Data-Driven Control of Insect Flapping Flight via Deep Reinforcement Learning

Tingsong Lu, Qiang Chen*, Yuming Fang*, *Fellow, IEEE*, Camille Le Roy, Xiaogang Jin, *Member, IEEE*, Zhigang Deng, *Senior Member, IEEE*

***Abstract*—Modeling and simulating realistic insect flight pose unique challenges due to the complex interaction between multi-degree-of-freedom wing kinematics and highly precise aerodynamic forces. To solve this challenge, this paper presents a bidirectional kinematics–aerodynamics coupled simulation framework for miniature insect flight. Our approach first models the kinematics of flying insects by parameterizing natural wingbeat cycles based on available real-world datasets. Subsequently, we compute aerodynamic forces utilizing an improved semi-empirical model, which extends from quasi-steady formulation by incorporating critical unsteady force components. To achieve closed-loop control for both kinematics and aerodynamics, we employ deep reinforcement learning to train a virtual insect to adaptively adjust flapping strategies in response to dynamic flight states. Finally, an integrated controller enables the simulated insect to autonomously regulate the wing motion and perform complex tasks such as visual obstacle avoidance. Extensive experiments and comparisons demonstrate that our framework can effectively generate physically plausible and autonomous insect flight across a variety of scenarios.**

***Index Terms*—Insect simulation, flapping flight, physics-based simulation, data-driven, deep reinforcement learning, kinematics-aerodynamics coupling.**

I. Introduction

REALISTIC simulation of living organisms, particularly flying creatures, has numerous potential applications such as virtual reality, entertainment, digital ecological simulation, and biodiversity modeling; but it remains a profoundly challenging yet fascinating research topic. In recent years, significant efforts have been devoted to simulating a wide range of flying species, from larger vertebrates including birds [1], [2] and bats [3], to smaller insects such as butterflies [4] and fruit flies [5].

To achieve realistic simulations of animal flight, prior research has proposed various approaches for modeling wing kinematics and aerodynamics. Early methods globally optimized wingbeat kinematics offline under a quasi-steady aerodynamic model to generate bird flight that follows a predefined trajectory [1]. Later, based on a quasi-steady aerodynamic algorithm, researchers proposed a data-driven paradigm to learn a continuous controller from motion capture data, enabling the simulation of a bird that autonomously matches its wing beats from training data [2]. However, this approach is highly dependent on the quality and diversity of the captured data, which limits its ability to generalize to maneuvers that are underrepresented in the training set. Building upon this foundational work, subsequent studies extended the application of these quasi-steady aerodynamic models to simulate the aerodynamics of butterfly flight, updating wingbeat parameters exclusively via manually designed functions [4]. However, the hand-crafted formulation may produce flight behaviors that deviate from physical reality, such as allowing an insect to ascend continuously without wing flapping. Furthermore, these approaches suffer from a common limitation: they rely on predefined control paradigms - either through global optimization for a specified path [1], match within a captured motion database [2], or manually designed laws [4]. As such, the resulting insect lacks a general-purpose learning mechanism to autonomously discover and adapt high-level flight strategies in complex or novel environments.

Compared to avian flight, insect flight presents additional challenges. First, the tiny size of insects renders the acquisition of comprehensive motion data, particularly accurate wing trajectories during flapping, highly challenging. Second, insect aerodynamics are predominantly governed by unsteady flow effects; consequently, the sole application of quasi-steady aerodynamic models is insufficient to account for the force generation required for realistic insect flight [6], [7].

To date, autonomously generating adaptive control policies for the tightly coupled kinematics-dynamics system underlying flapping insect flight remains a significant challenge, primarily due to the following reasons: (i) The aerodynamics in insect flight are strongly influenced by unsteady phenomena, making simplified quasi-steady approximations inadequate for accurate force prediction. (ii) The inherent requirement to tackle a high-dimensional, nonlinear control problem is induced by the tight bilateral coupling between multi-degree-of-freedom wing kinematics and the resulting unsteady aerodynamic forces.

To address these challenges, in this paper we propose a simulation framework for adaptive insect flight that emphasizes bidirectional coupling between kinematics and dynamics. Specifically, utilizing available insect wingbeat datasets, we first parameterize natural flapping cycles to generate physically plausible wing kinematics. We then compute the aerodynamic forces using an improved semi-empirical aerodynamic model, which extends the quasi-steady framework by incorporating

Tingsong Lu, Qiang Chen, and Yuming Fang are with the School of Computing and Artificial Intelligence, Jiangxi University of Finance and Economics, Nanchang 330013, China (e-mail: lutingsong@foxmail.com, qiangchen@jxufe.edu.cn, fa0001ng@e.ntu.edu.sg).

Camille Le Roy is with the Experimental Zoology Group, Wageningen University & Research, 6709 PG Wageningen, Netherlands (e-mail: camille.leroy@wur.nl).

Xiaogang Jin is with the State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310027, China (e-mail: jin@cad.zju.edu.cn).

Zhigang Deng is with the Department of Computer Science, University of Houston, Houston TX 77204, United States (e-mail: zdeng4@central.uh.edu).

*Corresponding author.

two critical unsteady force components capturing unsteady effects. The model's semi-empirical parameters are further refined using experimentally measured force data and computational fluid dynamics (CFD) simulation to enhance accuracy. Finally, we train insect agents to learn the kinematics-dynamics coupled control policy via deep reinforcement learning. Within the simulation environment, the virtual insect agents autonomously explore and learn the complex mapping between wing kinematics and flight dynamics. Extensive simulation experiments and comparisons demonstrate that our method can generate lifelike, autonomous insect flight motion across a variety of scenarios.

The main contributions of this work are summarized as follows:

- We propose an improved semi-empirical aerodynamic model for insect flight that incorporates two principal unsteady force mechanisms and leverages empirical data optimization to improve the accuracy of force prediction.
- We present a kinematics–dynamics coupled framework for insect flight that enables closed-loop control. Within this framework, the simulated insect can adaptively adjust its flapping strategies and perform complex flight tasks.

II. Related Work

In recent decades, researchers have investigated a wide range of motion control methodologies to animate virtual characters. Most studies have focused on walking animals including bipeds [8], [9] and quadrupeds [10], [11]. Flying creatures such as birds [1], [2] and insects [4], [12] have also been studied. The explored approaches encompass manually designed feedback control laws [4], data-driven control strategies [2], [8], and learning-based control methods [13], [14].

Flapping Wing Flight Simulation. Many physically based modeling and simulation approaches have been developed for a wide range of flapping flying organisms, including birds [1], [2], bats [3], [15], dragonflies [16], butterflies [4], [17], and honeybees [12]. These efforts aim to synthesize visually realistic and physically plausible flapping-wing flight behaviors, often through a combination of kinematic modeling and aerodynamic forces computing via a quasi-steady aerodynamic model. One of early representative works computed the wingbeat by physical principles to synthesize bird flight animations [1]. To enhance bird's wing kinematics realism, subsequent work incorporated motion capture data of real bird flights, allowing for the synthesis of more natural wing motion [2].

Inspired by the success of avian flight modeling, similar approaches have been adapted to simulate insect flight. Chen et al. [4] simulated butterfly wing cycle motions using simple cosine functions and computed aerodynamic forces via a quasi-steady model, although it lacks biomechanical fidelity, especially in replicating real butterflies' complex wing motion characteristics. Researchers proposed a bio-inspired model by integrating body roll control with quasi-steady aerodynamics to replicate realistic bee flight [12]. Unlike birds, insects fly in low Reynolds number regimes, where unsteady aerodynamic phenomena, such as additional mass and rotational circulation, are critical in the generation of aerodynamic forces. Consequently, the application of quasi-steady aerodynamic models to insect flight may result in inaccurate force predictions and unrealistic representations of flight dynamics. More importantly, these approaches are fundamentally constrained by their inability to support generalized learning, which precludes the autonomous discovery and adaptive optimization of high-level flight policies in novel environments.

Compared to previous work [1], [2], [4], [12], our approach differs in two fundamental aspects: (i) It augments the quasi-steady aerodynamic model by introducing two critical unsteady force terms, which better capture the unsteady aerodynamic effects essential for accurate insect flight simulation. (ii) It facilitates the autonomous learning of a generalizable control policy that dynamically maps high-dimensional flight states to optimal wing kinematics through continuous interaction with the environment.

Deep Reinforcement Learning for Control. In recent years, deep reinforcement learning (DRL) has emerged as a powerful framework for the simulation and control of physically-based agents, particularly in the domains of virtual character animation [18], [19]. Early reinforcement learning applications primarily addressed the design of virtual character controllers using motion capture data [20], as well as motion planning for physically simulated bipedal agents [21]. With the advent of deep neural networks, DRL has enabled scalable function approximation of both control policies and value functions, allowing agents to learn complex motor behaviors in high-dimensional, continuous environments [22]. DRL has proven effective in controlling physically simulated legged locomotion, enabling agents to acquire adaptive walking skills for planar bipeds and quadrupeds [23]. This capability is further extended to three-dimensional biped locomotion through a hierarchical control structure [24], where a high-level controller plans long-term goals such as footstep targets, while a low-level controller executes immediate motor actions to maintain stability and achieve these goals. In addition to applications in humanoid control and quadruped locomotion, DRL has also been utilized for learning control policies in flapping-wing flight [13] and aerobatic maneuvers of imaginary dragon [14].

Although prior studies [13], [14] employ DRL to simulate wing flapping motion, their objectives differ markedly from ours. In their earlier work [13], DRL is primarily applied to example-guided motor skill acquisition, where user-provided keyframes guide the learning of diverse flight behaviors through an evolutionary-aided exploration strategy. This line of research is subsequently extended in [14], where DRL is combined with self-regulated learning to achieve aerobatic maneuvers for an imaginary dragon by adaptively adjusting the sub-goals and rewards during training. In contrast, our work leverages DRL to learn autonomous control policies for realistic insect flapping flight by exploiting the coupling between kinematics and dynamics.

III. Overview of Our Approach

Our method is composed of three inter-related modules: kinematics modeling, dynamics computation, and kinematics-

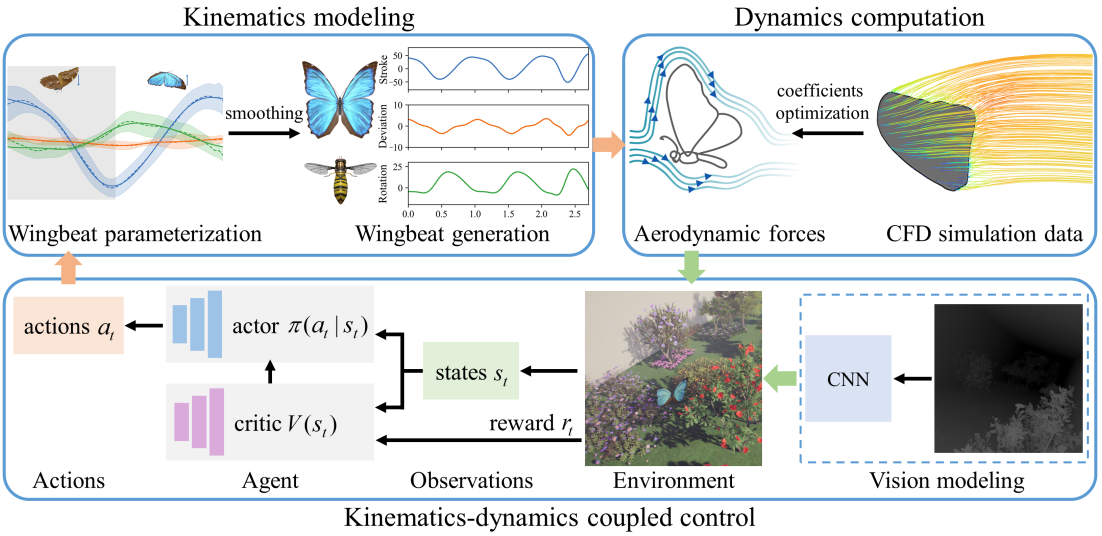


Fig. 1. Schematic view of our approach.

dynamics coupled control. Fig. 1 illustrates the pipeline of our approach, while Table I summarizes the main symbols and their definitions used in this paper. A brief overview of each module is provided below.

Kinematics modeling. We construct an insect mesh model rigged with a hierarchical hinge structure to animate wing flapping, parameterizing natural wingbeats as smoothly conjunctive cycle motions based on real-world datasets. Moreover, we create a wing flapping generator driven by frequency and amplitude parameters to simulate continuous and periodic wingbeats.

Dynamics computation. In addition to quasi-steady forces employed in the semi-empirical aerodynamic model, we systematically incorporate two unsteady forces to capture the intrinsic unsteady effects inherent to wingbeats. Subsequently, we construct CFD experimental datasets to optimize the coefficients of this aerodynamic model. Finally, leveraging the proposed aerodynamic model, we achieve control over insect motion through integration with wing-body coupling mechanisms.

Kinematics-dynamics coupled control. We model the vision-based perceptual ability of the insect using a virtual camera sensor, and then further construct a policy to train the insect in visually guided flight tasks. Leveraging real insect motion datasets, we map wing kinematics to aerodynamic dynamics and invert this mapping via DRL to enable real-time maneuvering control.

TABLE I
MAIN SYMBOLS AND DESCRIPTIONS USED IN THIS PAPER.

Symbols	Descriptions	Unit
ϕ	roll angle	deg
θ	pitch angle	deg
ψ	yaw angle	deg
β	stroke angle	deg
ζ	deviation angle	deg
η	rotation angle	deg
A	wingbeat amplitude	deg
f	wingbeat frequency	Hz
$\mathbf{v}$	velocity	m/s
$\mathbf{a}$	acceleration	m/s^2
$\boldsymbol{\omega}$	angular velocity	rad/s
α	angle of attack	deg
ρ	air density	kg/m^3

IV. Kinematics Modeling

We construct 3D models of two representative insect species with distinct flapping mechanisms: (i) the blue Morpho butterfly (*Lepidoptera*), characterized by low-frequency, large-amplitude wingbeats using two pairs of broad wings, and (ii) the hoverfly (*Diptera*), which exhibits high-frequency, small-amplitude wingbeats with a single pair of narrow, elongated wings. The wing morphological and kinematic characteristics of the blue Morpho and hoverfly, including wingbeat frequency and amplitude, are reported in [25]–[27]. Each insect model, aligned with physiological structures, comprises four components: head, thorax, abdomen, and wings (shown in Fig. 2), where wings are connected to the thorax via complex biological hinges [28]. Inspired by the work of [17], a hierarchical hinge structure is designed to drive flapping motion, as illustrated in Fig. 2. As shown in Fig. 3, each wing's flapping motion is defined by three rotational angles: stroke, deviation, and rotation. Concurrently, the insect's body exhibits six degrees of freedom in motion, with its orientation described by yaw, pitch, and roll angles.

A. Wingbeat Parameterization

The wingbeat of an insect can be generally decomposed into three time-varying angular components: stroke angle β, deviation angle ζ, and rotation angle η (shown in Fig. 3(a)). Inspired by previous work [29], we construct a natural wingbeat cycle by extracting the average motion from multiple wingbeat cycles and fitting it using a Fourier series. This fitted result is defined as the baseline wingbeat cycle. Capitalizing on the Fourier series' exceptional ability to capture periodic patterns, we truncate the expansion after the third harmonic

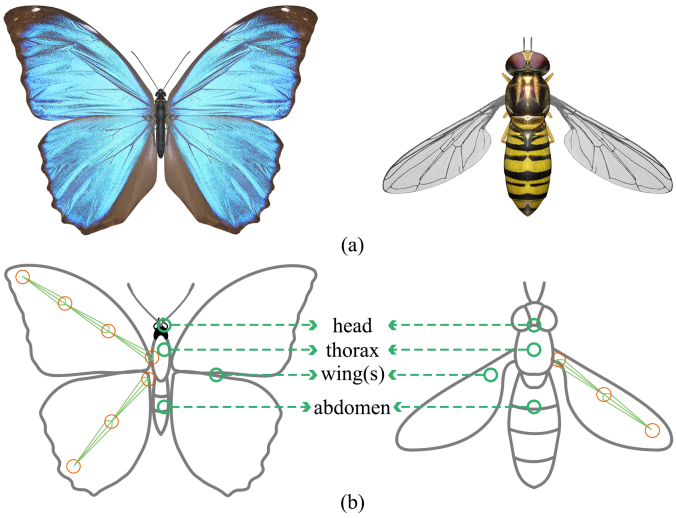


Fig. 2. Schematic illustration of two 3D insect models and their anatomical structures. (a) shows the constructed 3D models for two selected insects: a blue Morpho butterfly and a hoverfly. (b) shows the insects' anatomical structures and the hierarchical hinges rigged with their wings.

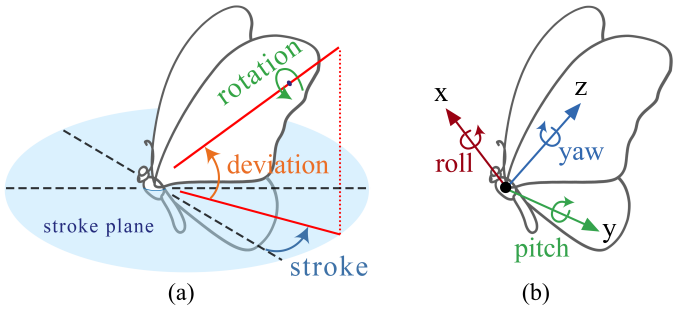


Fig. 3. (a) illustrates the three flapping angles defined in the wing coordinate system; (b) depicts the three body posture angles in the body coordinate system.

to achieve a pragmatic compromise between reconstruction accuracy and parametric efficiency. The angles of the baseline wingbeat cycle are expressed as a periodic function $W(t) = a_0 + \sum_{n=1}^{3} [a_n \cos(2\pi nt) + b_n \sin(2\pi nt)]$, where $W(t)$ denotes the wingbeat angle, a_0 is the mean wing angle over a cycle, a_n and b_n are the Fourier coefficients of the n-th harmonic, and t is dimensionless time normalized to one wingbeat cycle. Using the blue Morpho butterfly datasets [30] as an example, Fig. 4 demonstrates the fitting process for baseline wingbeat cycles, including stroke, deviation, and rotation angles.

To generate dynamically varying wingbeat motions, we introduce a frequency–amplitude modulation mechanism, which generalizes the baseline wingbeat cycle by adjusting its frequency and amplitude. The extended wingbeat motion is expressed as follows:

$$W^*(A^*, f^*, t) = a_0 + \frac{A^*}{A_b} \sum_{n=1}^{3} \left[a_n \cos\left(2\pi n \frac{f^*}{f_b} t\right) + b_n \sin\left(2\pi n \frac{f^*}{f_b} t\right) \right], \quad * \in \{\beta, \zeta, \eta\} \quad (1)$$

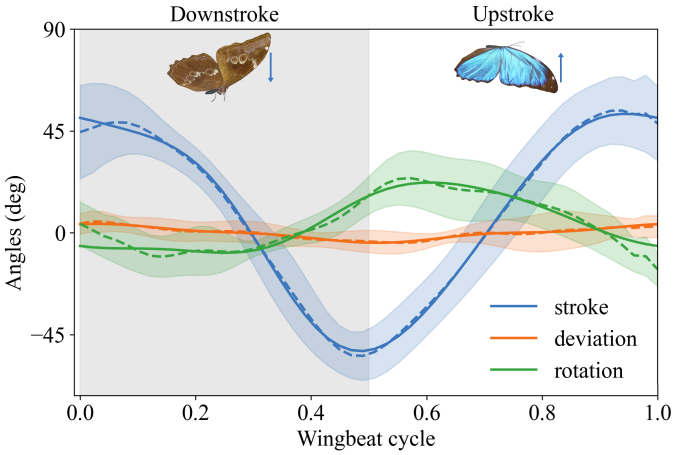


Fig. 4. Baseline wingbeat cycles of stroke, deviation, and rotation motion using Fourier series fitting. The dotted lines and the shadow regions denote the mean values and standard deviations of real wingbeat data, respectively.

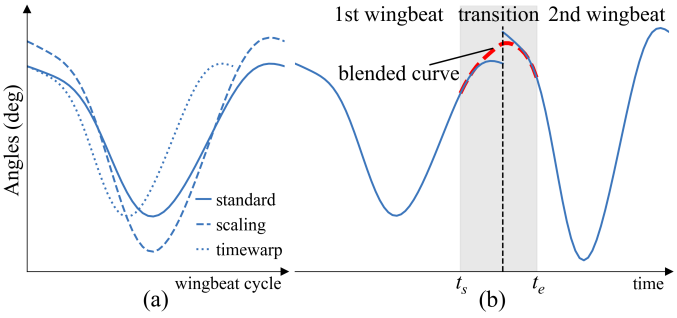


Fig. 5. (a) illustrates the generation of diverse flapping cycles by applying amplitude scaling and frequency time-warping to a baseline wingbeat cycle; (b) demonstrates the smooth transition between consecutive wingbeat cycles.

where A_b and f_b denote the amplitude and frequency of the baseline wingbeat cycle; A^* and f^* represent the varying amplitude and frequency for three angular components that vary over time, i.e. stroke angle β, deviation angle ζ and rotation angle η, respectively. Fig. 5(a) shows how variations in A^* and f^* affect the resulting flapping motion within a cycle.

However, when the wingbeat amplitude varies continuously, discontinuities may occur between adjacent flapping cycles. To mitigate this, we introduce a sigmoid-based transition function that ensures smooth interpolation of amplitude across successive wingbeats. Specifically, for a transition from the i-th to the $(i+1)$-th wingbeat over the interval $[t_s, t_e]$, the smoothed amplitude is computed as follows:

$$A^*(t) = A^*{}_i + (A^*_{i+1} - A^*_i) \frac{1}{1 + \exp\left[-\gamma\left(\frac{t - t_s}{t_e - t_s} - \frac{1}{2}\right)\right]}, \quad (2)$$

where A_i and A_{i+1} denote the amplitudes of the i-th and $(i+1)$-th wingbeats, respectively; $* \in \{\beta, \zeta, \eta\}$ denotes the angular components; γ is a smoothing factor that controls the transition sharpness; t_s and t_e represent the start and end times of the transition interval. Unlike the method in [1], which requires recomputing the motion twice within the transition region to achieve smooth flapping, our method eliminates the

need for forward recomputation and facilitates more efficient and seamless blending between consecutive wingbeat cycles (shown in Fig. 5(b)).

V. Dynamics Computation

Insect flight dynamics is primarily governed by aerodynamic forces generated through wingbeat kinematics. In order to model the aerodynamic force accurately, we propose a comprehensive aerodynamic model by combining: (i) quasi-steady forces modeled with the blade element model [31], [32], and (ii) unsteady contributions from added mass effects and rotational circulation force. The aerodynamic coefficients are optimized using empirical datasets including visually measured experimental data [33] and our CFD simulation dataset (refer to the supplemental document) to improve the accuracy of aerodynamic force prediction. Subsequently, the torques generated by wing dynamics are applied to the body to induce yaw, pitch, and roll motions, thereby achieving dynamic coupling between the insect's wings and body.

A. Aerodynamic Modeling

Following the blade element method [31], [32], a wing is discretized into a series of equally spaced spanwise blade elements, as illustrated in Fig. 6(a), with each element modeled as a two-dimensional airfoil. Aerodynamic forces and torques acting on each element are then computed using a quasi-steady aerodynamic model. The total aerodynamic force and torque exerted on the flapping wing are subsequently obtained by integrating the contributions from all blade elements.

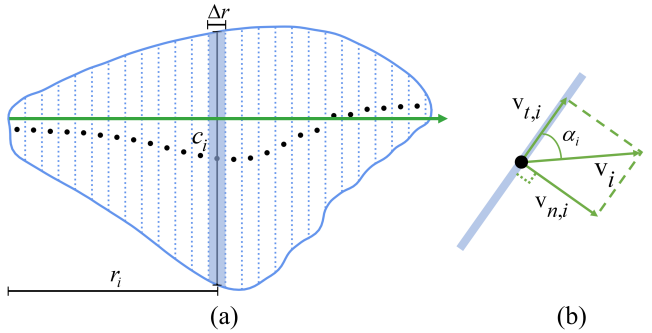


Fig. 6. (a) The i-th blade element is located at radial distance r_i from the wing base, with chord length c_i and spanwise width Δr. Its half-chord point (black circle) is taken as the representative location for evaluating aerodynamic forces. (b) The angle of attack α is defined as the arctangent of the ratio between the normal velocity component $\mathbf{v}_n$ and the tangential velocity component $\mathbf{v}_t$.

Quasi-steady force. Based on quasi-steady aerodynamic theory [34], [35], the simplified aerodynamic forces generated by insect wingbeat motion primarily consist of lift and drag forces. Inspired by previous studies of bird flapping simulations [1], [2] and insect wingbeat simulations [4], [17], we model the instantaneous lift and drag forces acting on the i-th blade element as follows:

$$\begin{cases} F_{l,i} = \dfrac{1}{2}\rho c_i \|\mathbf{v}_i\|^2 C_{l,i}(\alpha_i)\Delta r, \\ F_{d,i} = \dfrac{1}{2}\rho c_i \|\mathbf{v}_i\|^2 C_{d,i}(\alpha_i)\Delta r, \end{cases} \tag{3}$$

where $F_{l,i}$ and $F_{d,i}$ are the translational lift and drag forces for the i-th blade element, respectively; c_i and Δr are the chord length and spanwise width of the i-th blade element, respectively; ρ is an air density constant; and $\mathbf{v}_i$ is the air velocity over the surface of the i-th blade element.

Different species of flying insects exhibit different lift and drag coefficients due to interspecific variations in wing morphology and kinematic patterns [7]. In addition, intraspecific differences in wing geometry, such as shape and aspect ratio, can also lead to some variations in these aerodynamic coefficients [36]. Inspired by prior studies [6], [33], the lift coefficients $C_l(\alpha)$ and drag coefficients $C_d(\alpha)$, which vary with the angle of attack α, are commonly expressed in terms of two semi-empirical coefficients. The functional forms are given by:

$$\begin{cases} C_{l,i}(\alpha_i) = C_{P_\alpha} \sin\alpha_i \cos\alpha_i, \\ C_{d,i}(\alpha_i) = C_{P_\alpha} \sin^2\alpha_i + C_{D_0}, \end{cases} \tag{4}$$

where C_{P_α} represents the sensitivity of the quasi-steady pressure force coefficient to the angle of attack, defined as the derivative of the pressure coefficient C_P with respect to α at $\alpha = 0$; the drag coefficient offset C_{D_0} is equal to the drag coefficient at $\alpha = 0$. And α_i is the i-th blade element's angle of attack, which can be computed as follows:

$$\alpha_i = \tan^{-1}\left(\frac{\|\mathbf{v}_{n,i}\|}{\|\mathbf{v}_{t,i}\|}\right), \tag{5}$$

where $\mathbf{v}_{n,i}$ and $\mathbf{v}_{t,i}$ are the components of the air velocity along the normal of the i-th blade element surface and along the tangent direction, respectively (illustrated in Fig. 6(b)). Fig. 7 plots the relation between the angle of attack α and the coefficients $C_l(\alpha)$ and $C_d(\alpha)$.

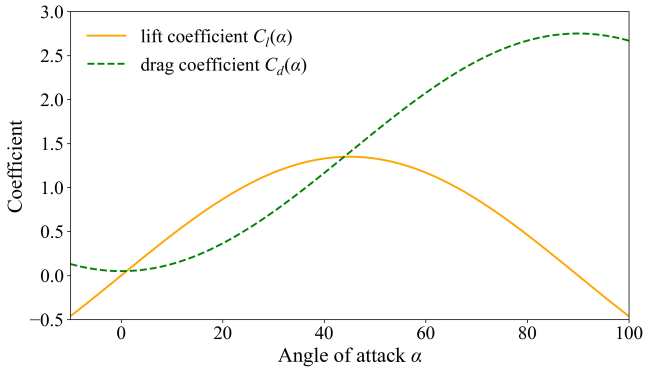


Fig. 7. Empirically fitted lift coefficients $C_l(\alpha)$ and fitted drag coefficients $C_d(\alpha)$ plotted against the wing's local angle of attack α.

Unsteady force. While quasi-steady aerodynamic models have proven effective in capturing the general trends of force generation during insect flapping flight, they inherently neglect key unsteady mechanisms that become prominent during rapid wing accelerations and stroke reversals [37]. In particular, the dynamic interaction between the wing and the surrounding fluid leads to transient forces that cannot be explained by steady-state assumptions alone. To address these limitations, we incorporate two major unsteady aerodynamic components:

the added mass force and the rotational circulation force [38], [39].

Added mass force. Due to the superposition of the wing's instantaneous acceleration with the ambient flow field, the added mass force arising in a real fluid can be effectively approximated using inviscid flow theory, as described in classical potential flow formulations [40], [41]. Classical inviscid flow theory [42], [43] predicts the surface-normal component of the added mass force acting on a thin flat plate with chord length c and spanwise width Δr as:

$$F_{a,i} = -\frac{\pi}{4}\rho c_i^2 \|\dot{\mathbf{v}}_{n,i}\|\Delta r, \tag{6}$$

where $F_{a,i}$ is the added mass force for the i-th blade element; and $\|\dot{\mathbf{v}}_{n,i}\|$ is the magnitude of the acceleration along the normal of the i-th blade element surface.

Rotational circulation force. The rotational circulation force arises during the rapid rotational motion of insect wings, particularly near stroke reversal. When the wing rotates around its leading edge at high angular velocities, it generates a rotational circulation that is distinct from the quasi-steady circulation induced by translational motion [39]. Inspired the previous works [6], [44], the rotational circulation force can be defined as follows:

$$F_{r,i} = \rho c_i^2 \dot{\eta}_i \|\mathbf{v}_{n,i}\| C_r \Delta r, \tag{7}$$

where $F_{r,i}$ is the rotational circulation force for the i-th blade element; and $\dot{\eta}_i$ is the angular velocity that represents the time derivative of the wing's rotation angle. The coefficient C_r represents the rotational circulation contribution, which in the classical theory is defined as $C_r = \pi(3/4 - \hat{x}_0)$, where $\hat{x}_0$ is the normalized position of the rotational axis. In practice, C_r should be regarded as an semi-empirical parameter and optimized against experimental data, as it may vary with wing morphology.

Resultant forces and torques. Let $\mathbf{F}_{ji} = \sum_{*\in\{l,d,a,r\}} F_{*,i} \cdot \hat{\mathbf{n}}_{*,i}$ be the resultant aerodynamics force acting on the i-th blade element of the j-th wing, where $\hat{\mathbf{n}}_*$ represents the unit direction vector associated with each force component. Then the instantaneous force and torque acting on the skeleton from the j-th wing are:

$$\mathbf{F}_j = \sum_i \mathbf{F}_{ji}, \quad \text{and} \quad \boldsymbol{\tau}_j = \sum_i \mathbf{r}_{ji} \times \mathbf{F}_{ji}, \tag{8}$$

where r_{ji} is the moment arm from the center of body mass to the center of the i-th blade element of the j-th wing. Additionally, the resultant aerodynamic force $\mathbf{F}^r = \sum_j \mathbf{F}_j$ serves as the primary driver of the insect's locomotion, facilitating forward velocity generation and position updates. The corresponding resultant aerodynamic torque $\boldsymbol{\tau}^r = \sum_j \boldsymbol{\tau}_j$ induces rotational motion about the body axes, thereby modulating the insect's orientation through roll, pitch, and yaw control.

B. Model Coefficients Optimization

As described above, our aerodynamic model involves three semi-empirical coefficients (i.e., C_{P_α}, C_{D_0}, and C_r). Instead of assigning these coefficients empirically, which may result in a reduced accuracy, we determine their values through particle swarm optimization (PSO). In this procedure, experimental aerodynamic datasets are employed as the ground-truth reference. The discrepancy between the experimental measurements and our aerodynamic model predictions is quantified using the mean squared error (MSE) metric ϵ, defined as follows:

$$\epsilon = \frac{1}{N_c N_s} \sum_{c=1}^{N_c} \sum_{s=1}^{N_s} \|\mathbf{F}_{c,s}^{exp} - \mathbf{F}_{c,s}^{r}\|^2, \tag{9}$$

where $\mathbf{F}_{c,s}^{exp}$ and $\mathbf{F}_{c,s}^{r}$ denote the experimental aerodynamic force and the resultant force predicted by our aerodynamic model, respectively, at time sample s within wingbeat cycle c. N_c is the total number of cycles, and N_s is the number of discrete time samples per cycle.

Using our CFD simulation dataset as a case study, the PSO algorithm is configured with a swarm size of 100 particles and a maximum of 500 iterations to ensure adequate convergence toward the optimum. The algorithm iteratively refines the coefficients with the objective of minimizing ϵ. In our experiments, convergence is achieved after approximately 370 iterations, yielding the optimal parameter set: $C_{P_\alpha} = 2.61$, $C_{D_0} = 0.09$, and $C_r = 0.41$, which resulted in a minimum error of $\epsilon = 0.0205$.

C. Wing-body Coupling

In insect flight, body roll and yaw motions are induced by torque asymmetries between the left and right wings, while asymmetries between the upstroke and downstroke phases within each wingbeat cycle give rise to periodic pitch oscillations. The rotational dynamics of the body under aerodynamic torques are described by the Euler's equations of motion in classical mechanics. To explicitly solve for body angular accelerations, the equations are reformulated as:

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}\left(\boldsymbol{\tau}^r - \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega})\right), \tag{10}$$

where $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\omega}}$ are the angular velocity and angular acceleration about body axes, respectively. Specifically, the angular velocity vector is defined as $\boldsymbol{\omega} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^{\mathrm{T}}$, where ϕ, θ, and ψ denote the roll, pitch, and yaw angles of the body, respectively (as illustrated in Fig. 3(b)). Inspired by the work [45], we calculate the moment of inertia $\mathbf{I}$ for the body as a prolate spheroid, the moment of inertia of the prolate spheroid about the primary axes is:

$$\mathbf{I} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad \text{and} \quad \begin{cases} I_x = \frac{m}{5}(b_w^2 + b_d^2), \\ I_y = \frac{m}{5}(b_l^2 + b_d^2), \\ I_z = \frac{m}{5}(b_l^2 + b_w^2), \end{cases} \tag{11}$$

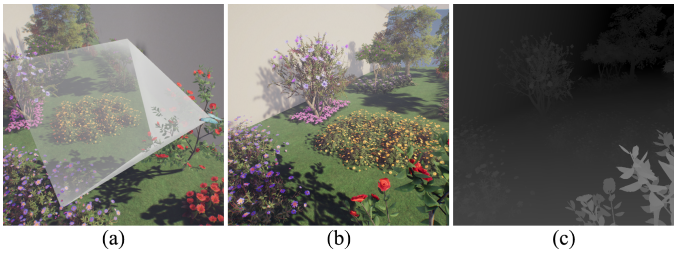
where I_x, I_y and I_z are the component of inertia along the x, y, and z axes, respectively; m is the total mass of the insect; b_l, b_w and b_d represent the length, width, and depth of the insect body, respectively. In addition, wings exhibit passively coupled kinematics that follow the insect's body rotations as a result of their hinge attachment to the thorax.

VI. Kinematics-Dynamics Coupled Control

In control systems, the dynamics of a system are regulated by adjusting its kinematic inputs to achieve desired outcomes. Specifically, insect flight dynamics are generated by wing kinematics, and wing kinematics parameters can be inferred backwards in order to achieve the desired dynamics. Therefore, we map inputs (wing kinematics) to outcomes (aerodynamic dynamics) and invert this mapping for real-time control via deep reinforcement learning. Specifically, in order to better approximate real-world situations, we first model the visual system of the insect and then construct a policy for deep reinforcement learning.

A. Vision Modeling

Most insects rely on compound eyes as their primary visual sensory organs, which facilitate critical behavioral responses such as obstacle avoidance through the perception of depth. Therefore, we use a virtual camera (eye) sensor to model the eyes and render a square lattice of 64×64 pixels with an adjustable field and depth of view (see Fig. 8). The visual input is first processed by a convolutional visual module, and then fed into a policy network (as illustrated in Fig. 9).



(a) (b) (c)

Fig. 8. (a) The camera sensor's view frustum is employed for the simplified modeling of the insect visual field. (b) The vision images from the viewpoint of the insect in flight. (c) The vision depth image is generated from the z-buffer data of the rendering pipeline.

B. Policy Construction

The policy is designed to enable physically simulated insect agents to perform controlled flapping flight while achieving specific objectives. The training procedure is organized into three levels of increasing complexity: beginner, intermediate, and expert. At the beginner level, the agent learns to maintain stable flight and basic maneuvering skills, such as forward flight, climbing, gliding, and turning, by interacting with the environment. At the intermediate level, the policy aims to imitate natural insect flight by learning from reference insect motion data presented as target positions and orientations, allowing it to reproduce biologically realistic trajectories and kinematic patterns. At the expert level, the policy is augmented with a visual perception module, enabling the agent to perceive the environment and learn obstacle avoidance behaviors while maintaining insect-like flight characteristics. Across all levels, the policy maps observed states to control actions that drive the agent toward the desired motion and task objectives.

States and Actions. Each state $s_t \in \mathcal{S}$ (the state space) encapsulates the environmental information perceived by an insect agent at time step t, and is formally represented as a tuple $(\mathbf{d}_t, b_t, \mathbf{v}_t, \mathbf{a}_t, \mathcal{I}_t)$. $\mathbf{d}_t$ is the displacement vector from the insect agent's position to the goal position; b_t is the agent's body posture (i.e., body angles); $\mathbf{v}_t$, $\mathbf{a}_t$ are the velocity and acceleration of agent, respectively; and $\mathcal{I}_t$ is a visual observation vector at current step t. Each action $a_t \in$ action space $\mathcal{A}$ specifies a control input through which the insect agent interacts with the environment. In the virtual environment, insect flight is driven exclusively by aerodynamic forces generated through flapping wing kinematics. Therefore, the action a_t can be expressed as a tuple $(A_t^l, A_t^r, f_t^l, f_t^r)$, where A_t^l, A_t^r denote the wingbeat amplitude of left wings and right wings, respectively, and f_t^l and f_t^r represent their corresponding wingbeat frequencies at time step t.

Reward. Reward r is designed to encourage insect-like flight behavior while maintaining balance, avoiding collisions, and remaining consistent with biological observations. The total reward at time step t is defined as follows:

$$r_t = w_t^g r_t^g - w_t^k r_t^k - w_t^d r_t^d - w_t^b r_t^b - w_t^c r_t^c, \tag{12}$$

where each w_t denotes a learnable weight that balances the contribution of the corresponding reward component. These reward weights terms are initialized to predefined values and jointly optimized with the policy network by maximizing the reinforcement learning objective via gradient-based updates. To prevent the learnable weights from becoming negative or collapsing to zero during training, we incorporate two mechanisms. First, each weight is parameterized using a Softplus transformation: $w_t = \log(1 + \exp(\hat{w}_t))$, where $\hat{w}_t$ is an unconstrained learnable parameter. This ensures that all reward weights remain strictly positive throughout training, preserving the intended penalizing effect of the negative reward terms. Second, we apply L2 regularization to the weight parameters, which encourages them to stay near their initial values while still allowing adaptive balancing during learning. The r_t^g is a positive task reward, while r_t^k, r_t^d, r_t^b, and r_t^c represent penalty terms that penalize undesirable behaviors such as unrealistic kinematics, excessive dynamics, or instability. The goal reward r_t^g encourages the insect agent to fly towards its goal position:

$$r_t^g = \exp\left(-\frac{\|\mathbf{d}_t\|^2}{2\delta^2}\right), \tag{13}$$

where $\mathbf{d}_t$ is the relative displacement vector from the agent's current position to the target position, and δ controls the spatial extent of the reward influence. To ensure that the agent produces biologically realistic wingbeat patterns, we define the kinematic reward r_t^k that penalizes deviations from the empirical distributions of the wingbeat frequencies and amplitudes observed in a real dataset of flying insects. The kinematic reward is given by:

$$r_t^k = \left(\frac{A - \mu_A}{\sigma_A}\right)^2 + \left(\frac{f - \mu_f}{\sigma_f}\right)^2, \tag{14}$$

where A and f denote the agent's wingbeat amplitude and frequency, respectively; μ_A, μ_f and σ_A, σ_f are the corresponding means and standard deviations of a real dataset. The kinematic reward encourages the agent's wingbeat patterns to converge toward averaged movements, consistent with the observation

that insects maintain stable amplitudes and frequencies during flight to optimize energy efficiency. In contrast to the relatively stable kinematic parameters, the velocities and accelerations during insect flight exhibit continuous variations. To encourage biologically plausible flight dynamics, we define a dynamics reward r_t^d that constrains the magnitudes of both velocities and accelerations:

$$r_t^d = \sum_{\lambda \in \Lambda} \Big[\max\left(0, \lambda - \lambda_{max}\right)^2 + \max\left(0, \lambda_{min} - \lambda\right)^2 \Big], \quad (15)$$

where $\Lambda = \{\|\mathbf{v}\|, \|\mathbf{a}\|\}$, with $\mathbf{v}$ and $\mathbf{a}$ representing the instantaneous velocity and acceleration of the agent. This term evaluates whether the magnitudes of both velocities and accelerations fall within the empirical range $[\lambda_{\min}, \lambda_{\max}]$ observed in real insects, encouraging the agent to reproduce the natural diversity of flight dynamics. The balance reward r_t^b penalizes abrupt rotational movements, especially roll motion, to keep the insect balanced during flight:

$$r_t^b = \|\boldsymbol{\omega}\|^2 + \|\dot{\boldsymbol{\omega}}\|^2 + \|\dot{\phi}\|^2 + \|\ddot{\phi}\|^2, \quad (16)$$

where $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\omega}}$ are the angular velocity and angular acceleration of the flying insect's body; $\dot{\phi}$ and $\ddot{\phi}$ are angular velocity and angular acceleration of the roll motion of the insect's body. The collision avoidance rewards r_t^c encourages the flying insect to keep a distance from detected obstacles:

$$r_t^c = \sum_{i=1}^{O} \max\left(0, \frac{d_s - d_i}{d_s}\right)^2, \quad (17)$$

where O denotes the number of detected obstacles; d_s is the agent's visual sensing range; and d_i represents the distance from the agent to the i-th obstacle. In our vision-based framework, the distance d_i is derived directly from the depth image generated by the virtual camera. For each detected obstacle instance, we compute the minimum depth value among its corresponding pixels. This approach naturally handles complex geometries (e.g., trees) by capturing the actual surface geometry visible from the agent's viewpoint, without requiring explicit geometric simplification.

Network. The simulation of realistic insect flight behaviors can be divided into a series of tasks where insect agents need to choose optimal actions from the continuous action space based on the current state. Within the Actor–Critic framework, we maintain two networks: an actor network that learns the policy π, and a critic network that estimates the value function $V(s)$. The policy π is modeled as a neural network (Figure 9) that maps a state s to a Gaussian distribution over action $\pi(a_t|s_t)$. The agent's visual observation $\mathcal{I}_t$ is processed by a visual encoder comprising three convolutional layers and a dense layer before being passed to a fully-connected network with other observation components. The value function $V(s_t)$ is represented as another deep neural network with similar architecture except the output layer is a single linear unit. We train the policies using PPO with a clipped surrogate objective [46]. The advantage for the policy gradient is calculated by GAE [47]. The algorithm learning procedure is described in detail in the supplemental document.

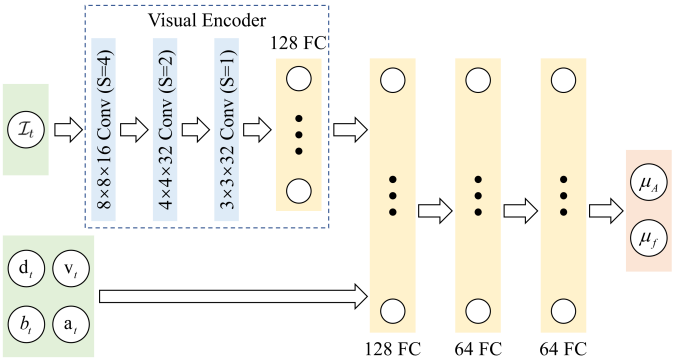


Fig. 9. Architecture of the used policy network.

VII. Results and Evaluations

We implemented our approach in Unity Engine with ML-Agents toolkit [48]. To evaluate its effectiveness, we simulated the flight motion of two types of flying insects, including a Morpho butterfly and a hoverfly. The approach was applied to emulate their flight behaviors in a variety of virtual environments. Experimental results, comparative analyses, quantitative evaluations, and user study outcomes are detailed in the subsequent sections. For animation results, please refer to the enclosed supplemental demo video.

A. Qualitative Results and Visual Comparisons

Visual obstacle avoidance learning. The beginner-level training focused on acquiring basic maneuvering skills, whereas the intermediate-level training was designed to imitate natural insect flight based on real flight datasets (see the supplemental document). At the expert level, the training incorporated visual obstacle avoidance, which was conducted in two progressive stages. In the first stage, the insect agent was exposed to simple geometric obstacles, such as cylinders and cubes. In the second stage, the obstacles were replaced by more complex virtual plants, including flowers and trees. In this phase, the state representation is augmented with depth images rendered from a virtual camera. The reward function retains the goal, kinematic, dynamic, and balance rewards from previous levels with a collision avoidance term that penalizes proximity to obstacles. Importantly, no explicit avoidance maneuvers are predefined; instead, the agent learns to autonomously adjust its wing kinematics through trial-and-error interactions while maintaining biologically plausible flight characteristics. Through this training, the insect learns to roughly perceive the relative positions of obstacles. Based on this perception, the insect autonomously adjusts its wing kinematics to generate aerodynamic forces that guided its flight away from obstacles, thereby achieving effective obstacle avoidance (shown in Fig. 10).

Comparison with a real-world blue Morpho butterfly video. In this comparative experiment, we simulated a virtual Morpho butterfly and directly compared it with real-world video of a Morpho butterfly, as provided in [25]. As shown in Fig. 11, the simulated Morpho butterfly, exhibits significant periodic whole-body pitching, which is due to intermittent lift generation due to low-frequency wingbeats. This phenomenon

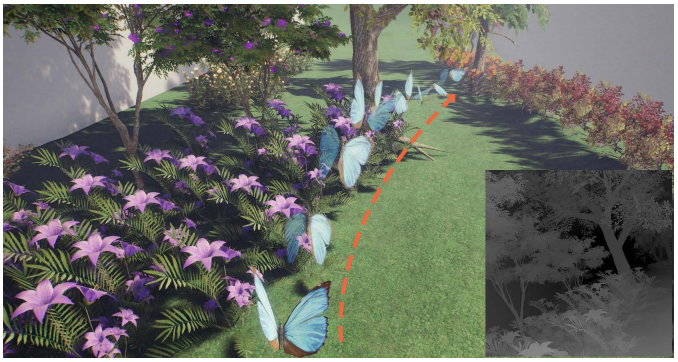


Fig. 10. The simulated insect flies freely in a virtual natural environment, avoiding obstacles through visual perception. The lower-right presents the corresponding visual depth image.

of pitching motion is a natural consequence of wing-body coupling vibrations, similar to that of a real butterfly.

Fig. 11. Comparison of flight sequences between a real-world butterfly (top) and a virtual butterfly generated by our method (bottom). The virtual butterfly exhibits pitch oscillations during flight due to the decoupling of kinematics-dynamics, which is similar to the flight motion of the real-world one.

Comparison with a real-world hoverfly video. Unlike butterflies, hoverflies maintain remarkably stable body movements during flight, which is due to continuous lift generation caused by high-frequency wingbeats with a single pair of narrow and elongated wings. Fig. 12 presents a direct comparison of flight motion between the simulated hoverfly and its real-world counterpart. As demonstrated in this comparison, the hoverfly simulated with our approach reproduced realistic flight motions that closely resemble those observed in real individuals.

Comparison with the state of the art. We compared our method with a recent state-of-the-art approach for butterfly flight simulation [4]. Unlike our framework, the approach in [4] does not couple kinematics and dynamics under the underlying physical principles, but instead updates wingbeat kinematics through manually designed laws. As a result, the generated flight behaviors are inconsistent with physical reality, such as a butterfly ascending continuously without flapping (see Fig. 13(a)). In contrast, as illustrated in Fig. 13(b), our method enables the butterfly to ascend by accelerating its wingbeats and to descend during gliding, thereby adhering to the physical principles observed in natural flight.

Comparison with ground truth. We also compared our model with the ground truth data [25]. As shown in Fig. 14, the discrepancy between our model and ground truth data is smaller than that observed for the butterfly flight simulation

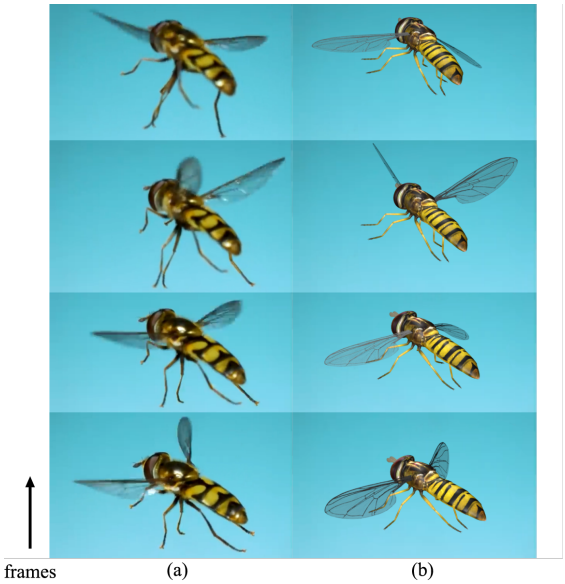


Fig. 12. Comparison of flight motion between a real-world hoverfly (a) and a simulated hoverfly generated by our approach (b). The simulated hoverfly demonstrates stable flight dynamics with realistic wingbeat kinematics that closely match video observations.

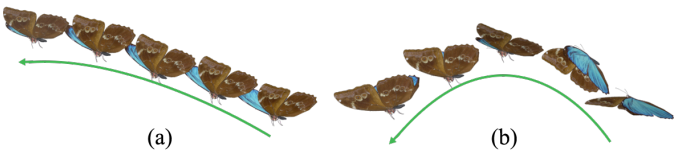


Fig. 13. (a) The simulated butterfly exhibits sustained gliding ascent in [4], which deviate from physical reality. (b) The virtual butterfly in our method ascends by accelerating its wingbeats and descends during gliding.

reported in [4]. In addition, our method can generate a wider variety of insect flight motion results beyond the ground truth data. Visual comparisons are provided in the supplemental demo video.

B. Quantitative Evaluation

Accuracy of the aerodynamic model. We evaluated the accuracy of our aerodynamic model by comparing it with previous approaches [4], [33] using the error metric ϵ between empirical datasets and model predictions. Using our CFD simulation dataset of the Morpho butterfly, we compared our model, which augments the quasi-steady model with added mass and rotational circulation forces, with the simplified quasi-steady model in [4]. Additionally, based on experimentally measured datasets of the hoverfly [33], we benchmarked our model against the semi-empirical model in [33], which accounts for quasi-steady and added mass force. As summarized in Table II, our model consistently yielded lower mean squared errors between the predicted aerodynamic forces and the ground-truth data than the baseline methods [4], [33].

Fidelity of flight motions. Inspired by prior research on insect swarm simulations [49], we adopt a quantitative approach to evaluate simulated insect flight motions using real-world flight datasets. This approach relies on discrete probability

TABLE II
COMPARISON OF OUR AERODYNAMIC MODEL WITH PREVIOUS AERODYNAMIC MODELS [4], [33].

Models	MSE (mN^2) ↓	
	CFD datasets	Measured datasets
Chen et al. [4]	0.0419	–
Walker et al. [33]	–	0.0262
Ours	**0.0205**	**0.0217**

density distribution functions (PDFs) generated from time-varying metrics, which capture the global characteristics of insect flight. Specifically, we design kinematic and dynamic metrics to assess the fidelity of wing motions and whole-body dynamics, respectively. Wing kinematics are characterized by the wingbeat amplitude A and frequency f, while whole-body dynamics are represented by velocity $\|\mathbf{v}\|$, acceleration $\|\mathbf{a}\|$, angular velocity $\|\boldsymbol{\omega}\|$, and angular acceleration $\|\dot{\boldsymbol{\omega}}\|$. Accordingly, our evaluation model is formulated as follows, incorporating these discrepancy terms:

$$E = 1 - \frac{1}{|K|}\sum_{\kappa \in K} E_\kappa, \tag{18}$$

where $K = \{A, f, \|\mathbf{v}\|, \|\mathbf{a}\|, \|\boldsymbol{\omega}\|, \|\dot{\boldsymbol{\omega}}\|\}$ is the set of evaluation metrics, and E_κ measures the discrepancy between the discrete PDFs of the real-world data and simulation results for metric κ. The E_κ is defined as:

$$E_\kappa = \frac{1}{2}\|P_\kappa^{real} - P_\kappa^{sim}\|_1, \tag{19}$$

where P_κ^{real} and P_κ^{sim} represent the discrete PDFs of the metric κ obtained from real-world data and our simulation model, respectively. To ensure comparability, both the real-world data and simulation results are normalized using pooled min–max feature scaling, thereby establishing a common numerical range. For each normalized metric, the discrete PDF is estimated by dividing the range into k equal-width bins, where k is determined following the Freedman–Diaconis rule. The probability density in each bin is then computed as:

$$P_{\kappa,i} = |S_{\kappa,i}|/|S_\kappa|, \tag{20}$$

where $|S_{\kappa,i}|$ denotes the number of samples falling into the i-th bin, and $|S_\kappa|$ is the total number of samples for metric κ.

Based on the quantitative evaluation method described above, we compared our model with the state-of-the-art butterfly simulation [4] using the blue Morpho butterfly dataset [30]. Fig. 14 presents the discrepancy values between the discrete PDFs of the real-world data and simulation results generated by our model and by the SOTA work [4]. The overall consistency values computed using Eq. 18 are 0.71 for our model and 0.54 for the SOTA work. These results demonstrate that our model achieves significantly higher fidelity to the ground-truth data, indicating that the simulated motions produced by our model more closely resemble real butterfly flight behaviors. This improvement can be attributed to the difference in how unsteady aerodynamic effects are modeled. While prior work [4] employs heuristic artificial vortex forces to enhance lift for artificial chaotic behavior presentation, these terms lack direct coupling with wing acceleration and rotation dynamics. In contrast, our physics-based unsteady components (added mass and rotational circulation forces) are derived from fluid dynamics principles and respond instantaneously to transient kinematic changes. This acceleration-dependent formulation enables more accurate force prediction during rapid maneuvers, such as stroke reversals and sharp turns, which are critical for reproducing biologically realistic flight trajectories. Consequently, the heuristic nature of artificial vortex forces may lead to systematic deviations from real flight dynamics, as reflected in the lower consistency value (0.54) observed for work [4].

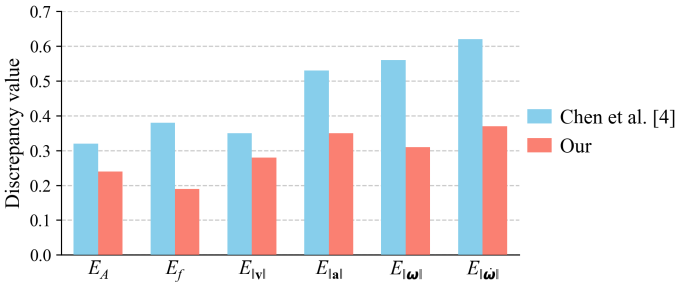


Fig. 14. Discrepancy values of different approaches (ours and Chen et al. [4]) are compared with the real-world datasets (ground-truth), where the discrepancy value is given by Eq. 19.

C. User Study

To qualitatively assess the simulation results, we conducted a user study comparing three approaches: (I) the state-of-the-art butterfly locomotion method [4], (II) our method without the kinematics–dynamics coupling component, and (III) our complete method. We adopted a paired comparison evaluation scheme [50], [51], as it allows participants to select the perceptually superior stimulus between two options, thereby reducing the likelihood of forced or unreliable judgments. The simulated insect flights generated by the three approaches were presented in immersive VR environment. From these results, we constructed two comparison pairs, namely Method I vs. Method III and Method II vs. Method III. Participants then viewed the stimuli through a VR headset and indicated their preference by casting votes.

A total of 30 student volunteers (13 female, 17 males; ages from 18 to 38 years old) from a university campus participated in the user study. All the participants are STEM majors and have no prior experience on animation or simulation. Prior to the experiment, all participants were informed about the purpose, procedures, and potential risks of the study. Based on the collected user votes, the aggregated results are presented in Figure 15. The preference votes from the paired comparisons reveal a clear advantage for our complete method. Specifically, the comparison between Method I and Method III resulted in votes of 4 and 26, respectively; while Method II versus Method III had 5 and 25 votes, respectively, in the second comparison (Method II vs. Method III). To statistically validate these preferences, a two-tailed one-sample t-test was applied to the voting data. The resulting p-values are smaller than 0.01 in both comparisons, confirming that our complete method

(Method III) is statistically significantly better than both our method without the kinematics–dynamics coupling (Method II) and the approach in [4] (Method I).

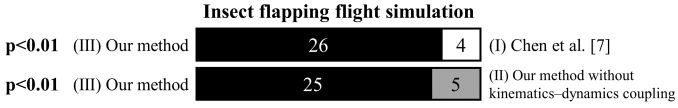


Fig. 15. The aggregated user voting result.

D. Ablation Study

Kinematic modeling. We compared our complete model using the proposed wingbeat parameterization against a model where wing motion is instead driven by a simple cosine function, as commonly adopted in prior work [4], [17]. As illustrated in the supplemental demo video, our kinematic modeling produces smooth and continuous wingbeat motions across consecutive wingbeat cycles, resulting in more realistic and natural flapping behavior. In contrast, the simple cosine function fails to ensure temporal continuity when wingbeat amplitudes vary between cycles, leading to abrupt changes in wing motion. These discontinuities induce large instantaneous variations in aerodynamic forces, which in turn cause oscillatory and unstable flight motion. As shown in Table III, flight simulations using the simple cosine function exhibit significantly larger errors in linear acceleration compared to real insect flight data, whereas our wingbeat parameterization achieves a substantially better agreement with ground truth.

Unsteady force components. To evaluate the significance of unsteady aerodynamic forces, we conducted an ablation study by comparing two configurations: (i) our complete model, and (ii) the same model without unsteady force components. For both configurations, flight simulations were performed under identical deep reinforcement learning control framework, training settings, and reward design. The resulting flight trajectories were compared with real insect flight data using the quantitative evaluation metrics. As reported in Table III, removing unsteady force components results in substantially larger discrepancies in wingbeat kinematics, with significantly increased errors in wingbeat frequency and amplitude, indicating that the learned policy deviates markedly from real insect flight behaviors. This failure to follow the input trajectory is primarily caused by reduced controllability due to inaccurate force prediction. Specifically, unsteady forces provide critical additional lift during stroke reversals and rapid accelerations. Without these components, the mapping between wing kinematics and resulting aerodynamic forces becomes systematically biased. Although the DRL controller attempts to compensate for the reduced lift by increasing wingbeat frequency and amplitude, such a compensation is global and cannot restore the precise force modulation required for reproducing realistic trajectories. Consequently, the learned policy deviates from biologically realistic flight patterns, leading to degraded maneuverability and unrealistic trajectories, particularly in scenarios where unsteady effects are critical.

kinematics–dynamics coupling. We further investigated the role of kinematics–dynamics coupling by comparing our complete model against a model in which wing kinematics are decoupled from the dynamic state during control. As illustrated in the demo video, the insect simulated with our complete model adaptively adjusts its wing kinematic strategies in response to dynamic states, enabling effective obstacle avoidance. In contrast, the insect simulated without kinematics–dynamics coupling fails to adapt its wing kinematics, despite perceiving obstacles ahead, leading to unsuccessful avoidance behavior.

TABLE III
ABLATION STUDY OF KINEMATIC MODELING (KM) AND UNSTEADY FORCE COMPONENTS (UF), EVALUATED AGAINST REAL FLIGHT DATA.

Models	$E_A \downarrow$	$E_f \downarrow$	$E_{\|\mathbf{v}\|} \downarrow$	$E_{\|\mathbf{a}\|} \downarrow$	$E_{\|\boldsymbol{\omega}\|} \downarrow$	$E_{\|\dot{\boldsymbol{\omega}}\|} \downarrow$
Our	**0.24**	**0.19**	**0.28**	**0.35**	**0.31**	**0.37**
Our w/o KM	0.29	0.26	0.35	0.68	0.35	0.42
Our w/o UF	0.54	0.62	0.33	0.39	0.36	0.40

VIII. Discussion and Conclusion

In this paper we present a data-driven control framework for insect flapping flight that emphasizes the bidirectional coupling between wing kinematics and aerodynamic dynamics. It integrates biologically plausible wingbeat generation, an improved semi-empirical aerodynamic model, with deep reinforcement learning-based control. First, we employ a limited set of real-world insect wingbeat recordings to parameterize natural flapping cycles using Fourier series, enabling the generation of physically plausible and temporally continuous wing kinematics. Second, we extend conventional quasi-steady aerodynamic models by incorporating two unsteady force components and optimizing the model using CFD simulation data. This approach improves the accuracy of aerodynamic force predictions for complex flapping-wing motions. Third, deep reinforcement learning is employed to allow virtual insect agents to learn the coupled mapping between wing kinematics and flight dynamics, with real flight trajectory data serving as references to guide the discovery of plausible control behaviors. Through this interactive training paradigm, agents acquire robust control policies capable of adaptive behaviors such as goal tracking and obstacle avoidance. By integrating these components, our framework enables realistic insect flight simulations, with numerous potential applications in virtual reality and computer graphics.

Despite the effectiveness and generality of our approach, the current framework presents several limitations: (1) *Limited realism in wingbeat kinematics.* Due to the scarcity of detailed insect wing kinematics data, we fit Fourier series to the average flapping motions observed in a small number of recordings. The generated motions reflect the overall trends of natural wingbeats but may lack sufficient realism and variability of real-world flapping strategies exhibited across different species or behavioral contexts. (2) *High cost of aerodynamic model optimization.* Our semi-empirical aerodynamic model is optimized using empirical datasets, including CFD simulation dataset and experimentally measured data. The generation of CFD simulation dataset is time-consuming and computationally intensive. Furthermore, the small size of insects makes

the acquisition of comprehensive motion data challenging, resulting in a scarcity of experimentally measured data.

As future work, we plan to acquire high quality whole-body kinematics data of insects, with particular focus on accurately capturing wing deformation during flight. These high-fidelity datasets will enable us to better understand the underlying flight mechanisms and further improve the realism of our simulation framework. Additionally, we plan to develop more efficient data-driven aerodynamic models capable of generalizing across insect species with reduced dependence on empirical datasets. Ultimately, we envision our system as a foundation for building lifelike, adaptive virtual insects applicable to scientific simulation and biomimetic robotics.

ACKNOWLEDGMENTS

We sincerely appreciate Dr. Camille Le Roy for sharing the blue Morpho butterfly flight datasets and offering thoughtful suggestions on the experimental design and result interpretation, all of which have greatly contributed to the smooth progress and improvement of this research. Qiang Chen was supported by the National Natural Science Foundation of China (Grant No. 62262024), and Science Foundation of the Jiangxi Province of China (Grant No. 20232BAB202023, 20252BAC250014). Yuming Fang was supported in part by the National Natural Science Foundation of China (Grant No. U24A20220, 62132006) and Science Foundation of the Jiangxi Province of China (Grant No. 20252BAC230003). Tingsong Lu was supported in part by the Innovation Fund Designated for Graduate Students of Jiangxi Province (Grant No. YC2025-B140). Xiaogang Jin was supported by the National Natural Science Foundation of China (Grant No. 62472373).

REFERENCES

[1] J.-c. Wu and Z. Popović, "Realistic modeling of bird flight animations," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 888–895, 2003.

[2] E. Ju, J. Won, J. Lee, B. Choi, J. Noh, and M. G. Choi, "Data-driven control of flapping flight," *ACM Transactions on Graphics*, vol. 32, no. 5, pp. 1–12, 2013.

[3] S. Wang, X. Zhang, G. He, and T. Liu, "Numerical simulation of unsteady flows over a slow-flying bat," *Theoretical and Applied Mechanics Letters*, vol. 5, no. 1, pp. 5–8, 2015.

[4] Q. Chen, T. Lu, Y. Tong, G. Luo, X. Jin, and Z. Deng, "A practical model for realistic butterfly flight simulation," *ACM Transactions on Graphics*, vol. 41, no. 3, pp. 1–12, 2022.

[5] R. Vaxenburg, I. Siwanowicz, J. Merel, A. A. Robie, C. Morrow, G. Novati, Z. Stefanidi, G.-J. Both, G. M. Card, M. B. Reiser *et al.*, "Whole-body physics simulation of fruit fly locomotion," *Nature*, vol. 643, pp. 1312–1320, 2025.

[6] M. H. Dickinson, F.-O. Lehmann, and S. P. Sane, "Wing rotation and the aerodynamic basis of insect flight," *Science*, vol. 284, no. 5422, pp. 1954–1960, 1999.

[7] S. P. Sane, "The aerodynamics of insect flight," *Journal of Experimental Biology*, vol. 206, no. 23, pp. 4191–4208, 2003.

[8] Y. Lee, S. Kim, and J. Lee, "Data-driven biped control," *ACM Transactions on Graphics*, vol. 29, no. 4, pp. 1–8, 2010.

[9] Y.-Y. Tsai, W.-C. Lin, K. B. Cheng, J. Lee, and T.-Y. Lee, "Real-time physics-based 3d biped character animation using an inverted pendulum model," *IEEE transactions on visualization and computer graphics*, vol. 16, no. 2, pp. 325–337, 2009.

[10] H. Zhang, S. Starke, T. Komura, and J. Saito, "Mode-adaptive neural networks for quadruped motion control," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1–11, 2018.

[11] Y.-S. Luo, J. H. Soeseno, T. P.-C. Chen, and W.-C. Chen, "Carl: Controllable agent with reinforcement learning for quadruped locomotion," *ACM Transactions on Graphics*, vol. 39, no. 4, pp. 38–1, 2020.

[12] Q. Chen, W. Guo, Y. Fang, Y. Tong, T. Lu, X. Jin, and Z. Deng, "A bio-inspired model for bee simulations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 31, no. 4, pp. 2073–2085, 2025.

[13] J. Won, J. Park, K. Kim, and J. Lee, "How to train your dragon: example-guided control of flapping flight," *ACM Transactions on Graphics*, vol. 36, no. 6, pp. 1–13, 2017.

[14] J. Won, J. Park, and J. Lee, "Aerobatics control of flying creatures via self-regulated learning," *ACM Transactions on Graphics*, vol. 37, no. 6, pp. 1–10, 2018.

[15] A. Hedenström and L. C. Johansson, "Bat flight: aerodynamics, kinematics and flight morphology," *Journal of Experimental Biology*, vol. 218, no. 5, pp. 653–663, 2015.

[16] R. J. Bomphrey, T. Nakata, P. Henningsson, and H.-T. Lin, "Flight of the dragonflies and damselflies," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 371, no. 1704, p. 20150389, 2016.

[17] Q. Chen, Z. Deng, F. Li, Y. Fang, T. Lu, Y. Tong, and Y. Zuo, "Real-time wing deformation simulations for flying insects," in *ACM SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024, pp. 1–11.

[18] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Transactions On Graphics*, vol. 37, no. 4, pp. 1–14, 2018.

[19] K. Hu, B. Haworth, G. Berseth, V. Pavlovic, P. Faloutsos, and M. Kapadia, "Heterogeneous crowd simulation using parametric reinforcement learning," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 4, pp. 2036–2052, 2021.

[20] J. Lee and K. H. Lee, "Precomputing avatar behavior from human motion data," in *ACM SIGGRAPH Symposium on Computer Animation*. Goslar, DEU: Eurographics Association, 2004, p. 79–87.

[21] S. Coros, P. Beaudoin, and M. van de Panne, "Robust task-based control policies for physics-based characters," *ACM Transactions on Graphics*, vol. 28, no. 5, p. 1–9, Dec. 2009.

[22] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[23] X. B. Peng, G. Berseth, and M. Van de Panne, "Terrain-adaptive locomotion skills using deep reinforcement learning," *ACM Transactions on Graphics*, vol. 35, no. 4, pp. 1–12, 2016.

[24] X. B. Peng, G. Berseth, K. Yin, and M. Van De Panne, "Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–13, 2017.

[25] C. Le Roy, D. Amadori, S. Charberet, J. Windt, F. T. Muijres, V. Llaurens, and V. Debat, "Adaptive evolution of flight in morpho butterflies," *Science*, vol. 374, no. 6571, pp. 1158–1162, 2021.

[26] C. Le Roy, N. Tervelde, T. Engels, and F. T. Muijres, "Adaptations in wing morphology rather than wingbeat kinematics enable flight in small hoverfly species," *eLife*, Jun. 2025.

[27] X. L. Mou, Y. P. Liu, and M. Sun, "Wing motion measurement and aerodynamics of hovering true hoverflies," *Journal of Experimental Biology*, vol. 214, no. 17, pp. 2832–2844, 2011.

[28] J. M. Melis, I. Siwanowicz, and M. H. Dickinson, "Machine learning reveals the control mechanics of an insect wing hinge," *Nature*, vol. 628, no. 8009, pp. 795–803, 2024.

[29] I. Nagesh, S. M. Walker, and G. K. Taylor, "Motor output and control input in flapping flight: a compact model of the deforming wing kinematics of manoeuvring hoverflies," *Journal of the Royal Society Interface*, vol. 16, no. 161, p. 20190435, 2019.

[30] C. Le Roy, N. Silva, R. Godoy-Diana, V. Debat, V. Llaurens, and F. T. Muijres, "Divergence of climbing escape flight performance in morpho butterflies living in different microhabitats," *Journal of Experimental Biology*, vol. 225, no. 15, p. jeb243867, 2022.

[31] M. Osborne, "Aerodynamics of flapping flight with application to insects," *Journal of Experimental Biology*, vol. 28, no. 2, pp. 221–245, 1951.

[32] H. Glauert, *The elements of aerofoil and airscrew theory*. Cambridge university press, 1983.

[33] S. M. Walker and G. K. Taylor, "A semi-empirical model of the aerodynamics of manoeuvring insect flight," *Journal of the Royal Society Interface*, vol. 18, no. 177, p. 20210103, 2021.

[34] T. Weis-Fogh, "Quick estimates of flight fitness in hovering animals, including novel mechanisms for lift production," *Journal of Experimental Biology*, vol. 59, no. 1, pp. 169–230, 1973.

[35] C. P. Ellington, "The aerodynamics of hovering insect flight. i. the quasi-steady analysis," *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, vol. 305, no. 1122, pp. 1–15, 1984.

[36] S. Bhat, J. Zhao, J. Sheridan, K. Hourigan, and M. Thompson, "Aspect ratio studies on insect wings," *Physics of Fluids*, vol. 31, no. 12, 2019.

[37] S. P. Sane and M. H. Dickinson, "The control of flight force by a flapping wing: lift and drag production," *Journal of Experimental Biology*, vol. 204, no. 15, pp. 2607–2626, 2001.
[38] J. R. Usherwood and C. P. Ellington, "The aerodynamics of revolving wings i. model hawkmoth wings," *Journal of Experimental Biology*, vol. 205, no. 11, pp. 1547–1564, 2002.
[39] S. P. Sane and M. H. Dickinson, "The aerodynamic effects of wing rotation and a revised quasi-steady model of flapping flight," *Journal of Experimental Biology*, vol. 205, no. 8, pp. 1087–1096, 2002.
[40] D. J. Acheson, *Elementary fluid dynamics*. Oxford University Press, 03 1990.
[41] G. Birkhoff, *Hydrodynamics*. Princeton University Press, 2015.
[42] L. Sedov, C. Chu, H. Cohen, B. Seckler, and J. Gillis, "Two-dimensional problems in hydrodynamics and aerodynamics," *Physics Today*, vol. 18, no. 12, pp. 62–63, 1965.
[43] P. Cinnella and P. M. Congedo, "Inviscid and viscous aerodynamics of dense gases," *Journal of Fluid Mechanics*, vol. 580, pp. 179–217, 2007.
[44] C. Qian, Y. Fang, F. Jia, J. Yan, Y. Liang, and T. Li, "Toward practical autonomous flight simulation for flapping wing biomimetic robots with experimental validation," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 5084–5095, 2025.
[45] Y. Meresman and G. Ribak, "Elastic wing deformations mitigate flapping asymmetry during manoeuvres in rose chafers (protaetia cuprea)," *Journal of Experimental Biology*, vol. 223, no. 24, p. jeb225599, 2020.
[46] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
[47] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.
[48] A. Juliani, V.-P. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar, and D. Lange, "Unity: A general platform for intelligent agents," *arXiv preprint arXiv:1809.02627*, 2020.
[49] X. Wang, J. Ren, X. Jin, and D. Manocha, "Bswarm: biologically-plausible dynamics model of insect swarms," in *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. New York, NY, USA: Association for Computing Machinery, 2015, pp. 111–118.
[50] X. Ma and Z. Deng, "Natural eye motion synthesis by modeling gaze-head coupling," in *2009 IEEE Virtual Reality Conference (VR)*. IEEE, 2009, pp. 143–150.
[51] J. Sanyal, S. Zhang, G. Bhattacharya, P. Amburn, and R. Moorhead, "A user study to compare four uncertainty visualization methods for 1d and 2d datasets," *IEEE transactions on visualization and computer graphics*, vol. 15, no. 6, pp. 1209–1218, 2009.

Tingsong Lu. He is currently a Ph.D. candidate at Jiangxi University of Finance and Economics. He received the B.Sc. and M.Sc. degrees from East China Jiaotong University. His research interests include insect swarm simulation, human computer interaction, and motion capture. His related work has been published in leading journals, such as ACM Transactions on Graphics and IEEE Transactions on Visualization and Computer Graphics, as well as in top-tier conferences, including SIGGRAPH.

Qiang Chen (Member, IEEE). He is currently an Associate Professor with Jiangxi University of Finance and Economics, Nanchang, China. He received his BSc and MSc degrees in Software Engineering in 2006 and 2012, respectively. He earned his PhD degrees in 2021. His current research interests include crowd simulation, insect swarm simulation, VR/AR, etc. The corresponding papers have been published in major international journals such as ACM Transactions on Graphics, IEEE Transactions on Visualization and Computer Graphics, and top conferences such as SIGGRAPH.

Yuming Fang (Fellow, IEEE). He received the BE degree from Sichuan University, Chengdu. China, the MS degree from the Beijing University of Technology, Beijing, China, and the PhD degree from Nanyang Technological University, Singapore. He is currently a professor with the School of Information Management, Jiangxi University of Finance and Economics, Nanchang, China. His research interests include visual attention modeling visual quality assessment, computer vision, and 3D image/video processing. He serves on the editorial board of the Signal Processing: Image Communication. He is a fellow of IEEE.

Camille Le Roy. He is a post-doctoral researcher in the Experimental Zoology Group at Wageningen University, Netherlands, where he investigates the evolution and biomechanics of insect flight. He received his Ph.D. in Evolutionary Biology from the Natural History Museum of Paris in 2020, focusing on the evolution of flight in Morpho butterflies. His research combines high-speed stereoscopic videography, 3D reconstruction techniques, and biomechanical analyses to understand animal locomotion and morphological adaptation. Dr. Le Roy has published in leading journals including Science and Nature Communications, and received the Young Scientist Award from the Society for Experimental Biology in 2022.

Xiaogang Jin (Member, IEEE). He received the BSc degree in computer science and the MSc and PhD degrees in applied mathematics from Zhejiang University, P. R. China, in 1989, 1992, and 1995, respectively. He is a professor with the State Key Laboratory of CAD&CG, Zhejiang University. His current research interests include virtual reality, traffic simulation, collective behavior simulation, cloth animation, virtual try-on, digital human, implicit surface modeling and applications, creative modeling, computer-generated marbling, sketch-based modeling, and computer games. He was the recipient of the ACM Recognition of Service Award in 2015 and the Best Paper Awards from CASA 2017 and CASA 2018. He is a member of the IEEE and ACM.

Zhigang Deng (Senior Member, IEEE). He is Moores Professor of Computer Science at University of Houston, Texas, USA. His research interests include computer graphics, computer animation, virtual humans, and HCI. He earned his Ph.D. in Computer Science at the Department of Computer Science at the University of Southern California in 2006. Prior that, he also completed B.S. degree in Mathematics from Xiamen University (China), and M.S. in Computer Science from Peking University (China). Besides serving as the conference general or program co-chairs for CASA 2014, SCA 2015, MIG 2022, PG 2023, and CGI 2024, he has been an Associate Editor for IEEE Transactions on Visualization and Computer Graphics, Computer Graphics Forum, etc. He is a Distinguished member of ACM and a Senior member of IEEE.