patterns. The continuum model, initially proposed by Carling,[28] has gained considerable attention in research, encompassing optimization studies across various hydrodynamic scenarios, such as steady-state cruising,[14] accelerated starting,[15] and collective swimming.[29–31] However, the hydrodynamic performance of the articulated model has received less exploration, except for its application as a standard validation case.[32–34]

While optimization based on the continuum model yields patterns more closely resembling actual fish swimming, the articulated model boasts unique advantages. First, it has been previously researched through potential flow theory,[35] which is highly suitable as our surrogate model. Second, the use of the articulated model is favored in the field of robotics, given constraints imposed by real-world materials and mechanical structures.[36,37] Third, hinges can be configured as revolute joints with passively induced pitch motion, simulating passive appendages such as the alulae of flying animals and fins of aquatic swimmers.[38] Finally, the internal dynamics (muscle behavior from a biological standpoint and actuator design from a robotics perspective) are also worthy of elucidation.

The most effective approach to address multibody systems is rigid multibody dynamics.[24] In a given rigid multibody model with $n_{dof} \in \mathbb{R}^n$ degrees of freedom, generalized coordinates are employed to describe the system's state. We use the symbols $\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), \ddot{\boldsymbol{q}}(t), \tau(t) \in \mathbb{R}^{n_{dof}}$ to, respectively, denote the vectors of generalized positions, velocities, accelerations, and forces at time t. We will omit the argument t since the dynamics is considered at a given instant of time. The dynamics of rigid multibody systems, which establishes the relationship between forces and induced accelerations, is expressed through the following equation using generalized coordinates

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \tau_{int} + \tau_{ext}. \tag{9}$$

Here, the matrix $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n_{dof} \times n_{dof}}$ represents the generalized inertia matrix, and the term $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n_{dof}}$ accounts for the generalized bias force, including the Coriolis and centrifugal forces. τ_{int} and τ_{ext} are vectors representing the generalized internal forces (including spring forces, damping forces, and driving forces) and the generalized external forces (resulting from gravity and surrounding fluids). This second-order differential equation can be integrated using numerical methods, such as the Runge–Kutta method.

Distinguished by the nature of motion actuation, hinges can be broadly categorized into two types: passive and active. In a given rigid-body system, the algorithm for passive hinges involves calculating the acceleration response to an applied force, a process known as forward dynamics. On the other hand, for active hinges, the calculation involves determining the force required to produce a given acceleration, termed inverse dynamics. In this study, we utilize the articulated body algorithm (ABA) for forward dynamics and the recursive Newton–Euler algorithm (RNEA) for inverse dynamics computation.[24]

As depicted in Fig. 1, for the floating-base system under investigation, the relative motions between each pair of adjacent hinge-linked components are prescribed, while the global movement driven by hydrodynamic forces remains unknown. To facilitate free swimming, allowing unconstrained translational and rotational motion, two prismatic joints q_x, q_y, and a revolute joint q_z are introduced between the fixed base and the head segment, all of which are passive joints. Subsequently, forward dynamics are applied to the passive joints, while inverse dynamics are performed on the remaining active joints. The use of generalized coordinates proves to be more efficient and adaptable, eliminating the need to rederive control equations even if the configuration undergoes modifications, such as altering the structural tree or changing hinge types. We validates this fluid–multibody interaction system in Appendix B.

C. Surrogate model

As proposed in Ref. 35, the surrogate model only account for the momentum exchange between the articulated body and the surrounding fluid in the context of a potential flow model, thus avoiding the complexity of considering all the details of the fluid medium. The articulated body is immersed in a perfect fluid, which means fluid particles will slip along the boundaries of the solid. Under this assumption, the dynamic equations governing the body motion can be written solely in terms of the body variables, without the explicit inclusion of the fluid variables. This is appropriate in the case of not considering the interactions between fishes, since we are mainly interested in the swimming performance of the fish rather than the surrounding flow.

Here, we consider the general case of the n-link eel-like model shown in Fig. 1. For the sake of clarity, supposing that the bodies $\mathscr{B}_i, i = 1, 2, \ldots, n$ are identical and made of a homogeneous material with mass density equal to that of the fluid density $\rho_s = \rho_f$. They have an ellipsoidal geometry with semi major-axis a and semi minor-axis b. The joints are placed at a distance e away from the tips of the ellipses along their major axes. Their angles, defined as $q_i, i = 1, 2, .., n - 1$, are shape variables describing the rotations of the latter link $\mathscr{B}_{i+1}$ relative to the previous one $\mathscr{B}_i$.

It is convenient to introduce a fixed inertial frame $\{\boldsymbol{e}_1, \boldsymbol{e}_2\}$ and co-rotating body-fixed frames $\{\boldsymbol{b}_1, \boldsymbol{b}_2\}$ attached at the center of mass of each solid link. Let $\boldsymbol{g}$ denote the locomotion variables (q_ω, q_x, q_y),

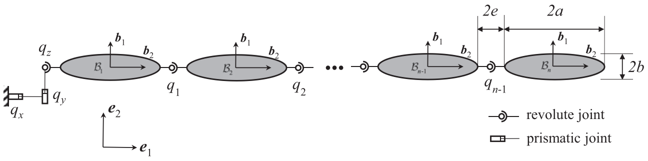


FIG. 1. The diagram of n-link eel-like structure. The joints q_x, q_y, and q_z are floating base coordinates, while q_1, q_2, $\cdots$, q_{n-1} are generalized coordinates of the multibody system. Elliptical bodies, each with major-axis $2a$ and minor-axis $2b$, are separated by a distance $2e$.

such that $\dot{\boldsymbol{g}} = (\dot{q_\omega}, \dot{q_x}, \dot{q_y})$ represent the corresponding rotational and linear velocities. It is obvious the net locomotion $\boldsymbol{g}$ is the results of the variation of shape variables $(q_1, q_2, \ldots, q_{n-1})$, and the configuration of this fish system can be fully described by the combination of both.

In the absence of external forces and moments, the kinetic energy of the whole system T can be written as the sum of the energies of the solid links $T_{\mathscr{B}_i}$ and the fluid T_f,

$$T = \sum_{n=1}^{n} T_{\mathscr{B}_i} + T_f. \tag{10}$$

The kinetic energy of the ith link $T_{\mathscr{B}_i}$ can be written in the following form:

$$T_{\mathscr{B}_i} = \frac{1}{2} \xi_i^T \mathbb{I}^s \xi_i, \quad i = 1, 2, \ldots, n, \tag{11}$$

where $\xi_i = (\omega_i, u_i, v_i)^T$ is the velocity of link $\mathscr{B}_i$ expressed in the $\mathscr{B}_i$-fixed frame. By coordinate transformation, each ξ_i can be represented by a combination of the locomotion variables and the shape variables.[24] The solid inertia matrix $\mathbb{I}^s$ is a 3×3 diagonal matrix with diagonal entries (I, m, m), where $I = m(a^2 + b^2)/4$ is the body moment of inertia and $m = \rho_s \pi ab$ is the mass of the ellipse.

Since we assume the fish is immersed in an infinite domain of incompressible and irrotational fluid, the velocity field can be obtained by take the gradient of a potential function ϕ, which is the solution to Laplace's equation $\nabla^2 \phi = 0$. Previous studies[35,39] have shown that following a standard procedure the kinetic energy of the fluid T_f can be written as

$$T_f = \frac{1}{2} \xi_i^T \mathbb{I}_{ij}^f \xi_i, \quad i = 1, 2, \ldots, n, \tag{12}$$

where $\mathbb{I}_{ij}^f$ is the 3×3 added inertia matrix. Here, we make the assumption that the three identical elliptical bodies are hydrodynamically decoupled, implying that the presence of other bodies does not influence the added masses associated with a given body. Therefore, one has diagonal added inertia matrixes $\mathbb{I}_{11}^f = \mathbb{I}_{22}^f = \mathbb{I}_{33}^f = \mathbb{I}^f$ with entries (I^f, m_1^f, m_2^f), where I^f, m_1^f and m_2^f are given by[40]

$$I^f = \rho_f \pi (a^2 - b^2)^2 / 8, \quad m_1^f = \rho_f \pi b^2, \quad m_2^f = \rho_f \pi a^2. \tag{13}$$

Consequently, the total kinetic energy Eq. (10) can be simplified to $T = \frac{1}{2} \xi_i^T \mathbb{I} \xi_i$, where $\mathbb{I} = \mathbb{I}^s + \mathbb{I}^f$ is also diagonal with nonzero entries $(j, m_1, m_2) = (I + I^f, m + m_1^f, m + m_2^f)$. For the neutrally buoyant bodies, the Lagrangian L is equal to the total kinetic energy T, which is a function of the aforementioned locomotion variables $\boldsymbol{g}$ and shape variables $q_1, q_2, \ldots, q_{n-1}$ and their corresponding velocities. Therefore, the dynamic equations governing these variables can be given by the Lagrange equations,

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\boldsymbol{g}}}\right) - \frac{\partial L}{\partial \boldsymbol{g}} = 0, \tag{14a}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \tau_i, \quad i = 1, 2, \ldots, n-1, \tag{14b}$$

where $\tau_i, i = 1, 2, \ldots, n-1$ are the forcing torques exerted on the revolute joint q_i [refer to Ref. 35 for the specific form of Eq. (14)]. According to Eq. (14a), we can obtain a first-order differential equation, which can be integrated to obtain net rigid fish motion given $(q_1, q_2, \ldots, q_{n-1})$. Also, the corresponding needed torques can be computed based on Eq. (14b).

D. Deep reinforcement learning

DRL is a mathematical framework designed for solving sequential decision-making problems.[41] In this framework, an *agent* predicts an *action* a_t based on the observation from the current *state* s_t to receive some heuristic-driven *reward* r_t. The ultimate objective of the agent is to maximize the cumulative reward by learning a policy $\pi(a_t|s_t)$. The optimization problems, which are essentially single-step decision-making problems, can be similarly addressed by DRL. Here, the single-step version of the policy-based algorithm, proximal policy optimization (PPO), is implemented in the following chapters (details about PPO are given in Appendix A).

1. Single-step PPO and its enhancement

In the context of the single-step version, the concepts of DRL and the PPO algorithm can be simplified, as only one action (i.e., the selected parameters) needs to be taken in the initial state.[18] For instance, the trajectory τ now consists of (s_0, a_0, r_0), where the initial state s_0 is constant, and the initial reward r_0 precisely represents the objective function. Consequently, the advantage function is also independent of the initial state, and its formula can be reduced to

$$A^{\pi_\theta}(a_0) \sim r_0 - V^{\pi_\theta}(s_0) = r_0 - \mathbf{E}_{\tau \sim \pi_\theta}(r_0). \tag{15}$$

Here, $V(s_0)$ is equivalent to the expectation of the initial reward r_0 following the policy π_θ. Therefore, the critic network associated with $V(s)$ is unnecessary, as we can approximate it by averaging all the evaluated rewards r_0 each time before the policy is updated. Thus, we only need to update the actor network, denoted as $a \sim \pi_\theta(\cdot|s_0)$. Essentially, the optimization procedure is equivalent to the convergence process to the optimal actor network parameters θ_{opt}, which maps the initial state s_0 to the optimal action a_{opt}.

Typically, the output of the actor network is normalized due to the activation function of the output layer. It is subsequently scaled into practical ranges of the specific problem, implying that the parameter space will be box-shaped. However, for constrained optimization problems, it is evident that rescaling alone falls short of addressing this task. Traditionally, we can explicitly add a penalty term in the objective function to enforce constraints in a soft manner, although this may have some drawbacks in terms of numerical stability and interpretability. Moreover, it means that the bound set for the problem may be violated when evaluating possible candidates.

Given the expensive simulation cost, our goal is to allocate as many computational resources as possible to evaluate solutions within the feasible domain. In this way, during the exploration phase, the agent can take more feasible actions to gain an overall understanding of the objective function and identify the most likely regions where the optimal solution may exist. To impose generic linear constraints on the output of the actor network in a hard manner, we borrow ideas from geometry, specifically the concepts of polyhedra and polytopes. The set of solutions to k linear inequality constraints in m dimensions is called a polyhedron, denoted as follows:

$$\mathscr{C} = \left\{ x | Ax \leq b, A \in \mathbb{R}^{k \times m}, b \in \mathbb{R}^k \right\} \subseteq \mathbb{R}^m. \tag{16}$$

This is the halfspace or H-representation by defining a polyhedron as the intersection of halfspaces represented by linear inequalities. However, a polyhedron can also be described in terms of points (vertices) and generating vectors (rays) according to the Minkowski–Weyl theorem, which is called the vertex or V-representation. Its mathematical form reads

$$\mathscr{C} = \{x | x = conv(V) + cone(R)\}, \tag{17}$$

where $conv(\cdot)$ is the convex hull of a set of vertices $V = \{v_1, v_2, \ldots, v_{n_v}\}$ and $cone(\cdot)$ is the conical hull of a set of rays $R = \{r_1, r_2, \ldots, r_{n_r}\}$

$$conv(V) = \sum_{i=1}^{n_v} \lambda_i v_i, \lambda_i \geq 0, \sum_{i=1}^{n_v} \lambda_i = 1,\ cone(R) = \sum_{j=1}^{n_r} \mu_j r_j, \mu_j \geq 0. \tag{18}$$

Algorithmic switching between these two representations can be achieved using the double description method.[42] While checking whether $x \in \mathscr{C}$ (corresponding to a linear programing problem) is straightforward in the H-representation, the V-representation is better aligned with our requirements, particularly in its ability to sample points inside the constraint set $\mathscr{C}$.

The specific implementation of enforcing hard constraints is outlined as follows: we need to perform some mapping to the output layer of the actor network, as depicted in Fig. 2. First, the actor network outputs n_v coefficients λ_i for the convex combination of the vertices and n_r coefficients μ_j for the conical combination of the rays. Second, activation functions (i.e., softmax function and absolute function) are applied to these coefficients, respectively, to ensure compliance with the requirements in Eq. (18). Finally, through linear combination, we obtain the tuning parameters $x \in \mathbb{R}^m$ that satisfy specific linear constraints.

It is noteworthy that the second step can be regarded as a specially tailored network layer. However, its coefficients (i.e., vertices V and rays R) are precomputed based on the specific linear inequalities and do not participate in the process of updates and backpropagation. In other words, this treatment only requires a finite time using the double description method at initialization, with no additional time expenses at training and test time.

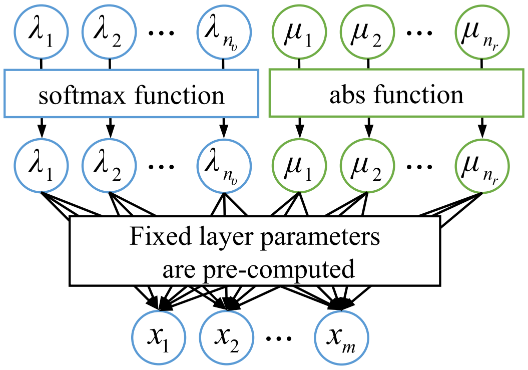


FIG. 2. Network architecture and details of the mapping procedure of our modified single-step DRL algorithm.

2. Controlled transfer learning

In addition to constraining parameter space to save exploration time, TL can serve as a technique to facilitate more effective learning. It offers a framework for transferring knowledge gained from solving a source task (a low-cost environment based on surrogate model) to enhance the performance of a related target task (a high-cost environment based on high-fidelity solver).

Previous work[23] which proposed CTL mainly focused on the application of CTL on multi-fidelity environments and did not deepen our understanding of its working mechanism when it is applied to single-step DRL. Although the environment of the source task utilized both in our work and theirs are the potential flow model, it is evident that this model is more suitable for the high Reynolds numbers flow considered in their work and differs significantly from our actual situation (fish swimming under moderate and low Reynolds numbers flow). However, in Sec. III, we will demonstrate that these qualitative similarities are sufficient to guide the agent from an initialized random policy to an intermediate sub-optimal policy. Furthermore, our contribution lays in developing a framework to combine the DRL algorithm and the LBM solver and allow to apply to complex dynamic control tasks without much modification.

As illustrated in Fig. 3, our study adopts the parameter-based approach transferring the knowledge by copying the weights of the policy from the source task (potential flow model) to the target task (LBM solver). It is noticed that excessive learning on the source task may lead to overfitting, which may hinder the generalizability and lead to negative transfer for the target task. Hence, a variance ratio α is defined to determine when to conclude training on the source task and proceed with weight transfer

$$\alpha = \frac{\sigma_i}{\max(\sigma_i, \sigma_{i-1}, \ldots, \sigma_1)}, \tag{19}$$

where σ_i represents the variance of chosen actions outputted by the policy network. Different from Ref. 23, the variance of rewards is not used for the construction of the criterion. Because we find the limitation of the number of parallel environments leads to inaccurate estimation of the variance of rewards, especially for cases with drastically changing reward functions. It is apparent that the variance of the actions decreases during the training process as long as the agent converges to a global optimal policy. Predictably, the variance ratio will decrease from $\alpha = 1$ initially and approach $\alpha = 0$ gradually when the optimal results are determined. Therefore, we define a cut-off value $\alpha_c = 0.3$, under which we assume the agent has gathered enough knowledge and learn a crude policy for the surrogate model. Then, this policy will be transferred to the target task and fine-tuned based on our high-fidelity environment. We reveal the working mechanism of our optimization framework through an test problem in Appendix C, where we also give intuitive optimization efficiency improvement in terms of reduction in high-fidelity evaluations.

III. OPTIMIZATION

A. Problem formulation

Our optimization study is conducted based on the articulated model described above. As illustrated in Fig. 1, the neutrally buoyant fish is characterized by a kinematic chain comprising n elliptical bodies interconnected by a sequence of revolute joints. Since in our study we

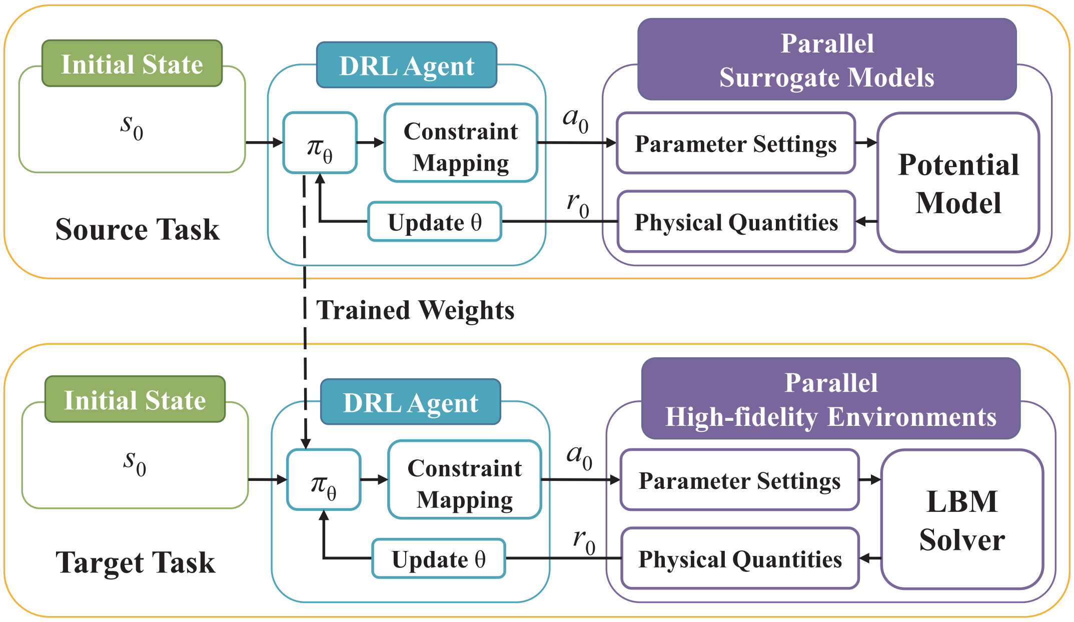


FIG. 3. Incorporation of model-based knowledge into DRL-based optimizer via transfer learning.

mainly focus on the steady swimming, it is a common approach to use a single sinusoid to describe the periodic motion of joint i,[43,44]

$$q_i(t) = A_i \cos(2\pi f_i t + \phi_i), \quad i = 1, 2, \ldots, n-1, \tag{20}$$

where A_i, f_i, and ϕ_i denote the pitch amplitude, frequency, and phase, respectively. Here, we arbitrarily designate the ellipsoid connected to the inertia frame as the head segment, with a reference phase set to $\phi_1 = 0$. Furthermore, a consistent frequency is applied across all joints to ensure synchronized body movements. This results in a total of $2n - 3$ parameters needed to define a complete swimming gait.

Here, we select the 3-link equal-length configuration as the reference case, with a chord length of $2a = 1$ and a density of $\rho = 1$ serving as the reference length and density scale, respectively. The configuration parameters, such as the segment number and length, will be varied in a series of parametric studies to examine their impact on optimal swimming gaits. To maintain consistency with the reference case, we specify that all ellipsoids possess a minor axis of $2b = 0.1$, with the sum of their major axes equating to 3. To adjust the major axis length, the equidistant gap is set to $2e = 0.1$ ensure that it is sufficiently small relative to the major axes.

In the subsequent study, three grid levels are used in the multi-block treatment. The coarsest grid covers the computational dimension $[-12, 12] \times [-12, 36]$ (scaled by $2a$) and two finer domains are nested inside with dimensions $[-8, 8] \times [-4, 12]$ and $[-4, 4] \times [-2, 6]$, respectively. The grid spacing for each level is half of the next coarser level and the finest grid resolution is $\Delta x = 0.01$. All simulations are conducted with a constant viscosity of $\nu = 2 \times 10^{-4}$ and undulation frequency $f = 0.015$. As our reference Reynolds number is fixed at $Re_{ref} = 2aU_{ref}/\nu = 100$, the velocity is non-dimensionalized with $U_{ref} = \nu Re_{ref}/c = 0.02$. This moderate Reynolds number is representative for the free swimming of small fish and larvae[45] and allows for a relatively large parameter space.

The resulting propulsion Reynolds number $Re_p = \bar{U}_t c/\nu$ based on the mean translational speed of the centroid of the fish $\bar{U}_t$ is in the range 60–200 approximately. The undulation Reynolds number has been deprecated, as it becomes challenging to determine its undulation amplitude for models with more than three segments. The swimmer accelerates from rest and gradually approaches an equilibrium state where drag forces balance out the thrust produced by its gait. The optimization cases are run for 30 cycles, and the objective function values are computed using time-averaged values during the last cycle.

B. Algorithm setup

We primarily focus on maximizing two crucial performance metrics in an evolutionary scenario: swimming efficiency and cruising speed. These objectives align with the practical biological functions of aquatic creatures, such as migrating (for efficient swimming) or hunting and escaping (for fast swimming).

Before exploring the optimization process, it is essential to quantify the consumed power to compare the performance of the mechanism. Assuming no energy losses from internal friction, the total work is input into the entire system through torques generated at each ideal hinge. The mathematical formulation of the total input power is given by $P_{\text{tot}} = \sum_{i=1}^{n-1} \tau_i \dot{q}_i$.

For the efficiency metrics,[6] it is typically evaluated using the Froude efficiency η, defined as the ratio of thrust power P_{thrust} to input power P_{tot}. There is no clear consensus on how thrust and drag should be decomposed for undulatory swimming, as their producing regions are distributed across the entire body and vary with time.[46] Hence, we postulate that the time integral of P_{thrust} is positively related to the kinetic energy of translational motion E_k. We do not explicitly calculate the exact value of thrust itself but characterize the efficiency by the ratio $\eta \sim E_k/W_{\text{tot}}$. We then define the objective function as the amount of input work W_{tot} transformed into E_k during an undulation period T,

$$F_\eta = \frac{E_k}{W_{tot}} = \frac{0.5m\bar{U}_t^2}{\int_t^{t+T} P_{tot}\,dt}. \qquad (21)$$

For the speed metrics, the objective function comprises the mean translational velocity $\bar{U}_t$ and a penalty term designed to constrain the maximum input work in each period

$$F_U = \bar{U}_t 10 * H(W_{tot} - W_{target}) * (W_{tot} - W_{target})^2, \qquad (22)$$

where $H(\cdot)$ is the Heaviside function. The second term, motivated by natural physiological limits, is essential. Early trials indicated that directly maximizing $\bar{U}_t$ led to rather unnatural motion patterns, despite rapid swimming speeds. After adding the penalty term, the algorithm converges to the fastest gait with the specified input work limit. The input work threshold W_{target} is chosen as 24 based on preliminary computational experiments.

In addition, the strategy of parallel environments is adopted to collect experiences concurrently. Data transfer between the agent and the environment is facilitated using TCP/IP sockets to ensure real-time interaction. Additionally, this approach allows for distributed computing on different devices, such as running multiple fluid environments on CPU clusters and the single-step DRL optimizer on GPUs. The other hyperparameter settings remain the same as in our previous study[20] if not stated.

IV. RESULTS AND DISCUSSION

A. The reference case

For the reference case, the control parameters consist of two amplitude coefficients A_1 and A_2 one phase difference ϕ_2. To avoid redundancy in the parameter space, we assume that the pitching amplitude of q_1 is larger than that of q_2 and impose the constraints $\mathscr{C} = \{0 \leq A_2 \leq A_1 \leq 1, -2\pi \leq \phi_2 \leq 0\}$ on the parameter space. This choice is made because the pairs (A_1, A_2, ϕ_2) and $(A_2, A_1, -2\pi - \phi_2)$ represent identical swimming patterns with opposite forward directions. Although the linear inequalities are not intricate, the resulting design domain cannot be explicitly described by box constraints. We optimize the kinematics of this 3-link equal-length model with respect to the efficiency and speed metrics defined by Eqs. (21) and (22). The optimal solutions in Table I and its corresponding motion presented in Figs. 4(a) and 4(b) are obtained using eight parallel environments.

From Table I, we observe that A_1 converges to 1 (the maximum allowed value) in both situations, suggesting that, for 3-link swimmers, larger head motion is necessary for achieving efficient and fast swimming. When combined with specific tail flapping motion simultaneously, it results in entirely distinct swimming performances. The resulting two swimming patterns are visualized in Fig. 5 using velocity vectors and vorticity contours. In general, both wakes consist of vortices of alternating sign shed at the tail per undulation cycle. This is exactly the reverse Kármán vortex street commonly observed in the wake of fish and robotic swimmers.[1]

TABLE I. Optimized results and performance metrics of the 3-link equal-length model.

	A_1	A_2	ϕ_2	$2a$	$\bar{U}_t$	W_{tot}	F_η
Efficient	1.00	0.79	−1.68	1.0	0.87	10.25	1.75%
Fast	1.00	0.94	−1.14	1.0	1.18	24.00	1.37%

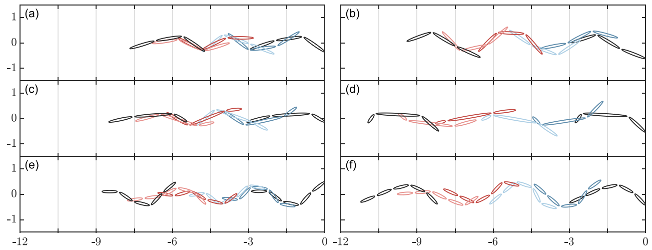
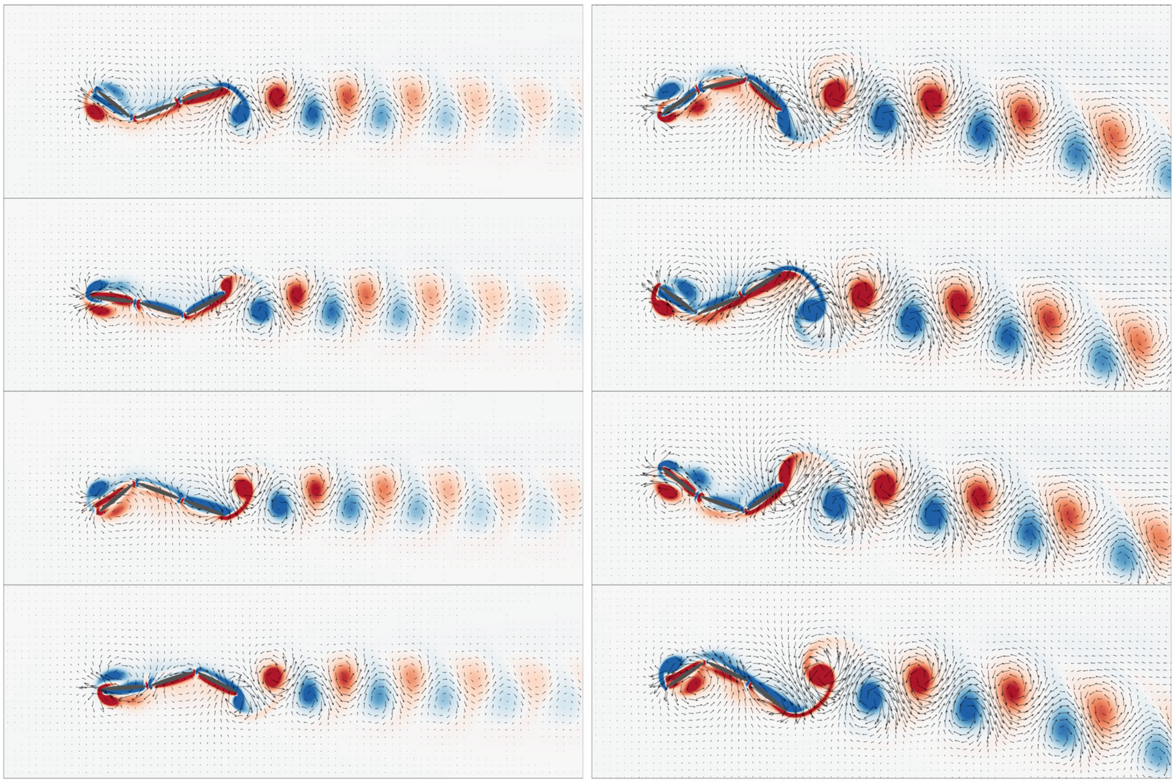


FIG. 4. Swimming kinematics of the optimal swimmers during two cycles as seen from above in the laboratory frame. Top to bottom: (a) and (b) 3-link equal-length, (c) and (d) 3-link variable-length, and (e) and (f) 5-link equal-length cases. Left column: (a), (c), and (e) efficient mode; right column: (b), (d), and (f) fast mode. The fish speed is magnified by two times for better clarity.

However, a notable distinction is observed in the alignment of the downstream vortices' centers between the efficient and fast modes. In the efficient mode, the vortices align well, contributing to an organized wake, whereas in the fast mode, they tend to shed in pairs, inducing deflected wakes. Similar behavior has been documented in studies on the wake of a flapping foil at high Strouhal numbers.[47,48] For a stationary flapping foil in a uniform flow, increased flapping frequency or amplitude leads to a pronounced inclination of the reverse Kármán vortex street in the downstream wake. This is often accompanied by the generation of sustained lateral forces and yawing motion. In our study, the centroid motion of the free-swimming model typically traces a zig–zag path in any direction due to the unpredictable yaw angle. Our optimization objective is to maximize translational velocity (resultant velocity combining longitudinal and lateral velocities). Therefore, the presence of a non-zero lateral mean velocity is reasonable for the fast mode, and this yawing motion aligns with conclusions drawn by previous literature.

From a hydrodynamic perspective, we can analyze the corresponding swimming mechanisms of both modes. In Figs. 4(a) and 4(b), it is evident that the undulation amplitude in the fast mode is larger, corresponding to a higher flapping Reynolds number. Previous studies have revealed a positive correlation between flapping Reynolds number and propulsion Reynolds number.[49,50] Thus, a larger undulation amplitude in the fast mode produces stronger trailing-edge vortices, leading to a higher propulsion speed. The properties of the trailing-edge vortices closely govern the symmetry-breaking behavior of the deflected wake: Greater asymmetry, contained in stronger trailing-edge vortex strength, creates a preference for the deflected wake.[47]

In the efficient mode, besides the smaller undulation amplitude of the posterior part, another significant feature is the distinct vortex shedding mechanism at the head segment. Despite both modes having identical head joint motion, the leading-edge vortices generated in the fast mode are not effectively reabsorbed by the boundary layer along the head segment. Instead, they directly detach and form secondary vortices, resulting in some energy loss into the flow field. In contrast, in the efficient mode, the fish synchronizes the body undulation to


(a)Efficient mode

(b)Fast mode

FIG. 5. Velocity field and contours of vorticity normal to the image plane (ω_z) for 3-link equal-length model in an entire swimming cycle after the fish has reached the asymptotic mean swimming speed. Left column: (a) efficient mode; right column: (b) fast mode.

make full use of those leading-edge vortices, enabling more energetically efficient forward propulsion.

B. 3-link variable-length case

In this section, we explore the variations in the characteristics of the most efficient and fastest swimmer when the lengths of the three links are altered. In addition to the three kinematic parameters considered in Sec. IV A, the control parameters now encompass two additional geometric parameters, namely, $2a_1$ and $2a_3$, where $2a_1$ and $2a_3$ represent the major axis lengths of the head and tail segments, respectively. Consequently, the major axis length of the intermediate segment can be calculated as $2a_2 = 3 - 2a_1 - 2a_3$. The constraints on

TABLE II. Optimized results and performance metrics of the 3-link variable-length model.

	A_1	A_2	ϕ_2	$2a_1$	$2a_2$	$2a_3$	$\bar{U}_t$	W_{tot}	F_η
Efficient	1.00	0.61	−1.53	0.94	1.46	0.60	0.95	6.40	3.29%
Fast	1.00	0.71	0.00	0.40	1.72	0.88	1.52	24.00	2.26%

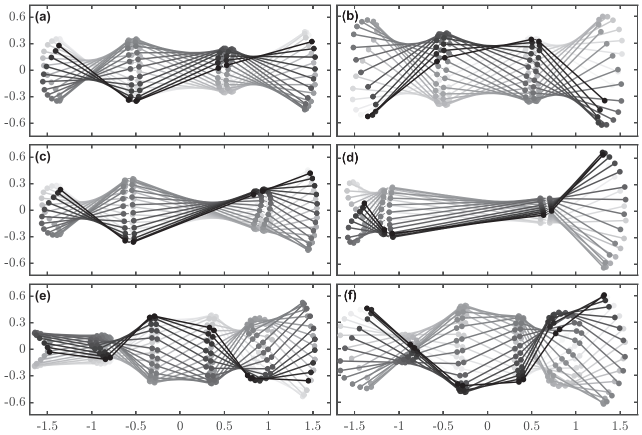


FIG. 6. Midline deformation of the optimal swimmers during one cycle. Top to bottom: (a) and (b) 3-link equal-length, (c) and (d) 3-link variable-length, and (e) and (f) 5-link equal-length cases. Left column: (a), (c), and (e) efficient mode; right column: (b), (d), and (f) fast mode.

the kinematic parameters remain consistent with Sec. IV A. For the geometric parameters, it is ensured that the major axis lengths of all segments fall within the interval [0.4, 1.8], expressed as $\mathscr{C} = \{0.4 \leq 2a_1, 2a_2, 2a_3 \leq 1.8\}$.

From the results in Table II alone, it is noteworthy that the optimal A_1 for both scenarios again converged to the maximum allowed value of 1, indicating the persistence of employing large amplitude oscillations at the head. However, a notable difference with Table I is observed in that, with the allowance for variations in the length of each segment, optimal performance can now be achieved not only by adjusting the phase of tail flapping but also by altering the position of body curvature. The kinematics and midline motion for both optimal modes are reproduced in Figs. 4(c), 4(d), 6(c), and 6(d). Naturally, the hydrodynamic performance shows a significant improvement compared to the reference case after introducing additional geometric degrees of freedom. For the efficient mode, adjusting the length by properly shortening the head and lengthening the mid-section nearly doubles the efficiency. This improvement is coupled with a 37.54% reduction in the total input work and an 8.25% increase in cruising speed. As for the fast mode, under the premise of $W_{tot} = 24$, not only has the cruising speed increased by 28.12% but there has also been a substantial increase in nearly 64% in terms of efficiency.

To explore the mechanism by which segment length influences hydrodynamic performance, a detailed analysis of their motion behaviors and flow field is required. For the efficient mode (the variable-length case), the phase lag of the tail joint is close to $\pi/2$, and the motion combination of the last two segments resembles that of a flexible flapping plate. In other words, the tail segment exhibits a slight bending opposite to the motion direction of the intermediate one in each half cycle. This functions similarly to bird feathers, where passive bending effectively acts as a deflected trailing-edge flap of the wing, introducing camber and thereby increasing total thrust. Additionally, its optimal tail segment is shorter than that of the equal-length model, which implies smaller greater mass and rotational inertia. Consequently, the tail motion needs to overcome lower inertial forces, leading to reduced input power and increased power extraction. From the comparison of the consumed input power curve of each hinge in Fig. 7, we are convinced that the restriction of equal segment length may require more input power, thus resulting in worse efficiency.

Next, let us examine the optimized fast mode. Surprisingly, its head and tail movements are nearly in phase, and the optimized head length reaches the lower limit we set at 0.4. The short head uses a larger flapping amplitude to counterbalance the torque generated by the long tail, which functions akin to a reaction wheel controlling the body's attitude. Consequently, the longest mid-section strives to remain parallel to the forward direction, minimizing pressure drag as much as possible and thus facilitating high-speed swimming. This can also be observed in the midline motion in the center-of-mass frame in Fig. 6(d) and the instantaneous pressure contour in Fig. 8.

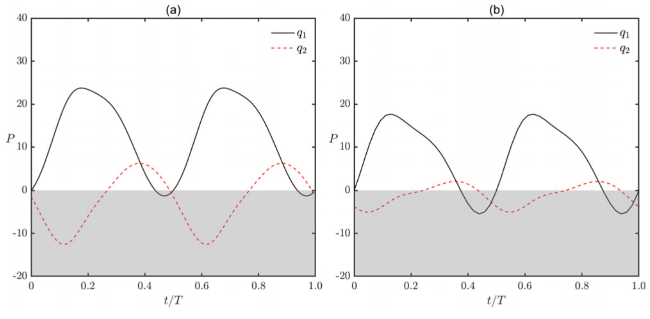


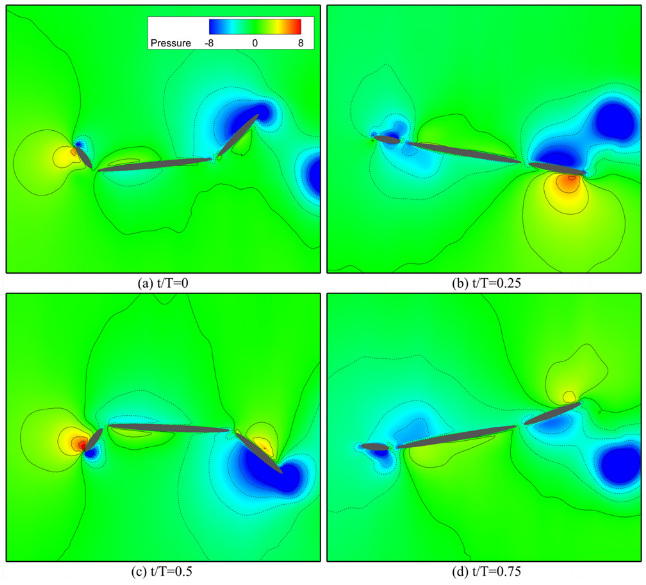
FIG. 7. The joint power curves for actuators q_1 and q_2 during one undulatory cycle for the efficient mode: (a) 3-link equal-length model and (b) 3-link variable-length model.

The midline motion of the fast mode exhibits characteristics reminiscent of the carangiform mode. Lateral deformations are primarily confined to the tail, with almost no deformations in the mid-section of the body. However, it differs from the carangiform mode due to its intense oscillating motion at the head. Although this type of head motion is generally not encountered in nature, it is reasonable when the morphological factor is considered. Most real fishes typically have large and wide heads that gradually become thinner moving toward the tail. This means that they alter their rotational inertia by concentrating mass toward the anterior part of the body, eliminating the need for significant head oscillations to balance tail torque. Therefore, the optimization results of the fast mode should be considered a compromise due to the limited degrees of freedom in fish shape.

This inspires us that the morphological aspects of the fish body are equally indispensable in the optimization process, with further potential for enhancing hydrodynamic performance. Anyway, our results are consistent with previous research on the optimal link-length ratio of a robotic fish, which also showed that models with variable-length segments perform better than those with equal-length segments.[51]

C. 5-link equal-length case

Finally, we aim to understand how the segment number affects the hydrodynamic performance of the articulated model. We increased



(a) t/T=0
(b) t/T=0.25
(c) t/T=0.5
(d) t/T=0.75

FIG. 8. Snapshots of instantaneous pressure coefficients for the fast mode of the 3-link variable-length model. Four different instants within one cycle are presented: (a) $t/T = 0.0$, (b) $t/T = 0.25$, (c) $t/T = 0.5$, and (d) $t/T = 0.75$.

TABLE III. Optimized results and performance metrics of the 5-link equal-length model.

	A_1	A_2	A_3	A_4	ϕ_2	ϕ_3	ϕ_4	$2a$	$\bar{U}_t$	W_{tot}	F_η
Efficient	0.92	1.00	1.00	1.02	−2.80	−3.92	−5.71	0.6	1.06	9.17	2.88%
Fast	0.25	1.00	0.89	1.00	−0.16	−1.05	−2.88	0.6	1.55	24.00	2.35%

the number of segments from three to five, guided by the previous work suggesting that five segments are sufficient to describe the kinematics extracted from actual fish data with at least 99% accuracy.[25]

There are a total of seven control variables, including four amplitude coefficients A_1, A_2, A_3, A_4 and three relative phase differences ϕ_2, ϕ_3, ϕ_4. This poses a challenge for optimization algorithms in high-dimensional parameter exploration, and thus, ten parallel environments are concurrently simulated. Moreover, the constraints imposed on this high-dimensional space are $\mathscr{C} = \{0 \le A_1, A_2, A_3, A_4 \le 1, -\pi/2 \le \phi_4 \le \phi_3 \le \phi_2 \le 0\}$, grounded in conclusions drawn from real fish data. The phases of each joint motion lag successively as their respective positions shift backward, ensuring that the undulatory traveling wave formed by the whole system moves rearward.

From Table III, it is evident that the optimal performance of 5-link equal-length model for both modes is promoted to a certain extent compared to the reference case. The maximal efficiency and cruising speed of this 5-link model were superior to the reference case by 64.33% and 30.62%, respectively. Figures 4(e), 4(f), 6(e), and 6(f) illustrate the geometry and kinematics of the optimal swimmer more intuitively.

With the presence of more joints in the current fish model, the head segment no longer bears the sole responsibility of generating thrust and provides more flexibility in motion. In the efficient mode, a notable feature is the small lateral displacement of the head segment in the center-of-mass frame. This is achieved by curving the head segment in the opposite direction to the tail, ensuring excellent alignment of the head posture with the swimming direction. In the fast swimming mode, the anterior half of the body hardly bends, and the first two segments almost form a unified entity. Additionally, the posterior part of the body exhibits large curvatures in both modes, adjusting the angle of the tail tip to facilitate smoother vortex shedding during changes in the tail direction.

From a hydrodynamic perspective, the mechanism of efficiency improvement by the 5-link model differs from that of the 3-link variable-length model. In Fig. 9(a), we plot the evolution of the input power of each hinge in one cycle for the 5-link model. The resulting curves reveal several intervals of negative power for multiple joints, indicating a better capacity to harness power from the flow field for energy conservation. This phenomenon has been identified in earlier experimental studies on specific fish species in nature.[52,53] According to these studies, fishes can capture negative work and exploit it to reduce the overall cost of motion due to the elasticity of their muscles and tendons.

It is noted that our definition of total input power suggests that the negative work can offset the cost of positive work. Figure 9(b) illustrates the quantitative statistics of positive and negative work for each model. We can observe that, the 3-link model, whether equal-length or variable-length, shows a consistent positive-to-negative work ratio around 3.5. In contrast, the 5-link model demonstrates a ratio close to 1.76, nearly halving the value. Although the negative work in the 5-link model offsets over half of the positive work, the substantial positive work requirements result in a total input power even higher than that of the 3-link variable-length model. In summary, the 3-link variable-length model enhances efficiency by reducing the overall input work required for hinge rotations, while the 5-link model, with the aid of more hinges, allows the entire system to better recover energy from the flow field to improve efficiency.

While for the fast locomotion of the 5-link model, explaining its underlying mechanism is much more easier. As depicted in Fig. 6(f), its midline kinematics resembles that of the C-start escape mechanism.[15] With more revolute joints, the body can bend into larger curvatures, capturing a significant amount of fluid that is subsequently accelerated backward through body undulation, resulting in an acceleration of the whole system in the opposite direction. Thus, by transferring more momentum to the fluid, each stroke of this motion results in a burst of acceleration to balance the drag-induced deceleration. The high average speed is maintained by cyclically repeating this process.

In conclusion, the optimization results accord with our intuition since the increased flexibility from additional joints enhances performance in both scenarios. Surprisingly, the optimization results for the 3-link variable-length model even surpass those for the 5-link equal-length model in certain aspects. This suggests that, for the studied parsimonious model, altering morphological parameters may be more crucial to performance improvement than increasing the number of links.

V. CONCLUSION

While fish are flexible and have high degrees of freedom at the same time,[54–56] fish robots are often made from multiple segments with rigid (to accommodate batteries, electronics, sensors, motors, and cables) parts. Given technological limitations, our research provides insights and generalizability regarding several key questions: the partitioning strategy of the fish body and the optimal placement of actuation points. We examine the most parsimonious robot design that can

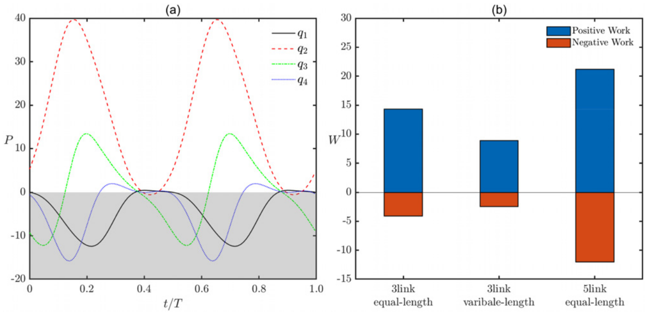


FIG. 9. (a) The joint power curves for actuators q_1, q_2, q_3, and q_4 during one undulatory cycle for the efficient mode of 5-link equal-length model; (b) integral results of positive and negative work over one cycle for three models.

accurately replicate the movements of the fish. However, since the Reynolds number and the optimization objective of our research problem are fixed, our design cannot fit all fish species and should vary to model different behaviors effectively.

The presented work introduces a novel computational framework for optimizing the swimming gaits of a self-propelled articulated swimmer. This framework combines a fluid–structure interaction solver for multibody systems with a customized RL-based optimization algorithm. The dynamic equations of the multibody system in generalized coordinates are advanced using the Runge–Kutta scheme. The fluid is solved using the LBM, and the presence of the multibody system is accounted for using the IBM. The resulting approach is benchmarked on self-propulsion induced by active joint motion cases to validate the coupled dynamics between the flow and the articulated structure.

Considering the intricacies involved in designing swimming gaits, we choose the emerging single-step DRL technique combined with controlled transfer learning as the optimization algorithm. For the source task, the agent initially optimizes motion parameters in the potential model environment and dynamically terminates to obtain a sub-optimal policy. Then, it is transferred to the LBM environment for fine-tuning to complete the final steps of training. This step-by-step approach requiring fewer evaluations on the high-fidelity environments significantly reduces computational time, but more importantly creates a physics-informed reinforcement learning framework. In addition, the previous statistical study on real fish data offers some reasonable physiological constraints that can help exclude nonphysical solutions and explore the design space more effectively. As the original algorithm was only suitable for box constraints, we introduce the V representation transformation which is capable of handling linear constraints in a strict manner.

The developed framework has been applied to gait optimization of an articulated swimmer model at moderate Reynolds numbers. Within the optimization study, a systematic investigation was conducted on two key factors influencing the optimized results: the length and number of linkages. By examining the impact of these two factors on the optimal kinematics of the mechanism, a series of new insights are provided on the design and behavior of both natural and robotic swimmers. Variable segment lengths enable a subtle bending motion of the tail similar to bird feathers, enhancing the total thrust generation for efficient swimming. While for fast swimming, a short head with a large flapping amplitude counterbalances torque from the tail to adjust the body posture, minimizing pressure drag and facilitating high-speed swimming. The flexibility offered by more segment numbers is utilized differently for efficient and fast swimming modes. In the efficient mode, the head always bends in the opposite direction to the tail, in order to align well with the swimming direction and exploit negative work from the flow field. Conversely, for the fast mode, the whole body bends into large curvatures, which resembles the C-start acceleration mechanism.

In terms of our future work, we can focus on improving the DRL-based optimization algorithm by addressing nonlinear constraints and exploring the use of transfer learning from simulations to experiments. Additionally, ongoing efforts involve refining fish body models by incorporating the skin between each body and introducing more degrees of freedom in morphology.[57] It is noteworthy that the present approach can be readily extended to three dimensions without requiring many methodological changes. Our work has concentrated on steady swimming motion, but fishes have different movements for other higher-level unsteady behavior control (e.g., path following, acceleration, and sharp turn). Due to the modular structure of our framework, simply replacing the DRL-based optimizer with the original DRL algorithm enables our framework to readily explore these diverse situations. This will provide a more comprehensive understanding of fish locomotion and contribute to the development of versatile and adaptive robotic swimming systems.

ACKNOWLEDGMENTS

This work was supported by the Natural Science Foundation of China (NSFC) (Grants Nos. 11772326 and 11972342). The authors thank Dr. Yusheng Jiao for fruitful discussions and suggestions about the potential flow model. Simulations are performed at the Supercomputing Center of the University of Science and Technology of China.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Chengyun Wang: Conceptualization (equal); Investigation (equal); Methodology (equal); Writing – original draft (equal). **Haibo Huang:** Funding acquisition (equal); Resources (equal); Supervision (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

APPENDIX A: PROXIMAL POLICY OPTIMIZATION

In the original PPO formulation, an actor network is introduced to approximate the policy π_θ, where θ represents the weights and biases of the neural network.[58] Considering a trajectory defined as a sequence of state-action-reward triples $\tau = (s_0, a_0, r_0, \ldots, s_T, a_T, r_T)$, its expected cumulative reward can be expressed as $J(\pi_\theta) = \mathbf{E}_{\tau\sim\pi_\theta}[R(\tau)]$. Here, $R(\tau) = \sum_{t=0}^{T} \gamma^t r_t$ represents a discounted expected reward over T time steps, and $\gamma \in [0, 1]$ is a discount factor weighing the importance of present and future rewards. To enhance the overall performance of the agent, the actor network can be updated using gradient ascent, as follows:

$$\theta = \theta + \alpha \nabla_\theta J(\pi_\theta), \tag{A1}$$

where α is the learning rate and $\nabla_\theta J(\pi_\theta)$ is the policy gradient. According to the policy gradient theorem and expected grad-log-prob lemma, this term can be substituted in the form of an expectation by

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} \log\left(\pi_\theta(a_t|s_t)\right) A^{\pi_\theta}(s_t, a_t)\right]. \tag{A2}$$

Note that $A^{\pi_\theta}(s_t, a_t)$ represents the advantage function, describing how much better or worse the action a_t is than other actions on average at state s_t based on the current policy. To evaluate the expected value of the return in-state s, the state-value function $V(s)$ is introduced, typically predicted by another critic network. The advantage function $A^{\pi_\theta}(s_t, a_t)$ can then be approximated as

$$A^{\pi_\theta}(s_t, a_t) \sim r_t + \gamma V^{\pi_\theta}(s_{t+1}) - V^{\pi_\theta}(s_t). \tag{A3}$$

It is important to note that two challenges may arise if we directly update the actor network using Eq. (A2). The first challenge is the low efficiency of the data obtained from old policies, and the second is the high sensitivity to the step size. However, PPO addresses these issues by employing a clipped surrogate loss. To ensure the ability to update with data from the old policy $\pi_\theta^{'}$ and maintain similarity between π_θ and $\pi_\theta^{'}$, Eq. (A2) is modified as follows:[59]

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\min\left(\frac{\pi_\theta^{'}(a_t|s_t)}{\pi_\theta(a_t|s_t)}A^{\pi_\theta}(s_t,a_t),\right.\right.$$
$$\left.\left.\text{clip}\left(\frac{\pi_\theta^{'}(a_t|s_t)}{\pi_\theta(a_t|s_t)}, 1-\epsilon, 1+\epsilon\right)A^{\pi_\theta}(s_t,a_t)\right)\right], \tag{A4}$$

where the clip function enforces the ratio between new and old policy within $[1-\epsilon, 1+\epsilon]$ and ϵ is a hyperparameter determining how far away the new policy is allowed to go from the old.

APPENDIX B: ENVIRONMENT SOLVER VALIDATION

We validate our environment solver using a representative problem inspired by biological locomotion: the free swimming of an articulated 3-link fish model.[33] The model represents a special case with $n = 3$ in Fig. 1. Each elliptical body has a density of ρ_s and an aspect ratio of $a/b = 10$. The distance between each body is set to $2e = 0.4$. We utilize the same prescribed motion to control the two active revolute joints q_1 and q_2 as specified in[33]

$$q_1(t) = -\cos(t - \pi/2),\ q_2(t) = -\cos t. \tag{B1}$$

While enforcing active motion on the revolute joint supplies zero net force in the translation direction, the system successfully moves in space due to motion planning. To maintain consistency with the previous study,[33] the undulation Reynolds number (defined from the peak angular velocity of the hinges and the ellipse chord length, $Re_u = \dot{q}_{max}(2a)^2/\nu$ is adopted and set to be 200. All quantities are scaled by the reference length $2a$ and velocity $2a\dot{q}_{max}$. The computation domain remains the same as in Sec. III A, and the finest grid spacing Δx is set to be 0.005. The initial centroid position of the intermediate ellipse is located at the origin.

Many studies have validated this case, but some literature assumes bodies to be massless. Here, we treat them as neutrally buoyant (i.e., $\rho_s = \rho$), consistent with the configurations reported in Refs. 34 and 60. Therefore, we compare the resulting position and velocity components of the central body over 20 time units to both references in Fig. 10. Note that the downward tilt of the overall trajectory is due to the initial bias of the hinges based on Eq. (B1): one hinge is initially straight, while the other is deflected.

As depicted in Fig. 10, the consistency is noteworthy for the rotation angle and y-direction marching distance, with our study closely aligning with the results of Refs. 34 and 60. However, the x-direction marching distance in our study lies between them. This discrepancy may be attributed to the utilization of different numerical schemes: Ref. 34 employed the vortex particle method in a weakly coupled scheme, whereas Ref. 60 utilized the finite difference method in a monolithic scheme. In general, the overall results exhibit excellent agreement, particularly in the early stages.

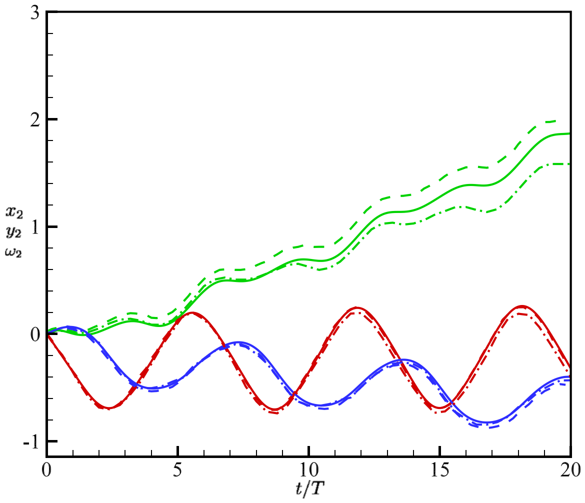


FIG. 10. Time history of centroid position x_2, y_2 and rotation angle ω_2 of the central body: present results (solid), results in Ref. 34 (dashed), and results in Ref. 60 (dotted–dashed). This position plot depicts x_2 in green, y_2 in red, and ω_2 in blue.

APPENDIX C: OPTIMIZATION TEST PROBLEM

For a better understanding of the underlying mechanisms behind transfer learning and constraint handling, we consider a simple test problem based on the problem formulation in Sec. III A. Here, the oscillation amplitudes of both joints A_1 and A_2 to are be optimized and the relative phase difference is fixed at $\pi/2$ as in the validation case. The coefficients A_1, A_2 span a two-dimensional design space corresponding to a range of swimming modes.

The optimization goal is to find an optimum fish mode that maximizes the propulsive efficiency. Thus, the efficiency metric defined in Eq. (21) is employed both for the source and target task. To demonstrate how the algorithm guarantees strictly feasible candidates during evaluations, the following constraint set is imposed: $\mathscr{C} = \{0 \leq A_2 \leq A_1 \leq 1\}$. In terms of computational time, each simulation using LBM to simulate $t/T = 30$ takes approximately 5 min on 18 cores in comparison to the surrogate model which spends less than 1 s.

In the following, we aim to examine the convergence behavior of our proposed algorithm. Figures 11(a) and 11(b) depict the distribution of sampled candidates over six environments for the source task in the original and latent design spaces, respectively. We can find that the V representation transformation essentially applies a certain mapping from the two-dimensional $A_1 - A_2$ plane into the plane defined by $\lambda_1 + \lambda_2 + \lambda_3 = 1$ in the three-dimensional space. Actually, activation functions can also be viewed as a kind of mapping to impose constraints. For instance, softmax function encodes the simplex constraints and sigmoid

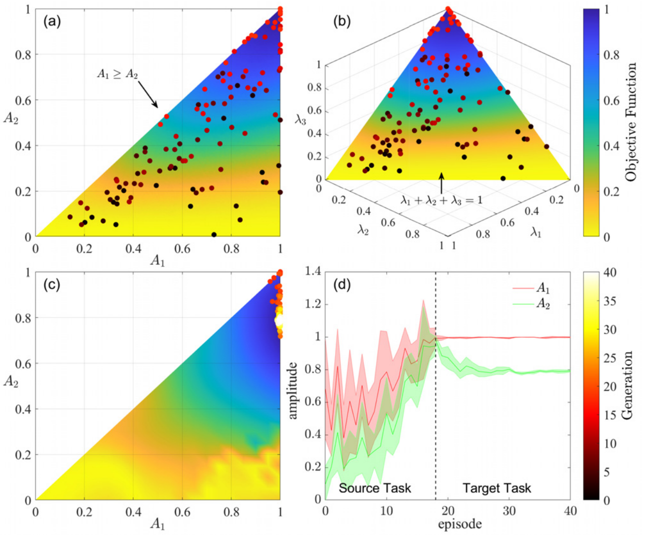


FIG. 11. The trajectory of the sampled points denoted by filled circles in the (a) the original design space for the source task, (b) the latent design space for the source task, and (c) the original design space for the target task. The colors of each circle represent the corresponding episode. The cloud diagram representing the normalized objective function values in the background is interpolated using the scatter grid data. (d) The action convergency of A_1 and A_2 in the whole process. The lines and shaded regions, respectively, denote the mean and standard deviations over six environments of each variable.

function encodes the upper and lower bounds. Therefore, our constraint mapping process can be regarded as a specialized activation function.

Here, Figs. 11(c) and 11(d) show the sampled candidates for the target task and the average convergence during the whole process. Compared with Fig. 11(a), it is noticed that our optimizer explores in a wide range of the design space during the source task, and after CTL, it efficiently marches toward the global optimum during the target task in fewer than 30 episodes. Figure 12 illustrates another optimization attempt conducted without CTL, from which we can observe a significant reduction of episode evaluation by over 30%. Our previous research has shown that sufficient sampling during the exploration phase is crucial to gain an overall understanding of the objective function.[20] We achieved a substantial acceleration in convergency since the training time consumed on the surrogate model is significantly less than that consumed on the LBM solver. This mitigates the significant computational burden associated with the episode-hungry nature of DRL.

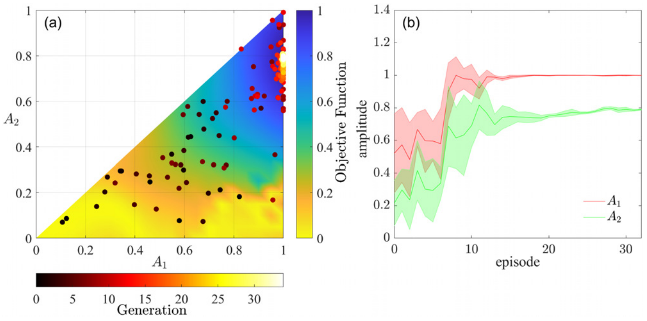


FIG. 12. (a) The trajectory of the sampled points denoted by filled circles in the original design space for the case that directly interacting with the high-cost environment from a random policy. (b) The action convergency of A_1 and A_2 in the whole process. The colors and symbols are consistent with the legend of Fig. 11.

To some extent, training on the source task replaces part of the exploration phase, and the perceived information about the region where the optimum may exist will be imprinted into the agent as a starting policy. From this policy, the agent can quickly find the optimum on the target task, as long as the objective functions of the source and target task share similar qualitative characteristics. Significantly, due to the simplifying assumptions of the surrogate model, the global optimum of the two tasks are qualitatively located in the same region, but quantitatively located at different positions, as shown in the contours of Figs. 11(a) and 11(c). However, by using the variance ratio as a criterion for transfer learning, we succeed preventing overfitting from occurring, since the source task training automatically terminated at an appropriate episode without converging to the maximum value in the upper right corner of Fig. 11(a).

In conclusion, the agent can start from scratch, and good performances can be achieved at the cost of enormous computational load. However, a randomly generated initial policy can be too hard to be optimized and it may never be improved particularly when the objective function is highly nonlinear and multimodal. It is useful to pre-train the policy so that agent can start from a robust and reasonable initial policy. For the following optimization problems in higher-dimensional parameter spaces, we will employ this physics-informed RL framework to save computational time as much as possible.

REFERENCES

[1] M. S. Triantafyllou, G. Triantafyllou, and D. Yue, "Hydrodynamics of fishlike swimming," Annu. Rev. Fluid Mech. **32**, 33–53 (2000).

[2] W. M. Van Rees, M. Gazzola, and P. Koumoutsakos, "Optimal shapes for anguilliform swimmers at intermediate Reynolds numbers," J. Fluid Mech. **722**, R3 (2013).

[3] D. Scaradozzi, G. Palmieri, D. Costa, and A. Pinelli, "BCF swimming locomotion for autonomous underwater robots: A review and a novel solution to improve control and efficiency," Ocean Eng. **130**, 437–453 (2017).

[4] D. Roper, S. Sharma, R. Sutton, and P. Culverhouse, "A review of developments towards biologically inspired propulsion systems for autonomous underwater vehicles," Proc. Inst. Mech. Eng., Part M **225**, 77–96 (2011).

[5] R. W. Whittlesey, S. Liska, and J. O. Dabiri, "Fish schooling as a basis for vertical axis wind turbine farm design," Bioinspiration Biomimetics **5**, 035005 (2010).

[6] M. Lighthill, "Note on the swimming of slender fish," J. Fluid Mech. **9**, 305–317 (1960).

[7] T. Y.-T. Wu, "Hydromechanics of swimming propulsion. Part 1. Swimming of a two-dimensional flexible plate at variable forward speeds in an inviscid fluid," J. Fluid Mech. **46**, 337–355 (1971).

[8] J. Newman, "The force on a slender fish-like body," J. Fluid Mech. **58**, 689–702 (1973).

[9] S. J. Lighthill, *Mathematical Biofluiddynamics* (SIAM, 1975).

[10] D. Terzopoulos, X. Tu, and R. Grzeszczuk, "Artificial fishes: Autonomous locomotion, perception, behavior, and learning in a simulated physical world," Artif. Life **1**, 327–351 (1994).

[11] J. Song, Y. Zhong, H. Luo, Y. Ding, and R. Du, "Hydrodynamics of larval fish quick turning: A computational study," Proc. Inst. Mech. Eng., Part C **232**, 2515–2523 (2018).

[12]J. Song, Y. Zhong, R. Du, L. Yin, and Y. Ding, "Tail shapes lead to different propulsive mechanisms in the body/caudal fin undulation of fish," Proc. Inst. Mech. Eng., Part C **235**, 351–364 (2021).

[13]A. Abouhussein and Y. T. Peet, "Computational framework for efficient high-fidelity optimization of bio-inspired propulsion and its application to accelerating swimmers," J. Comput. Phys. **482**, 112038 (2023).

[14]S. Kern and P. Koumoutsakos, "Simulations of optimized anguilliform swimming," J. Exp. Biol. **209**, 4841–4857 (2006).

[15]M. Gazzola, W. M. Van Rees, and P. Koumoutsakos, "C-start: Optimal start of larval fish," J. Fluid Mech. **698**, 5–18 (2012).

[16]G. Tokić and D. K. Yue, "Optimal shape and motion of undulatory swimming organisms," Proc. R. Soc. B **279**, 3065–3074 (2012).

[17]W. M. Van Rees, M. Gazzola, and P. Koumoutsakos, "Optimal morphokinematics for undulatory swimmers at intermediate Reynolds numbers," J. Fluid Mech. **775**, 178–188 (2015).

[18]J. Viquerat, J. Rabault, A. Kuhnle, H. Ghraieb, A. Larcher, and E. Hachem, "Direct shape optimization through deep reinforcement learning," J. Comput. Phys. **428**, 110080 (2021).

[19]H. Ghraieb, J. Viquerat, A. Larcher, P. Meliga, and E. Hachem, "Single-step deep reinforcement learning for open-loop control of laminar and turbulent flows," Phys. Rev. Fluids **6**, 053902 (2021).

[20]C. Wang, P. Yu, and H. Huang, "Reinforcement-learning-based parameter optimization of a splitter plate downstream in cylinder wake with stability analyses," Phys. Rev. Fluids **8**, 083904 (2023).

[21]M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," J. Mach. Learn. Res. **10**, 1633–1685 (2009).

[22]X. Yan, J. Zhu, M. Kuang, and X. Wang, "Aerodynamic shape optimization using a novel optimizer based on machine learning techniques," Aerosp. Sci. Technol. **86**, 826–835 (2019).

[23]S. Bhola, S. Pawar, P. Balaprakash, and R. Maulik, "Multi-fidelity reinforcement learning framework for shape optimization," J. Comput. Phys. **482**, 112018 (2023).

[24]R. Featherstone, *Rigid Body Dynamics Algorithms* (Springer, 2014).

[25]S. E. Fetherstonhaugh, Q. Shen, and O. Akanyeti, "Automatic segmentation of fish midlines for optimizing robot design," Bioinspiration Biomimetics **16**, 046005 (2021).

[26]Z. Guo, C. Zheng, and B. Shi, "Discrete lattice effects on the forcing term in the lattice Boltzmann method," Phys. Rev. E **65**, 046308 (2002).

[27]X. Yang, X. Zhang, Z. Li, and G.-W. He, "A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations," J. Comput. Phys. **228**, 7821–7836 (2009).

[28]J. Carling, T. L. Williams, and G. Bowtell, "Self-propelled anguilliform swimming: Simultaneous solution of the two-dimensional Navier–Stokes equations and Newton's laws of motion," J. Exp. Biol. **201**, 3143–3166 (1998).

[29]M. Gazzola, P. Chatelain, W. M. Van Rees, and P. Koumoutsakos, "Simulations of single and multiple swimmers with non-divergence free deforming geometries," J. Comput. Phys. **230**, 7093–7114 (2011).

[30]G. Novati, S. Verma, D. Alexeev, D. Rossinelli, W. M. Van Rees, and P. Koumoutsakos, "Synchronisation through learning for two self-propelled swimmers," Bioinspiration Biomimetics **12**, 036001 (2017).

[31]S. Verma, G. Novati, and P. Koumoutsakos, "Efficient collective swimming by harnessing vortices through deep reinforcement learning," Proc. Natl. Acad. Sci. U. S. A. **115**, 5849–5854 (2018).

[32]J. D. Eldredge, "Numerical simulation of the fluid dynamics of 2D rigid body motion with the vortex particle method," J. Comput. Phys. **221**, 626–648 (2007).

[33]J. D. Eldredge, "Dynamically coupled fluid–body interactions in vorticity-based numerical simulations," J. Comput. Phys. **227**, 9170–9194 (2008).

[34]C. Bernier, M. Gazzola, R. Ronsse, and P. Chatelain, "Simulations of propelling and energy harvesting articulated bodies via vortex particle-mesh methods," J. Comput. Phys. **392**, 34–55 (2019).

[35]E. Kanso, J. E. Marsden, C. W. Rowley, and J. B. Melli-Huber, "Locomotion of articulated bodies in a perfect fluid," J. Nonlinear Sci. **15**, 255–289 (2005).

[36]B. Bayat, A. Crespi, and A. Ijspeert, "Envirobot: A bio-inspired environmental monitoring platform," in *IEEE/OES Autonomous Underwater Vehicles (AUV)* (IEEE, 2016), pp. 381–386.

[37]P. Liljebäck and R. Mills, "Eelume: A flexible and subsea resident IMR vehicle," in *Oceans 2017-Aberdeen* (IEEE, 2017), pp. 1–4.

[38]J. Toomey and J. D. Eldredge, "Numerical and experimental study of the fluid dynamics of a flapping wing with low order flexibility," Phys. Fluids **20**, 073603 (2008).

[39]E. Kanso and J. E. Marsden, "Optimal motion of an articulated body in a perfect fluid," in *Proceedings of the 44th IEEE Conference on Decision and Control* (IEEE, 2005), pp. 2511–2516.

[40]J. N. Newman, *Marine Hydrodynamics* (The MIT Press, 2018).

[41]R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, 2018).

[42]K. Fukuda and A. Prodon, "Double description method revisited," in *Franco-Japanese and Franco-Chinese Conference on Combinatorics and Computer Science* (Springer, 1995), pp. 91–111.

[43]J. Tan, Y. Gu, G. Turk, and C. K. Liu, "Articulated swimming creatures," ACM Trans. Graph. **30**, 1–12 (2011).

[44]A. Wiens and M. Nahon, "Optimally efficient swimming in hyper-redundant mechanisms: Control, design, and energy recovery," Bioinspiration Biomimetics **7**, 046016 (2012).

[45]P. W. Webb, "Simple physical principles and vertebrate aquatic locomotion," Am. Zool. **28**, 709–725 (1988).

[46]W. W. Schultz and P. W. Webb, "Power requirements of swimming: Do new methods resolve old questions?" Integr. Comp. Biol. **42**, 1018–1025 (2002).

[47]R. Godoy-Diana, C. Marais, J.-L. Aider, and J. E. Wesfreid, "A model for the symmetry breaking of the reverse Bénard–von Kármán vortex street produced by a flapping foil," J. Fluid Mech. **622**, 23–32 (2009).

[48]D. J. Cleaver, Z. Wang, and I. Gursul, "Bifurcating flows of plunging aerofoils at high Strouhal numbers," J. Fluid Mech. **708**, 349–376 (2012).

[49]M. Gazzola, M. Argentina, and L. Mahadevan, "Scaling macroscopic aquatic locomotion," Nat. Phys. **10**, 758–761 (2014).

[50]X. Lin, J. Wu, and T. Zhang, "Self-directed propulsion of an unconstrained flapping swimmer at low Reynolds number: Hydrodynamic behaviour and scaling laws," J. Fluid Mech. **907**, R3 (2021).

[51]J. Yu, L. Wang, and M. Tan, "Geometric optimization of relative link lengths for biomimetic robotic fish," IEEE Trans. Rob. **23**, 382–386 (2007).

[52]F. Hess and J. Videler, "Fast continuous swimming of Saithe (*Pollachius Virens*): A dynamic analysis of bending moments and muscle power," J. Exp. Biol. **109**, 229–251 (1984).

[53]T. P. Johnson, D. A. Syme, B. C. Jayne, G. V. Lauder, and A. F. Bennett, "Modeling red muscle power output during steady and unsteady swimming in largemouth bass," Am. J. Physiol.-Regul., Integr. Compar. Physiol. **267**, R481–R488 (1994).

[54]C. Wardle, J. Videler, and J. Altringham, "Tuning in to fish swimming waves: Body form, swimming mode and muscle function," J. Exp. Biol. **198**, 1629–1636 (1995).

[55]B. C. Jayne and G. V. Lauder, "Are muscle fibers within fish myotomes activated synchronously? Patterns of recruitment within deep myomeric musculature during swimming in largemouth bass," J. Exp. Biol. **198**, 805–815 (1995).

[56]J. D. Altringham and D. J. Ellerby, "Fish swimming: Patterns in muscle function," J. Exp. Biol. **202**, 3397–3403 (1999).

[57]J. Kajtar and J. Monaghan, "On the swimming of fish like bodies near free and fixed boundaries," Eur. J. Mech.-B **33**, 1–13 (2012).

[58]R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Processing Systems 12* (MIT Press, 1999).

[59]J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv:1707.06347 (2017).

[60]R. Yang, *Monolithic Methods and Versatile Applications for Numerical Fluid-Structure Interaction Studies* (University of California, Los Angeles, CA, 2020).