Reinforcement Learning of Active Colloidal Particles

Masterarbeit

Vorgelegt von **Veit-Lorenz Heuthe** an der

Mathematisch Naturwissenschaftliche Sektion,

Fachbereich Chemie.

1. Gutachter: Prof. Clemens Bechinger

2. Gutachter: Prof. Christine Peter

Konstanz den 18. November 2020

E R K L Ä R U N G

Ich versichere hiermit, dass ich die anliegende Masterarbeit mit dem Thema:

Reinforcement Learning of Active Colloidal Particles

selbständig verfasst und keine anderen Hilfsmittel und Quellen als die angegebenen benutzt habe.

Die Stellen, die anderen Werken (einschließlich des Internets und anderer elektronischer Text- und Datensammlungen) dem Wortlaut oder dem Sinn nach entnommen sind, habe ich in jedem einzelnen Fall durch Angabe der Quelle bzw. der Sekundärliteratur als Entlehnung kenntlich gemacht.

Weiterhin versichere ich hiermit, dass die o.g. Arbeit noch nicht anderweitig als Abschlussarbeit einer Abschluss-Prüfung eingereicht wurde. Mir ist ferner bekannt, dass ich bis zum Abschluss des Prüfungsverfahrens die Materialien verfügbar zu halten habe, welche die eigenständige Abfassung der Arbeit belegen können.

_____

(Unterschrift)

_____

(Ort, Datum)

Table of Contents

Abbreviations

- ML - Machine Learning
- RL - Reinforcement Learning
- AOD - Acousto-Optical Deflector
- ANN - Artificial Neural Network
- PPO - Proximate Policy Optimization
- MSD - Mean Square Displacement

1 Introduction

Machine learning (ML) as a powerful tool for decision making and data analysis has found its way into many aspects of our lives like process industry[1] and healthcare[2]. With their ability to handle large datasets, ML algorithms have established in science, too as a method to make predictions even for models that lack an explicit description[3]. Active matter science is particularly often dealing with complex systems in absence of capable models and can therefore be expected to greatly benefit from the application of ML[4]. The subject of this research field are so called active particles, whose motion is driven by constant uptake and conversion of energy from their environment[5]. Since this property is very common to organisms in nature, it is not surprising that patterns of natural behavior like chemotaxis[6], group formation[7-8] or even swirling[9] have already been reproduced in active particle systems. So far, such phenomena were studied by proposing rules for particle interaction and then investigating the emergent states. A more promising way to identify the principles underlying emergent behavior would be to define a specific collective state and then determine the rules necessary for achieving it. Multi agent robotics are facing similar challenges and often rely on reinforcement learning (RL)[10], a class of ML algorithms, for finding strategies of how robots can collectively solve a given task. RL algorithms, that rely on automatic improvement in a specific task solely from experience, are therefore well suited for modelling emergent patterns in combination with active matter. But so far, there is to the authors knowledge only one experimental study combining machine learning and active matter, where a single active particle is steered by an RL algorithm. The aim of the present work is to give an RL algorithm the control over the motion of multiple active swimmers, that are ought to solve a collective task. The algorithm is implemented in a way, that reproduces the situation of natural organisms with individually acting entities that rely on limited perception. As a task, the RL controlled particles should learn how to rotate a much larger, rod-shaped object by means of their propulsion. Due to the big difference between the particles' and the rod's size, this task demands for a collective solution. Additionally, the rod's movements allowed for the determination of the pushing force of a single active particle since they are a direct, measurable response to the forces exerted on it.

2 Theory and Methods

Since this work is aiming at a combination of active colloidal motion and reinforcement learning, these two concepts are explained in this chapter. In the part about active colloidal motion, this phenomenon and its experimental realization in this work are outlined. The theory of reinforcement learning is explained in a little more detail with a focus on the basic principle and how the used algorithm was implemented to be able to control the motion of active Brownian swimmers.

2.1 Active Colloidal Motion

It is known since the 19th century that matter on the small scale is in constant motion even in equilibrium with its environment[11]. Although, only since the end of the 20th century the idea of self-propelled, non-equilibrium motion of small particles is present[12]. This so-called active Brownian motion refers to particles, that can take up energy from their environment and convert it to propulsion[5]. Until the experimental introduction of active Brownian swimmers (or microswimmers) this property was exclusive to biological entities. It was therefore realized right from the beginning that microswimmers are an ideal framework to study the complex dynamics of living systems in a minimalistic framework[12]. And indeed, microswimmer systems are observed to reproduce individual and collective phenomena, that are inherent to motile organisms, like phototaxis, group formation or swirling[13]. In the present work, individual and precise steering of multiple particles was crucial since the aim was to give a reinforcement learning algorithm the control over multiple particles' motion. In the next two sections, the microswimmer system and its experimental realization used in this work are explained.

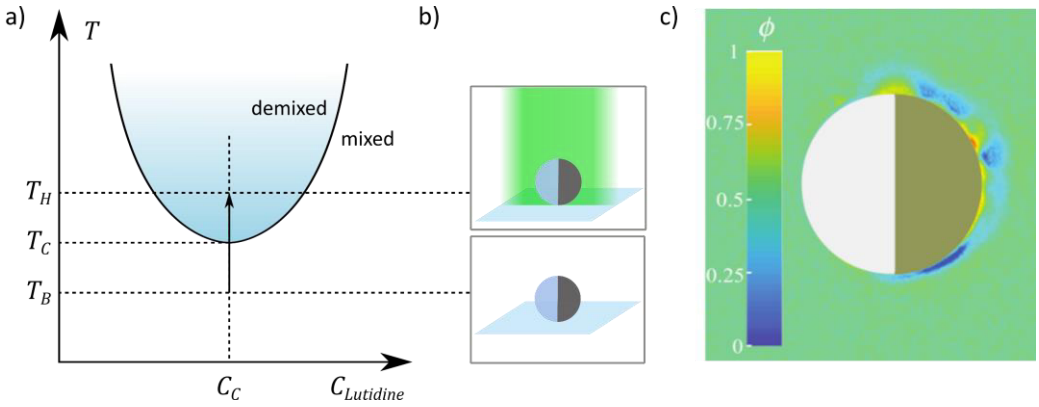

Figure 1: The propulsion mechanism: a) Phase diagram of water lutidine mixtures with a critical concentration of $C_c = 28.6$ wt% and a critical temperature $T_c = 33.9$ °C. b) Illuminating a particle at bath temperature T_B leads to heating of its cap to a temperature T_H above T_c. This induces demixing of the solvent as depicted in c) in an experimental image with color-coded intensity ϕ[14]. The asymmetry of the demixing bubble induces self-phoretic propulsion of the particle.

2.1.1 Light induced Demixing driven Microswimmers

Today, there exists a broad variety of ways to fuel microswimmers, that are covered in reviews like Ref. [5]. But there are only a few systems at hand, that allow for the precise control of the particles' motility[15-17], which was an important requirement for the microswimmers in this work. The active particles used here are driven by self-phoresis[18]. For this, silica particles of about 6 µm in diameter were half coated with carbon and then immersed in a mixture of water containing 2,6-lutidine in the critical concentration $C_c = 28.6$ wt%. For experiments, a bath temperature of $T_B = 28.5$ °C was maintained. Illumination of the so-called Janus particles heats up their carbon cap to a temperature T_H above the critical temperature $T_c = 33.9$ °C of the water lutidine mixture. This leads to the formation of an asymmetric demixing zone around the particles[14, 18] (illustrated in Fig 1). The emerging gradients in concentration and chemical potential produce a net-force on the colloid, that makes it move through the solution. Since the light intensities necessary for heating of the carbon cap above T_c are very low (below 10 µW/µm^2), optical forces can be excluded. Most importantly, this microswimmer system allows for tuning the propulsion velocity by the light intensity.

Controlling the motility of the particles with homogeneous illumination would not have allowed for making the particles move independently, which was a fundamental

requirement for the aim of this work. To achieve individual motion, the microswimmers were illuminated separately using focused laser beams with a beamwidth of only 10 µm. By this means, each particles' propulsion velocity could be controlled on its own. Additionally, control of the particles' orientation was necessary for a complete determination of their behavior. In the case of the used microswimmers, particle rotation can be induced with an intensity gradient ∇I in the illumination, as illustrated in Fig 2[19]. The emerging different propulsion forces on either of the swimmer's sides result in a net torque exerted on it. In the next section, the experimental realization of individual particle illumination and asymmetric illumination profiles are explained.

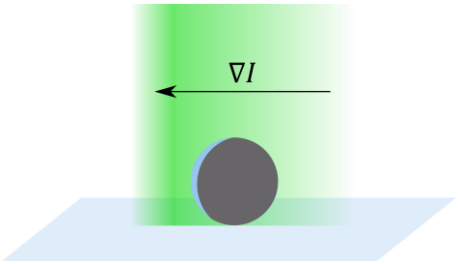

Figure 2: Illumination of a particle with an intensity gradient ∇I exerts a torque on it, allowing for controlled rotations.

2.1.2 The Experimental Setup

For steering the above introduced microswimmers in experiment, the particles had to be illuminated precisely and individually. Fig 3 schematically depicts the main components of the setup used for this purpose. An acousto-optical deflector (AOD) was used to position a laser beam at defined positions inside a measurement cell, which contained the microswimmers. In an AOD, light is diffracted by acoustic waves travelling through a crystal[20]. By changing the frequency of these waves, the angle of diffraction can be controlled. To finally translate the lights' direction to a position in the measurement cell, a 4f-optic and an objective were employed. Since an AOD can only deflect the beam by one angle, the particles were illuminated sequentially at 100 kHz. The remixing dynamics of water and lutidine in the micrometer range take place on the time scale of milliseconds[21] and are therefore much slower than the laser pulse durations. Because of this, the illumination occurs quasi-continuous in terms of the propulsion mechanism. To realize the asymmetric illumination necessary for controlled

particle rotation, two laser spots of different intensity were pointed to the two sides of a particle, as illustrated in Fig 4. A digital camera was used to keep track of the particles' positions and orientations at 5 frames per second. This information was processed to update the laser positions based on how the particles should move in the next frame. This feedback loop allows for online steering of the particles according to any desired rule. Previous studies using this setup with different rule definitions were already able to show how versatile and well applicable this system is for modelling the emergence of collective patterns amongst microswimmers[7, 9, 22]. In contrast to these studies, the fixed rules were replaced by a machine learning algorithm in the present work.

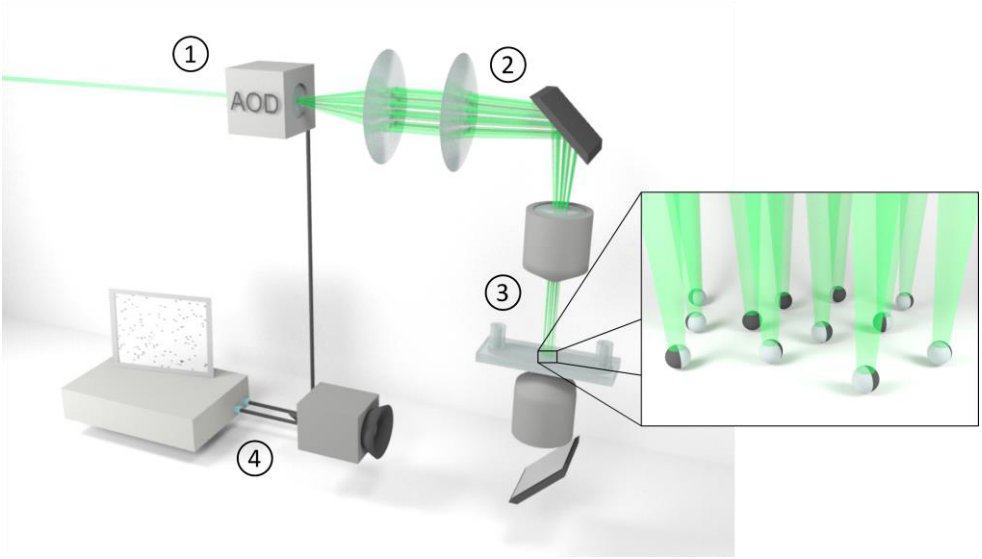

Figure 3: The setup used to steer microswimmers: With the help of a 4f-optic (2) and an objective, an AOD (1) controls the position of a laser beam in the sample cell (3). A camera records pictures online (4), which are evaluated by a computer to determine the next laser positions based on how the particles should move in the next frame. This circular flow of information creates a feedback loop allowing for live steering of the microswimmers according to any rule specified.

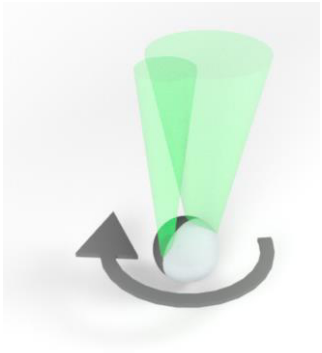

Figure 4: The particles were illuminated by two spots of different intensity at either end of the cap to reorient them in a controlled way.

2.2 Reinforcement Learning

Software tasks are becoming more and more complex. At the same time, elaborate computation and large datasets are becoming more and more easily available. Thus, it is no wonder that the development of software using machine learning rather than manually programming desired outputs for every possible input has seen a big boom in the last decade[23]. The general idea of machine learning is to create an algorithm, that can improve in a specific task solely through experience in this task. In this field of artificial intelligence, reinforcement learning (RL) is a highly promising and intensively studied class of algorithms, that does not rely on extensive training using artificially labeled datasets, but rather on learning by doing[24]. This semi-supervised approach requires only minimal feedback to the software during training in the form of a so-called reward or penalty, which is a measure of how well the algorithm is performing in the task it should achieve. Due to this form of autonomous data acquisition, RL algorithms are particularly useful for practical tasks where a program is supposed to choose an action based on a situation it is facing, just like a player in a board game. And indeed, RL based programs have by now beaten the best human players in complex, strategic games like chess and Go[24], but could also be economically used e.g. for controlling electric power systems in order to reduce costs and enhance efficiency[25]. In robotics, RL algorithms are used to enable groups of agents to act collectively[10]. For this reason, an RL algorithm was chosen in this work to take control of the movement of active swimmers, trying to solve a collective task. In the following sections, the basics of RL are briefly discussed together with the algorithm used in this work and the details of its implementation.

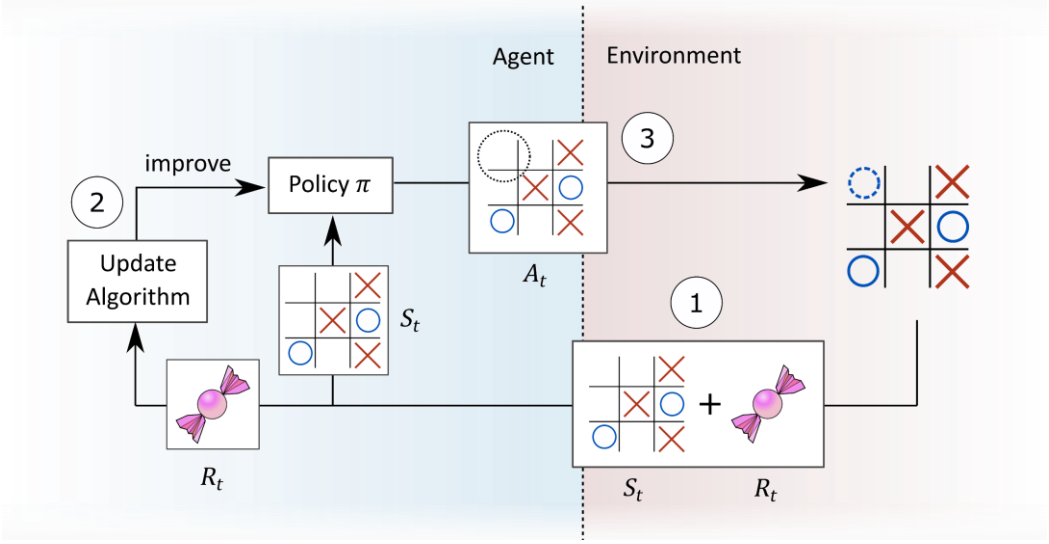

Figure 5: Flow chart showing how an RL algorithm works on the example of a Tic Tac Toe game: 1) The agent receives two reactions of the environment to his last action: the current state S_t and a reward R_t (symbolized by the sweet) based on the last action of the agent. 2) The agent processes the information transported by the reward R_t to evaluate and update its policy. 3) The agent chooses an action A_t based on the current state S_t according to its policy.

2.2.1 Basic Principles of Reinforcement Learning

In RL an algorithm is always viewed as an agent, that is interacting with an environment[26]. The agent chooses actions based on a policy, while the environment is the system the agent is interacting with. Even though the type of system and interactions can cover a whole spectrum from only rearranging data in a digital environment to controlling process parameters in an industrial plant, the training of an RL algorithm follows a general workflow, which is schematized in Fig 5. Like in a game of Tic Tac Toe, the training of an RL agent can be viewed as the agent and the environment taking turns in changing the current state, e.g. in Tic Tac Toe the constellation of markers on the board. This allows RL to be treated in the mathematical framework of a Markov decision process, where every step the agent takes is independent of the steps previously taken.

One step of interaction starts with the environment giving the agent the current state S_t. Then, the agent determines its next action A_t based on S_t and according to its policy π. The policy π is a function, that takes the current state S_t as an input and returns the probability of every possible action A_t to be chosen in this state. For

learning, the agent's policy π must be changed since the same policy always produces the same probability distribution amongst the actions. Therefore, the environment returns an additional number R_t, which is called the reward. This reward is a measure of how well the last action (A_{t-1}) agreed to what the agent is supposed to achieve. In the Tic Tac Toe example, the reward could be $R_t = 1$ for a turn in which the agent wins the game and $R_t = 0$ for every other turn. This parameter is the only way of communicating the aim, the algorithm is supposed to achieve and is used to update the policy π. For successful training, the policy must be changed in a way, that ensures the future actions to be closer to the desired behavior. By giving the agent higher rewards for actions that are more beneficial for the achievement of the task, one can translate every task to the problem of maximizing the reward. This means, the agents' policy needs to be updated in a way that favors actions that lead to a high reward. One way to make the agent maximize its reward is the use of a so-called state-action value function, which assigns a value $Q(A_t, S_t)$ to every possible action A_t in every possible state S_t the agent can encounter. This value Q quantifies, how valuable it is to choose the action A_t in the state S_t regarding the achievement of the given task. Since the reward is always defined to be higher for better actions, the state-action value Q can be defined in terms of the reward. This is done using the expectation value of the cumulated rewards R following the choice of A_t at time t after encountering state S_t. The equation

$$Q(A_t, S_t) = E_\pi \left[\sum_k^\infty R_{t+k} | \, S_t, A_t \right] \qquad (1)$$

expresses this mathematically. $E[\,]$ is a conditional expected value with the conditions that firstly the agent is continuously acting according to the policy π and secondly that the action A_t is chosen at time t in the state S_t. For the Tic Tac Toe example, the state-action value function corresponds to a list of values of every possible move for each combination of symbols on the board. Moves that bring the agent closer to a win would having a higher value than moves allowing the opponent to win. With the use of a value function, one possible policy would be to deterministically choose the action with the highest value:

$$\pi(A_t = a \mid S_t = s) = \begin{cases} 1 : A_t = \underset{A}{\operatorname{argmax}}\, Q(S_t, A) \\ 0 : else \end{cases} . \qquad (2)$$

In Eq. (2), the argmax function returns the action A_t of all possible actions at time t for which the state-action value function $Q(S_t, A_t)$ is maximal. The policy π returns the probability of choosing action A_t at time t with the condition of the current state (at time t) being $S_t = s$. With this policy, the problem of selecting the right action in every situation breaks down to finding the so-called optimal value function Q^*, in which the best action for every state has the highest value. Since explicitly calculating the optimal value function is unfeasible for most tasks, the expectation value in Eq. (1) can be simply approximated by an average of the rewards obtained in the trajectory samples the agent encounters during training. The thus obtained value function converges to the optimal value function Q^* in the limit of infinitely long training, meaning the limit of exhaustive sampling of the whole state-action space with trajectories[26]. Therefore, an RL algorithm that plays Tic Tac Toe and updates its state-action value function every time it gets a reward will reliably play better and better with more and more training.

This example of training an RL agent is only one of many possible ways to implement and improve a policy. More sophisticated and efficient algorithms usually rely on more complex policies and policy updates[27]. However, the basic idea of generating experience during training and then updating a policy according to the returned rewards is shared by all of them including the one used in this work.

2.2.2 Machine Learning and Colloidal Matter

In the research field of colloidal and especially active matter the studied systems are often much too complex to be grasped by simple models. This demands for data-analysis algorithms that are capable of handling large datasets and make predictions even without the availability of an explicit model[4]. Machine learning therefore seems to be ideal for helping to understand this part of soft matter physics. There are indeed already studies employing machine learning for the improvement of image and video analysis or predicting complex patterns in microscopic motion. While most of the algorithms used in the field so far are relying on supervised learning and are used for analysis, only little has been done using machine learning to generate experimental

data. The application of machine learning and especially reinforcement learning to control the motion of active particles is appealing for two reasons: Firstly, there are big efforts in using active matter for modelling emergent phenomena[28]. RL could help here to find strategies of motion that lead to desired behavior[4], similar to the way it is used in robotics to train multiple agent systems[10]. Secondly, modelling the complex interactions of multiple entities purely by simulations has shortcomings regarding the adequate reproduction of noise in the environment[29]. In active matter, thermal noise is always present and has a significant influence on the motion of particles. Active particles therefore allow for the modelling of the behavior of autonomous individuals like robots with real noise but minimal experimental efforts.

Besides studies on simulation of navigation of active particles with the help of machine learning[30-32], there is to the authors knowledge only one example of a machine learning algorithm physically controlling the motion of objects on the microscopic level[33]. In the mentioned study, a self-thermophoretic active swimmer in form of a microparticle coated with gold nanoparticles was steered using local laser illumination of the particle surface. A RL algorithm using a simple tabular state-action value function as outlined in the previous section controlled the motion of the microswimmer. After training, it was able to navigate the particle to one fixed corner of a two-dimensional grid world as illustrated in Fig 6.

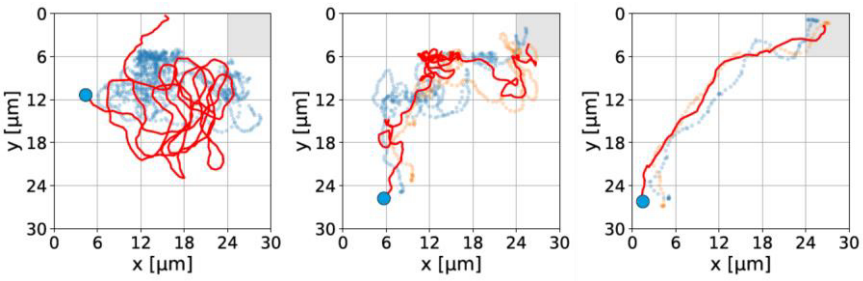

Figure 6: A RL algorithm managed to navigate an active swimmer to one specific field (marked in grey) of a five by five grid world in Ref [33]. The three pictures represent the particles trajectory at different stages of learning.

The aim of the present work was to apply RL control not only to one but multiple particles, which allows for the emergence of collective strategies for solving a task. Namely, an algorithm should rotate an about 100 µm long rod by means of controlling the motion of around twenty microswimmers with diameters of about 6 µm. This task

demanded for a more sophisticated RL algorithm capable of handling the much more complex state space compared to the example above. The algorithm used in this work, Proximate Policy Optimization (PPO), is not only capable of handling complex and continuous state and action spaces but is also very efficient and requires less training time. In the next section, the details of the implementation of machine learning in this work are explained together with an outline of the used algorithm.

2.2.3 Representation of the States and Actions

The microswimmers used in this work should act as "intelligent" individuals, in the same way as a collective of organisms or robot swarms. Every particle should therefore have individual perception and be acting on its own, without paramount knowledge of the current state, namely the exact positions of all other particles and the rod. Figure 7 sketches the model of a particle based RL agent used in this work.

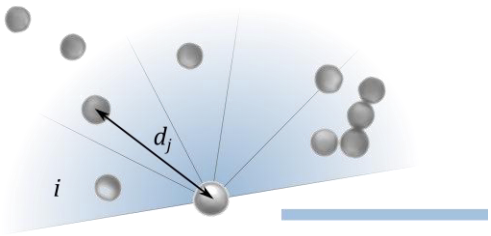

Figure 7: Implementation of sight: Five vision cones i are assigned to each particle. For each vision cone, the RL agent is given five numbers corresponding to the number of particles in each cone, weighed by their distance d_j to the observing particle (Eq. (5)). (Scalebar: 50 µm)

Each microswimmer is given five virtual vision cones. For each cone i, an observable O_i is calculated according to

$$O_i = \sum_j \frac{1}{1 + d_j} \tag{3}$$

with the sum over all particles j within cone i and the corresponding particle distances to the reference particle d_j. This implementation of sight already proved useful in mimicking patterns of collective behavior that are also found in nature[7, 9, 22]. For the

perception of the rod, it was considered to consist of densely packed particles as illustrated in Fig 8. The "rod-particles" were then used to calculate a second, independent set of observables similar to Eq. (3). This made a total of ten numbers representing the RL agents' complete perception of their environment. Based on this ten-dimensional observable vector, the RL algorithm had to decide on one of the actions *propulsion*, *passive*, *rotate right* or *rotate left*. To maximize the learning efficiency, the particles shared the exact same RL algorithm for choosing actions, allowing the algorithm to learn from the experience of all the particles simultaneously. It is obvious that this high-dimensional, continuous state space would not allow for an algorithm relying on a state-action value function as outlined above, simply because there are too many states an agent could encounter.

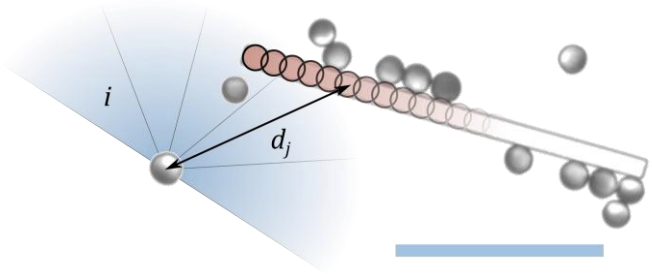

Figure 8: Implementation of the perception of the rod: The rod is considered to consist of many closely packed particles. For every particle, a second set of observables is calculated from these "rod-particles" according to Eq. (3). (Scalebar: 50µm)

2.2.4 The used RL Algorithm (PPO)

In contrast to the example of board games, the state space the RL algorithm was facing in this work was continuous, since all ten observables of each particle could assume any real-numbered value. The way of dealing with such environments in agent-based machine learning is to use a policy, that does not list an action for every state. It rather computes an action directly from the observables using function approximation methods. This is called policy approximation[26]. The algorithm employed in this work for steering the microswimmers was introduced by Schulman et al. in 2017 and is called Proximate Policy Optimization (PPO)[34]. It was implemented for the present work by Emanuele Panizon. PPO is a state of the art algorithm, that has already succeeded in challenging tasks like video and board games[35]. Figure 9 shows a flow

chart sketching the internal training and decision procedures of this algorithm. The key elements are two artificial neural networks (ANN), which are referred to as *actor* and *critic*, respectively. They both feature three hidden layers with 32, 16 and 16 nodes and have ten input nodes for the ten observables. As the name suggests, the actor ANN decides, which action to choose in which state. It does so by producing four output values, which are normalized to yield the probabilities of the four possible actions directly from the ten given observables as inputs. Both ANNs are trained using the given reward and by means of gradient ascent. The critic ANN is used to generate an expected reward according to the current observables. Therefore, it is optimized to precisely predict, which set of observables are yielding which reward. The updates of the actor network rely on the difference of the output of the *critic* and the actual reward. The goal of the *actor's* training is to enhance the probability of actions that led to higher rewards then the *critic* predicted and reduce the probabilities of actions that went worse than expected. The updates of both actor and critic are not done every time the agent decides on a new action. Instead, the observables and rewards acquired over a certain time are saved and then used for multiple update steps. This is necessary, since the observable and reward samples obtained during a single update step are afflicted with strong noise. Only the evaluation of a large set of these quantities allows the RL algorithm to change in the right way to improve performance.

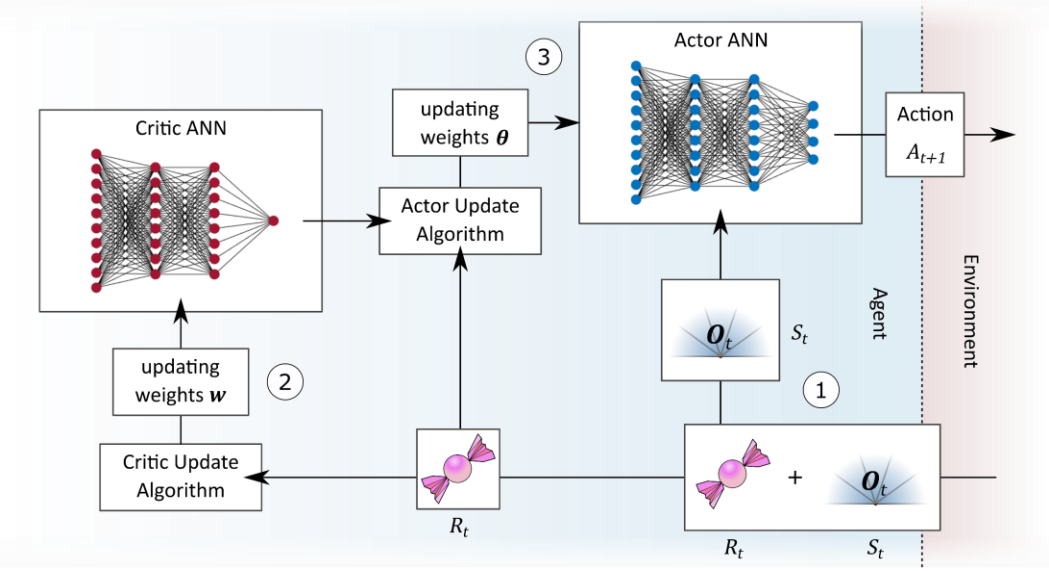

Figure 9: Schematic representation of the RL algorithm used in this work: 1) A reward R_t (symbolized by the sweet) and an observable vector $\boldsymbol{O}_t$ are given to the algorithm. The policy is implemented as an ANN (so called actor) and directly produces the probabilities of the four possible actions A_t as a function of the ten-dimensional observable state. 2) The reward is used to update the weights $\boldsymbol{w}$ of the critic ANN which produces an estimate state value. 3) The weights $\boldsymbol{\theta}$ of the actor ANN are updated using the current reward and the expected reward computed by the critic ANN.

2.2.5 The Implementation of the Reward

Since the reward is the only way of communicating the objective to a RL agent, the ultimate success of learning strongly depends on its definition. In the experiments, each particle was given a reward every time its action was updated. For the task, the particles should solve in this work, the reward R_i of particle i is computed as

$$R_i = 1.5 \cdot H(3.5\sigma - d_i) \; + \; 100 \cdot H(3.5\sigma - d_i) \cdot T_{i,g} \cdot \Delta\Theta \qquad (4)$$

where H is the Heaviside function, σ the particle diameter, $T_{i,g}$ the geometric torque (meaning without any physical units) a particle exerts on the rod and $\Delta\Theta$ the change in the rod's orientation since the last update. The factors 1.5 and 100 in Eq. (4) were chosen to equalize the orders of magnitude of the two contributions to the reward. Particles can get a reward by two means. First, a general reward is given to particles close to the rod, namely closer than 3.5 particle diameters σ ,implemented by the first

term in Eq. (4). This contribution to the reward is intended to encourage the particles to interact with the rod for shorter training times. Additionally, exerting a torque $T_{i,g}$ on the rod is rewarded proportional to the magnitude $\Delta\Theta$ the rod has rotated since the last frame, represented by the second term in Eq. (4). The torque is calculated for every particle according to

$$T_{i,g} = \sin(\theta_i) \cdot \Delta x_i - \cos(\theta_i) \cdot \Delta y_i \tag{5}$$

with the orientation angle of the particle θ_i and the distances of the particle to the rod center in x and y direction Δx_i and Δy_i as shown in Fig 10.

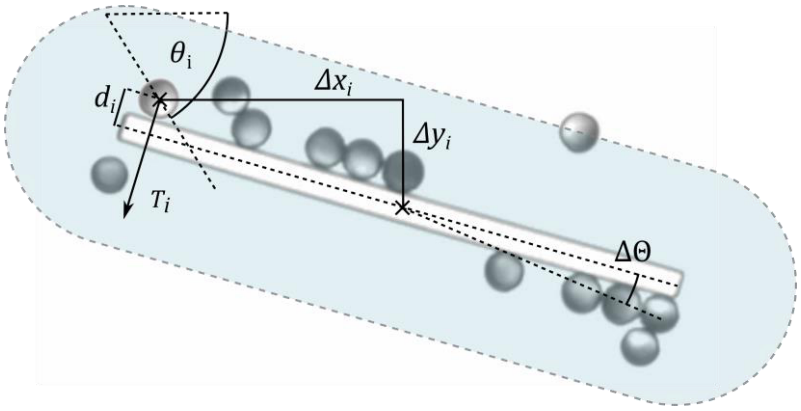

Figure 10: The reward for every particle is made up by a basic contribution for being in the vicinity of the rod (shortest distance d_i smaller than 3.5 particle diameters), as indicated by the blue-shaded region. Δx_i and Δy_i are the positions of particle i in x and y with respect to the rods center and θ_i its orientation. The geometric torque $T_{i,g}$ is calculated from these quantities according to Eq. (5) and rewarded proportional to the rods rotation $\Delta\Theta$ since the last update.

3 Results and Discussion

In this work, a RL algorithm was given the control over the motion of multiple microswimmers. These particle based RL agents ought to collectively solve the task of rotating a rod shaped, Brownian particle. For this task, each of them could only use the information about its environment gathered via five virtual vision cones and choose from the set of actions: *propulsion*, *passive*, *rotate left* and *rotate right*. The next sections cover the results of these experiments, starting with the fabrication of the used particles and rods. Then the result of the learning process is investigated and compared to a simulation. By changing one of the experimental parameters, the efficiency of the particles' effort was enhanced. In the last section, the experiments are evaluated from a different point of view. Here, the rod was used as a probe to determine the propulsion force of a single microswimmer.

3.1 Fabrication of Microswimmers and Rods

As described above, the microswimmers used in this work consist of silica colloids half coated with carbon. For their fabrication, SiO_2 microspheres ($\sigma = 6.27\,\mu m$, Microparticles GmbH) were washed with deionized water and then dripped on microscope slides in the form of a dilute dispersion to yield particle monolayers upon drying. Afterwards, a carbon coater (Leica EM ACE600) was used to deposit an 80 nm carbon layer on the particles, leaving their down-facing side uncoated. The now so-called *Janus* particles were removed from the microscope slides in water with short sonification pulses. Rod shaped particles with a length of 96 µm and a 4 µm x 4 µm cross-section were printed using a 3D printer with sub micrometer resolution (Photonic Professional GT, Nanoscribe GmbH). For the experiments, the rods were stabilized using the surfactant Ploronic F127. Samples were prepared by mixing and washing small amounts of Janus particles and rods in an aqueous solution of 28.6 wt% 2,6-lutidine. After washing, the colloids were redispersed in the same critical mixture and filled into a flat silica cell with a thickness of 200 µm (HellmaAnalytics High Precision Cell).

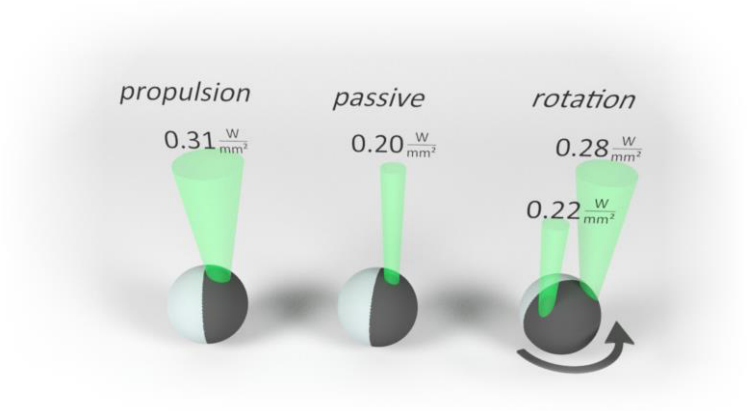

Figure 11: The illumination used for the different actions from left to right: *propulsion* using one bright laser spot on the cap, *passive* with a spot of low intensity on the cap and *rotations* employing two spots of different intensity on either side of the cap.

3.1.1 Characterization of the Microswimmers' Motion

Since in the RL experiments the particles ought to execute the actions *propulsion*, *passive*, *rotate right* and *rotate left*, the particles' response to these actions was tested. The uncoated side of the swimmers is considered their front since with the utilized illumination intensities, the particles move in this direction. Laser spots with different intensities and positions with respect to the particle center were used to realize the four actions (illustrated in Fig 11): For *propulsion* and *passive*, the capped side of a particle was illuminated using a single spot with intensities of 0.31 W/mm^2 and 0.20 W/mm^2, respectively. In the case of the *passive* action, the illumination served the purpose of preventing the particles rotational diffusion. For rotations, the left and right side (with respect to the particles orientation) were illuminated with different intensities. A low intensity beam (0.22 W/mm^2) was pointed to the side, the particle should rotate to and a high intensity beam (0.28 W/mm^2) to the side the particle should rotate away from. All laser spots had a beam waist of about 10 µm. Over the course of 20 seconds, the illumination of each particle was kept constant with respect to its position, before switching to the next action. The rotation was restricted to $\pi/3$, meaning that whenever a particle had rotated by that angle, its action was changed to *passive* for the rest of the 20 seconds. For half an hour, random actions were assigned to each particle and the particles motion recorded on a microscope video. The particles' responses were evaluated from the videos by constructing a trajectory for

each particle and comparing the particle's position and orientation before and after each action. The results of this experiment are shown in Fig 12 in the form of histograms displaying the magnitudes of translation and rotation during all actions, respectively. In the case of the actions *propulsion* and *passive* only the events in which a particle was further apart from every other particle than one particle diameter σ (6.27 µm) were considered. This should prevent particles from blocking each other's motion. The distributions of the particle displacements during both actions *propulsion* and *passive* show a distribution with a mean displacement of 11.4 ± 6.5 µm and 5.9 ± 3.7 µm, respectively. The large standard deviations show that the particles' response to an action is not deterministic but varies between particles and from time to time. During the action *passive* the particles do move due to the weak illumination. Even though this was not a desired behavior, the weak illumination was important to retain the particles orientation during this action. The small displacements were tolerable, since they were on average smaller than a particle diameter. The results of the rotational motion of the particles show sharp distributions for both the actions *rotate right* and *rotate left* at angles of $-\pi/3$ and $\pi/3$, respectively.

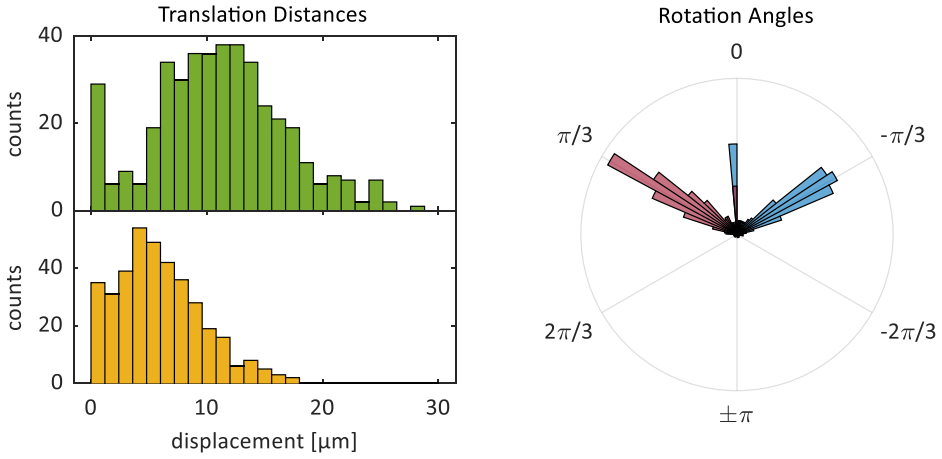

Figure 12: Evaluation of the particles' responses to the different actions. Left: histograms of the particles' displacement during *propulsion* (green) and *passive* (yellow). Right: Histograms of the change in orientation during the actions *rotate right* (blue) and *rotate left* (red). For all actions, the illumination was kept constant with respect to the particles position for 20 seconds.

The histograms in Fig. 12 show that there were some rare events that led to no motion at all. From the experiment videos it is evident that these events were caused by

particles temporarily sticking to the substrate or possessing an orientation in which the propulsion mechanism did not work. Overall however, the particles responded with the desired translations and rotations to the four different actions. This allowed for proceeding to the introduction of the rod-shaped, Brownian particles, the microswimmers were intended to interact with.

3.1.2 Interaction of Microswimmers with Brownian Rods

Besides the particles' motion, their capability to move the rod by means of their propulsion was a crucial requirement for the success of this work. To find out, whether this was possible, some particles were steered against a rod from one side. Figure 13 shows a series of snapshots showing the rod and the particles pushing it. When the particles started to touch the rod, it began to move with significant speed in the direction in which the particles were pushing it. The rod did slow down the particles compared to their free motion, but even as little as three particles were enough to bring the bigger object to constant motion. Also, there was no aggregation of the particles at the rods or clustering of the rods themselves. Thus, there were no modifications regarding the preparation of the microswimmers and rods necessary, before starting the RL experiments explained in the next section.

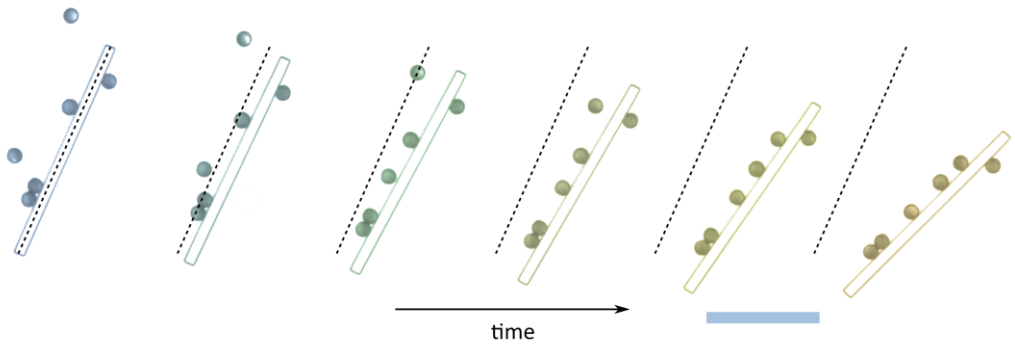

Figure 13: Snapshots of a rod and particles pushing it (shaded in a different color in each frame) over the course of two minutes. The pictures were displaced for clarity and the dashed lines represent the position of the rod in the first frame. (Scalebar: 50 µm)

3.2 Reinforcement Learning Experiments

In the RL experiments, a machine learning algorithm was trained in steering microswimmers to rotate a rod with about 100 µm in length. For this task, 20 to 25 microswimmers were selected. Special care was taken to include only particles, that reacted properly to the actions described above. During the experiments, the particles had to stay within a circular area with 260 µm in diameter. When one of the swimmers reached the edge of this measurement field, it was driven inside again, until it was closer than 115 µm to the center. The RL algorithm was trained within several runs. At the start of each run, the swimmers were used to push the rod to the middle of the measurement circle. Before the control of the particles' motion was given to the RL algorithm, the microswimmers were placed in two rows close to the rod as a starting configuration (Fig. 14). A run was terminated when one of the following events occurred: 1) The particles had a mean distance of more than 65 µm from the rod, since the particles could only learn by interacting with the rod. 2) The particles had a mean distance bigger than 70 µm from the center of the measurement circle. This was intended to inhibit the aggregation of particles at the edges. 3) The rod was displaced further than 90 µm from the center. This should prevent the particles from pushing the rod to a position from which they were not able to push it back to the middle again.

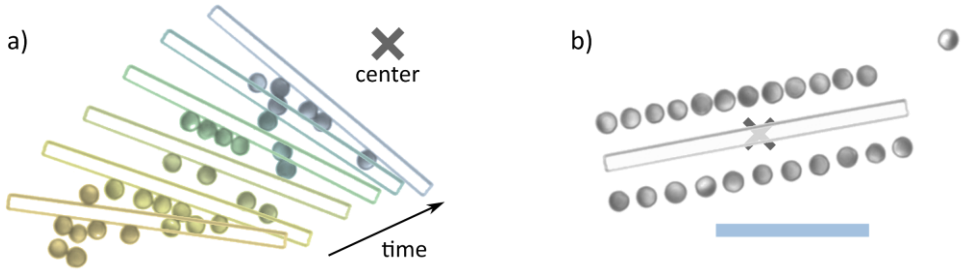

Figure 14: Initialization of the RL experiments: a) The microswimmers were used to push the rod to the middle of the measurement circle (illustrated by the cross). b) When the rod reached the middle, the particles were positioned close to the rod before starting the experiments. (Scalebar: 50 µm)

During the experimental runs, the particles' actions were updated every 20 seconds. This time allowed the particles to significantly move the rod in between two updates. It is important that the rods orientation changes are not dominated by noise for the adequate determination of rewards. On average, the particles moved around two

particle diameters within one *propulsion* action, as discussed above. The magnitude of rotation was limited to $\pi/5$ during one action, which is the angle of one vision cone. This should prevent too large changes in the observables during one action. For updating the actions, the observables corresponding to the vision cones and a reward according to Eq. (4) were computed for every particle. These values were handed to the RL algorithm for deciding on the next actions. The RL algorithm was trained every 10 minutes. Then, it used all the observables and corresponding rewards acquired during this time to update the actor and critic ANNs.

3.2.1 Successful Rotation of the Rod

The most natural quantifier for the learning progress of a RL agent is the reward[26]. Since the reward always must be designed to have higher values when the RL algorithm performs better, a rise in reward is always connected to an improvement in performance. Particles were rewarded for both close proximity to the rod and exerting a torque on it, as defined in Eq. (5). Figure 15 shows the time evolution of the total reward R_{tot} of all particles per update step and its average value R_{av} for the first seven runs of the RL experiments. All curves show a decreasing reward R_{tot} for the first 5 to 20 minutes. This stems from positioning the particles close to the rod prior to each run, where they were rewarded for their mere presence. Since some of the particles eventually leave the rewarding zone around the rod, the overall reward per update step drops. After this initial phase, the curves fluctuate around an average value. This average reward per update step R_{av} increases from around 15 per update step during the first run to around 40 in the last three runs (inset in Fig 15). The fluctuations in R_{tot} increase in magnitude, too. A possible cause of this could be more and more particles interacting with the rod during the training. The rising rewards indicate that the particles' performance improved during these seven runs. However, there is no clear saturation in the reward, and it cannot be sure, if the learning process was completed.

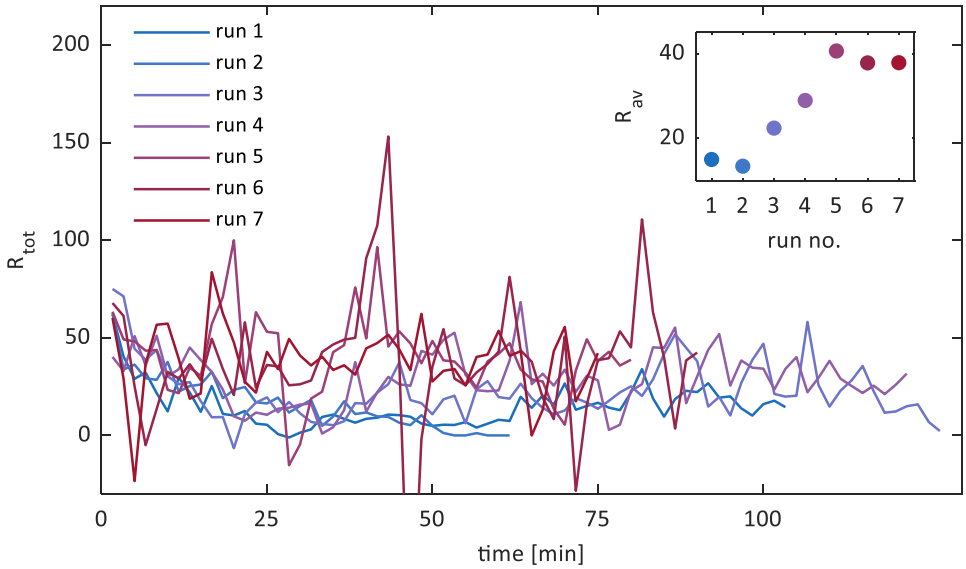

Figure 15: Main graph: the ensemble reward R_{tot} (summed over all particles), averaged over five update steps. Inset: the mean reward R_{mean} (averaged over the whole run) for the first seven runs of the RL training.

Even though the reward did clearly not saturate, the experiment videos confirmed that the particles learned[1] to rotate the rod and thus the goal of these experiments was reached. Figure 16 shows snapshots of the swimmers' efforts while rotating the rod which reveal the particles' strategy. The particles push from both sides against the opposite ends of the rod. This is probably the most intuitive way for solving the given task. The particles rotated the rod clockwise in all runs of this experiment. In another learning cycle, the particles preferred to rotate the rod anti-clockwise. Since the reward was implemented symmetrically with respect to the direction of rotation, this choice was random. The reward promotes particle efforts that align with the least change in rod orientation. Therefore, it is likely that a small random rotation of the rod gets amplified. Keeping this in mind, it appears most probably that the preferred direction develops due to random fluctuations in the beginning of training.

[1] Of course, the silica colloids themselves in this experiment did not learn anything. However, since each particle was steered on its own based on information gathered from it's perspective, the resulting situation does not differ from particles possessing an internal computation unit and acting on their own.

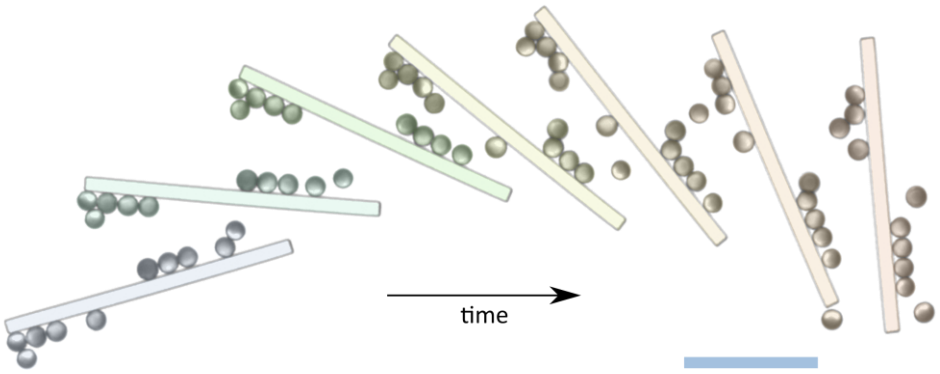

Figure 16: Microswimmers pushing the rod to a rotation. The snapshots were displaced and colour-shaded for better visibility. During the depicted time span of 4 minutes, the rod's centre of mass did not move more than 4 µm from its original position. (Scalebar: 50 µm)

To investigate the learning progress from a different perspective than the reward, the rods rotation was quantified. In Fig 17 the time development of the rod's orientation Θ over time throughout the different runs is shown together with its average angular velocity ω_{av} in each run. During the first two runs, the orientation of the rod did not change more than $\pi/2$. From the third run on, the curves of Θ against time get steeper and the angular velocity of the rod increases from run to run. Eventually, ω reaches a maximum in the sixth run, during which the particles managed to turn the rod by more than two full rotations. This development reflects the change in the average reward. It can therefore be concluded that the reward was defined in the right way to promote particle motion that would lead to a rotation of the rod.

Both the reward and the rod's rotation only represent the ultimate result of the particles' efforts but shed no light on the way the particles achieved it. The next section covers results, that allow for deeper understanding of the learning process from the perspective of the particles' behavior that finally allowed them to turn the rod.

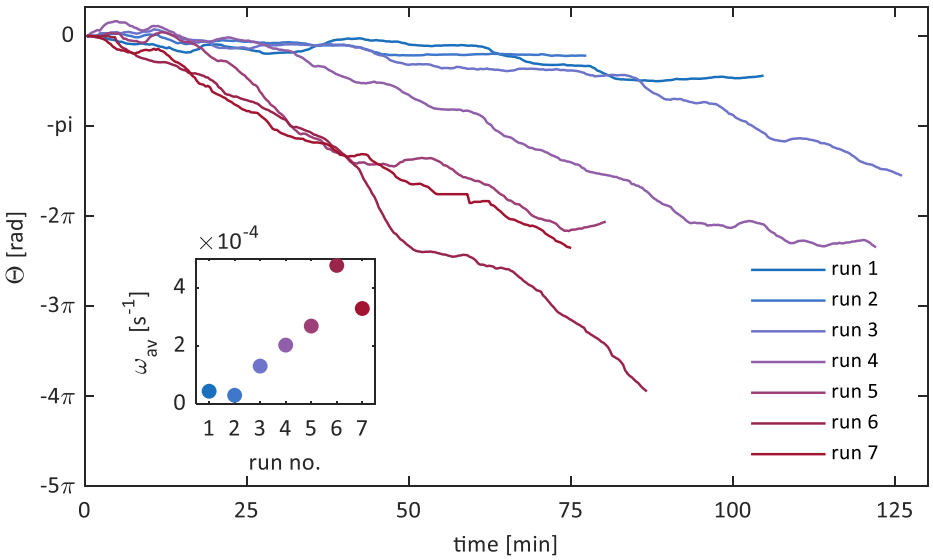

Figure 17: Time development of the rods' orientation Θ during the seven runs of training. The inset shows the average angular velocity ω_{av} of the rod for each run.

3.2.2 Results of Training in Terms of Particle Behavior

It is intuitive that for eventually being able to rotate the rod, the particles at first have to get close to it. This is why the reward defined in Eq. (4) featured a contribution from the mere presence of a particle close to the rod. To find out, whether the particles did indeed learn to stay close to the rod, the particles' distances to the rod D during training were evaluated. D in this case is the shortest distance of a particle to any point of the rod's long axis. Further, the average fraction f_R of particles inside the reward zone was evaluated for every run. Figure 18 shows the results of this analysis in the form of a probability distribution of D together with the evolution of f_R throughout the runs. For the first two runs, the particle-rod distances D are evenly distributed between 0 µm and 100 µm. The probability to find a particle at even larger distances to the rod is very small, due to the finite size of the measurement circle. For the runs later in training, the curves show a decreasing probability to find particles far away from the rod. Simultaneously, a strong increase in probability for distances that are smaller than the reward cutoff of 22 µm (black dashed line in Fig 18) is visible. The values of f_R show the same trend. This fraction increases from around 0.2 for the first run to around 0.6 for the last of the seven runs. These results show that the swimmers learned to stay close to the rod during training. As previously pointed out, this was a first important

step, since only if the particles interacted with the rod, they could learn how to rotate it.

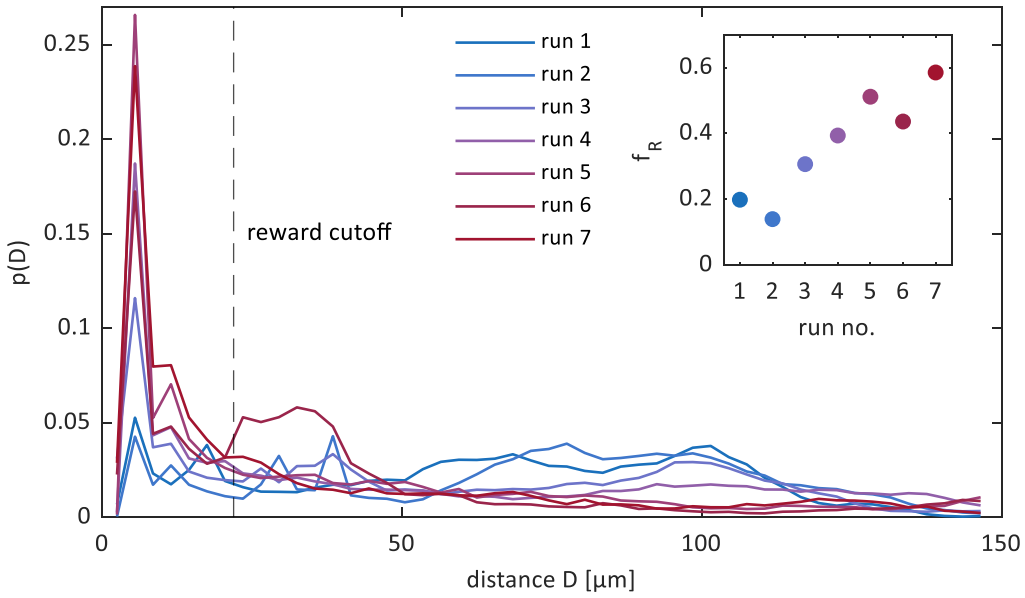

Figure 18: Probability distributions of the shortest distance D of a particle to any point of the rod's long axis. The black dashed line displays the distance up to which a particle was still rewarded for its proximity to the rod. Inset: The average proportion of particles f_R that were closer to the rod than the reward cutoff.

One peculiar feature which is shared by all the distributions in Fig 18 is a spike at a distance of about 5.5 µm. This peak arises from the particles, which are in direct contact with the rod. The distance of these particles to the rod's long axis is determined by the sum of half a particle diameter ($\sigma/2 = 3.14$ µm) and half a rod diameter (2 µm). It is well known that even in absence of steering control, active Brownian swimmers accumulate at solid surfaces[36] due to their slow reorientation times compared to their velocity. This general behavior of active swimmers was probably helping the particles in the experiments to stay in the vicinity of the rod.

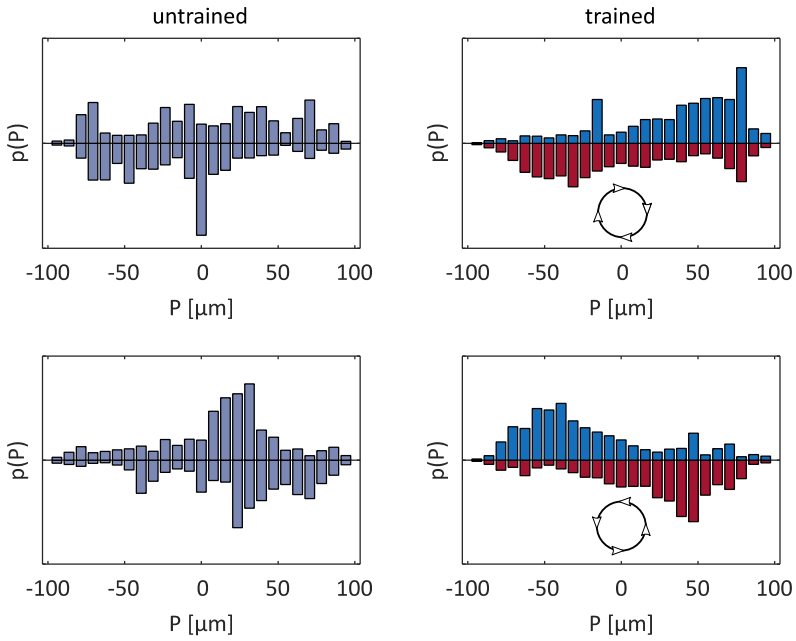

Figure 19: Probability distributions of the microswimmers' position P along the rod at the beginning (purple) and the end of the training (red and blue). The distributions above and below the x-axes account for particles on different sides of the rod. For the particles below the rod, the distributions are mirrored at the x-axis. The upper and lower graphs correspond to experiments, in which the microswimmers turned the rod clockwise and anti-clockwise, respectively. The circular arrows indicate the direction of rotation.

Obviously, staying close to the rod is not a sufficient strategy for rotating it. One can expect that the way the particles distribute around the rod plays a role in the solution of that task. Therefore, the probability distribution of the particles' position P along the rod was analyzed before and after training. For this, the rod was considered to be oriented horizontally in each frame. This divided the microswimmers in two groups above and below the rod, respectively. Then, the particle positions were evaluated separately for the two groups with respect to the rods position. The results are shown in Fig 19 for particles that turned the rod clockwise and anti-clockwise, respectively. Before learning, the microswimmers do not prefer any part of the rod. In this case, the particles are randomly distributed over the whole length. The distributions corresponding to successful experiments on the other hand, show a higher probability for finding a particle at the ends of the rod than at the middle. For the two sides of the rod, the distributions are anti-symmetric. In experiments, where the rod was turned clockwise for example, particles above the rod favored to stay at its left end and particles below it favored to stay at its right end. These distributions reflect an intuitive

way of solving the task efficiently, since at the ends of the rod a particle can exert the highest torque using its limited force. This strategy was by no means implemented in the reward definition or any other information given to the RL agent. It was therefore spontaneously emerging from the RL algorithm adopting its behavior to the task.

Another, more direct indicator of the particles' efforts to rotate the rod is the torque the particles exert on it. Since the real torques are not easily evaluable, the geometric torque T_g already used in the reward definition (Eq. (5)) was analyzed. This quantity was obtained from the experiment videos by evaluating the particles' distance d_j to the rod center and orientations ϕ_j relative to the rod. The total geometric torque $T_{tot,g}$ was obtained for every video frame using

$$T_{tot,g} = \sum_j d_j \sin(\phi_j) \tag{6}$$

where the summation is over all particles j in direct or indirect contact (via another particle) to the rod. For this evaluation, only particles were considered that performed a *propulsion* action. Figure 20 shows probability distributions of $T_{tot,g}$ and the absolute value of its average $|T_{av,g}|$ for each of the seven training runs. The distributions for the first two runs are sharp peaks at $T_{tot,g} = 0$. With more training, the distributions become broader with higher probability for non-zero torques. Additionally, the center of the distributions shifts to negative values. This asymmetry is necessary for a rotation of the rod, since symmetrically distributed torques would cancel out. Positive angles were defined in the anti-clockwise direction and therefore the shift to negative values reflects the clockwise direction in which the particles turned the rod in this experiment. The average torque $|T_{av,g}|$ confirms that the net torque on the rod changed from values close to zero in the first three runs to finite, increasing values during training. Contrary to the previously discussed indicators, that only rely on the particles' positions, $T_{tot,g}$ also considers the particles' orientation. Its increasing average shows that the RL algorithm also learned to control the particles orientation and employed this parameter to solve the task. The time development of the torques agree very well with that of the rotational velocity of the rod and the reward, that were analyzed in the last section.

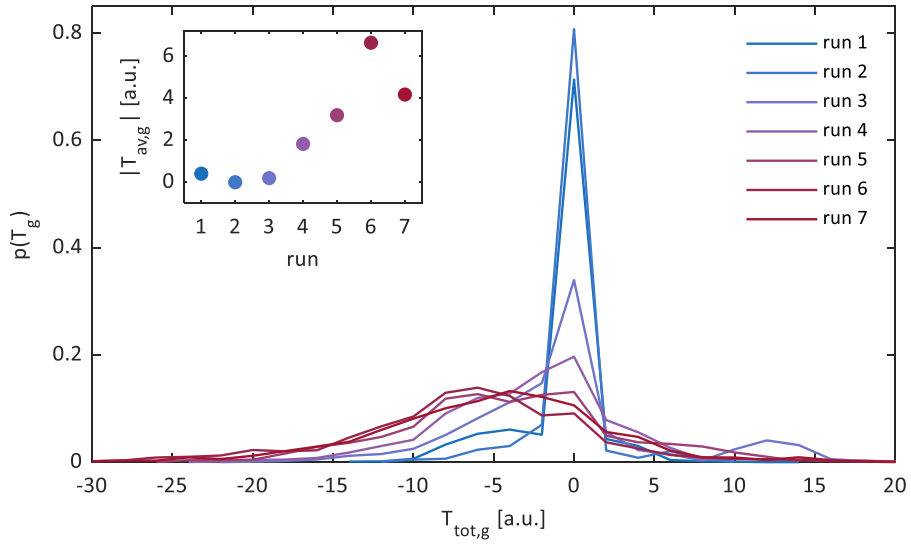

Figure 20: Probability distributions of the total geometric torque $T_{tot,g}$ the particles exerted on the rod during the different runs. Inset: the absolute value $|T_{av,g}|$ of the mean torque plotted against the run number.

The particles distance to the rod D, their distribution along the rod $p(P)$ and the torques they exerted on it $T_{tot,g}$ all changed during training. In conclusion, the particles' strategy can be broken down into three key steps: 1) The particles learned to stay close to the rod. This behavior was also induced by the reward definition to make the interaction with the rod attractive. 2) The microswimmers learned to prefer the rod's ends, which makes sense for maximizing the exerted torque. Since the reward did not feature any term connected to the particle distribution along the rod, this behavior evolved spontaneously during training. 3) Additional to their positioning, the particles learned to orient towards the rod and to push against it, which ultimately allowed them to turn the rod. These behaviors evolved over time, parallel to the mean angular velocity ω_{av} and the mean reward R_{av} during each run. This poses the learning process to be a gradual improvement rather than a sudden breakthrough. All the indicators of the particles' learning progress still show imperfections after training. For example, the highest average fraction f_R of swimmers inside the rewarding zone around the rod was only 60 % and the torque distributions still show some torques opposing the rod's rotation even in the last run. It is therefore probable that even better results regarding the rod's rotation could be obtained with longer training times. Longer experiments proved difficult due to changing experimental conditions, that led to sample drifts or other disturbing influences. Achieving longer training times with steady conditions

should be one of the aims of further studies combining this microswimmer system with RL.

In conclusion an RL agent was successfully trained to rotate a rod by steering multiple active swimmers during the experiments discussed above. The algorithm was able to solve this task without any knowledge of the overall situation, just relying on limited perception in the form of five vision cones per particle. In the next section, the outcome of a simulation of the same system is compared to the experimental results.

3.2.3 Comparison to Simulation

As already pointed out in a previous section, the particles' responses to the actions in the RL experiments were not deterministic but rather divers. It is intuitive that e.g. a particle not moving forward when it is ought to diminishes the overall efficiency of the particle's attempts to rotate the rod. Therefore, it can be anticipated that more defined particle responses would ultimately lead to a more efficient solution of the task. For comparison, simulations in which the particles did response deterministically were done. In these simulations, the same RL algorithm was employed, also using only five vision cones as observables and the previously mentioned four actions. All the simulations for this work were done by Emanuele Panizon. Figure 21 shows representative snapshots of a simulated run after 150 training runs.

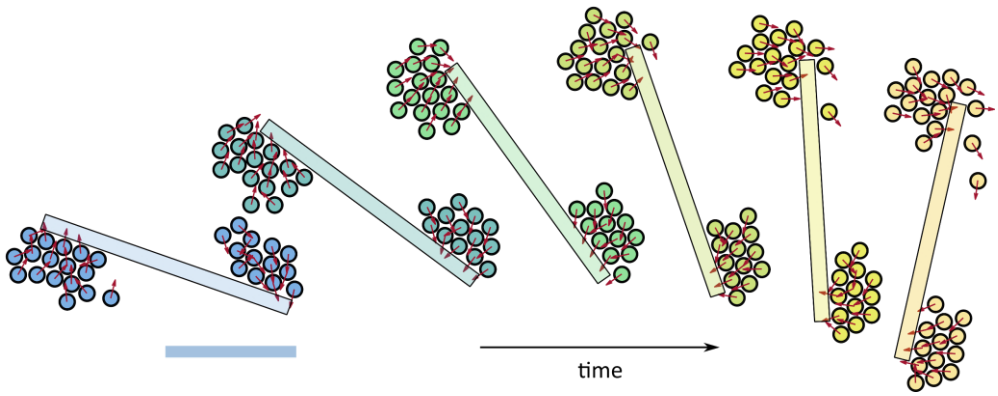

Figure 21: Snapshots of a simulated trajectory after training for 150 runs. For clarity, the frames were displaced and colored; the rod's center of mass did not move more than 1 µm within the depicted time span of 8 minutes of equivalent time. The particles' orientations are indicated by the red arrows. (Scalebar 50 µm)

In simulation, the trained particles show nearly perfect behavior regarding the indicators discussed in the previous section: All particles are in contact with the rod directly or via other particles. The swimmers form two closely packed groups at the ends of the rod on opposite sides. Also, the particle orientations are very well aligned and pointing towards the rod end. This maximizes the torque the particles exert on the rod. Overall, the situation in Fig 21 can be considered to be a nearly optimal solution to the problem of rotating the rod. This shows that the RL algorithm and its implementation with limited perception and discrete actions of the swimmers did not pose a limit to the solution efficiency in the experiment. It is more likely that the imperfections in the particle responses or the limited training time restricted the success of the particles' efforts.

Interestingly, the situation after only seven training runs in the simulation did not differ too much from the seventh and last training run in the experiment. Figure 22 depicts snapshots of both the simulation in this stage and the experiment. In both, only a part of the microswimmers stays gathered at the rod. Further, the real and simulated swimmers close to the rod show preference for the ends. However, in both cases the microswimmers are not perfectly accumulated there, but still scattered along the rod, contrasting the simulation after exhaustive training (Fig 21). Therefore, simulated and experimental particles with the same training time show similar behavior. This indicates that despite the success in turning the rod, the learning of the particles in experiment had probably not converged to its final state yet. In conclusion, longer experiments could achieve even better results. This result also begs the question, whether the diverse action responses of the experimental microswimmers disturb the training or not. If deterministic responses would facilitate learning, one would expect the simulation to show better results than the experiment with similar training times. Longer training of the algorithm in experiments could also answer this question, since it would allow for a quantitative comparison between the learning progress in simulation and experiment and differences in the final states could be investigated.

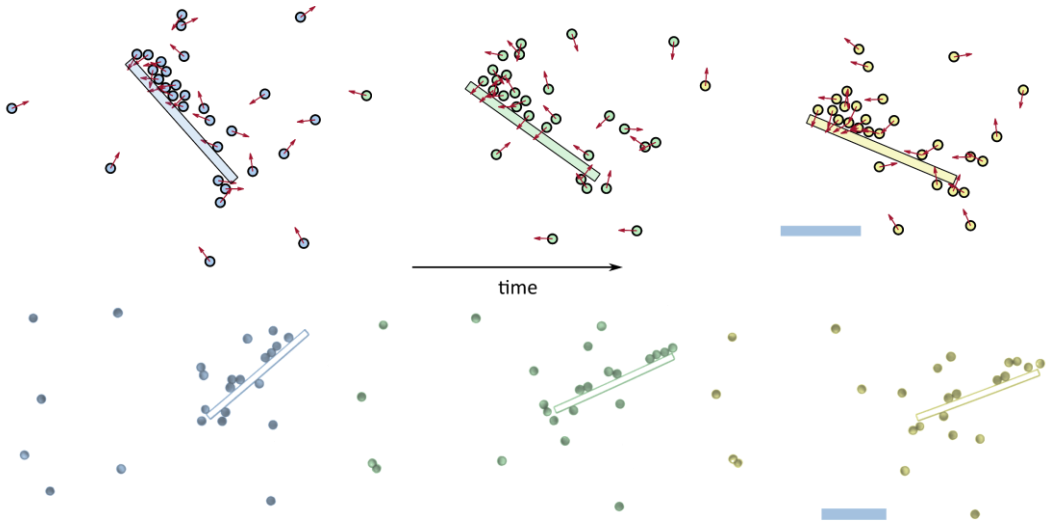

Figure 22: Snapshots of the seventh training run in both simulation (to row) and experiment (bottom row). For clarity, the snapshots are colored and for simulations, the particle orientations are indicated by the red arrows. (Scalebars: 50 μm)

3.2.4 -Improving the Efficiency of the Particle Efforts

When comparing the motion of particles in simulation and experiment, one significant difference stood out. In both cases, particles that did not push perpendicular against the rod, sled along it until they slipped of it's end. However, in simulation particles in this situation were soon able to reorient and move to the other end of the rod. In the experiment on the other hand, particles that sled of the rod often moved far away from it and took much longer time to find their way back to one of the rod's ends. Figure 23 illustrates this difference with snapshots of simulated and real particles in this situation. The reason for this shortcoming of the real particles was found to be a different ratio of translation to rotation during one action. The particle's rotation was limited to $\pi/5$ in both simulation and experiment since this was the angle of one of their vision cones. But while in simulations the particles traveled only around one diameter during a *propulsion* action, the real microswimmers moved on average about two diameters. If a particle that is trying to reorient does not only choose rotation actions but also moves forward, this results in a larger displacement, if the translation distance during one action is larger. Thus, the real swimmers tended to get further away from the rod when slipping of it. It is obvious that this shortcoming of the real particles diminished their efficiency of turning the rod.

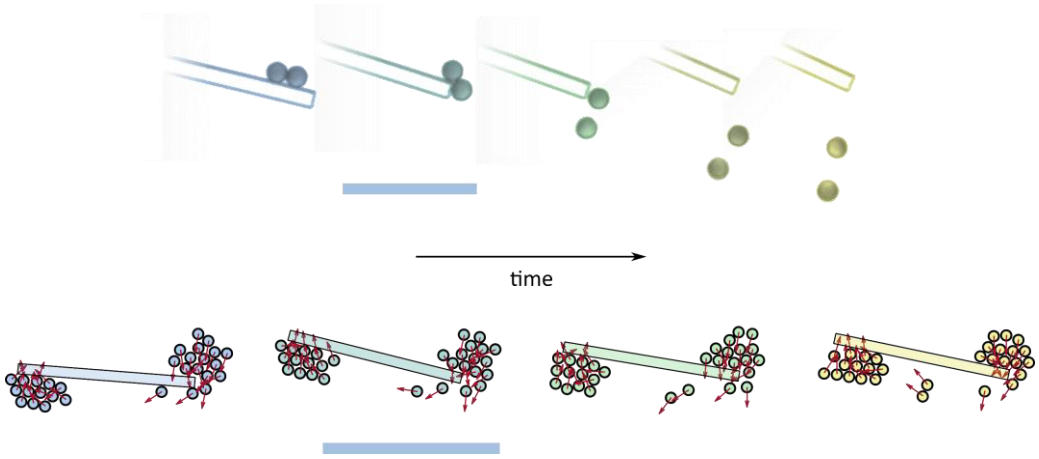

Figure 23: Snapshots of particles sliding of the rod in experiment (top row) and simulation (bottom row). The pictures are displaced and colored for clarity. The red arrows indicate the particles' orientation in the simulation snapshots. (Scalebars: top: 25 µm, bottom: 100 µm)

To prevent the microswimmers from displacing too far from the rod when slipping of their translation distances were reduced. This was achieved by reducing the time of one action from 20 seconds to 10 seconds. With this new action time, the particles were still able to rotate by $\pi/5$ in one action, but the translation decreased significantly. Figure 24 shows histograms of the displacements during *propulsion* and *passive* actions with a time of 10 seconds between two updates. The average translation distances during both *propulsion* and *passive* actions decreased by about 50 % to 6.0 µm and 2.7 µm, respectively. A new learning cycle was started with the action time set to 10 seconds. Figure 25 shows snapshots of the last run of training. It is evident from the videos of this experiment that the particles had far less troubles to stay in close vicinity of the rod. The average fraction f_R of particles inside the rewarding zone around the rod during the last run was as high as 0.85. This was about 40 % higher than the highest value for f_R (0.6) achieved by the particles trained with an action time of 20 seconds. The shorter translation distances also manifested in a higher average torque on the rod, since more particles were interacting with it (compare Fig 25 and Fig 16). The highest value of $|T_{av,g}|$ doubled with respect to the measurements using the longer action time. Reducing the action time had not only positive impacts on the performance of the RL algorithm. With the action time set to 10 seconds, it took about eight times as much training before the particles successfully rotated the rod. Additionally, the learning indicators investigated above did not continuously show a trend towards better performance, giving evidence for phases of

regression during training. As previously pointed out, the 20 seconds for one action was chosen to allow for significant motion of the rod during two updates. The rod's rotational noise probably contributed stronger to the rewards with less time between two updates. To profit from both the faster learning and the more precise steering, one should use a gradually decreasing action times in further experiments. In summary, the shorter action time allowed the particles to turn the rod much more efficient. The reduced translation distances allowed the particles to reorient with less displacement and therefore steer more precisely. The learning process itself however, took much longer time than before.

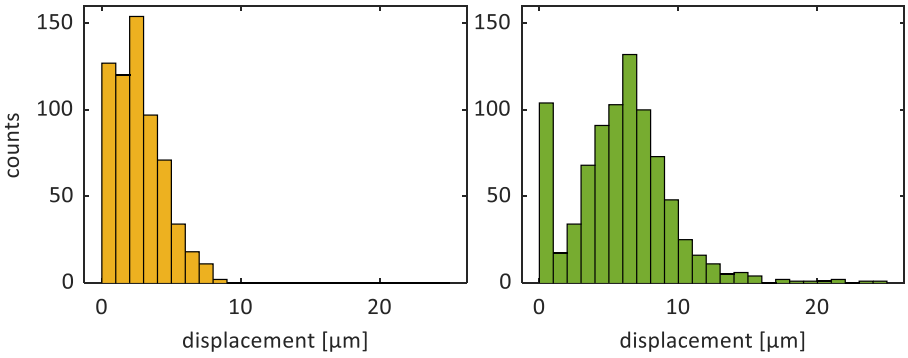

Figure 24: Translation distances of microswimmers in *passive* actions (left, yellow) and *propulsion* actions (right, green). The average translation distances are 2.7 µm and 6.0 µm, respectively.

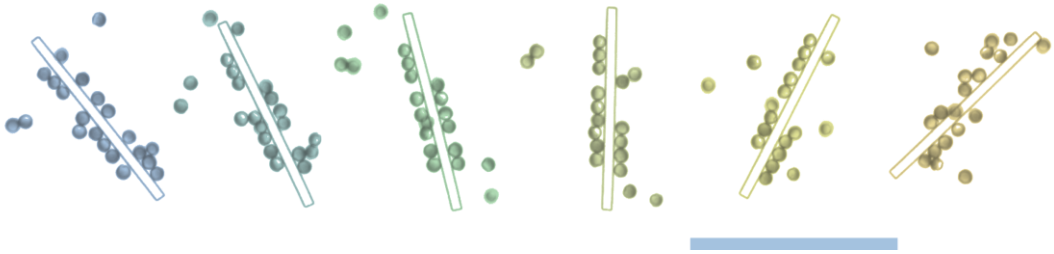

Figure 25: Snapshots of the particles trained with an action time of 10 seconds rotating a rod. The pictures were displaced and colored for clarity. During the depicted 4 minutes, the rod's center of mass did not move more than 4 µm. (Scalebar: 100 µm)

In summary, the combination of the controlled motion of active particles with an RL algorithm proved successful. It could be shown that the particles learned to stay close to the rod, position at its ends and then push against it in order to turn it. There is only one other example[33] achieving the experimental control of particle motion using RL and to the authors knowledge, the present work is to date the first microscopic multiagent system controlled by RL, that is implemented experimentally.

3.3 Determination of the Microswimmers' Pushing Force

In free motion, the propulsion force of an active swimmer can be determined from its velocity and hydrodynamic friction coefficient via the Stokes-relation. One could naively assume that the pushing force exerted by a microswimmer on an obstacle was the same as the propulsion force. However, the flow fields around a particle necessary for propulsion are influenced by nearby objects[28]. This can alter the force in situations, where the particle is pushing against another object like the rod in the RL experiments discussed above. Because their determination is by no means trivial, only little is known about the pushing forces of microswimmers. During the evaluation of the RL experiments, it stood out how strongly the total geometric torque $T_{tot,g}$ exerted by all particles on the rod, correlated with the speed of rotation of the rod ω (see Fig 26). Since the factors relating these two quantities include the pushing force of a microswimmer, this correlation presented the opportunity to use the rod as a probe for the determination of this force. In the next section, the determination of the pushing force of an active swimmer is discussed, starting with a model of the rod used for the calculations.

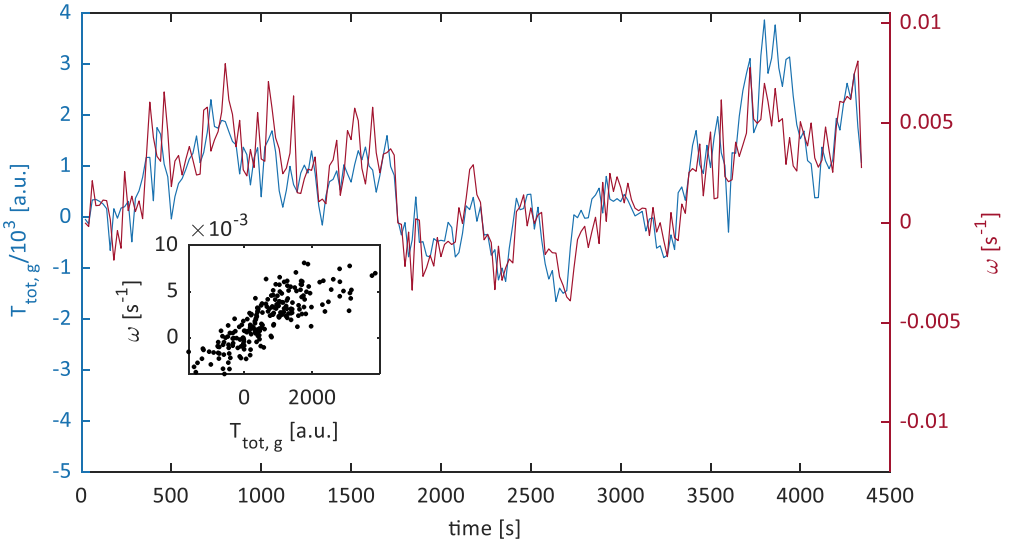

Figure 26: Time evolution of the total geometric torque $T_{tot,g}$ and the rotation velocity ω of the rod. The two quantities show very good correlation. Inset: Plot of ω against $T_{tot,g}$ during every update step, indicating a linear relationship.

3.3.1 Assumptions and Rod Model

The total geometric torque $T_{tot,g}$ is the torque T_{tot} the particles exert on the rod normalized to the pushing force F_{as} of an active swimmer:

$$T_{tot,g} = \sum_j d_j \sin(\phi_j) = \frac{T_{tot}}{F_{as}}, \tag{7}$$

where $j = 1, 2, 3, \ldots$ are the particles in direct or indirect contact to the rod and d_j and ϕ_j are their distance to the rod's center and orientation relative to the rod, respectively (see Fig 27). Using Eq. (7), $T_{tot,g}$ and the rod's angular velocity ω can be related by

$$\omega = \frac{T_{tot}}{R_r} = \frac{T_{tot,g} \cdot F_{as}}{R_r}, \tag{8}$$

where F_{as} is again the particle pushing force and R_r the rotational friction coefficient of the rod. Plots of ω against $T_{tot,g}$ do indeed show a linear relationship (see inset in

Fig 26). $T_{tot,g}$ and ω can be obtained from the experiment videos. With this data, Eq. (8) together with the knowledge of R_r would allow for the determination of the pushing force of a single active swimmer F_{as}. One way to measure the rod's rotational friction coefficient would be to observe it in free diffusion and calculate R_r from its rotational diffusion coefficient using the Einstein relation. However, this method proved unfeasible, because the rods did not diffuse to an evaluable extend due to their large size. Therefore, it was assumed that the rod's friction coefficient can be approximated by summing up the friction coefficients of its segments. For this, the rod was considered to consist of a row of cubes, whose friction coefficients γ_c were determined from their diffusion coefficients. This model allowed for the determination of R_r from the friction coefficients of the cubes and is sketched in Fig 27.

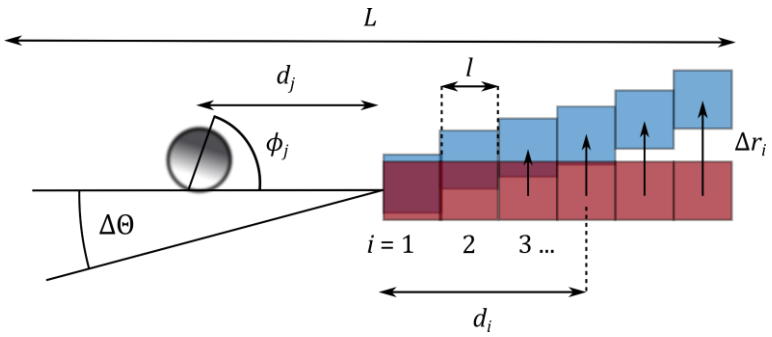

Figure 27: Principle of the approximation of the rotational friction coefficient of the rod R_r: The horizontal line and the red squares i with side length l at positions d_i represent the rod with length L in its initial position. After a small rotation $\Delta\Theta$, the rod is represented by the blue cubes. The rotation of the rod was considered to consist of translations Δr_i of the subunits. By summing up the frictional forces of the cubes and their lever to the center of the rod, the torque necessary for the rotation can be approximated. ϕ_j and d_j are the angles of microswimmers with respect to the rod and their distance to the rod center, respectively.

In the used model, to rotate the rod by an angle $\Delta\Theta$, one would have to translate all the cubes $i = 1, 2, 3, \ldots$ by a small distance Δr_i. The torque T_{tot} necessary to rotate the rod with a certain angular velocity $\omega = \frac{\Delta\Theta}{\Delta t}$ can therefore be expressed as a sum over the frictional forces of all cubes:

$$T_{tot} = R_r \cdot \frac{\Delta\Theta}{\Delta t} = \sum_{i=-\frac{L}{2l}}^{i=\frac{L}{2l}} \gamma_c \cdot |d_i| \cdot \frac{\Delta r_i}{\Delta t} = \gamma_c \cdot \frac{\Delta \sin(\Theta)}{\Delta t} \sum_{i=-\frac{L}{2l}}^{i=\frac{L}{2l}} d_i^2 \,, \tag{9}$$

with the sum over all cubes, the friction coefficient of a cube γ_c, the cubes' distances to the rod center d_i and velocities $\frac{\Delta r_i}{\Delta t}$. Simplifying the sum in Eq. (9) and approximating $\sin(\Theta)$ with Θ, the rotational friction coefficient R_r of the rod can be expressed in terms of the friction coefficient γ_c of a single cube:

$$R_r = \frac{1}{12} \cdot \gamma_c \cdot \frac{L^3}{l} \,. \tag{10}$$

Using the number of cubes $n = \frac{L}{l}$, this relation can be rewritten using the transversal friction coefficient of the rod Γ_t, which we assume to be the n-fold of a cubes friction coefficient γ_c:

$$R_r = \frac{1}{12} \cdot \Gamma_t \cdot L^2. \tag{11}$$

To determine the rods transversal friction coefficient, first the friction coefficient of a single cube had to be determined. Second, the assumption of simple additivity of the friction coefficients of particles comprising a bigger particle was tested, by investigating rod segments, that were larger than one cube, but still small enough to diffuse significantly and comparing their friction coefficient to the cubes'.

3.3.2 Experimental determination of the Rod's Friction Coefficient

This section covers the experimental determination of the friction coefficients of cubes and bigger segments of the rod from their diffusivity. The particles used in this experiments were produced by the same method as the rod (see section 3.1) and with the same cross-section, namely 4 µm x 4 µm. Different lengths of 4 µm (cubes), 8 µm (dimers), 12 µm (trimers) and 16 µm (quadrumers) were used. These shapes were selected on one hand to measure the friction coefficient of single cubes. On the other

hand, the assumption made in the previous section was tested; namely that the friction coefficient of a bigger object could be approximated by summing up the friction coefficients of the cubes comprising it. A sample was prepared by dispersing a mix of the particles of different length in a mixture of lutidine (28.6 wt%) in water. A measurement cell was filled with this dispersion and placed in the experimental setup that was also used for the RL experiments, to reproduce the conditions as accurately as possible. Figure 28 shows a microscope image of the cubes and cuboids.

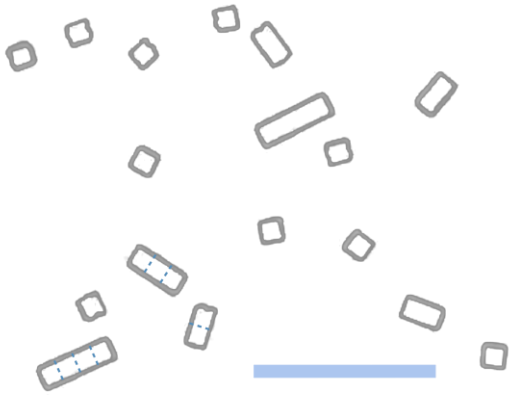

Figure 28: Exemplary picture of the particles of different shape used for the approximation of the friction coefficient of the rod. The dashed lines in the left lower corner visualize, how the cuboids can be considered to consist of multiple cubes. (Scalebar: 10 µm)

Over the course of several hours, a video of the freely diffusing particles was recorded and afterwards their positions were connected to trajectories. In the case of the cubes, the mean square displacement (MSD) was computed directly from the trajectories and fitted linearly to yield the cubes' diffusion coefficient D_c according to

$$MSD(t) = |\Delta x(t)^2|_t = 4D_c \cdot t\,. \tag{12}$$

The cuboids' anisotropic shape demanded for an additional step in the evaluation of their trajectories, since only their transversal diffusivity was of interest. Thus, their longitudinal and transversal motion had to be separated. The transversal displacements were evaluated by projecting the position of a cuboid in each frame on the short axis of the same cuboid in the frame before, as depicted in Fig 29. By summing up all these distances, one-dimensional trajectories were constructed. The

one-dimensional trajectories were then used to calculate the cuboids' transversal MSDs. To account for the reduced dimensionality, the resulting curves were fitted using a pre-factor of 2 instead of 4 in Eq (12). Fig 30 shows the MSDs of the four different particle types, together with the average slope of the curves of each shape (black dashed lines).

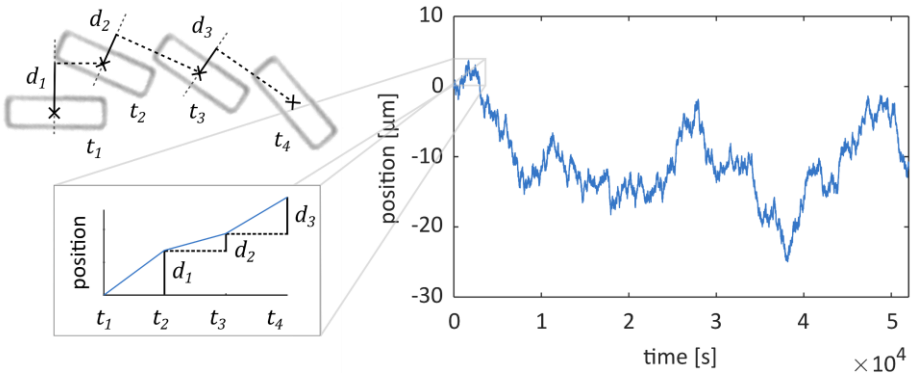

Figure 29: Construction of one dimensional, transversal trajectories of the anisotropic cuboids. The distances d_i a cuboid moved in its transversal direction between two frames are computed for every step by projecting its final position on its short axis in the initial position (left part of the figure). By summing up these displacements up to each frame, a one-dimensional trajectory was obtained (right part and inset).

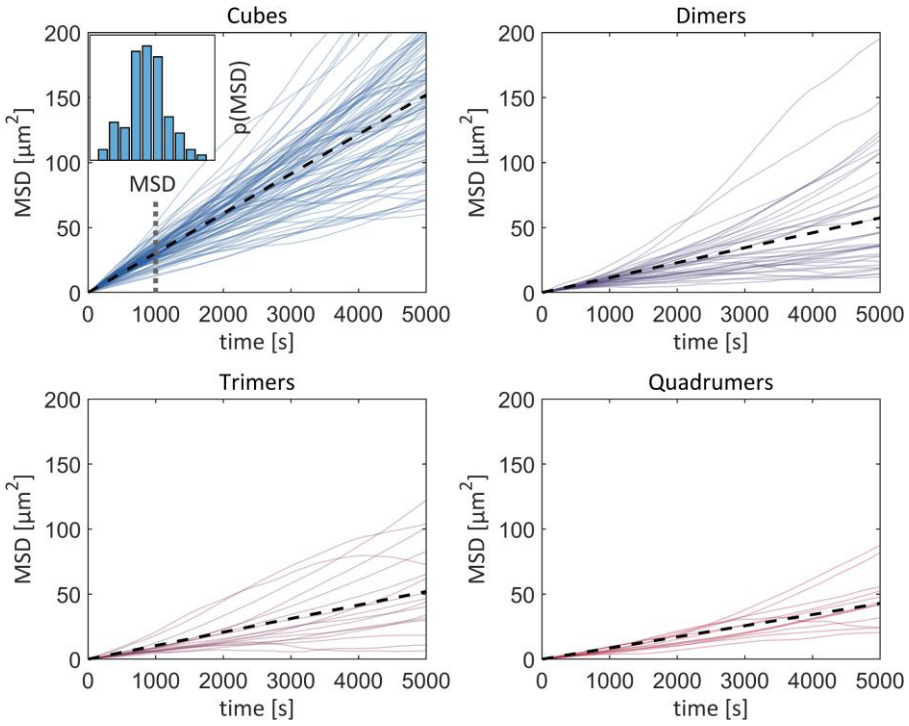

Figure 30: Mean Square Displacements of the four different kinds of particles observed in free diffusion. Only curves with a minimal length of 11000 seconds are shown and were fitted up to 5000 seconds. For the multiples of single cubes, the MSDs were computed from one dimensional trajectories of the diffusion transversal to the cuboids' orientation. Inset: probability distribution of the MSD of cubes at $t = 1000$ s (grey dotted line).

The MSDs of all kinds of particles show a diversity of slopes. Keeping in mind the small imperfections in the particles' shape and local differences in the surface of the measurement cell, the diverse measurement results can be anticipated. Additionally, the statistical nature of the MSD induces some degree of spreading in the slopes. Even the MSDs of a set of perfectly identical particles would not display the exact same slope for every particle. The probability of the cubes' MSDs at $t = 1000$ s (inset in Fig 30) is nevertheless sharply distributed around a mean value. Thus, the scattering of the curves in Fig 30 is still within an reasonable range for an evaluation of its average. To avoid any disturbance of the evaluation by the increasing noise at larger intervals t, only the data of trajectories with a minimal length of 11000 seconds was used. The individual MSDs were fitted linearly between $t_1 = 0$ and $t_1 = 5000$ seconds, since up to this time t_1 the average slope was independent of the upper bound. Fits using a higher t_2 were already influenced by the noise in the MSDs at larger times. Even though displaying diverse slopes, the MSDs in Fig 30 overall grow linear in the first 5000 seconds. This is a good indicator for undisturbed diffusion on this timescale. The

average slopes of the MSDs displayed in Fig 30 clearly decrease from cubes to tetramers. This indicates a higher friction coefficient for particles that are composed of more cubes, as expected based on the assumption of additivity of friction coefficients for assemblies of particles. The particles' diffusion coefficients D were obtained from the average slopes of the MSDs. Using the Einstein relation

$$\gamma = \frac{\mathrm{k_B}T}{D}\ , \tag{13}$$

with the Boltzmann constant $\mathrm{k_B}$ and the temperature T, the friction coefficients γ of the cubes and cuboids were obtained. Since in the case of the cuboids, only the transversal diffusivities were used, Eq. (13) yielded their transversal friction coefficient γ_t. Figure 31 shows the resulting values plotted against the number of cubes comprising each sort of particles.

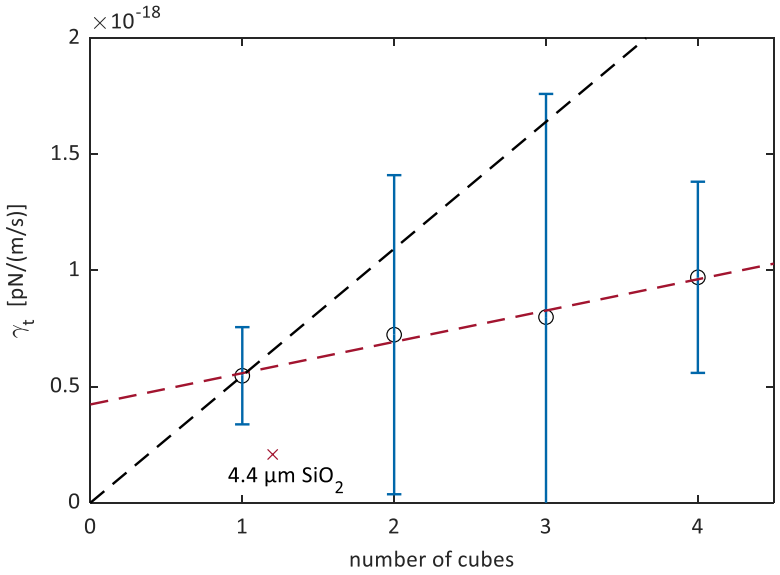

Figure 31: The transversal friction coefficients of differently sized particles plotted against the number of cubes comprising one particle. The red dashed line represents a linear fit. For comparison, the experimentally determined friction coefficient of a silica sphere with 4.4 µm diameter at the same conditions is shown. The black dashed line represents the original assumption of simple additivity of the friction coefficients.

The friction coefficients are in the order of 10^{-18} pNs/m, which agrees well with the friction coefficient of a silica sphere with 4.4 µm in diameter at the same conditions

($0.2 \cdot 10^{-18}$ pNs/m)[9]. The errors in the coefficients are large due to the diversity of slopes of the MSDs. Even though, the friction coefficients are clearly increasing with a higher number of cubes comprising each particle. A linear fit to the friction coefficients (red dashed line in Fig 31) displays a non-zero y-axes intersection and a slope of about the quarter of the friction coefficient of a single cube. This contrasts with the original assumption of simple additivity, represented by the black dashed line in Fig 31. In this assumption, the friction coefficients would have grown proportional to the number of cubes, resulting in an y-axis intersection at zero and a slope equal to the friction coefficient of a single cube. The non-proportionality of the friction coefficient and the number of cubes can be explained by considering the cubes' friction in more detail: The total friction coefficient of a particle can be understood as the sum over the frictional forces of all its surfaces with its surroundings. The connection of two cubes to a larger particle eliminates the surfaces, via which the particles get connected. Therefore, the resulting friction coefficient is smaller than the sum of the initial values, as illustrated in Fig 32. With this understanding, the way to approximate the transversal friction coefficient Γ_t of the rod was modified. Instead of using the n-fold of the cubes' friction coefficient γ_c, the linear fit to the measured friction coefficients of rod segments was extrapolated to $n =$ 24. This method together with Eq. (11) yielded $\mathrm{R}_r = 4.2 \cdot 10^{-16}$ Nms as the rotational friction coefficient of a rod with 96 µm length. In the next section, this value is used to determine the microswimmers' propulsion forces.

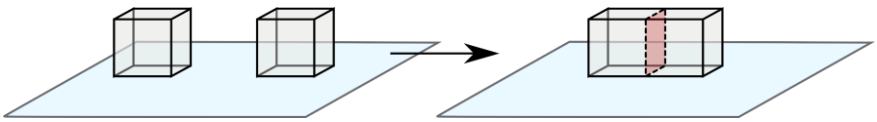

Figure 32: By connecting two cubes to a bigger particle, the connection surfaces (red-shaded) get eliminated. This leaves the total friction coefficient smaller than the sum of the friction coefficients of the initial particles.

3.3.3 Determination of the Pushing Force of Microswimmers

The value of the rotational friction coefficient R_r of the rod determined in the previous section allowed for the determination of the pushing force F_{as} of a single active particle in contact with the rod. The rotational velocity of the rod ω and the total geometric torque $T_{tot,g}$ exerted by the particles during each update step in the experiments were

determined from the experimental videos. The values of ω in dependance of $T_{tot,g}$ were fitted linearly using Eq. (8). From the slopes, the pushing force F_{as} of a single active swimmer was determined using the value of R_r determined above. A total of 18 experiments in which the particles managed to rotate the rod by more than π was evaluated by this means. Figure 33 displays the results in the form of a histogram. The determined pushing forces F_{as} of single active swimmers span a range from 0.025 pN to 0.125 pN with a mean value of 0.08 pN. These values are on the same order of magnitude as values previously reported for Janus-particles in a water-lutidine system[17]. The broad distribution of the forces is probably due to local differences in the sample cell, that led to differing rod friction coefficients in different experiments. For comparison, the propulsion force of the used microswimmers in undisturbed motion was determined from their average speed and their friction coefficient. This yielded a value of about 0.19 pN. Thus, the pushing force is by more than a factor of two smaller than the propulsion force in undisturbed swimming. This deviation is due to distortions of the swimmers' flow fields necessary for propulsion by the nearby rod[28]. Such differences between pushing and propulsion forces are not limited to the microswimmers used in this work, but could also be present in systems where the particles are driven by different mechanisms, since many examples of active motion rely on the formation of heterogeneities in the solution around a particle[5].

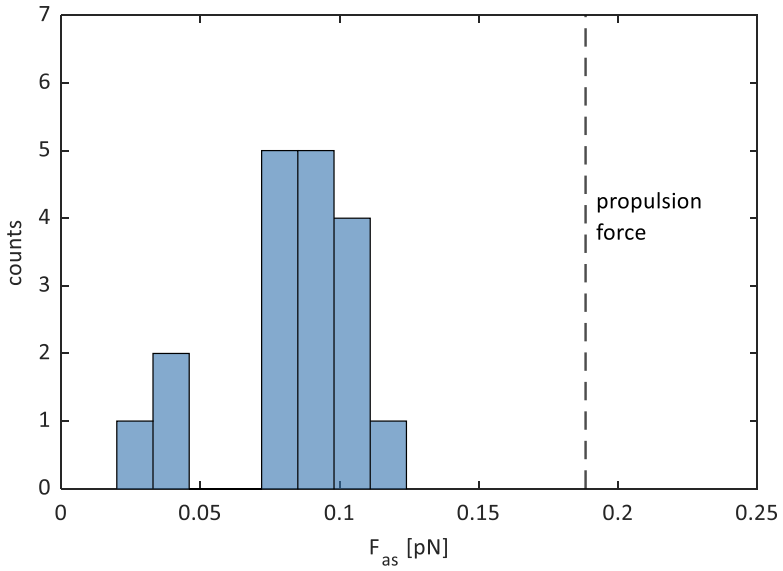

Figure 33: Histogram of the pushing forces F_{as} of single active particles pushing against the rod, determined from 18 different experiment runs. The black dashed line on the right is the propulsion force of an undisturbed active swimmer calculated from its friction coefficient and propulsion velocity.

In this chapter, the pushing forces exerted on the rod by single active swimmers were determined from the angular velocity of the rod in the RL experiments and the torque the particles exerted on it. An additional experiment had to be conducted to estimate the rotational friction coefficient of the rod, which was a crucial parameter in the evaluation of the experimental data. In the end, the presence of the rod close to the particles turned out to reduce the pushing forces by a factor of two with respect to the propulsion force. These results are important for considerations of active swimmers used in real applications, since there the particles will most probably encounter the contact to bigger objects. Further studies to validate these results could include simulations of the particle propulsion mechanism in the presence of additional surfaces, as well as experiments to investigate the influence of obstacles on different propulsion mechanisms.

4 Conclusion and Outlook

In this work, an RL algorithm was given control over the motion of multiple active particles as individual agents. To reproduce the situation of natural organisms, the microswimmers had only limited perception of their environment. This is to date the first experimental realization of a multi agent RL system with microscopic dimensions. The microswimmers were given the task of collectively rotating a much larger Brownian rod by means of their propulsion only using a certain set of actions. To communicate this to the RL algorithm, the particles were rewarded for presence close to the rod and exerting torques on it. During a training phase, the particles learned to stay close to the rod, to accumulate at its ends on opposite sides and to orient towards and push against it. With this strategy, the microswimmers managed to rotate the rod by more than two turns during an 80-minute experiment. In simulations of the same system, particles learned to rotate a rod even more efficiently. A comparison of the swimmers' behavior in simulation and experiment revealed that the real particles had troubles to reorient due to the large distance they travelled during one action. A reduction of this displacement resulted in a much higher efficiency of the particles' efforts with average torques being about twice as high as before. In addition to the RL experiments, the pushing force a single active particle exerted on the rod were determined. This was achieved by approximating the rod's rotational friction coefficient and evaluating its orientational response to the particles' position and pushing directions. The results showed that the force of a microswimmer pushing against an object is only half of its propulsion force in undisturbed motion, due to the rod's influence on the flow field around the particles.

Since during the experiments, convergence of the RL algorithm to an optimal solution could not be confirmed, further experiments should aim at longer training times. Future studies could also examine the compatibility of simulated training with the experiments and investigate, whether pre-trained particles have a learning advantage. It would also be interesting to face the microswimmers with more difficult tasks by e.g. modifying the shape of the object they are interacting with. The successful implementation of multiagent RL in experiments also opens new opportunities for the general field of emergent behavior in active matter. By facing a group of agents with a task rather than proposing interaction rules, collective states could be studied as the

result of optimization processes. There are also interesting research perspectives regarding the pushing forces of active particles. The results obtained in this work should be validated by more accurate measurements. Additionally, the difference between propulsion and pushing force for other types of microswimmers should be examined as an important characteristic of active particles.

Acknowledgements

My biggest thanks go to Prof. Clemens Bechinger for taking me into his group and making this project possible. I want to thank him for very uncomplicated and kind support whenever I needed it. I am also very grateful to him for always taking his time to discuss every aspect of the project in detail and helping me a lot with the writing.

I want to thank Prof. Christine Peter very much for being the second appraiser of this thesis.

I am most grateful to Robert for being my supervisor for this work and putting in a lot of time to explain the setup to me. I also want to thank him for sharing his knowledges about coding and computers with me whenever I had any question and for taking the time to help me writing this thesis.

This work would not have been possible without Emanuele, who implemented the RL algorithm and did all the simulations. I also want to thank him very much for answering a lot of questions I had about reinforcement learning and helping me with the writing.

For making the micro-rods, -cubes and -cuboids I want to thank Jakob very much. I would have been in big trouble, if the rods would not have been as well behaving as they were, thanks to Jakob's expertise.

The rest of the group I want to thank very much for a lot of interesting discussions in the coffee corner and making the time I spent with this thesis most enjoyable.

Finally, I want to thank Julia for supporting me the whole time of my thesis, especially in stressful times. I am also grateful for her proof reading and many corrections.

Literature

[1] Z. Ge, Z. Song, S. X. Ding, B. Huang, *IEEE Access* 2017, *5*, 20590-20616.

[2] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Shen, Y. Wang, *Stroke Vasc Neurol* 2017, *2*, 230-243.

[3] J. Pathak, B. Hunt, M. Girvan, Z. Lu, E. Ott, *Phys Rev Lett* 2018, *120*, 024102.

[4] F. Cichos, K. Gustavsson, B. Mehlig, G. Volpe, *Nature Machine Intelligence* 2020, *2*, 94-103.

[5] C. Bechinger, R. Di Leonardo, H. Löwen, C. Reichhardt, G. Volpe, G. Volpe, *Reviews of Modern Physics* 2016, *88*, 045006.

[6] O. Pohl, H. Stark, *Phys Rev Lett* 2014, *112*, 238303.

[7] F. A. Lavergne, H. Wendehenne, T. Bäuerle, C. Bechinger, *Science* 2019, *364*, 70-74.

[8] L. Barberis, F. Peruani, *Phys Rev Lett* 2016, *117*, 248001.

[9] T. Bäuerle, R. C. Löffler, C. Bechinger, *Nat Commun* 2020, *11*, 2547.

[10] E. Yang, D. Gu, tech. rep, 2004.

[11] R. Brown, *The Philosophical Magazine* 1828, *4*, 161-173.

[12] T. Vicsek, A. Czirok, E. Ben-Jacob, I. I. Cohen, O. Shochet, *Phys Rev Lett* 1995, *75*, 1226-1229.

[13] U. Erdmann, W. Ebeling, L. Schimansky-Geier, F. Schweitzer, *The European Physical Journal B* 2000, *15*, 105-113.

[14] J. Ruben Gomez-Solano, S. Roy, T. Araki, S. Dietrich, A. Maciolek, in *arXiv e-prints*, 2020, p. arXiv:2006.02546.

[15] A. P. Bregulla, H. Yang, F. Cichos, *ACS Nano* 2014, *8*, 6542-6550.

[16] H. R. Jiang, N. Yoshinaga, M. Sano, *Phys Rev Lett* 2010, *105*, 268302.

[17] G. Volpe, I. Buttinoni, D. Vogt, H.-J. Kümmerer, C. Bechinger, *Soft Matter* 2011, *7*.

[18] I. Buttinoni, G. Volpe, F. Kümmel, G. Volpe, C. Bechinger, *J Phys Condens Matter* 2012, *24*, 284129.

[19] C. Lozano, B. Ten Hagen, H. Löwen, C. Bechinger, *Nat Commun* 2016, *7*, 12828.

[20] J. Moroz, Google Patents, 1986.

[21] J. R. Gomez-Solano, S. Samin, C. Lozano, P. Ruedas-Batuecas, R. van Roij, C. Bechinger, *Sci Rep* 2017, *7*, 14891.

[22] T. Bäuerle, A. Fischer, T. Speck, C. Bechinger, *Nat Commun* 2018, *9*, 3232.

[23] M. I. Jordan, T. M. Mitchell, *Science* 2015, *349*, 255-260.

[24] M. Botvinick, S. Ritter, J. X. Wang, Z. Kurth-Nelson, C. Blundell, D. Hassabis, *Trends Cogn Sci* 2019, *23*, 408-422.

[25] M. Glavic, R. Fonteneau, D. Ernst, *IFAC-PapersOnLine* 2017, *50*, 6918-6927.

[26] R. S. Sutton, A. G. Barto, *Reinforcement learning: An introduction*, MIT press, 2018.

[27] L. P. Kaelbling, M. L. Littman, A. W. Moore, *Journal of Artificial Intelligence Research* 1996, *4*, 237-285.

[28] A. Zöttl, H. Stark, *Journal of Physics: Condensed Matter* 2016, *28*.

[29] N. Jakobi, P. Husbands, I. Harvey, in *European Conference on Artificial Life*, Springer, 1995, pp. 704-720.

[30] L. Biferale, F. Bonaccorso, M. Buzzicotti, P. Clark Di Leoni, K. Gustavsson, *Chaos* 2019, *29*, 103138.

[31] S. Colabrese, K. Gustavsson, A. Celani, L. Biferale, *Phys Rev Lett* 2017, *118*, 158004.

[32] E. Schneider, H. Stark, *Epl-Europhys Lett* 2019, *127*.

[33] S. Muiños-Landin, K. Ghazi-Zahedi, F. Cichos, 2018, p. arXiv:1803.06425.

[34] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, 2017, p. arXiv:1707.06347.

[35] Y. Wang, H. He, X. Tan, in *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference, Vol. 115* (Eds.: P. A. Ryan, G. Vibhav), PMLR, Proceedings of Machine Learning Research, 2020, pp. 113--122.

[36] G. Li, J. X. Tang, *Phys Rev Lett* 2009, *103*, 078101.