DISS. ETH NO. 29586

REINFORCEMENT LEARNING AND OPTIMAL EXPERIMENTAL DESIGN FOR MODELING AND CONTROL OF FISH SCHOOLING HYDRODYNAMICS

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES

(Dr. sc. ETH Zürich)

presented by

PASCAL PHILIPPE MARTIN WEBER
MSc Physics, ETH Zürich

born on 18 December 1990

accepted on the recommendation of

Prof. Dr. P. Koumoutsakos, examiner
Prof. Dr. N. Li, co-examiner

2023

Pascal Philippe Martin Weber: *Reinforcement Learning and Optimal Experimental Design for Modeling and Control of Fish Schooling Hydrodynamics,*
© 2023

DOI: 10.3929/ethz-b-000637048

This thesis is dedicated to Heinz†

ABSTRACT

This doctoral thesis investigates the modeling and control of fish hydrodynamics. The study consists of two main components: computational modeling of flow fields and sensory cues, and understanding the optimality principles driving fish behavior. The thesis emphasizes the incorporation of hydrodynamic interactions to accurately represent the fish's environment. The research utilizes reinforcement learning, Bayesian inference, and high-performance computing to analyze natural behavior and flow fields. The insights gained from this study have potential applications in autonomous robot swimmers and may inspire new experiments in biology. Efficient and scalable implementations of computation fluid dynamics, reinforcement learning, and Bayesian inference algorithms are developed to address the computational challenge and pave the way for future advancements in the field.

ZUSAMMENFASSUNG

Diese Doktorarbeit untersucht die Modellierung und Kontrolle der Hydrodynamik von Fischen. Die Arbeit besteht aus zwei Teilen: der computergestützen Modellierung von Strömungsfeldern und sensorischen Wahrnehmungen sowie dem Verständnis der optimalen Prinzipien, die das Verhalten von Fischen steuern. Die Arbeit legt besonderen Wert auf die Einbeziehung hydrodynamischer Wechselwirkungen, um die Umgebung der Fische präzise darzustellen. Die Forschung nutzt Reinforcement Learning, Bayes'sche Inferenz und Hochleistungsrechnen, um das natürliche Verhalten und die Strömungsfelder zu analysieren. Die gewonnenen Erkenntnisse aus dieser Studie haben potenzielle Anwendungen in autonomen Roboter-Schwimmern und können neue Experimente in der Biologie inspirieren. Effiziente und skalierbare Implementierungen von rechengestützer Fluiddynamik, Reinforcement Learning, und Bayes'scher Inferenz werden entwickelt, um den Herausforderungen der Berechnung gerecht zu werden und den Weg für zukünftige Fortschritte in diesem Bereich zu ebnen.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude and appreciation to all those who have contributed to the completion of this doctoral thesis. This journey would not have been possible without the support, guidance, and encouragement from numerous individuals and institutions:

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Petros Koumoutsakos, for making this thesis possible and providing guidance, support, and insightful feedback throughout the entire research process. His expertise, attention to detail, and drive to constantly pushing the boundaries have been instrumental in shaping this thesis and me as a researcher. Also, I would like to extend my sincere thanks to my co-supervisor, Prof. Na Li, for her valuable time, expertise, and constructive criticism.

I am thankful to ETH Zürich, the Swiss Supercomputing Center (CSCS), and EuroHPC JU for providing me with with a first-class academic environment and the needed computational resources to pursue my doctoral studies. The access to research facilities, lectures, and events has been excellent.

Furthermore, I would like to express my appreciation to my collaborators, colleagues, and fellow members of the CSElab – Daniel Wälchli, Michail Chatzimanolakis, Sergio Martin, Guido Novati, Pantelis Vlachas, Ivica Kicic, Martin Boden, Fabian Wermelinger, Lucas Amoudruz, Ermioni Papadopoulou, Sergey Litvinov, Petr Karnakov, Athena Economides, Susanne Lewis, Xin Bian, Sebastian Kaltenbach, Georgios Arampatzis, Jens Honore Walther, Jacopo Canton, Julija Zavadlav, Siddhartha Verma, and Costas Papadimitriou – for their camaraderie, intellectual discussions, and shared experiences. Their collaborative spirit have made this research endeavor exceptional.

Lastly, I want to express my gratitude to the incredible people who form the bedrock of my life: my beloved family and friends. Your unwavering support and boundless love have been invaluable. You have been my guiding stars and your unyielding encouragement, your unwavering loyalty, and uplifting spirits have been invaluable source of joy. Thank you for everything!

TABLE OF CONTENTS

1 Introduction 1

2 Modeling of Swimmers in Incompressible Flows 5

2.1 Governing Equations 6
2.2 Forces and Swimming Efficiency 7
2.3 Shapes and Swimmers 8
2.4 Numerical Method 10
2.5 Adaptive Mesh Refinement 12
2.6 Parallelization 14
2.7 Verification and Validation 15

3 Deep Reinforcement Learning 19

3.1 Remember and Forget for Experience Replay (ReF-ER) 21
3.2 ReF-ER for Multi-Agent Reinforcement Learning 25
  3.2.1 Formalism 25
  3.2.2 SISL Environments 29
  3.2.3 Evaluation 31
  3.2.4 ReF-ER and Scientific Multi-Agent Reinforcement Learning . 34
3.3 A Bayesian Perspective on Uncertainties in DRL 37
  3.3.1 Bayesian Inference 38
  3.3.2 Reinforcement Learning 39
  3.3.3 Bayesian Deep Reinforcement Learning 40
  3.3.4 Results 47

4 Bayesian Optimal Experimental Design 55

4.1 Information Gain and Expected Utility 56
  4.1.1 Formalism 56
  4.1.2 Estimation 57
  4.1.3 Optimization 58

4.2 Optimal Sensor Placement for Schooling Swimmers 60
4.2.1 Schooling Formations 62
4.2.2 Flow Sensors 62
4.2.3 Measurement Model 63
4.2.4 Inferring the Number of Swimmers 65
4.2.5 Results 66

5 Controlling Artificial Swimmers using DRL 79

5.1 Reinforcement Learning for Artificial Swimmers. 81
5.2 Multi-Task RL to Swim in Reverse and Forward Kármán Wakes . 83
5.2.1 Environment 84
5.2.2 Results 85
5.3 Transfer Learning for Schooling Swimmers with MARL 95
5.3.1 Results 96
5.3.2 Schooling Hydrodynamics 101

6 Discussion and Outlook 105

A Appendix for Chapter Bayesian Optimal Experimental Design . 115

A.1 Optimal Sensor Placement for Schooling Swimmers 115
A.1.1 Configurations 115
A.1.2 The posterior Is Not Symmetric 115
A.2 Optimal Allocation of Limited Test Ressources 121
A.2.1 Introduction 121
A.2.2 Epidemiological Model 123
A.2.3 Optimal Testing 125
A.2.4 Error Model 126
A.2.5 Informed Priors 127
A.2.6 Results 128
A.2.7 Bayesian Inference from randomized testing 139
A.2.8 Expected Information Gain 142
A.2.9 Estimation of the Expected Information Gain 145
A.2.10 Optimal Location and Time of Testing 146
A.2.11 Quantification of Uncertainty 146

B Appendix Deep Reinforcement Learning 161
B.1 V-RACER for discrete action space 161
B.2 Clipped Normal Distribution 161
B.2.1 Importance Weight 162
B.2.2 Kullback-Leibler Divergence 162
B.3 Algorithms 167

Bibliography 171

1

INTRODUCTION

> *If you can't solve a problem, then there is an easier problem you can solve: find it.*
>
> — George Pólia

This doctoral thesis presents a comprehensive exploration of computational methods, Bayesian statistics, and reinforcement learning (RL) techniques, aiming to understand complex systems. The motivation behind this research lies in the need for effective tools and methodologies to analyze and comprehend intricate phenomena in the fluid dynamics of fish. By leveraging computational methods and advanced statistical techniques, this thesis seeks to provide novel insights and practical applications. The thesis is divided into four chapters, each addressing different aspects and applications:

In the first chapter, the numerical method is introduced that enables the simulation of incompressible flows around complex and self-deforming obstacles, with a specific focus on artificial swimmers. This chapter outlines the governing equations, describes the employed numerical method, and presents the verification and validation of the implementation. By providing a reliable computational framework, this research facilitates a detailed understanding of the fluid dynamics of fish.

In the second chapter, deep reinforcement learning (DRL) techniques take center stage. The thesis extends the Remember and Forget for Experience Replay (ReF-ER) algorithm to Multi-Agent RL (MARL) and benchmarks its performance on collaborative environments, demonstrating superior results compared to state-of-the-art algorithms. Furthermore, the integration of Bayesian inference with off-policy actor-critic DRL algorithms is explored, and a rigorous evaluation of existing Bayesian deep learning methods for DRL tasks is presented. These investigations provide valuable insights into the application of DRL techniques to understand systems with multiple interacting agents and the quantification of uncertainties in DRL by its fusion with Bayesian inference.

The third chapter concentrates on Bayesian optimal experimental design and its application to examine the optimal sensor distribution on an artificial swimmer's surface to identify a leading group of swimmers. It showcases the efficacy of Bayesian experimental design in improving predictions and decision-making processes. Furthermore, it provides guidance in the selection of the state for DRL of the hydrodynamics of fish.

The fourth chapter highlights two applications of DRL, focusing on algorithmic understanding of natural behavior. The first application examines energy harvesting by fish in wakes of bluff bodies, while the second investigates the hydrodynamics of schooling fish. Through these studies, the thesis offers new perspectives and a unified framework for analyzing natural behavior, contributing to our understanding of complex systems in diverse domains.

In summary, this doctoral thesis makes contributions to the fields of computational fluid dynamics, Bayesian optimal experimental design, and deep reinforcement learning. By introducing novel methodologies and testing them in computationally challenging applications, this research enables a deeper understanding of complex systems and fosters advancements in various domains. It shows how the combination of computational methods, Bayesian inference, and DRL approaches offers a powerful toolkit for analyzing and comprehending intricate phenomena across different domains.

2

MODELING OF SWIMMERS IN INCOMPRESSIBLE FLOWS

> *Truth is much too complicated to allow anything but approximations.*
>
> — John von Neumann

In the following we present the numerical method that is suitable to simulate the incompressible flow around complex, self-deforming obstacles such as artificial swimmers. Besides the governing equation and numerical method, we present verification and validation of the implementation.

ACKNOWLEDGMENTS The software CUBISMUP is based on prior work of Guido Novati, who provided important assistance during the early stages of this thesis. Many improvements of the numerical method, and the addition of adaptive mesh refinement was lead by Michalis Chatzimanolakis, who thought the author many lessions on the art of numerical modelling. The associated collaboration gave rise to two papers [1, 2], on which this chapter is based. The computational resources for this work was provided by the Swiss National Supercomputing Centre (CSCS) under project s929.

2.1 Governing Equations

We model incompressible fluids with multiple deforming geometries using the incompressible Navier-Stokes equations with Brinkman penalization to enforce the no-slip, no-through boundary conditions on the surface of deforming obstacles [3–5]

$$\begin{aligned} \frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u} &= -\frac{\nabla p}{\rho} + \nu\Delta\boldsymbol{u} + \lambda\sum_{s=1}^{N_s}\chi^{(s)}(\boldsymbol{u}^{(s)}-\boldsymbol{u}),\\ \boldsymbol{\nabla}\cdot\boldsymbol{u} &= 0. \end{aligned} \tag{1}$$

Here, we denote the velocity field by $\boldsymbol{u} : \Omega \times [0, T] \to \mathbb{R}^3$, the pressure field by $p : \Omega \times [0, T] \to \mathbb{R}^3$, the kinematic viscosity by ν, and density of the fluid by ρ. Brinkman penalization models the fluid-structure interaction by introducing a penalization-term, with coefficient λ. The obstacles $s = 1, \dots, N_s$, are modeled by the characteristic function $\chi^{(s)}$, that is $\chi^{(s)} = 1$ inside the obstacle $\Omega^{(s)}$ and $\chi^{(s)} = 0$ outside the obstacle $\Omega \setminus \Omega^{(s)}$. The obstacle velocity-field $\boldsymbol{u}^{(s)} \in \mathbb{R}^3$ consists of translation, rotation, and deformation

$$\boldsymbol{u}^{(s)} = \boldsymbol{u}^{(s)}_{\text{trans}} + \boldsymbol{u}^{(s)}_{\text{rot}} + \boldsymbol{u}^{(s)}_{\text{def}}. \tag{2}$$

While the deformation velocity $\boldsymbol{u}^{(s)}_{\text{def}}$ is prescribed, the translation and rotation velocity is computed from the fluid linear momentum

$$\boldsymbol{u}^{(s)}_{\text{trans}} = \frac{1}{m}\int_{\Omega}\chi^{(s)}\boldsymbol{u}\,\mathrm{d}V, \tag{3}$$

and angular momentum

$$\boldsymbol{u}^{(s)}_{\text{rot}} = \boldsymbol{\omega}^{(s)} \times \boldsymbol{r} \quad \text{with} \quad \boldsymbol{\omega}^{(s)} = \frac{1}{I}\int_{\Omega}\chi^{(s)}(\boldsymbol{r}\times\boldsymbol{u})\,\mathrm{d}V, \tag{4}$$

where $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{x}^{(s)}$ is the displacement from the obstacle center of mass $\boldsymbol{x}^{(s)}$ and the mass m and moment of inertia I are computed as

$$m = \int_{\Omega}\chi^{(s)}\,\mathrm{d}V, \qquad I = \int_{\Omega}\|\boldsymbol{r}\|^2\chi^{(s)}\,\mathrm{d}V. \tag{5}$$

This allows updating the location $\boldsymbol{x}^{(s)}$ and angle $\phi^{(s)}$ of the obstacle by integrating the equations of motion

$$\frac{\mathrm{d}\boldsymbol{x}^{(s)}}{\mathrm{d}t} = \boldsymbol{u}^{(s)}, \quad \frac{\mathrm{d}\phi^{(s)}}{\mathrm{d}t} = \boldsymbol{\omega}^{(s)}. \tag{6}$$

We note, that for non-deforming obstacles the penalization term acts as a Lagrange multiplier enforcing the translation and rotation motion of the obstacle on the fluid. For deforming obstacles it acts as an elastic response propelling the fluid out of the obstacle.

2.2 Forces and Swimming Efficiency

The total force acting on a solid body is

$$\boldsymbol{F} = \int\limits_{\partial\Omega_s} (2\mu D \cdot \boldsymbol{n} - p\boldsymbol{n}) \mathrm{d}S \,, \tag{7}$$

where $\mathrm{d}S$ denotes the infinitesimal surface element with normal $\boldsymbol{n}$, μ is the dynamic viscosity, $\partial\Omega_s$ is the surface of the obstacle and $D = \frac{1}{2}\left(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^\top\right)$ the strain-rate tensor. The first term corresponds to viscous forces, and the second to pressure-induced forces.

For the computation, the surface integral is expressed as a volume integral

$$\boldsymbol{F} = \int\limits_{\Omega} (2\mu D \cdot \boldsymbol{n} - p\boldsymbol{n}) \delta(\mathrm{S_d}) \, \mathrm{d}\Omega \,, \tag{8}$$

where δ is the Dirac delta and $\mathrm{S_d}$ is the signed distance function from the surface of the body to any point in Ω. Since $\chi^{(s)} = H(\mathrm{S_d})$, where H is the Heaviside function, we find that $\delta(\mathrm{S_d}) = \frac{\mathrm{d}\chi}{\mathrm{d}\boldsymbol{n}} = \nabla\chi \cdot \boldsymbol{n}$ where the normal vector is computed from the signed distance function as $\boldsymbol{n} = \frac{\nabla \mathrm{S_d}}{|\nabla \mathrm{S_d}|}\big|_{\partial\Omega_s}$. Thus, the total force can be effectively computed from

$$\boldsymbol{F} = \int\limits_{\Omega} (2\mu D \cdot \boldsymbol{n} - p\boldsymbol{n})(\nabla\chi^{(s)} \cdot \boldsymbol{n}) \, \mathrm{d}\Omega \,. \tag{9}$$

When computing viscous forces the penalization method underestimates velocity gradients near walls and it was suggested to compute the gradients on a “lifted" surface [6]. We employ a similar approach: the necessary gradients are computed two grid points away from the surface and are then extrapolated back to it through a second-order Taylor expansion. In the Taylor expansion, the derivatives are computed with second-order one-sided differences, facing away from the wall. From the force the drag F_D and drag coefficient C_D can be computed

$$F_D = \boldsymbol{F} \cdot \frac{\boldsymbol{u}^{(s)}}{|\boldsymbol{u}^{(s)}|} \,, \qquad C_D = \frac{2F_D}{\rho|\boldsymbol{u}^{(s)}|^2 A} \,, \tag{10}$$

where A is a characteristic area and $\rho = 1$ is the fluid density.

2.3 Shapes and Swimmers

In the following we discuss the parametrization for the obstacles. The general computational pipeline consists of computing the

BOUNDING BOX to avoid computing the characteristic function in regions where there are no obstacles.

SIGNED DISTANCE from the obstacle's surface. Negative values correspond to grid points outside and positive values to grid points inside the obstacle.

CHARACTERISTIC FUNCTION which is obtained by applying the Heaviside function to the signed distance function.

Simple Shapes

The characteristic of the "simple shapes" describing in this subsection is that they are non-deforming. As we will see in the following section, this difference implies a vanishing right-hand side for the Poisson equation (eq. (22)), e.g. that the computed velocity field is divergence free everywhere.

CYLINDER For a cylinder with radius R the signed distance at $(x, y) \in \Omega$ is computed as

$$d(x, y) = R^2 - (x^2 + y^2) \tag{11}$$

HALFDISK For a half-disk with radius R the signed distance at $(x, y) \in \Omega$ is computed as

$$d(x, y) = \begin{cases} -x\,, & \text{if } x > 0 \\ R^2 - (x^2 + y^2)\,, & \text{else} \end{cases} \tag{12}$$

Artificial Swimmers

The shape of a swimmer of length L is parameterized according to the arc-length $s \in [0, L]$ along the centerline of the body. We distinguish two ways to impose the motion with period T on the fish:

CARLING FISH The first follows Carling [7], which computes the y-coordinate $y(s, t)$ of the centerline via a sinusodial wave

$$y(s, t) = \frac{4}{33}(s + 0.03125L) \sin\left(\frac{2\pi t}{T} - \frac{2\pi s}{L}\right) . \tag{13}$$

STEFAN FISH The second formulation follows Stefan Kern [8], where the self-propelling motion is imposed via a sinusoidal variation of the centerline curvature $k(s,t)$ according to

$$k(s,t) = A(s)\sin\left(\frac{2\pi t}{T} - \frac{2\pi s}{L}\right) . \tag{14}$$

Here $A(s)$ denotes the amplitude which linearly increases from $A(0) = 0.82/L$ to $A(L) = 5.7/L$. The centerline coordinate is computed from the curvature by integrating the Frenet equations [9].

During the computation of the midline, the deformation velocity [9] is computed. This deformation velocity implies a non-divergence free velocity field inside the obstacle.

From the midline we construct the shape of the swimmer using the width given by

$$w(s) = \begin{cases} \sqrt{2w_h s - s^2}\,, & 0 \le s < s_b \\ w_h - (w_h - w_t)\left(\frac{s-s_t}{s_t-s_b}\right)\,, & s_b \le s < s_t \\ w_t \frac{L-s}{L-s_t}\,, & s_t \le s \le L \end{cases}\,, \tag{15}$$

where $w_h = s_b = 0.04L$, $s_t = 0.95L$ and $w_t = 0.01L$. The same ideas used for the fish can also be used to model other undulating bodies, like for example a NACA hydrofoil, whose width would be parameterized via

$$\begin{aligned} w(s) = 5t\Big[&0.2969\sqrt{s} - 0.1260s - 0.3516s^2 \\ &+ 0.2843s^3 - 0.1015s^4\Big]\,, \end{aligned} \tag{16}$$

where t is the maximum thickness as a fraction of the chord and gives the last two digits in the NACA 4-digit denomination.

2.4 Numerical Method

The governing eq. (1) are solved using a modified pressure projection method [1, 10]. In the following, the temporal discretization is made explicit by denoting the field at timestep t using a superscript. Although everything is written using a constant time-step Δt, in practice the time-step is adopted to satisfy the Courant–Friedrichs–Lewy (CFL) condition. Starting from the initial flow configuration, the solver performs the following steps:

PUT OBJECTS ON GRID First, the characteristic function $\chi^{(s),t+1}$ and deformation velocity $\boldsymbol{u}_{\text{def}}^{(s),t+1}$ for the obstacles is created:

1. The velocity $\boldsymbol{u}_{\infty}^{t}$ of the frame of reference is computed

$$\boldsymbol{u}_{\infty}^{t} = \frac{1}{N}\sum_{s'} \boldsymbol{u}_{\text{lin}}^{(s'),t} . \tag{17}$$

   Where $\boldsymbol{u}_{\text{lin}}^{(s'),t}$ are the linear velocities of N obstacles $s' \in \{1,\dots,N\} \subseteq \{1,\dots,N_s\}$ with which the frame of reference moves.

2. The obstacle center of mass $\boldsymbol{x}^{(s),t+1}$ and orientations $\theta^{(s),t+1}$ are updated using the linear $\boldsymbol{u}_{\text{lin}}^{(s'),t}$ and angular $\omega^{(s),t}$ velocity computed from eq. (6). In order to compute the integrals, a second-order accurate approximation of the characteristic function [11] is used. The equation of motion for the swimmer are then integrated using forward Euler

$$\begin{aligned} \boldsymbol{x}^{(s),t+1} &= \boldsymbol{x}^{(s),t} + \Delta t\, \boldsymbol{u}_{\text{lin}}^{(s),t} , \\ \theta^{(s),t+1} &= \theta^{(s),t} + \Delta t\, \omega^{(s),t} . \end{aligned} \tag{18}$$

3. Based on the new location and orientation the characteristic function $\chi^{(s),t+1}$ is computed from the signed distance function and the result is put on the grid.

ADVECTION AND DIFFUSION After creating the shapes we perform advection and diffusion of the flow field in the whole domain using the second-order midpoint rule

$$\begin{aligned} \boldsymbol{u}^{t+1/2} &= \boldsymbol{u}^{n} + \frac{1}{2}\Delta t \left(\nu \Delta \boldsymbol{u}^{t} - (\boldsymbol{u}^{t}\cdot\nabla)\boldsymbol{u}^{t}\right) , \\ \boldsymbol{u}^{*} &= \boldsymbol{u}^{t} + \Delta t \left(\nu \Delta \boldsymbol{u}^{t+1/2} - (\boldsymbol{u}^{t+1/2}\cdot\nabla)\boldsymbol{u}^{t+1/2}\right) . \end{aligned} \tag{19}$$

Here, the diffusion terms are discretised with centered second-order finite differences and advection terms are handled with an upwind fifth-order WENO scheme [12].

PENALIZATION We apply the penalization force by performing an implicit time-step

$$\boldsymbol{u}^{**} = \boldsymbol{u}^{*} + \lambda \Delta t \sum_{s=1}^{N_s} \chi^{(s),t+1} \left(\boldsymbol{u}^{(s),t+1} - \boldsymbol{u}^{**} \right) . \tag{20}$$

Since the implicit time-stepping is stable for any $\lambda > 0$, this allows for arbitrarily large values for the penalisation coefficient. As the penalisation coefficient tends to infinity, eq. (1) converge to the incompressible Navier-Stokes equations with tangential and slip velocities on the surface of the solid body in the order of $\lambda^{-1/2}$ and λ^{-1} respectively [13]

PRESSURE PROJECTION Here we conclude the timestep with pressure-projection

$$\boldsymbol{u}^{t+1} = \boldsymbol{u}^{**} - \Delta t \, \nabla p^{t+1} , \tag{21}$$

where the pressure field is obtained by solving the Poisson equation

$$\nabla^2 p^{t+1} = \frac{1}{\Delta t} \nabla \cdot \boldsymbol{u}^{**} . \tag{22}$$

The Poisson equation arises when taking the divergence of eq. (21) and requiring that the deformation $\boldsymbol{u}_{\text{def}}^{(s),t+1}$ of the obstacles impose a non-divergence-free flow field $\nabla \cdot \boldsymbol{u}^{t+1} = \sum_{s=1}^{N_s} \chi^{(s),t+1} \nabla \cdot \boldsymbol{u}_{\text{def}}^{(s),t+1}$. The poisson equation is discretised with a conservative, second-order accurate discretisation of the divergence operator [14]. All unknown values are concatenated in a vector. The arising linear system $A\boldsymbol{\phi} = \boldsymbol{b}$ is solved by using the preconditioned biconjugate gradient stabilised method [15], with a custom preconditioner [1].

2.5 Adaptive Mesh Refinement

The discretization is done on an adaptive block-structured grids implemented in the CubismAMR library [1]. CubismAMR is an adaptive version of the Cubism library [16–18], which partitions the simulation domain into cubes of uniform resolution that are distributed to multiple compute nodes. These cubes are further divided into blocks for cache-optimised parallelism. CubismAMR organizes these blocks in an octree data structure (for three-dimensional simulations) or a quadtree data structure (for two-dimensional simulations), which allows for grid refinement or compression in different regions. In contrast to uniform grids or body fitted meshes, this allows to dynamically adopting the grid to capture the emerging structures in a flow field.

The grid is composed of square blocks, each with the same number of cells. Each block has a locally uniform resolution that is defined by its level of refinement $\ell = 0, \dots, L-1$ as $h_\ell = 2^{-\ell} h_0$, where h_0 is the coarsest grid spacing possible. At mesh refinement from level ℓ to level $\ell + 1$ a block is divided into four blocks, whereas mesh compression from level ℓ to $\ell - 1$ is achieved by combining four blocks to one. Blocks that are adjacent are not allowed to differ by more then one refinement level. This allows arranging the blocks of the grid in a quadtree data structure. Whether a block should be refined or compressed is determined every few timesteps. Our simulations use the vicinity to solid body surfaces and the magnitude of vorticity as criteria for mesh refinement or compression. Additional examples of such criteria could be the magnitude of pressure gradients, or the magnitude of Wavelet detail coefficients [19, 20]. Following the multiresolution framework by [21, 22], restriction and prolongation operators are used to map values between blocks of different resolutions. The restriction operator $\mathcal{R}$ is used during mesh compression to replace four blocks at level $\ell + 1$ by one block at level ℓ. Grid point values at level ℓ are computed by averaging. The prolongation operator $\mathcal{I}$ of one block at level ℓ to four blocks at level $\ell + 1$ is defined via a third-order Taylor expansion, where derivatives are approximated with second-order central finite difference schemes. For the approximation of spatial derivatives, we use finite difference schemes. This is done by creating a frame of uniform resolution around each gridpoint [23] through the interpolation of ghost cell values. At the interface between different refinement levels ℓ and $\ell + 1$, this requires the interpolation of ghost cell values from the coarse to the fine level and vice-versa. From a fine to a

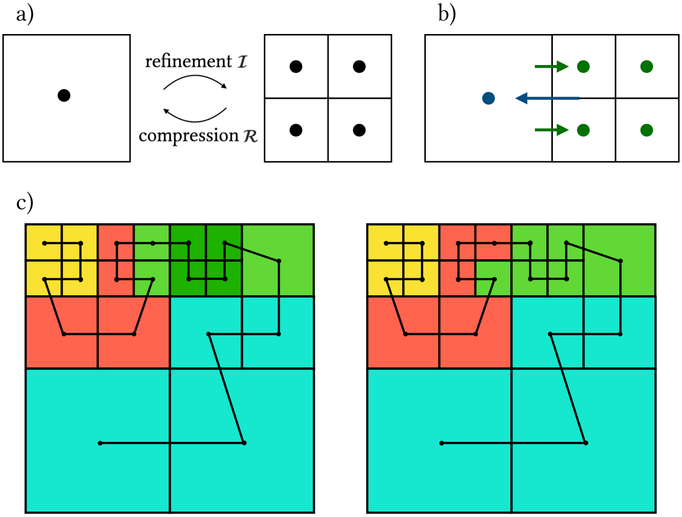


FIGURE 1: Illustration of the AMR. Figure a) illustrates of how cells are refined or compressed. b) shows the fluxes at coarse-fine-interface. In order to ensure conservation, the sum of green fluxes is replacing the blue flux. Figures c) illustrate load-balancing using diffusion of work. The black line illustrates the Hilbert curve. The colour represents the process that owns the block. The block in dark green got refined and thus the green process has more work than the other three processes. After diffusion of work the load is again equally distributed.

coarse level this is done by averaging. For the ghost cells that need to be interpolated from a coarse level to a finer one, we use third-order accurate quadratic interpolation as proposed by [24] and [25]. When used with a second-order finite difference scheme for approximation of first and second derivatives, this guarantees second and first order accuracy respectively. Whenever needed we follow [26] and [27] and use a second-order accurate conservative discretisation of the divergence operator. Whenever a cell is next to cells of different resolutions, missing values are interpolated. This results in a non-conservative discretisation of the divergence operator at the interface between two different refinement levels ℓ and $\ell+1$, as the flux computed for level ℓ need not be equal to the sum of the two fluxes at level $\ell+1$ that make up the same cell face. The situation is illustrated in fig. 1. Conservation is achieved by replacing the flux at level ℓ by the sum of the fluxes at level $\ell+1$.

2.6 Parallelization

CubismAMR is parallelised with the Message Passing Interface (MPI) programming model. The quadtree data structure is traversed by a space-filling Hilbert curve [28], that assigns each block a unique index. Initially, blocks are distributed to different MPI processes based on their index along the Hilbert curve, the locality property of which guarantees that each process will own blocks that are spatially close to each other. Load-imbalance between different processes, introduced because of mesh compression or refinement, is handled by redistributing work through a one-dimensional diffusion-based scheme. As was first proposed by [29], the number of blocks N_p^t of process p at timestep t is updated as

$$N_p^{t+1} = N_p^t + c(N_{p+1}^t - 2N_p^t + N_{p-1}^t)\,. \tag{23}$$

c is a user-defined constant, set to $c = 0.25$ in the present work. This load-balancing scheme limits communication between consecutive processes along the one-dimensional Hilbert curve while gradually redistributing workload. In fig. 1 (bottom) we illustrate the process in two-dimensions. Additionally, all blocks are evenly redistributed to all processes based on the load-imbalance ratio (defined as the ratio between the maximum and the minimum number of blocks any process has).

2.7 Verification and Validation

We simulate the impulsively started flow around a cylinder, for Reynolds numbers in the range $[550, 9500]$. All the simulations presented in this section use a reference time unit $T = \frac{D}{2U}$, where D is the cylinder diameter and U its velocity. They were performed in a $[0, 40D] \times [0, 20D]$ domain and the cylinder was placed at $(10D, 10D)$. The timestep was controlled by a Courant number of 0.45, based on the minimal grid spacing and the maximal velocity present in the domain. The coarsest grid possible had a resolution of 128×64. The penalisation coefficient was set to $\lambda = 10^7$. Grid refinement and compression were based on vorticity magnitude. The grid points within $0.1D$ vicinity to the cylinder surface were refined to the maximum resolution allowed.

To verify consistency and convergence, we fix the Reynolds number at Re$=$ 1 000 and perform simulations with increasing resolution, by changing the maximum levels of refinement allowed, denoted by L. The simulation with the finest resolution (with $L = L^* = 9$ levels) is used as a reference solution for two quantities of interest: the instantaneous cylinder drag coefficient $C_D^L(t)$ and the vorticity field $\omega^L(\tau)$ with $\tau = 10T$. For the drag, the error is defined as

$$\varepsilon_{C_D}^L = \frac{1}{t_{\max} - t_{\min}} \int_{t_{\min}}^{t_{\max}} |C_D^L(t) - C_D^{L^*}(t)| \mathrm{d}t \,, \tag{24}$$

with $t_{\min} = 0.01T$ and $t_{\max} = 10T$ and for the vorticity field as

$$\varepsilon_\omega^L = |\omega^L(\tau) - \omega^{L^*}(\tau)| \ \ , \ \ \tau = 10T \,. \tag{25}$$

		Drag coefficient		Vorticity	
L	$\hat{N}^L$	$\varepsilon_{C_D}^L$	Rate	ε_ω^L	Rate
4	89	0.167	1.01	0.524	1.78
5	119	0.124	1.97	0.313	2.26
6	166	0.0469	2.94	0.125	2.63
7	242	0.0154	2.23	0.0486	2.04
8	381	0.00779	1.51	0.0238	1.59
9	637	0	–	0	–

TABLE 1: Verification study errors and convergence rates, impulsively started cylinder at Re$=$ 1 000.

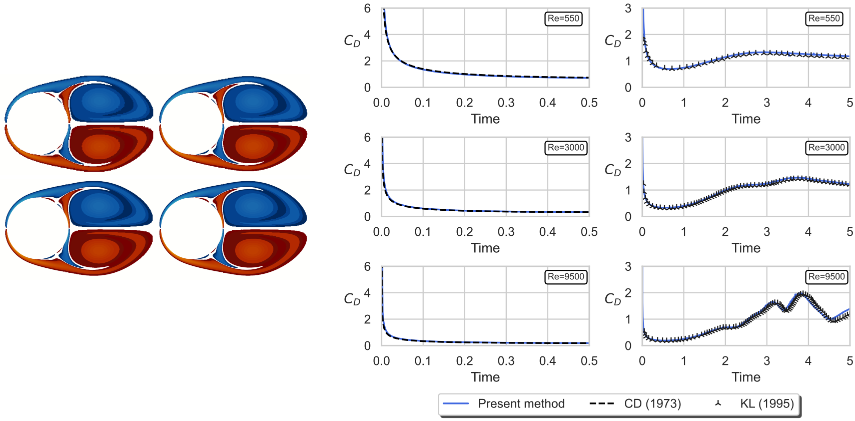


FIGURE 2: Left: Convergence of vorticity field with increasing resolution at Re= 1 000. From top left to bottom right: 6,7,8 and 9 levels of refinement. Right: Drag coefficient, validation results. Comparison of early drag history with analytical solution by [30] (left) and with simulations by [31] (right), for various Reynolds numbers.

Following [32] we also define the average number of gridpoints per dimension

$$\hat{N}^L = \left(\frac{1}{t_{\max} - t_{\min}} \int_{t_{\min}}^{t_{\max}} N^L(t) \right)^{1/2}, \tag{26}$$

as a measure of the computational cost, where $N^L(t)$ is the instantaneous number of grid points when L refinement levels are used. A summary of the simulations performed for this verification study is shown in table 1. Convergence of the vorticity field is visualised in the left of fig. 2, where it can be seen that the solutions for 8 and 9 refinement levels are indistinguishable from each other. To validate our method, we compare our results against the drag coefficient of the impulsively started cylinder at early times as computed analytically by [30] and against simulations by [31], for longer times. As shown in the right of fig. 2, our results are in excellent agreement with the references they are compared against.

3

DEEP REINFORCEMENT LEARNING

> *[..] to eliminate coercion: to apply controls by changing the environment in such a way as to reinforce the kind of behavior that benefits everyone.*
>
> — B. F. Skinner

The following we present reinforcement learning (RL) and it's application to systems with multiple agents. Furthermore, we present a combination of RL with Bayesian inference:

REMEMBER AND FORGET EXPERIENCE REPLAY FOR MULTI-AGENT REINFORCEMENT LEARNING We present the extension of the Remember and Forget for Experience Replay (ReF-ER) algorithm to Multi-Agent Reinforcement Learning (MARL). ReF-ER was shown to outperform state of the art algorithms for continuous control in problems ranging from the OpenAI Gym to complex fluid flows. In MARL, the dependencies between the agents are included in the state-value estimator and the environment dynamics are modeled via the importance weights used by ReF-ER. In collaborative environments, we find the best performance when the value is estimated using individual rewards and we ignore the effects of other actions on the transition map. We benchmark the performance of ReF-ER MARL on the Stanford Intelligent Systems Laboratory (SISL) environments. We find that employing a single feed-forward neural network for the policy and the value function in ReF-ER MARL, outperforms state of the art algorithms that rely on complex neural network architectures.

A BAYESIAN PERSPECTIVE ON UNCERTAINTIES IN DEEP REINFORCEMENT LEARNING Deep reinforcement learning promises new ways for automated discoveries in science and engineering. For reliable conclusions, the learned black-box controllers require an accurate estimation of the uncertainties. Most existing work focuses on testing the robustness by sampling the initial condition of the environment, while the aleatoric uncertainty addresses the exploration-exploitation dilemma. On the other hand, the epistemic uncertainty is evaluated a-posteriori by doing multiple experiments with different random seeds, an infeasible technique for computationally expensive applications. Bayesian inference is considered the gold standard for uncertainty quantification. However, there has yet to be a rigorous evaluation of the existing Bayesian deep learning methods for off-policy deep reinforcement learning. We close the gap by describing the integration of Bayesian inference with the remember and forget for experience replay algorithm and discussing approximation techniques to generalize the presented formalism to other algorithms. The following benchmarks on the MuJoCo continuous control tasks of the OpenAI Gym show that the discussed approach is applicable to assess the uncertainties during training and testing.

ACKNOWLEDGMENTS The presented methods extend V-RACER with ReF-ER by Guido Novati as published in [33]. The code Korali was initially developed and published by Georgios Arampatzis, Sergio Martin, and Daniel Wälchli for Bayesian Inference and Optimization [34]. The author joined the development when it was extended to support deep reinforcement learning. The author is thankful to the fruitful and very stimulating collaboration with Sergio Martin and Daniel Wälchli. The work presented in this chapter is based on [35, 36]. The computational resources were provided by the Swiss National Supercomputing Centre (CSCS) under project s929.

3.1 Remember and Forget for Experience Replay (ReF-ER)

Reinforcement Learning (RL) solves sequential decision making processes formalized by Markov Decision Processes (MDPs). An MDP is defined by the tuple $(\mathcal{S},\ \mathcal{A},\ r,\ D)$, defining the state $s \in \mathcal{S}$, the action $a \in \mathcal{A}$, the reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and transition map $D : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$. Given an initial state distribution $p(s)$ and a stochastic policy $\pi(a|s)$ we can define the Q- and state-value function V

$$\begin{aligned} Q^\pi(s,a) &= \mathbb{E}_{a_t\sim\pi(\cdot|s_t)}\left[\sum_{t=0}^{\infty}\gamma^t r_t | s_0 = s, a_0 = a\right], \\ V^\pi(s) &= \mathbb{E}_{a\sim\pi}\left[Q^\pi(s,a)\right]. \end{aligned} \tag{27}$$

where $\gamma \in [0,1)$ is the discount factor and $r_t = r(s_t, a_t)$. The optimal policy $\pi^\star$ is defined as

$$\pi^\star = \arg\max_\pi V^\pi(s)\,, \quad \forall s \in \mathcal{S}\,. \tag{28}$$

In order to find the optimal policy, the agents interact with the environment at discrete timesteps t. At every timestep, the agents take actions a_t based on the observation of a states s_t. Thereby the environment transitions into new states s_{t+1} and returns rewards r_t.

Off-policy methods are among the most sample efficient methods for RL. They reuse past *experience*, which is tuples consisting of states, actions, and rewards (s_t, a_t, r_t) to learn the optimal policy. The experiences are collected as *episodes* $\mathcal{E}_k = \{(s_t, a_t, r_t)\}_{t=1}^{T_k}$ and stored in a *replay memory* $\mathcal{RM} = \{\mathcal{E}_k\}_{k=1}^{K}$. A successful algorithm to find the optimal policy for computationally demanding optimal control problems is V-RACER with ReF-ER [33]. Here, a single neural network is trained to approximate the policy $\pi_{\boldsymbol{\omega}}(a|s)$ and value function $V^\pi_{\boldsymbol{\vartheta}}(s)$. We denote the respective weights by $\boldsymbol{\omega}$ and $\boldsymbol{\vartheta}$. Note, that the parameters $\boldsymbol{\omega}$ and $\boldsymbol{\vartheta}$ only differ in the output layer.

For the value function the weights $\boldsymbol{\vartheta}$ of the neural network $V^\pi_{\boldsymbol{\vartheta}}(s)$ are updated to minimize

$$\mathcal{L}(\boldsymbol{\vartheta}) = \mathbb{E}_d\left[\left(V^\pi_{\boldsymbol{\vartheta}}(s_t) - \hat{V}^{\text{tbc}}_t\right)^2\right]. \tag{29}$$

Here, the target $\hat{V}^{\text{tbc}}_t$ is based on the on-policy returns estimator Retrace [37]

$$\hat{Q}^{\text{ret}}_t = r_t + \gamma V^\pi(s_t) + \gamma\bar{\rho}_t(\hat{Q}^{\text{ret}}_{t+1} - Q^\pi(s_t, a_t))\,. \tag{30}$$

Since V-RACER approximates $Q^\pi(s_t, a_t) \approx V^\pi(s_t)$ the V-trace estimator [38] becomes

$$\hat{V}_t^{\text{tbc}} = V^\pi(s_t) + \bar{\rho}_t \left[r_t + \gamma \hat{V}_{t+1}^{\text{tbc}} - V^\pi(s_t) \right] , \tag{31}$$

and can be related to the on-policy returns estimator Retrace via

$$\hat{Q}_t^{\text{ret}} = r_t + \gamma \hat{V}_{t+1}^{\text{tbc}} . \tag{32}$$

We learn the weights ω of the policy $\pi_\omega(s|a)$ by maximizing the expected advantage

$$\mathcal{L}(\omega) = \mathbb{E}_d \left[\rho_t(\omega) \left(\hat{Q}_t^{\text{ret}} - V^\pi(s_t) \right) \right] . \tag{33}$$

Taking the gradient with respect to the weights ω yields the off-policy gradient [39]. The importance weight $\rho_t(\omega)$ is given by

$$\rho_t(\omega) = \frac{\pi_\omega(a_t|s_t)}{\pi(a_t|s_t)} . \tag{34}$$

Here the denominator denotes the policy that was used when s was sampled. The truncated importance weight $\bar{\rho}_t = \min(1, \rho_t)$ is used for the value estimate. In order to update the policy, a mini-batch of experiences is sampled from the replay memory. For each of these experiences a gradient $\hat{\boldsymbol{g}}(\omega)$ is computed. ReF-ER avoids negative impact from off-policy data by classifying experiences based on the importance weight ρ. The criterion $\frac{1}{c_{\max}} < \rho < c_{\max}$ depends on the cut-off $c_{\max}$ that is annealed during training. The gradient is then regularized via

$$\hat{\boldsymbol{g}}^{\text{R}}(\omega) = \begin{cases} \beta \hat{\boldsymbol{g}}(\omega) - (1-\beta)\hat{\boldsymbol{g}}^{\text{KL}}(\omega) \,, & \text{on-policy} \\ -(1-\beta)\hat{\boldsymbol{g}}^{\text{KL}}(\omega) \,, & \text{else.} \end{cases} \tag{35}$$

The factor β is adjusted according to the fraction of off-policy samples $n_{\text{off-policy}}$ in the replay memory

$$\beta \leftarrow \begin{cases} (1-\eta)\beta & \text{if } n_{\text{off-policy}} > n^\star \\ (1-\eta)\beta + \eta, & \text{otherwise.} \end{cases} \tag{36}$$

Here η denotes the learning rate used by Adam. This allows maintaining a target fraction $n^\star$ of off-policy experiences in the replay memory. The regularizer is the gradient of the Kullback-Leibler divergence between the current and the past policy [40]

$$\hat{\boldsymbol{g}}^{\text{KL}}(\omega) = \nabla_\omega D_{\text{KL}} \left(\pi_k \| \pi_\omega \right) . \tag{37}$$

3.2 Remember and Forget Experience Replay for Multi-Agent Reinforcement Learning

State of the art deep reinforcement learning (RL) algorithms approximate the optimal value function and policy using deep neural networks. This approach has been showcased in the playing of Atari games [41], board games like Shōgi, Chess, and Go [42]. More recently Multi-Agent Reinforcement Learning (MARL) has been applied to multiplayer games such as poker [43] and prominent computer games such as Dota 2 and StarCraftII [44, 45]. Tasks that require multiple agents to collaborate often rely on a generalization of single agent RL algorithms and employ complex neural network architectures for the value estimation. Deep MARL presents several algorithmic challenges. Learning individual policies is hard in environments with multiple learning agents that can be non-stationary. Moreover, the inclusion of collaborative behaviour by an averaging of the rewards yields a credit assignment problem, hindering the reinforcement of beneficial behaviour. Finally, most games are restricted to a discrete action space. More recently the extension of MARL to scientific computing and complex systems (scientific multi-agent reinforcement learning (SciMARL) [46, 47]) has showcase the importance of control in continuous, high-dimensional action spaces. To the best of our knowledge, generalizations of well-established algorithms for continuous action RL is limited [48, 49].

We address these challenges by revisiting the relationships and interactions between multiple agents. We follow the centralized training with decentralized execution paradigm [50, 51] (CTDE). Agents learn using the experiences from all their peers, while at execution time, the learned policy is used to make decisions based on a single agent's state information. The inclusion of the observations from all agents addresses the non-stationary issues, and the adoption of ReF-ER handles effectively far-policy experiences [33]. Furthermore, the learning rule can be modified to systematically examine the credit assignment problem. Lastly, ReF-ER allows to model the interaction strength between agents via the importance weight.

3.2.1 Formalism

Multi-Agent Reinforcement Learning (MARL) aims to find the optimal policy in a Multi-Agent Markov Decision Process (MAMDP). A MAMDP is a tuple $(\mathcal{S}, \mathcal{A}, \boldsymbol{r}, N, D)$ consisting of the states $\boldsymbol{s} \in \mathcal{S}^N$, actions $\boldsymbol{a} \in \mathcal{A}^N$, and rewards $\boldsymbol{r} \in \mathbb{R}^N$ observed by the $N \in \mathbb{N}_+$ agents in the environment. Here we

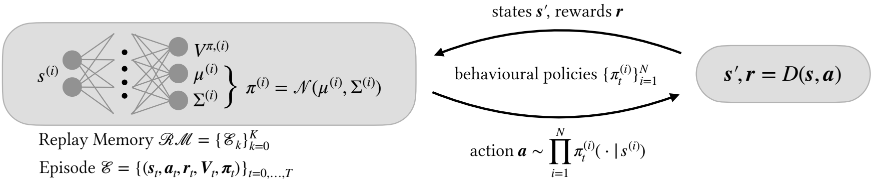

states $\boldsymbol{s}'$, rewards $\boldsymbol{r}$
$s^{(i)}$
$V^{\pi,(i)}$
$\mu^{(i)}$
$\Sigma^{(i)}$
$\pi^{(i)} = \mathcal{N}(\mu^{(i)}, \Sigma^{(i)})$
behavioural policies $\{\pi_t^{(i)}\}_{i=1}^N$
$\boldsymbol{s}', \boldsymbol{r} = D(\boldsymbol{s}, \boldsymbol{a})$
Replay Memory $\mathscr{RM} = \{\mathscr{E}_k\}_{k=0}^K$
Episode $\mathscr{E} = \{(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{r}_t, \boldsymbol{V}_t, \boldsymbol{\pi}_t)\}_{t=0,\dots,T}$
action $\boldsymbol{a} \sim \prod_{i=1}^N \pi_t^{(i)}(\,\cdot\,|\,s^{(i)})$

FIGURE 3: Schematic of the MARL loop.

consider a homogeneous setting, where all agents have the same state space, action space, and reward function. The agents' individual states, actions, and rewards are denoted by $s^{(i)} \in \mathcal{S}$, $a^{(i)} \in \mathcal{A}$, and $r^{(i)} \in \mathbb{R}$ for $i = 1, \dots, N$. The transition map is given by $\boldsymbol{r}, \boldsymbol{s}' = D(\boldsymbol{s}, \boldsymbol{a})$, i.e. it depends on the states and actions from all agents. This is contrary to environments in which agents take actions sequentially and the state is updated after each action. In MAMDP we specify the initial state distribution $p(\boldsymbol{s})$ and the stochastic policy $\pi(\boldsymbol{a}|\boldsymbol{s})$. The later is the probability distribution over the actions of the agents given the states. We assume that the policy factorizes as

$$\pi(\boldsymbol{a}|\boldsymbol{s}) = \prod_{i=1}^{N} \pi(a^{(i)}|s^{(i)}) \,. \tag{38}$$

This allows for *decentralized execution*, where the agent samples an action solely based on its state. The vector-valued state-action-value $\boldsymbol{Q}$ function and state-value $\boldsymbol{V}$ are defined as

$$\boldsymbol{Q}^{\pi}(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \boldsymbol{r}_t | \boldsymbol{s}_0 = \boldsymbol{s}, \boldsymbol{a}_0 = \boldsymbol{a}\right] , \quad \boldsymbol{V}^{\pi}(\boldsymbol{s}) = \mathbb{E}_{\pi}\left[\boldsymbol{Q}^{\pi}(\boldsymbol{s}, \boldsymbol{a})\right] . \tag{39}$$

where $\gamma \in [0, 1)$ is the discount factor and $\boldsymbol{r}_t = \boldsymbol{r}(\boldsymbol{s}_t, \boldsymbol{a}_t)$. Assuming equal importance for all agents yields the scalar state-value that is expressed in terms of the individual state value functions $V^{\pi}(s^{(i)})$

$$P\left[\boldsymbol{V}^{\pi}(\boldsymbol{s})\right] = \frac{1}{N} \sum_{i=1}^{N} V^{\pi}(s^{(i)}) \,. \tag{40}$$

The optimal policy $\pi^{\star}$ in the MAMDP maximizes the average of the state-values for all agents

$$\pi^{\star} = \arg\max_{\pi} \frac{1}{N} \sum_{i=1}^{N} V^{\pi}(s^{(i)}) \,,\ \forall \boldsymbol{s} \in \mathcal{S}^N \,. \tag{41}$$

In MARL the agents interact with the environment as depicted in fig. 3. At timestep t, the agents take actions $\boldsymbol{a}_t$ based on the states $\boldsymbol{s}_t$. Thereby the environment transitions into new states $\boldsymbol{s}_{t+1}$ and returns rewards $\boldsymbol{r}_t$. An *experience*, consists of states, actions, and rewards $(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{r}_t)$. Upon termination of the episode $\mathcal{E}_k = \{(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{r}_t)\}_{t=1}^{T_k}$ the experiences are stored in the replay memory $\mathcal{RM} = \{\mathcal{E}_k\}_{k=1}^{K}$. In the following d denotes the distribution of the experiences in the replay memory.

We extend V-RACER with ReF-ER [33] for MARL, training a neural network that approximates the state-value and the policy parameters. Using the experiences in the replay memory, we train the neural network with Adam [52]. In the following, $\pi_{\omega}(a|s)$ and $V^{\pi}_{\boldsymbol{\vartheta}}(s)$ denote the policy and state-value function with the respective weights $\boldsymbol{\omega}$ and $\boldsymbol{\vartheta}$. Note, that the parameters $\boldsymbol{\omega}$ and $\boldsymbol{\vartheta}$ only differ in the output layer.

3.2.1.1 Value Learning

The relationships between the agents is included in the learning process by introducing a scalarization function $f : \mathbb{R}^N \to \mathbb{R}$ in the on-policy returns estimator Retrace [37]

$$\hat{Q}^{\text{ret}}_{t,f} = f(\boldsymbol{r}_t) + \gamma V^{\pi}_f(\boldsymbol{s}_t) + \gamma \bar{\rho}_t (\hat{Q}^{\text{ret}}_{t+1,f} - Q^{\pi}_f(\boldsymbol{s}_t, \boldsymbol{a}_t)) \,. \tag{42}$$

V-RACER approximates $Q^{\pi}_f(\boldsymbol{s}_t, \boldsymbol{a}_t) \approx V^{\pi}_f(\boldsymbol{s}_t)$ and hence the V-trace estimator [38] becomes

$$\hat{V}^{\text{tbc}}_{t,f} = V^{\pi}_f(\boldsymbol{s}_t) + \bar{\rho}_t \left[f(\boldsymbol{r})_t + \gamma \hat{V}^{\text{tbc}}_{t+1,f} - V^{\pi}_f(\boldsymbol{s}_t) \right] \,, \tag{43}$$

and it can be related to the on-policy returns estimator Retrace via

$$\hat{Q}^{\text{ret}}_{t,f} = f(\boldsymbol{r})_t + \gamma \hat{V}^{\text{tbc}}_{t+1,f} \,. \tag{44}$$

The truncated importance weight $\bar{\rho}_t$ will be defined in the following section. It balances the impact from off-policy data on the current estimator of the state-value.

For the function f we distinguish two cases. The first case, which we refer to as *individual*, assumes that the value estimate depends solely on the reward observed by agent i

$$f^{(i)}_{\text{individual}}(\boldsymbol{r}) = r^{(i)} \,, \quad V^{\pi}_{\text{individual},(i)}(\boldsymbol{s}) = V^{\pi}(s^{(i)}) \,. \tag{45}$$

In the second case, referred to as *cooperative*, the reward and the state-values are averaged

$$f_{\text{cooperative}}(\boldsymbol{r}) = \frac{1}{N}\sum_{i=1}^{N} r^{(i)}\,, \quad V^{\pi}_{\text{cooperative}}(\boldsymbol{s}) = \frac{1}{N}\sum_{i=1}^{N} V^{\pi}(s^{(i)})\,. \tag{46}$$

The weights ϑ of the neural network $V^{\pi}_{\vartheta}(s)$ are updated to minimize the loss

$$\mathcal{L}(\vartheta) = \mathbb{E}_d\left[\frac{1}{N}\sum_{i=1}^{N}\left(V^{\pi}_{\vartheta}(s_t^{(i)}) - \hat{V}^{\text{tbc}}_{t,f}\right)^2\right]. \tag{47}$$

Note, that both variants reduce to the original loss proposed in [33] when assuming a single agent.

3.2.1.2 Policy Gradient

Given the definition of $\hat{Q}^{\text{ret}}_{t,f}$, we learn the weights ω of the policy $\pi_\omega(s|a)$ by maximizing the expected advantage

$$\mathcal{L}(\omega) = \mathbb{E}_d\left[\frac{1}{N}\sum_{i=1}^{N}\rho_t(\omega)\left(\hat{Q}^{\text{ret}}_{t,f} - V^{\pi}(s_t^{(i)})\right)\right]. \tag{48}$$

Taking the gradient with respect to the weights ω yields the off-policy gradient [39]. The importance weights $\rho_t(\omega)$ reflect the assumed dynamics of the environment.

Adopting the *full dynamics* $\boldsymbol{r}^{t+1}, \boldsymbol{s}^{t+1} = D(\boldsymbol{s}^t, \boldsymbol{a}^t)$ and the factorization of the policy from eq. (73) yields the importance weight

$$\rho_t^N(\omega) = \prod_{i=1}^{N}\frac{\pi_\omega(a_t^{(i)}|s_t^{(i)})}{\pi(a_t^{(i)}|s_t^{(i)})}\,. \tag{49}$$

Here the denominator denotes the policy that was used when $s^{(i)}$ was sampled. When using this importance weight the value estimate and the policy update depend on the probability of the action of all agents.

In cases where interactions between the agents are negligible, the value estimate and the policy update should not be influenced by the probabilities of the other agents. This is achieved by using the importance weight

$$\rho_t(\omega) = \frac{\pi_\omega(a_t^{(i)}|s_t^{(i)})}{\pi(a_t^{(i)}|s_t^{(i)})}\,. \tag{50}$$

We denote it as the *local dynamics model*.

3.2.1.3 Remember and Forget for Experience Replay

In order to update the policy, a mini-batch of experiences is sampled from the replay memory. For each of these experiences a gradient $\hat{\boldsymbol{g}}(\boldsymbol{\omega})$ is computed. ReF-ER avoids negative impact from off-policy data by classifying experiences based on the importance weight ρ. The criterion $\frac{1}{c_{\text{max}}} < \rho < c_{\text{max}}$ depends on the cut-off c_{max} that is annealed during training. The gradient is then regularized via

$$\hat{\boldsymbol{g}}^{\text{R}}(\boldsymbol{\omega}) = \begin{cases} \beta\hat{\boldsymbol{g}}(\boldsymbol{\omega}) - (1-\beta)\hat{\boldsymbol{g}}^{\text{KL}}(\boldsymbol{\omega}), & \text{on-policy} \\ -(1-\beta)\hat{\boldsymbol{g}}^{\text{KL}}(\boldsymbol{\omega}), & \text{else.} \end{cases} \tag{51}$$

The factor β is adjusted according to the fraction of off-policy samples $n_{\text{off-policy}}$ in the replay memory

$$\beta \leftarrow \begin{cases} (1-\eta)\beta & \text{if } n_{\text{off-policy}} > n^{\star} \\ (1-\eta)\beta + \eta, & \text{otherwise.} \end{cases} \tag{52}$$

Here η denotes the learning rate used by Adam. This allows maintaining a target fraction $n^{\star}$ of off-policy experiences in the replay memory. The regularizer is the gradient of the Kullback-Leibler divergence between the current and the past policy [40]

$$\hat{\boldsymbol{g}}^{\text{KL}}(\boldsymbol{\omega}) = \nabla_{\boldsymbol{\omega}} D_{\text{KL}}\left(\pi_k \| \pi_{\boldsymbol{\omega}}\right) . \tag{53}$$

For details we refer to the publication of the ReF-ER algorithm [33].

3.2.2 SISL Environments

We benchmark the performance of our algorithms in the three cooperative Stanford Intelligent Systems Laboratory Environments (SISL) implemented in PettingZoo [48, 53] (see fig. 4).

In *Multiwalker* a package is located on top of three bipedal robots. The reward for each walker depends on the distance the package has traveled plus 130 times the change in the walker's position. All agents receive -100 reward if any walker or the package falls. The walker that falls is further penalized by -10 reward. Each walker exerts force on two joints in their two legs, giving a continuous action space represented as a four element vector $\mathcal{A}_{\text{multiwalker}} \subseteq \mathbb{R}^4$. The state consists of 31 real values $\mathcal{S}_{\text{multiwalker}} \subseteq \mathbb{R}^{31}$ consisting of simulated noisy linear data about the environment and

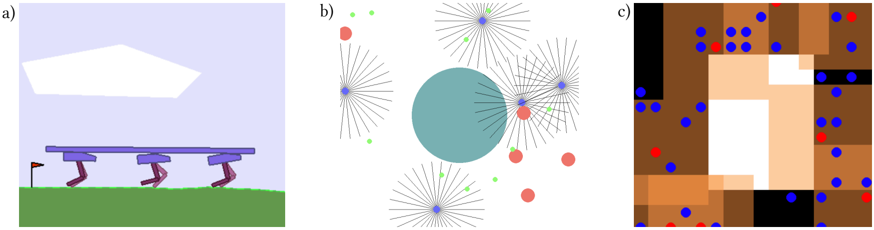


FIGURE 4: SISL environments: (a) Multiwalker, (b) Waterworld, and (c) Pursuit.

information about neighboring walkers. The environment ends after 500 steps or if the package or a walker falls.

In *Waterworld* five agents attempt to consume food while avoiding poison. There are ten moving poison targets, which have a radius of 0.75 times the radius of the agent. Furthermore there are five moving food targets with radius two times the size of the agent radius. In the center of the domain is a solid object. An agent obtains a shaping reward of 0.01 for touching food. The food can be consumed if two agents touch it simultaneously, in which case both participating agents obtain 10 reward. On the other hand, touching a poison target and consuming it gives -1 reward. After consumption both food and poison targets randomly reappear at another location in the environment. The agents have a continuous action space represented by two elements $\mathcal{A}_{\text{waterworld}} \subseteq \mathbb{R}^2$, which corresponds to horizontal and vertical thrust. In order to penalize unnecessary movement the agents obtain a negative reward based on the absolute value of the applied thrust. Each agents state results from 30 range-limited sensors, depicted by the black lines, which detect neighboring entities and result in a 242 element vector $\mathcal{S}_{\text{waterworld}} \subseteq \mathbb{R}^{242}$. The environment terminates after 500 steps.

In *Pursuit* there are 30 blue evaders and eight red pursuer agents in a 16×16 grid with an obstacle in the center, shown in white. Every time the pursuers fully surround an evader, each agent receives a reward of 5 and the evader is removed from the environment. In order to facilitate training, pursuers receive a shaping reward of 0.01 every time they touch an evader. The pursuers have a discrete action space, consisting of directions up, down, left, right or stay $\mathcal{A}_{\text{pursuit}} = \{\uparrow, \downarrow, \leftarrow, \rightarrow, \circ\}$. Each pursuer observes a 7×7 grid centered around itself, depicted by the orange boxes surrounding the red pursuer agents. For each of the gridpoints it receives three signals, the first signal indicates a wall, the second signal indicates the number of allies and the third signal indicates the number of opponents. Thus the observation space can be represented by $\mathcal{S}_{\text{pursuit}} \subseteq \mathbb{Z}^{147}$. The environment terminates after 500 steps or if all evaders are captured.

3.2.3 Evaluation

We benchmark the algorithms using five runs with 20 000 episodes each. For each run we compute the moving median of the cumulative reward averaged over all agents. The window size of the moving median is 100 episodes. In fig. 5 we plot the median of the medians and the 95% confidence interval of the five runs. Throughout this study, we apply the default parameters suggested in Novati & Koumoutsakos [33]. We set the discount factor $\gamma = 0.995$, we use a replay memory of size $2^{18} = 262\,144$ with initial random exploration with a randomly initialized neural network for $2^{17} = 131\,072$ experiences. The learning rate of the Adam optimizer is initialized to $\eta = 0.0001$ and the batchsize $B = 256$. For the approximation of the value function and the policy we use a neural network with two hidden layers of width 128. We initialize the ReF-ER factor $\beta = 0.3$, the target threshold $n^{\star} = 0.1$ and cut-off $c_{\text{max}} = 4$. In contrast to the original V-RACER, we use the clipped normal distribution for continuous action domains and introduce a discrete variant of V-RACER (see Supplementary Material). A pseudo code is available in appendix B. The SISL experiments where executed on a single CPU with 12 cores. The required training-times were approximately 48 hours per run. In total we performed 2x3x4x5=120 runs.

We distinguish the algorithms eq. (49) that calculate the importance weights as V-RACER *full dynamics individual*-FDI and *full dynamics cooperative*-FDCo, depending on the value estimator (eq. (45) and eq. (46)). Accordingly, we name the algorithms V-RACER *local dynamics individual*-LDI and *local dynamics cooperative*-LDCo, if we apply eq. (50) to calculate the importance weight. Finally, we distinguish training a single shared policy for all agents and one policy per agent. We sample a mini-batch for all variants and use the observations from all agents to train the neural networks using eq. (47) and eq. (48).

In fig. 5 we show the results for the three SISL environments. We find that the LDI algorithm results are superior to those of the other algorithmic variants. More specifically, while in the Multiwalker (a,d) environment the improvement is minimal, the advantage is clearly noticeable in the Waterworld (b,e) and Pursuit (c,f) environments. We argue that the FDI and LDI algorithms dominate their cooperative counterparts (FDCo and LDCo) because of the credit assignment problem: After averaging the rewards it

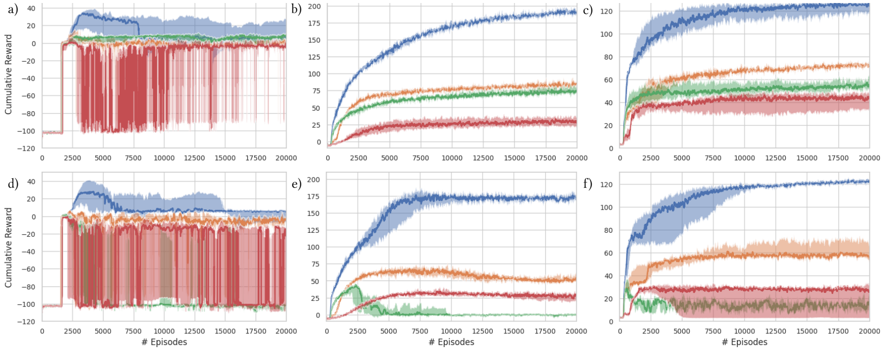


FIGURE 5: Median cumulative reward of 5 runs and 100 episodes, and averaged over all agents. The shading shows the 95% confidence interval. The results in the first row (a-c) are for a single shared policy, the second row (d-f) for one individual policy per agent. The first column (a,d) shows Multiwalker, the second (b,e) Waterworld, and the third (c,f) Pursuit. The lines correspond to LDI (—), FDI (—), LDCo (—), and FDCo (—).

TABLE 2: Maximum mean cumulative training reward, averaged over 5 runs and 100 episodes and state of the art by [49]. The number in the parenthesis denotes the episode in which it was achieved..

	Multiwalker	Waterworld	Pursuit
V-RACER (LDI, single policy)	**19.80 (3.2k)**	**194.57 (19.2k)**	**124.22 (20.8k)**
PPO, ApeX DDPG, ApeX DQN	13.67 (9.2k)	14.8 (6.3k)	77.63 (24k)
QMIX	-5.65 (17.8k)	1.62 (45.3k)	45.41 (59.6k)
MADDPG	-33.95(44.2k)	-1.81 (41k)	4.04 (28.8k)

is difficult to identify which actions contribute most to the success of the agents. Furthermore, sharing the experiences among all agents in a single policy is sufficient to resolve the non-stationary issue and using eq. (49) does not produce valuable information. In contrast, we find that taking the product of the individual importance weights harms the update. The relative order in terms of performance for the employed algorithms is similar when training multiple policies. Interestingly, the maximal performance of the LDI, reached during training, is worse for multiple policies. During the evaluation, however, the single policy variant is outperformed (see table 3). Another important difference arises for the LDCo. For multiple policies experience sharing does not remove the non-stationary issue. Together with the credit assignment problem, it hinders the convergence of the LDCo method. The FDCo alleviates this problem. By adding the information of the other agents in the importance weight (eq. (49)), the destructive effect of non-stationary and the credit assignment problem is resolved.

Additionally, we compare our results when training a single policy with the results in Terry *et al.* [49], which, to the best of our knowledge, represent the state of the art on the SISL environments. In Table 2 we show a comparison between the best results in Terry *et al.* [49] and our best performing alternative (LDI). We find that on all environments LDI outperforms the existing results by a margin.

In Multiwalker, we find that the LDI is the preferred algorithm. This can be especially seen during the first 8k episodes. After that, the returns for some runs drop significantly and yield a lower asymptotic median return. This behaviour is often observed in Walker-like environments. In the failing runs the agents start falling while trying to move faster and faster. In order to avoid this catastrophic failure the algorithm converges to a safe, but suboptimal policy. Nonetheless, the models that consider the local dynamics are the better performing alternatives. In the Multiwalker the LDI obtains a final return of 4.09, which is comparable to the attained return with the LDCo (3.73). The other variants do not achieve positive rewards (-31.68 for FDI and -38.11 for FDCo). We observe that in LDI certain runs identify a policy which consistently achieves a cumulative median return of around 20, however the algorithm does not learn the associated policy in every run. Here, the state of the art [49] achieves the highest cumulative reward when using a multi agent version of PPO [54]. From table 2 it can be seen, that the LDI outperforms the multi agent version of PPO.

In the second continuous action environment Waterworld, we see clear differences between the algorithms. The LDI outperforms the other variants

TABLE 3: Testing the optimal policy for LDI on the SISL environments. The mean, maximal, and minimal cumulative reward is computed over 50 episodes. We highlight the better result.

	Multiwalker			Waterworld			Pursuit		
	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min
One Policy	**1.0**	**1.9**	**0.4**	202.3	271.8	140.6	56.8	84.7	11.6
N Policies	-0.1	1.5	-2.2	**236.4**	**303.4**	**191.0**	**130.3**	**146.6**	**112.8**

significantly. We observe, that the FDI and the LDCo show similar performance. In Waterworld, it seems that correlating gradient updates (FDI) and rewards (LDCo) have similar negative effects. This effects seem to sum up when considering both correlations (FDCo). In Waterworld the state of the art is a MA version of ApeX DDPG [55]. The best variant of the proposed algorithm outperforms the existing results by 13X.

Finally, we discuss the results on the Pursuit benchmark. The negative effect arising from the credit assignment problem (LDCo) is larger than when correlating the update (FDI), which is contrary to our findings in Waterworld. Combining both assumptions (FDCo) still is the worst performing alternative. As can be seen in table 2, the present best performing alternative outperforms the state of the art (a MA version of ApeX DQN [55]). Comparing the maximal achieved mean returns shows that ReF-ER MARL provides 1.6X higher return.

3.2.4 ReF-ER and Scientific Multi-Agent Reinforcement Learning

We test LDI with multiple policies in the problem of fish schooling in the presence of strong hydrodynamic interactions, in high fidelity simulations of the Navier-Stokes equations [56]. We consider 20 self-propelled swimmers (figure fig. 6) interacting through their vortex wakes. The non-linear vortex interactions make this a challenging control problem [57]. Here, each swimmer observes a 16-dimensional state, encoding the ability to sense the environments via sight, flow sensing, and proprioception. Given the observation of the environment, the swimmers can control their motion via two actions that allow steering and changing the swimming velocity. Their reward is composed of the instantaneous swimming efficiency and a penalty term for collisions. In fig. 6 we show the trajectories relative to the center of mass

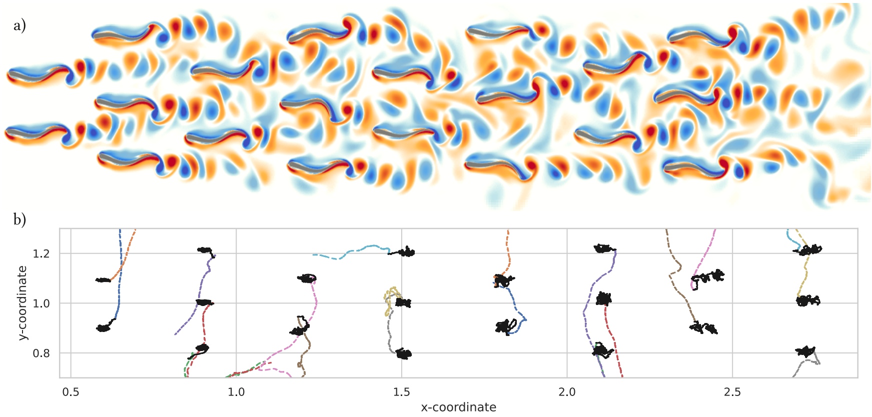


FIGURE 6: Structure of the school for 100 swimming periods. Figure (a) shows the swimmers in gray, where the colors visualize the vorticity field (blue negative and red positive). Figure (b) displays the trajectories of the controlled (black) and uncontrolled (dashed, color) swimmers relative to the center of mass of the school. An animation of both schools can be found in the supplementary materials.

of the school. Controlled schooling behaviour is observed for 100 tail beat periods whereas without control the swimmers collide at 1/5th of this time. Control for high fidelity simulations has a high computational cost, which requires asynchronous training and data collection. We use one compute node to train the neural networks and 64 independent simulations with two nodes for data collection. In total, the run took four days on 129 nodes of a Cray XC50 system equipped with a 12 core Intel® Xeon® CPU and one NVIDIA® Tesla® P100 GPU. This suggests the suitability of our method to expensive scientific computing applications.

3.3 A Bayesian Perspective on Uncertainties in Deep Reinforcement Learning

Uncertainty quantification is essential for scientific and engineering applications. It can be used to calibrate model parameters using experimental data, determine the sensitivity to the uncertainty of these parameters [58], and better understand the source of errors. In recent years, deep learning has proved successful in many domains of science and engineering [59, 60]. However, a rigorous assessment of the uncertainties in these models has been mostly overlooked due to many model parameters and the vast amount of data involved during the training process [61–63]. Efforts of Bayesian uncertainty quantification for deep learning models have primarily been focused on supervised learning [64]. While supervised learning uses a fixed dataset, in deep reinforcement learning (RL) the data is collected during training [65, 66]. During the training process and evaluation of an RL model, we need to clearly distinguish different sources of uncertainty: those coming from the stochastic environment, the noise added to the agents' actions to balance the exploration-exploitation trade-off, as well as the uncertainties from the stochastic training of the policy. An overview of the different sources of uncertainty is shown in fig. 7. In the present work, we focus on the last source of uncertainty, namely the stochastic training process.

The existing applications of Bayesian deep learning (BDL) to RL have been restricted to deep ensembles, dropout, and variants thereof. The use of ensembles has been focused on increasing the training performance and reducing variance in the estimates of the value function [67–74]. In contrast variance-enhancing methods, such as dropout are usually ignored as they

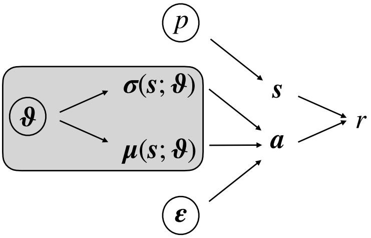


FIGURE 7: Sources of uncertainty in reinforcement learning. The circled elements denote sources of uncertainty and the arrows indicate dependencies between the variables. In gray, are the uncertainties that are targeted with the proposed Bayesian methodology.

can hinder training. However, they are established as cheap alternatives to the computationally expensive ensemble methods [75–78]. Markov Chain Monte Carlo (MCMC) methods, such as Langevin dynamics and Hamiltonian Monte Carlo, are sparsely discussed in the literature, as these methods are more challenging to implement and calibrate [79–81]. Although recent studies discuss the role of uncertainties, a Bayesian perspective is missing [82–86]. The present work provides an introduction to Bayesian uncertainty quantification in RL, and we use the MuJoCo [87] continuous control tasks in the OpenAI gym [88] in order to assess these methods during training and policy evaluation.

3.3.1 Bayesian Inference

In a Bayesian framework, parameters and model outputs are treated as random variables. It assumes the knowledge of the *likelihood* $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\vartheta})$ of a prediction $\boldsymbol{y}$ for some model $\mathcal{M}$ with input $\boldsymbol{x}$ and parameters $\boldsymbol{\vartheta}$. The *prior distribution* $p(\boldsymbol{\vartheta})$ can be assumed or learned from data and describes the degree of knowledge about the unknown parameters $\boldsymbol{\vartheta}$. We increase our knowledge about the parameters using data $D = \{(\boldsymbol{x}, \boldsymbol{y})_i\}_{i=1}^N$. Splitting the data into the inputs $X = \{\boldsymbol{x}_i\}_{i=1}^N$ and target values $Y = \{\boldsymbol{y}_i\}_{i=1}^N$ and applying Bayes' theorem we can compute the *posterior distribution* of the parameters

$$p(\boldsymbol{\vartheta}|X, Y) = \frac{p(Y|X, \boldsymbol{\vartheta})p(\boldsymbol{\vartheta})}{p(X, Y)} \,. \tag{54}$$

The denominator on the right-hand side is the *evidence*, or *marginal likelihood*. Assuming independent and identically distributed (i.i.d) data, we can write the likelihood as

$$p(Y|X, \boldsymbol{\vartheta}) = \prod_{i=1}^{N} p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{\vartheta}) \,. \tag{55}$$

During evaluation of the model, the prediction $\hat{\boldsymbol{y}}$ for an input $\hat{\boldsymbol{x}}$ is obtained by sampling the *posterior predicitive distribution*

$$p(\hat{\boldsymbol{y}}|\hat{\boldsymbol{x}}) = \int p(\hat{\boldsymbol{y}}|\hat{\boldsymbol{x}}, \boldsymbol{\vartheta})p(\boldsymbol{\vartheta}|X, Y)\, \mathrm{d}\boldsymbol{\vartheta} \,. \tag{56}$$

This is in large contrast to frequentist inference, where predictions are made using $p(\hat{\boldsymbol{y}}|\hat{\boldsymbol{x}}, \boldsymbol{\vartheta}^\star)$ based on a single set of parameters $\boldsymbol{\vartheta}^\star$ obtained by performing maximum a-posteriori estimation

$$\boldsymbol{\vartheta}^\star = \arg\max_{\boldsymbol{\vartheta}} \log p(\boldsymbol{\vartheta}|X, Y) \,. \tag{57}$$

This does not consider the distribution of the parameters and therefore does only allow estimating the uncertainties of the estimate under stringent assumptions such as a Gaussian distribution of the errors in the data.

3.3.2 Reinforcement Learning

Reinforcement Learning (RL) solves sequential decision-making processes formalized by Markov Decision Processes (MDPs). An MDP is defined by the tuple $(\mathcal{S},\ \mathcal{A},\ r,\ D)$ consisting of a state-space $s \in \mathcal{S}$, an action-space $a \in \mathcal{A}$, a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and an unknown, stochastic transition map $D(s'|a, s)$. For a given stochastic policy $\pi(a|s)$ we can define the state-value function

$$V^{\pi}(s) = \mathbb{E}_{D,\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s\right] . \tag{58}$$

Here $\gamma \in [0, 1)$ is the discount factor and we abbreviate $r_t = r(s_t, a_t)$ for time t. The goal is to find the optimal policy

$$\pi^\star = \arg\max_{\pi} V^{\pi}(s) , \quad \forall s \in \mathcal{S} . \tag{59}$$

Algorithms that compute the optimal policy are based on interactions with the environment. At every timestep, the agent takes an action a_t based on the observation of a state s_t. Thereby the environment transitions into a new state s_{t+1} and the reward r_t is computed. In off-policy actor-critic methods the states, actions, and rewards are stored in a replay memory. From the data stored in the replay memory the optimal policy $\pi(a|s; \boldsymbol{\vartheta})$ and value function $V(s; \boldsymbol{\vartheta})$ with parameters $\boldsymbol{\vartheta}$ is learned. State-of-the-art deep RL employs neural networks $\mathrm{NN}(s; \boldsymbol{\vartheta})$ as function approximator, where the weights $\boldsymbol{\vartheta}$ of the neural network are typically optimized using stochastic gradient descent

$$\boldsymbol{\vartheta}^{n+1} = \boldsymbol{\vartheta}^{n} - \eta \nabla_{\boldsymbol{\vartheta}} L(\boldsymbol{\vartheta}^{n}, D) . \tag{60}$$

Here η denotes the learning rate and $\nabla_{\boldsymbol{\vartheta}} L(\boldsymbol{\vartheta}^{n}, D)$ denotes the stochastic gradient estimator computed with respect to the parameters of the empirical loss computed over a mini-batch of samples D. Upon convergence, this approach yields a point estimate $\mathrm{NN}(s; \boldsymbol{\vartheta}^\star)$.

3.3.3 Bayesian Deep Reinforcement Learning

For Bayesian RL we propose learning the *posterior predictive policy*

$$\pi(a|s) = \int \pi(a|s;\boldsymbol{\vartheta})\, p(\boldsymbol{\vartheta}|D)\mathrm{d}\boldsymbol{\vartheta}\,. \tag{61}$$

The posterior distribution is defined as the negative exponential $p(\boldsymbol{\vartheta}|D) = \exp[-L(\boldsymbol{\vartheta}, D)]$ such that the point estimate discussed in section 5.1 corresponds to the maximum a-posteriori estimate. Without loss of generality, we assume that the policy is parameterized with mean $\boldsymbol{\mu}(s;\boldsymbol{\vartheta})$ and standard deviation $\boldsymbol{\sigma}(s;\boldsymbol{\vartheta})$. For the posterior predictive distribution, the *total uncertainty* of the action in state s can then be computed from

$$\Delta a(s)^2 = \mathrm{var}_a\left[\pi(a|s)\right]. \tag{62}$$

The total uncertainty can be decomposed into the *epistemic uncertainty*

$$\delta a(s)^2 = \mathrm{var}_{\boldsymbol{\vartheta}|D}\left[\boldsymbol{\mu}(s;\boldsymbol{\vartheta})\right], \tag{63}$$

and the *aleatoric uncertainty*

$$da(s)^2 = \Delta a(s)^2 - \delta a(s)^2\,. \tag{64}$$

Similarly, the value function V or state-action value function Q can be computed via

$$V(s) = \mathbb{E}_{\boldsymbol{\vartheta}|D}\left[V(s;\boldsymbol{\vartheta})\right], \quad Q(s,a) = \mathbb{E}_{\boldsymbol{\vartheta}|D}\left[Q(s,a;\boldsymbol{\vartheta})\right], \tag{65}$$

instead of using a point estimate.

We note, that in RL the stochastic form of the policy is introduced to cope with the exploration-exploitation dilemma [89]. Although algorithms like the used VRACER algorithm [33] learn the variance used to sample the action, the underlying objective is to maximize the cumulative reward, not to express the uncertainty of an action. In safety-critical applications, the correct assessment of uncertainty is essential, and as we will see in the following section how the present method allows a clearer picture.

3.3.3.1 Approximation

The expectation values in eqs. (61) and (65) can not be computed analytically. In order to compute an approximation, we assume that we are given $i =$

$1, \dots, N$ samples $\boldsymbol{\vartheta}^{(i)} \sim p(\cdot|D)$ from the posterior distribution. Using these samples we can use Monte Carlo integration to approximate

$$\pi(a|s) \approx \frac{1}{N} \sum_{i=1}^{N} \pi(a|s; \boldsymbol{\vartheta}^{(i)}) \,, \tag{66}$$

and

$$V(s) \approx \frac{1}{N} \sum_{i=1}^{N} V(s; \boldsymbol{\vartheta}^{(i)}) \,, \quad Q(s,a) \approx \frac{1}{N} \sum_{i=1}^{N} Q(s,a; \boldsymbol{\vartheta}^{(i)}) \,. \tag{67}$$

Under this approximation, an action is sampled by first sampling a categorical distribution with $p_i = 1/N$ for $i = 1, \dots, N$. The action is then found by sampling the realization $\pi(a|s; \boldsymbol{\vartheta}^{(i)})$ of the policy for parameters $\boldsymbol{\vartheta}^{(i)}$. For this approximation and assuming independence between the components of the action a_j the total uncertainty is

$$\Delta a_j(s) \approx \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(\mu_j^2(s; \boldsymbol{\vartheta}^{(i)}) + \sigma_j^2(s; \boldsymbol{\vartheta}^{(i)}) \right) - \hat{\mu}_j^2(s)} \,. \tag{68}$$

Where μ_j and σ_j are the mean and standard deviation of the j-th action and

$$\hat{\mu}_j(s) = \frac{1}{N} \sum_{i=1}^{N} \mu_j(s; \boldsymbol{\vartheta}^{(i)}) \,. \tag{69}$$

Furthermore, the epistemic uncertainty reads

$$\delta a_j(s) \approx \sqrt{\frac{1}{N} \sum_{i=1}^{N} \mu_j^2(s; \boldsymbol{\vartheta}^{(i)}) - \hat{\mu}_j^2(s)} \,, \tag{70}$$

while the aleatoric uncertainty becomes

$$da(s) \approx \sqrt{\frac{1}{N} \sum_{i=1}^{N} \sigma_j^2(s; \boldsymbol{\vartheta}^{(i)})} \,. \tag{71}$$

This is in accordance with our intuitive picture, where the noise from the stochastic policy becomes the aleatoric uncertainty, and the variance in the mean action for different samples of the posterior corresponds to the epistemic uncertainty.

3.3.3.2 Sampling Algorithms

In the following paragraphs, we will present the prominent methods from the literature, and discuss their applicability to sample the posterior distribution in deep RL models.

MONTE CARLO DROPOUT (MCD) One of the central problems in deep learning is overfitting. One method that was found to be an effective regularization techniques are Dropout [90] and DropConnect [91]. Here, the idea is to randomly set weights in the neural network to zero. Later work showed that dropout is a Bayesian approximation and can be used to compute samples from the predictive posterior distribution of the data [92, 93]. In RL, this was found to be ineffective due to inconsistencies arising between training and testing [78].

STOCHASTIC WEIGHT AVERAGING GAUSSIAN (SWAG) Upon convergence, SGD only changes the weights of the neural network in the vicinity of a local minimum. The idea of stochastic weight averaging is to average these weights in order to improve the predictive performance of deep neural networks [94]. In later work, this idea was extended to approximate a Gaussian distribution based on the samples [95]. While in RL this method is readily applicable given the most recent weights of an already trained model, during training, only the most recent samples can be considered.

DEEP ENSEMBLES (DE) Instead of training one neural network, we can configure the learning to train N neural networks. It was shown that the samples obtained like this allow a good approximation of the predictive posterior distribution [96]. These networks are not only initialized independently but also trained on separate mini-batches. Ensembles can be combined with all of the methods presented so far and should help sampling from multiple local minima [76, 86, 97, 98].

LANGEVIN DYNAMICS (LD) It was found that suitable choices of the learning rate guarantee convergence of SGD to a local minimum and that the parameters recorded on the trajectory are samples from an approximate posterior distribution [99]. This also applies to variants with momentum like the Adam optimizer [100]. Later it was shown, that adding the correct amount of noise to the trajectory of stochastic gradient descent yields samples from the full posterior distribution [101, 102]. Similarly to variants of SGD like

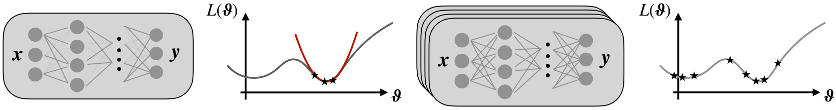


FIGURE 8: Visualization of sampling techniques. From left to right: MCD proposes to disable connections in the neural network at random. SWAG computes a Gaussian approximation of the posterior based on past samples. DE methods train several independent networks. LD and HMC are Markov-Chain Monte Carlo methods to sample the posterior distribution.

Adam, convergence can be improved by adopting the covariance of the added noise by the empirical covariance of the gradients [103–106].

HAMILTONIAN MONTE CARLO (HMC) Hamiltonian Monte Carlo (HMC) re-frames the inference problem in the framework of Hamiltonian mechanics. By introducing the energy and momentum variables, the Hamilton equations can be solved in order to compute proposals in a Metropolis–Hastings algorithm [107]. This approach requires the availability of gradient information, which are readily available for neural networks [108, 109]. With the increasing complexity of the neural network architectures and the size of the datasets, standard HMC was soon found to be infeasible and SGD-HMC was introduced [110, 111]. Similar to the approaches from Langevin dynamics, the geometry of the loss surface can be taken into account [112].

3.3.3.3 Policy Approximation

The present section shows some approaches to retain the Gaussian form of the policy that is required by many algorithms. Since the gradient of the importance weight can be computed based on eq. (66), the reason we present these approximations are mainly to approximate the KL-divergence [113–115] and other regularization terms that might rely on a Gaussian policy.

NAIVE From a formal point of view, off-policy methods like ReF-ER can be understand as a sampling from an ensemble of past policies [116], thus these algorithms should also applicable for the more general ensemble like eq. (66). This means that even though during inference we sample the action from eq. (61), we approximate

$$\pi(a|s) \approx \pi(a|s;\boldsymbol{\vartheta}), \quad V(s) \approx V(s;\boldsymbol{\vartheta}) \tag{72}$$

during the training in order to compute the policy gradient. We have observed that this requires clipping the KL gradient to 10^7, an approach that was also used in the baseline model to avoid numerical problems. We call this method the *Naive Model* (NM).

GAUSSIAN Another option is to approximate the mixture distribution arising from the Monte Carlo approximation in eq. (61) using a diagonal, clipped Gaussian distribution

$$\pi(a|s) \approx \mathcal{N}^c(\hat{\boldsymbol{\mu}}(s), \mathrm{diag}(\hat{\boldsymbol{\sigma}}^2(s))). \tag{73}$$

Using the chain rule we find the gradients to update the weights

$$\begin{aligned}\frac{\mathrm{d}L(\boldsymbol{\vartheta})}{\mathrm{d}\mu_j(s;\boldsymbol{\vartheta})} &= \frac{\partial L(\boldsymbol{\vartheta})}{\partial\hat{\mu}_j(s)}\frac{\partial\hat{\mu}_j(s)}{\partial\mu_j(s;\boldsymbol{\vartheta})} + \frac{\partial L(\boldsymbol{\vartheta})}{\partial\hat{\sigma}_j(s)}\frac{\partial\hat{\sigma}_j(s)}{\partial\mu_j(s;\boldsymbol{\vartheta})}, \\ \frac{\mathrm{d}L(\boldsymbol{\vartheta})}{\mathrm{d}\sigma_j(s;\boldsymbol{\vartheta})} &= \frac{\partial L(\boldsymbol{\vartheta})}{\partial\hat{\mu}_j(s)}\frac{\partial\hat{\mu}_j(s)}{\partial\sigma_j(s;\boldsymbol{\vartheta})} + \frac{\partial L(\boldsymbol{\vartheta})}{\partial\hat{\sigma}_j(s)}\frac{\partial\hat{\sigma}_j(s)}{\partial\sigma_j(s;\boldsymbol{\vartheta})}.\end{aligned} \tag{74}$$

We note that this requires that the current network is part of the set of samples. While deep ensembles and stochastic gradient Langevin dynamics readily contain the current set of parameters in the samples, for dropout and stochastic weight averaging this is not the case. For the latter two, the current weights of the network have to be included in the samples. While the first factors can be reused from the baseline algorithm (see appendix B.2), the latter has to be computed. Using eq. (67) to compute the value function implies a factor $1/N$. For the computation of the mean $\hat{\mu}_j(s)$ and variance $\hat{\sigma}_j^2(s)$ we can distinguish several approximations:

Total If we want to preserve the moments of the mixture distribution and incorporate the total uncertainty, we need to take eq. (68). In this case, the derivative of the mean $\hat{\mu}$ and standard deviation $\hat{\sigma}$ of the approximate predictive posterior distribution $\hat{\pi}$ with respect to the mean and standard deviation of the current policy gives

$$\begin{aligned}\frac{\partial\hat{\mu}_j(s)}{\partial\mu_j(s;\boldsymbol{\vartheta})} &= \frac{1}{N}, \quad &\frac{\partial\hat{\sigma}_j(s)}{\partial\mu_j(s;\boldsymbol{\vartheta})} &= \frac{1}{N}\frac{1}{\hat{\sigma}_j(s)}\left(\mu_j(s;\boldsymbol{\vartheta}_\mu) - \hat{\mu}_j(s)\right), \\ \frac{\partial\hat{\mu}_j(s)}{\partial\sigma_j(s;\boldsymbol{\vartheta})} &= 0, \quad &\frac{\partial\hat{\sigma}_j(s)}{\partial\sigma_j(s;\boldsymbol{\vartheta})} &= \frac{1}{N}\frac{\sigma_j(s;\boldsymbol{\vartheta})}{\hat{\sigma}_j(s)}.\end{aligned} \tag{75}$$

We call this method the *Gaussian with Total Uncertainty* (GTU).

Epistemic Instead of using the total variance, we can only consider the epistemic uncertainty and compute the standard deviation of the Gaussian approximation via eq. (70). In this case, the chain rule gives

$$\begin{aligned} \frac{\partial \hat{\mu}_j(s)}{\partial \mu_j(s;\boldsymbol{\vartheta})} &= \frac{1}{N}, \quad \frac{\partial \hat{\sigma}_j(s)}{\partial \mu_j(s;\boldsymbol{\vartheta})} = \frac{1}{N}\frac{1}{\hat{\sigma}_j(s)}\left(\mu_j(s;\boldsymbol{\vartheta}_\mu) - \hat{\mu}_j(s)\right), \\ \frac{\partial \hat{\mu}_j(s)}{\partial \sigma_j(s;\boldsymbol{\vartheta})} &= 0, \quad \frac{\partial \hat{\sigma}_j(s)}{\partial \sigma_j(s;\boldsymbol{\vartheta})} = 0. \end{aligned} \tag{76}$$

We call this method the *Gaussian with Epistemic Uncertainty* (GEU).

Aleatoric We can ignore the epistemic uncertainty and compute the standard deviation of the Gaussian approximation using eq. (71). In this case, the chain rule gives

$$\begin{aligned} \frac{\partial \hat{\mu}_j(s)}{\partial \mu_j(s;\boldsymbol{\vartheta})} &= \frac{1}{N}, \quad \frac{\partial \hat{\sigma}_j(s)}{\partial \mu_j(s;\boldsymbol{\vartheta})} = 0, \\ \frac{\partial \hat{\mu}_j(s)}{\partial \sigma_j(s;\boldsymbol{\vartheta})} &= 0, \quad \frac{\partial \hat{\sigma}_j(s)}{\partial \sigma_j(s;\boldsymbol{\vartheta})} = \frac{1}{N}\frac{\sigma_j(s;\boldsymbol{\vartheta})}{\hat{\sigma}_j(s)}. \end{aligned} \tag{77}$$

We call this method the *Gaussian with Aleatoric Uncertainty* (GAlU).

Average Instead of taking inspiration from the uncertainties, we can follow the same approach as for the mean and average the standard deviations resulting from the sampled parameters. In this case, the chain rule gives

$$\begin{aligned} \frac{\partial \hat{\mu}_j(s)}{\partial \mu_j(s;\boldsymbol{\vartheta})} &= \frac{1}{N}, \quad \frac{\partial \hat{\sigma}_j(s)}{\partial \mu_j(s;\boldsymbol{\vartheta})} = 0, \\ \frac{\partial \hat{\mu}_j(s)}{\partial \sigma_j(s;\boldsymbol{\vartheta})} &= 0, \quad \frac{\partial \hat{\sigma}_j(s)}{\partial \sigma_j(s;\boldsymbol{\vartheta})} = \frac{1}{N}. \end{aligned} \tag{78}$$

We call this method the *Gaussian with Average Uncertainty* (GAvU).

We note that mixing different approximations results in an inconsistency between the policy gradient and the KL-regularization. In these cases, we observed that the experiences in the replay memory become off-policy very fast, and therefore hinder learning. Furthermore, we found that using the total uncertainty, or the aleatory uncertainty, training the variance will not be successful. We observe, that the policies will reduce their variances,

MCD	SWAG	DE	LD
O(M)	O(NM+M)	O(1)	O(M)

(a) Sampling

Naive	Gaussian
O(1)	O(N)

MCD	SWAG	DE	LD
O(1)	O(1)	O(N)	O(1)

(b) Neural Network

TABLE 4: Scaling of the different algorithms. We regard the scaling of the computational cost in terms of the number of weights M and the number of posterior samples N.

which causes numerical problems. In our case we use squareplus$(x) = 0.5(x + \sqrt{1 + x^2})$ which only slowly goes to zero, thus requiring very large values in the hidden layers. Indeed the values get so large, that we observe an overflow. When changing the activation to ReLU$(x) = 0.5(x + \|x\|)$, the variance immediately drops to zero, and thus the variance of the individual policies is not trained anymore. In order to avoid this, we suggest disabling training the standard deviation for these methods.

3.3.3.4 Computational Cost

In the following analysis, two types of computational cost are distinguished. The first is the cost to sample the weights, and the other is the cost to train the neural network. The different algorithms are discussed in the following, a summary can be found in table 4. For sampling the parameter, the DE is the cheapest. In order to sample the parameters, we solely have to sample a categorical distribution to determine the index for one of the P policies $p \sim \{1, \dots, P\}$. The weights can then readily be read from memory. It is followed by MCD and LD. While MCD requires a uniformly distributed sample for every weight, LD requires sampling a Gaussian distribution for every weight. For the vanilla version of LD, where there is no additional noise added to the samples from the SGD trajectory, this cost even diminishes. The most expensive variant is SWAG. Here first the approximate multivariate Gaussian distribution is computed using N samples from the SGD trajectory, based on which a Gaussian distribution is sampled for every weight. In order to estimate the cost of inference and training of the neural network, we have to distinguish the cost from the respective learning and sampling methods. While all methods besides ensembles require just one set of neural network parameters, the ensemble method requires training N neural networks. From the learning algorithms, the naive variant is the cheapest. A categorial distribution is sampled to determine the sample to consider, then the neural network is forwarded for this set of parameters. For the Gaussian approximation and the full model, the neural network has to be forwarded for every of the N parameter samples. When combining the ensembles with the Gaussian approximation or the full model, we would naively find a complexity of $O(N^2)$, however, this can be reduced to $O(N)$ if we train all the neural networks in the ensemble using the same minibatch. In this case, we can evaluate the posterior distribution once (causing a cost of $O(N)$) and then compute the backward pass for each of the networks individually. Moreover, we note that the ensemble training can be parallelized and therefore the time-to-solution is rescued. Even though the computational cost is higher, in science and engineering applications, the evaluation of the environment is much slower than processing the neural network. The respective computation can be easily overlapped when collecting samples in parallel [117, 118]. Furthermore, optimization techniques that parallelize the computational load in order to reduce the time-to-solution can readily be applied.

3.3.4 Results

In order to learn the policy, we employ V-RACER with ReF-ER [33], an algorithm that has shown to be successful for computationally demanding optimal control problems (scientific multi-agent reinforcement learning (SciMARL) [46, 47]). We use the standard hyperparameters reported in the original publication and parameterize the policy as $\pi(a|s;\boldsymbol{\vartheta}) = \mathcal{N}^c(\boldsymbol{\mu}(s;\boldsymbol{\vartheta}), \mathrm{diag}(\boldsymbol{\sigma}(s;\boldsymbol{\vartheta}))^2)$ with the clipped normal distribution $\mathcal{N}^c$ (see appendix B.2). It will serve as our *baseline* method. For the comparison, we will restrict our attention to a subset of the methods presented in section 3.3.3.1. We regard DropConnect with varying dropout probability p, DE with a varying number N of policies, SWAG with a diagonal approximation of the posterior based on the M most recent weights collected from the SGD trajectory, and LD without adding noise. HMC is disregarded to avoid the large cost to tune its hyperparameters.

In order to ensure a fair comparison, all evaluation results under comparison were obtained using the same set of random seeds. This includes the seed for the initialization of the neural network and the sampling of the mini-batch during training, but also the seed in the environment. For the evaluation, we restricted our attention to three tasks from the openAI gym MuJoCo environments [87, 88]: Half-Cheetah-v4, Hopper-v4, and Swimmer-v4. We evaluate these environments using the sampling methods that are applicable to already trained models, i.e. MCD, SWAG, DE, and LD without additional noise. We furthermore benchmark NM during training. The code and scripts for all experiments are available under `https://github.com/cselab/korali/tree/UQRL`.

3.3.4.1 Testing Performance and Uncertainties

In the first section of the results, the sampling methods which can be used on a pre-trained model are tested. A collection of 16 policies is trained for ten million updates. For each of the training runs, we store the past 1'024 neural network parameters. This enables testing MC, SWAG, DE, and LD. Here, we replace the latest policy with the mixture distribution eq. (66) using a different number of samples out of the 1'024 stored hyperparameters and from the 16 trained policies. For each of the sampling methods we subsequently perform 32 runs in order to obtain the testing statistics. In

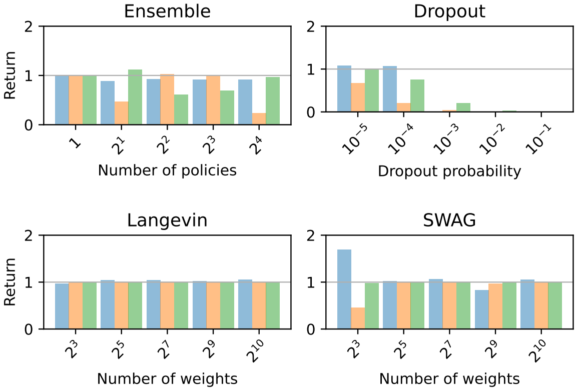


FIGURE 9: Normalized Testing returns for the different sampling methods. The blue bars (🟦) correspond to HalfCheetah-v4, the orange bars (🟧) to Hopper-v4, and the green bars (🟩) to Swimmer-v4.

order to compare the methods on the different environments, we define the normalized return as the return using the sampling method R divided by the return of the baseline method R_0. The results are presented in fig. 9. We observe, that the only method that can completely retain the performance of the original method over the range of samples tested is LD. For MCD, we observe a strong deterioration of the returns with increasing dropout probability. While SWAG performs almost en-par with the baseline for $N \geq 2^5$

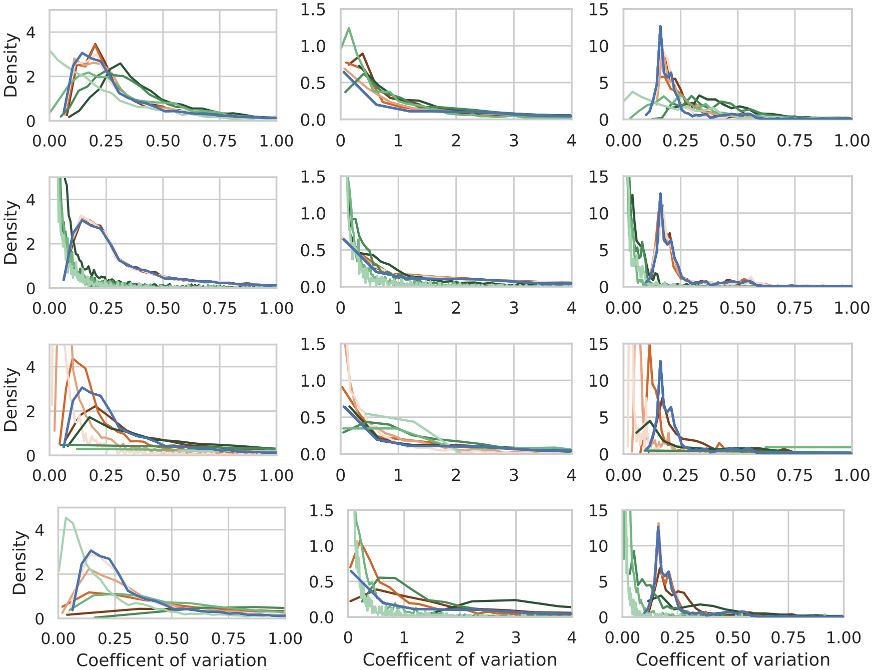


FIGURE 10: Coefficient of variation for the uncertainty estimates. The plots are grouped by the environment HalfCheetah-v4, Hopper-v4, and Swimer-v4 in columns and sampling methods DE, LD, SWAG, MCD in the rows. The blue line (—) shows the uncertainties $\sigma(s; \vartheta)$ learned for the baseline policy, the orange lines (—) show the aleatoric uncertainty (eq. (71)), and the green lines (—) show the epistemic uncertainty (eq. (70)). The four shades (from dark to light; from more to fewer samples/disturbance) correspond to different numbers of samples ($N = 1024, 512, 128, 32$ for SWAG and LD, $p = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$ for MCD, and $N = 16, 8, 4, 2$ for DE).

samples, we observe inconsistent behavior when using $N = 2^3$ samples. This indicates that $N \geq 2^5$ samples are needed in order to have a converged local approximation of the posterior distribution. For the DE, we do not find a consistent result: While the HalfCheetah environment has slightly below-baseline performance, the Hopper and Swimmer environments show variation in the attainable returns.

In order to better understand these variations, we consider the uncertainties resulting with the different sampling methods for all environments under comparison. In order to compare the uncertainties defined in section 3.3.3 we regard the coefficient of variation

$$c = \left| \frac{\text{uncertainty}}{\text{mean}} \right| , \tag{79}$$

which allows comparing the uncertainties of all actions independent of their magnitude. This allows comparison between environments and plotting c for all components of the actions. The resulting densities on one testing trajectory per environment is shown in fig. 10. We observe a notable differences for the different sampling methods. While for the DE, the aleatoric uncertainty of HalfCheetah and Swimmer is in close correspondence to the learned uncertainty $\sigma(s; \boldsymbol{\vartheta})$, it grows with increasing number of policies in the ensemble for the Hopper environments. While for HalfCheetah and Swimmer, the epistemic uncertainty unveils that the multiple trained policies are different local optima, which is reflected by the growing uncertainty. On the Hopper environment this is not evident. For LD the shift of the epistemic uncertainty is less severe, and also the aleatoric uncertainty matches the learned uncertainty very well. This is expected and shows the important property of of LD to create samples steam from the same local optima. The picture for SWAG is intriguing, while for low number of samples used to approximate the posterior distribution, the aleatoric uncertainty is peaked around zero, while the epistemic uncertainty is flat. when adding more samples this picture changes and the epistemic uncertainty becomes more peaked, while the aleatoric becomes flat. For MCD the failure becomes evident and the aleatoric uncertainty captures the learned uncertainty from the original model only for very low dropout probabilities. The epistemic uncertainty is unpeaked and indicated the main problem, namely that the actions become close-to random.

3.3.4.2 Training Performance

After analyzing the testing performance we compare the training performance for the samplers SWAG, DE, and LD. This excludes MCD which has been shown to introduce too much uncertainty in order to work during testing. The course of training for the first 2.5 million updates is shown in fig. 27. As can be seen, DE and LD reach the same level of maximal reward as the baseline algorithm, indicating that the introduced sampling methods do not disturb the training process. For SWAG this picture is not confirmed and increasing the number of samples hinders training. In the HalfCheetah environment this trend is clearly visible. For Hopper it is less pronounced and the runs with 32 and 128 samples still reach good rewards. Here, only the runs with 512 and 1024 samples are strongly lagging behind. Interestingly, this observation

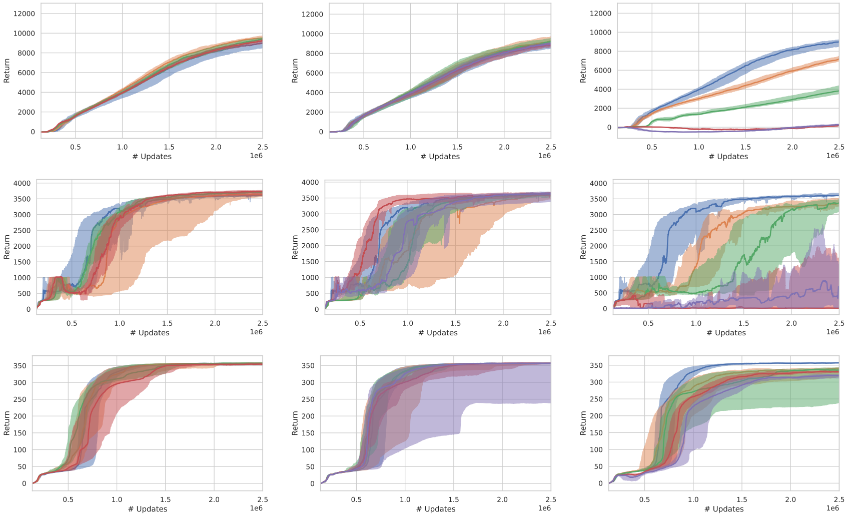


FIGURE 11: Training results for the different sampling methods. We perform ten training runs and plot the median and 60% confidence interval for each method with shading in the columns - top row HalfCheetah-v4, center row Hopper-v4, and bottom row Swimmer-v4. In each plot, the baseline result is displayed as blue line (—). The left column shows the results for DE, with two (orange —), four (green —), eight (red —), and sixteen (purple —) policies. The center and right columns show the results for Langevin and SWAG, with different number of samples: 32 (orange —), 128 (green —), 512 (red —), and 1024 (purple —).

is not confirmed for the Swimmer environments, although a delay can be observed the reached maximal reward after 2.5M updates is still at a decent level. The fact that SWAG seems to be problematic during training can be understood from different perspectives: In essence SWAG approximates the posterior locally by averaging the past N samples along the SGD trajectory. This implies that updates that reinforce the observed behavior are delayed and therefore the training process slowed down. Another point of view arises when considering the handling of off-policy experiences with ReF-ER: The averaging also implies a larger off-policy ratio, that hinders the training. These observations are invalid for the other methods under comparison. In both cases the sampling of the action takes place with policies that are currently being trained, and thus the off-policy method can readily handle the samples. The level of disturbance introduced by these methods can only be observed when looking at the confidence intervals. While for Half-Cheetah, the variance between the runs is almost the same, we can see clear differences for LD and DE in the Hopper and Swimmer environments. Also here the trend is towards larger uncertainties when increasing the number of samples. This is also in accordance with the expectations from our analysis of the uncertainties in the previous section.

4

BAYESIAN OPTIMAL EXPERIMENTAL DESIGN

> *Intellect distinguishes between the possible and the impossible; reason distinguishes between sensible and the senseless. Even the possible can be senseless.*
>
> — Max Born

Fish schooling implies an awareness of the swimmers for their companions. In flow mediated environments, in addition to visual cues, pressure and shear sensors on the fish body are critical for providing quantitative information that assists the tracking the proximity to other fish. Here we examine the distribution of sensors on the surface of an artificial swimmer so that it can optimally identify a leading group of swimmers. We employ Bayesian experimental design coupled with numerical simulations of the two-dimensional Navier Stokes equations for multiple self-propelled swimmers. The follower tracks the school using information from its own surface pressure and shear stress. We demonstrate that the optimal sensor distribution of the follower is qualitatively similar to the distribution of neuromasts on fish. Our results show that it is possible to identify accurately the center of mass and the number of the leading swimmers using surface only information.

ACKNOWLEDGMENTS This chapter is based on [119, 120]. For sake of brevity, we moved study for the optimal allocation of limited test resources to the appendix appendix A.2. The computational resources were provided by the Swiss National Supercomputing Centre (CSCS) under project s929.

4.1 Information Gain and Expected Utility

We define the information gain as the distance between the prior belief on the quantities of interest and the posterior belief after obtaining the measurements. Here, we choose as measure of the distance the Kullback–Leibler divergence between the prior and the posterior distribution.

4.1.1 Formalism

A measurement $\boldsymbol{y} \in \mathbb{R}^n$ can be expressed as,

$$\boldsymbol{y} = \boldsymbol{F}(\boldsymbol{s}, \boldsymbol{\vartheta}; \boldsymbol{\vartheta}') + \varepsilon \, . \tag{80}$$

Here, $\boldsymbol{F}(\boldsymbol{\vartheta}; \boldsymbol{s})$ denotes the prediction of a model of the system at for measurement parameters $\boldsymbol{s}$ (location, time, or any other parametrisation of the experimental configuration) and for some parameter of interest $\boldsymbol{\vartheta}$. The model can further depend on some nuissance parameters $\boldsymbol{\vartheta}'$, that are not of interest. We model the error term by a multivariate Gaussian distribution $\varepsilon \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}(\boldsymbol{s}))$ with zero mean and covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{s}) \in \mathbb{R}^{n \times n}$. In this case the likelihood of a measurement is given by,

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{\vartheta}, \boldsymbol{s}, \boldsymbol{\vartheta}') = & \frac{1}{\sqrt{(2\pi)^n \det(\boldsymbol{\Sigma}(\boldsymbol{s}))}} \\ & \exp\left(-\frac{1}{2}\left(\boldsymbol{y} - \boldsymbol{F}(\boldsymbol{s}, \boldsymbol{\vartheta}; \boldsymbol{\vartheta}')\right)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{s}) \left(\boldsymbol{y} - \boldsymbol{F}(\boldsymbol{s}, \boldsymbol{\vartheta}; \boldsymbol{\vartheta}')\right)\right) . \end{aligned} \tag{81}$$

The covariance matrix depends on the measurement parameters $\boldsymbol{s}$, which allows modeling correlations of measurements taken simultaneously. Introducing such correlations is importance to avoid clustering of sensors [121]. We wish to identify the locations $\boldsymbol{s}$ yielding the largest information gain about the unknown parameter $\boldsymbol{\vartheta}$ of the disturbance. A measure for information gain is defined through the Kullback–Leibler (KL) divergence between the prior belief of the parameter values and the posterior belief, i.e., after measuring the environment. The prior and posterior beliefs are represented through the density functions $p(\boldsymbol{\vartheta})$ and $p(\boldsymbol{\vartheta}|\boldsymbol{y}, \boldsymbol{s}, \boldsymbol{\vartheta}')$, respectively. The two densities are connected through Bayes' theorem,

$$p(\boldsymbol{\vartheta}|\boldsymbol{y}, \boldsymbol{s}, \boldsymbol{\vartheta}') = \frac{p(\boldsymbol{y}|\boldsymbol{\vartheta}, \boldsymbol{s}, \boldsymbol{\vartheta}')\, p(\boldsymbol{\vartheta}|\boldsymbol{s}, \boldsymbol{\vartheta}')}{p(\boldsymbol{y}|\boldsymbol{s}, \boldsymbol{\vartheta}')} \, , \tag{82}$$

where $p(\boldsymbol{y}|\boldsymbol{\vartheta}, \boldsymbol{s}, \boldsymbol{\vartheta}')$ is the likelihood function defined in Equation (81) and $p(\boldsymbol{y}|\boldsymbol{s}, \boldsymbol{\vartheta}')$ the marginal likelihood. We assume that the prior belief on the parameters $\boldsymbol{\vartheta}$ does not depend on the sensor locations and nuissance parameters, $p(\boldsymbol{\vartheta}|\boldsymbol{s},\boldsymbol{\vartheta}') \equiv p(\boldsymbol{\vartheta})$. The utility function is defined as [122],

$$\begin{aligned} u(\boldsymbol{s},\boldsymbol{y},\boldsymbol{\vartheta}') &:= D_{\mathrm{KL}}(p(\boldsymbol{\vartheta}|\boldsymbol{y},\boldsymbol{s},\boldsymbol{\vartheta}')||p(\boldsymbol{\vartheta})) \\ &= \int_{\mathcal{T}} \ln \frac{p(\boldsymbol{\vartheta}|\boldsymbol{y},\boldsymbol{s},\boldsymbol{\vartheta}')}{p(\boldsymbol{\vartheta})}\, p(\boldsymbol{\vartheta}|\boldsymbol{y},\boldsymbol{s},\boldsymbol{\vartheta}')\mathrm{d}\boldsymbol{\vartheta}\,. \end{aligned} \tag{83}$$

Here the integration is performed over the space of parameter of interest $\mathcal{T}$. The expected utility is defined as the expectation over the measurements and nuisance parameters

$$\begin{aligned} U(\boldsymbol{s}) &:= \mathbb{E}_{\substack{\boldsymbol{y}\sim p(\cdot|\boldsymbol{s},\boldsymbol{\vartheta}') \\ \boldsymbol{\vartheta}'\sim p(\cdot)}}\left[u(\boldsymbol{s},\boldsymbol{y},\boldsymbol{\vartheta}')\right] \\ &= \int_{\mathcal{Y}}\int_{\mathcal{P}} u(\boldsymbol{s},\boldsymbol{y},\boldsymbol{\vartheta}')\; p(\boldsymbol{y}|\boldsymbol{s},\boldsymbol{\vartheta}')p(\boldsymbol{\vartheta}')\mathrm{d}\boldsymbol{y}\,\mathrm{d}\boldsymbol{\vartheta}'\,, \end{aligned} \tag{84}$$

where $\mathcal{Y}$ is the domain of all possible measurements and $\mathcal{P}$ the space of nuisance parameters . Using Equation (82) and (83) the expected utility can be expressed as,

$$U(\boldsymbol{s}) = \int_{\mathcal{Y}}\int_{\mathcal{P}}\int_{\mathcal{T}} \ln\frac{p(\boldsymbol{y}|\boldsymbol{\vartheta},\boldsymbol{s},\boldsymbol{\vartheta}')}{p(\boldsymbol{y}|\boldsymbol{s},\boldsymbol{\vartheta}')}\; p(\boldsymbol{y}|\boldsymbol{\vartheta},\boldsymbol{s},\boldsymbol{\vartheta}')\; p(\boldsymbol{\vartheta})p(\boldsymbol{\vartheta}')\;\mathrm{d}\boldsymbol{y}\;\mathrm{d}\boldsymbol{\vartheta}\;\mathrm{d}\boldsymbol{\vartheta}'\,. \tag{85}$$

4.1.2 Estimation

The expected utility presented in eq. (85) admits no closed-form solution. Therefore we use numerical integration to estimate the expected utility for each sensor location. For sake of brevity we neglect the nuisance parameters in what follows.

CONTINOUS PARAMETER SPACES When $\boldsymbol{\vartheta}$ is a continuous random variable, the estimator for the expected utility can be obtained by approximating the two integrals by numerical integration. We note that for low dimensional integrals, using quadrature might be benefitial. More generally we use Monte Carlo integration with $N_{\boldsymbol{\vartheta}}$ samples from $p(\boldsymbol{\vartheta})$ and $N_{\boldsymbol{y}}$ samples from $p(\boldsymbol{y}|\boldsymbol{\vartheta},\boldsymbol{s})$ [123]. In this case the resulting estimator is given by,

$$\begin{aligned} U(\boldsymbol{s}) \approx \hat{U}(\boldsymbol{s}) = \frac{1}{N_{\boldsymbol{\vartheta}}N_{\boldsymbol{y}}}\sum_{j=1}^{N_{\boldsymbol{y}}}\sum_{i=1}^{N_{\boldsymbol{\vartheta}}}\Bigg[&\ln p(\boldsymbol{y}^{(i,j)}|\boldsymbol{\vartheta}^{(i)},\boldsymbol{s}) \\ &- \ln\left(\frac{1}{N_{\boldsymbol{\vartheta}}}\sum_{k=1}^{N_{\boldsymbol{\vartheta}}} p(\boldsymbol{y}^{(i,j)}|\boldsymbol{\vartheta}^{(k)},\boldsymbol{s})\right)\Bigg]\,. \end{aligned} \tag{86}$$

The samples are denoted as $\boldsymbol{\vartheta}^{(i)} \sim p_{\boldsymbol{\vartheta}}(\cdot)$ for $i = 1, \ldots, N_{\boldsymbol{\vartheta}}$ and $\boldsymbol{y}^{(i,j)} \sim p_{\boldsymbol{y}}(\cdot|\boldsymbol{\vartheta}^{(i)}, \boldsymbol{s})$ for $j = 1, \ldots, N_{\boldsymbol{y}}$.

DISCRETE PARAMETER SPACES When $\boldsymbol{\vartheta}$ is a discrete random variable taking values in the set $\{\boldsymbol{\vartheta}_1, \ldots, \boldsymbol{\vartheta}_{N_{\boldsymbol{\vartheta}}}\}$ the expected utility in Equation (85) is given by,

$$U(\boldsymbol{s}) = \sum_{i=1}^{N_{\boldsymbol{\vartheta}}} p(\boldsymbol{\vartheta}_i) \int_{\mathcal{Y}} \ln \frac{p(\boldsymbol{y}|\boldsymbol{\vartheta}_i, \boldsymbol{s})}{p(\boldsymbol{y}|\boldsymbol{s})} \, p(\boldsymbol{y}|\boldsymbol{\vartheta}_i, \boldsymbol{s}) \, \mathrm{d}\boldsymbol{y} \,. \tag{87}$$

An estimator of the given utility can be obtained by Monte Carlo integration using $N_{\boldsymbol{y}}$ samples from the likelihood distribution $p(\boldsymbol{y}|\boldsymbol{\vartheta}_i, \boldsymbol{s})$. The estimator is given by

$$\begin{aligned} U(\boldsymbol{s}) \approx \hat{U}(\boldsymbol{s}) = \frac{1}{N_{\boldsymbol{y}}} \sum_{j=1}^{N_{\boldsymbol{y}}} \sum_{i=1}^{N_{\boldsymbol{\vartheta}}} p(\boldsymbol{\vartheta}_i) \Bigg[& \ln p(\boldsymbol{y}^{(i,j)}|\boldsymbol{\vartheta}_i, \boldsymbol{s}) \\ & - \ln \left(\sum_{k=1}^{N_{\boldsymbol{\vartheta}}} p(\boldsymbol{\vartheta}_k) p(\boldsymbol{y}^{(i,j)}|\boldsymbol{\vartheta}_k, \boldsymbol{s}) \right) \Bigg] \,. \end{aligned} \tag{88}$$

where $\boldsymbol{y}^{(i,j)} \sim p_{\boldsymbol{y}}(\cdot|\boldsymbol{\vartheta}^{(i)}, \boldsymbol{s})$ for $j = 1, \ldots, N_{\boldsymbol{y}}$.

COMPUTATIONAL COST We remark that in cases where the evaluation of the model $\boldsymbol{F}(\boldsymbol{s}, \boldsymbol{\vartheta})$ is computationally expensive, the cost is mainly determined by the number of simulations $N_{\boldsymbol{\vartheta}}$. In the other extreme when the measurements are in very high-dimensional spaces, the cost is dominated by the computational burden to compute the $N_{\boldsymbol{y}}$ samples following the measurement error model in Equation (81).

4.1.3 Optimization

In order to determine the optimal measurement parameters we maximize the utility estimator $\hat{U}(\boldsymbol{s})$ described in Equation (86). It has been observed that the expected utility in the case of multiple equivalent sensors s often exhibit many local optima [121, 124]. Heuristic approaches, such as the sequential sensor placement algorithm described by [125], have been demonstrated to be effective to elevate this problem. Following [125], we perform the optimization iteratively, placing one sensor after the other. This procedure is formalized as

$$s_i^\star = \arg\max_{s} \ \hat{U}(\boldsymbol{s}) \qquad \text{where} \qquad \boldsymbol{s} = (s_1^\star, \ldots, s_{i-1}^\star, s) \,. \tag{89}$$

Besides the mentioned advantages, sequential placement allows to quantify the importance of each sensor placed and provides further insight into the resulting distribution of sensors.

4.2 Optimal Sensor Placement for Schooling Swimmers

Fish navigate in their habitats by processing visual and hydrodynamic cues from their aqueous environment. Such cues may serve to provide awareness of their neighbors as fish adapt their swimming gaits in groups. Early studies have shown that vision is a critical factor for fish schooling [126]. However, more recent studies have shown that even blinded fish can keep station in a school [127]. Such capabilities are of particular importance in flow environments where vision capabilities may be limited [128].The flow environment is replete with mechanical disturbances (pressure, shear) that can convey information about the sources that generated them. Fish swimming in groups have been found to process such hydrodynamic cues and balance them with social interactions [129, 130]. In order to detect mechanical disturbances in terms of surface pressure and shear stresses fish have developed a specialized organ, the lateral line system. The mechanoreceptors in the lateral line – allowing the sensing of the disturbances in water – are called neuromasts. A number of studies and experiments have shown that the functioning of the lateral line is crucial for several tasks [131, 132]. Experiments with trout in the vicinity of objects have shown its importance for Kármán gaiting and bow wake swimming as well as energy efficient station keeping [133, 134]. Using the information contained in the flow, the cylinder diameter, the flow velocity, and the position relative to the generated Kármán vortex street were quantified [135, 136]. Using blind cave fish, several studies have shown the importance of the lateral line to detect the location and the shape of surrounding objects and avoid obstacles [137–140]. In another study, the feeding behavior of blinded mottled sculpin was tested and it was found that they use their lateral line system to detect prey [141]. It was also found that blind fish manage to keep their position in schools and lose this ability with a disabled lateral line organ [142]. The importance of the lateral line was also shown for enhanced communication [143], the selection of habitats [144] and rheotaxis [145].

In this work, we mimic the mechanosensory receptors, more specifically the sub-surface ‘canal’ neuromasts and superficial neuromasts [146, 147]. The neuromast on the fish skin are used to detect shear stresses, where the ones residing in the lateral line canals are used to detect pressure gradients [148–152]. Due to the filtering nature of the canals, the detection of small hydrodynamic stimuli against background noise is improved for the subsurface neuromasts [153].

The effectiveness and versatility of the lateral line organ has yielded several bio-inspired artificial flow sensors [154–158]. Arranging these sensors in arrays on artificial swimmers has attracted attention to transform underwater sensing [128, 159–164]. Here, leveraging the intelligent distributed sensing inspired by the lateral line showed to be effective in robots moving in aquatic environments [165–170].

In order to better use and understand the capabilities of the artificial sensors several studies regarding the information content in the flow and optimal harvesting of this information were performed: The prevalence of information on the position of a vibrating source was shown to be linearly coded in the pressure gradients measured by the subsurface neuromasts [171]. Furthermore, it was shown that the variance of the pressure gradient is correlated with the presence of lateral line canals [172]. In [173], fish robots equipped with distributed pressure sensors for flow sensing were combined with Bayesian filtering in order to estimate the flow speed, the angle of attack, and the foil camber. Other studies have focused on dipole sources in order to develop methods that extract information and optimize the parameters of the sensing devices [174, 175]. In a recent study artificial neural networks were employed to classify the environment using flow-only information [176–179]. In order to find effective sensor positions weight analysis algorithms were employed [180].

Following an earlier work for detection of flow disturbances generated from single obstacles [181], we examine the optimality of the spatial distribution of sensors in a self-propelled swimmer that infers the size and the relative position of the leading school. We combine numerical simulations of the two-dimensional Navier–Stokes equation and Bayesian optimal sensor placement to examine the extraction of flow information by pressure gradients and shear stresses and the optimal positioning of associated sensors. The work demonstrates the capability of sensing a rather complex system using information of shear and pressure. Such information is available both, to biological organisms and artificial swimmers. We remark that the work does not aim to reproduce biological systems but rather reveal algorithms that may be applicable to robotic systems. At the same time, we find that the identified optimal sensor locations for the two-dimensional artificial swimmers have similarities to biological systems indicating common governing physical mechanisms for the hydrodynamics of natural and artificial swimmers.

4.2.1 Schooling Formations

The tail-beating motion that propels forward a single swimmer generates in its wake a sequence of vortices. The momentum contained in the flow field induces forces which swimmers in schooling formation must overcome to maintain their positions in the group [182]. In this study, we maintain the schooling formation for multiple swimmers by employing closed-loop parametric controllers. The tail beating frequency $T_{p,i}$ of each swimmer i is increased or decreased if it lags behind or surpasses respectively a desired position Δx_i in the direction of the school's motion,

$$T_{p,i} = T_p(1 - \Delta x_i)\,. \tag{90}$$

The mean school trajectory is adjusted by imposing an additional uniform curvature $k_{C,i}$ along each swimmer's midline in order to minimize its lateral deviation Δy_i and its angular deflection $\Delta\theta_i$,

$$k_{C,i} = [\Delta y_i, \langle\Delta\theta_i\rangle]_- + [\langle\Delta y_i\rangle, \Delta\theta_i]_- + [\langle\Delta y_i\rangle, \langle\Delta\theta_i\rangle]_-\,. \tag{91}$$

Here, $\langle\cdot\rangle$ defines an exponential moving average with weight $\delta t/T_p$, which approximates the integral term found in PI controllers and

$$[a, b]_- = \begin{cases} |a|b\,, & \text{if } ab < 0\,, \\ 0\,, & \text{otherwise}\,. \end{cases} \tag{92}$$

The formulation in Equation (91) indicates that if both the lateral displacement and the angular deviation are positive (or both negative) the swimmer will gradually revert to its position in the formation. Conversely, if Δy_i and $\Delta\theta_i$ have different signs the displacement has to be corrected by adding (or subtracting) curvature to the swimmer's midline.

4.2.2 Flow Sensors

We distinguish two types of sensors on the swimmer body. The superficial neuromasts detect flow stresses and the subcanal neuromasts pressure gradients [156, 183, 184]. From the numerical solution of the 2D Navier–Stokes equation we obtain the flow velocity $\mathbf{u} = (u, v)$ and the pressure p at every point of the computational grid. The surface values of these quantities are obtained through a bi-linear interpolation from the nearest grid points. We perform offline analysis by recording the interpolated pressure p and flow velocity **u** in the vicinity of the body. We remark that we have neglected points near the end of the body to reduce the influence of large flow gradients that are generated by the motion and sharp geometry of the tail. The shear stresses are computed on the body surface using the local tangential velocity in the two nearest grid points. Moreover, we compute pressure gradients along the surface by first smoothing these pressure along the surface using splines implemented in SciPy [185, 186].

In the present experiment setup, we consider a group of swimmers followed by a single swimmer. The follower needs to identify (i) the relative location **r** of the center of mass and (ii) the population n_f of the leading group. We denote with $\boldsymbol{\vartheta} = \mathbf{r}$ or $\boldsymbol{\vartheta} = n_f$ these unknown quantities and allow the follower to update its prior belief $p(\boldsymbol{\vartheta})$ about the leading group of swimmers by collecting measurements on its sensors. These sensors are distributed symmetrically on both sides of the swimmer and are represented by a single point on its mid-line. We denote the k-th measurement location at the upper and the lower part with $\boldsymbol{x}_1(s_k)$ and $\boldsymbol{x}_2(s_k)$, respectively. The corresponding measurements are denoted by y_k^1 and y_k^2, respectively (see Figure 12 for a sketch of the setup).

4.2.3 Measurement Model

The measurement model is specified in accordance with eq. (81) with a covariance matrix is given by,

$$\Sigma_{ij}(\boldsymbol{s}) = \begin{cases} \sigma^2 \exp\left(-\frac{\|\boldsymbol{x}_1(s_i)-\boldsymbol{x}_1(s_j)\|}{\ell}\right), & \text{if } 1 \le i,j \le n, \\ \sigma^2 \exp\left(-\frac{\|\boldsymbol{x}_2(s_{i-n})-\boldsymbol{x}_2(s_{j-n})\|}{\ell}\right), & \text{if } n < i,j \le 2n, \\ 0, & \text{otherwise}, \end{cases} \tag{93}$$

where $\ell > 0$ is the correlation length and σ is the correlation strength. For all the cases described in this work, the correlation length is set to one tenth of the swimmer length $\ell = 0.1L$. The correlation strength is set to be two times the average of the signals coming from the simulations,

$$\sigma = \frac{1}{n N_{\vartheta}} \sum_{j=1}^{2n} \sum_{i=1}^{N_{\vartheta}} |\boldsymbol{F}(\boldsymbol{\vartheta}^{(i)}; s_j)|, \tag{94}$$

where $\boldsymbol{\vartheta}^{(i)}$ are samples from the distribution $p(\boldsymbol{\vartheta})$. We remark that the covariance matrix must be symmetric and positive definite. To ensure positive

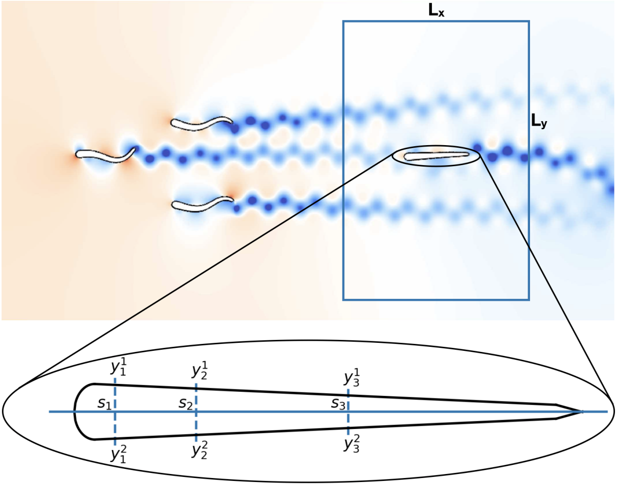


FIGURE 12: Simulation setup used for determining the optimal sensor distribution on a fish-like body. The follower is initially located inside the rectangular area. The number of swimmers in the leading group is varied between one and eight. The sensor-placement algorithm attempts to find the arrangement of sensors $\boldsymbol{s}$ that allows the follower to determine with lowest uncertainty the relative position $\mathbf{r}$ and the number of swimmers n_f in the leading group of swimmers. For each sensor s_k the swimmer collects measurements y_k^1 and y_k^2 at locations $\boldsymbol{x}_1(s_k)$ and $\boldsymbol{x}_2(s_k)$ on the skin, respectively.

definiteness we have to take special care to the case where we pick a sensor location twice. Notice that when $s_i = s_j$ for $i \neq j$, a non-diagonal entry equals the diagonal entry and positive definiteness is violated. We handle this case by setting the argument of the exponential in Equation (93) to 10^{-7} when $s_i = s_j$. This form of the correlation error reduces the utility when sensors are placed too close together and prevents excessive clustering of the sensors [121, 187].

4.2.4 Inferring the Number of Swimmers

Let $\boldsymbol{\varphi}$ be the random variable representing one of the group configurations. Each group configuration is associated with a unique number $\boldsymbol{\varphi}_{i,\ell}$ for $\ell = 1, \dots, n_i$, where n_i is the total number of configurations containing i swimmers. With this notation, $\boldsymbol{\varphi}$ takes values in the set $\{\boldsymbol{\varphi}_{i,\ell} \,|\, i = 1, \dots, 8\,, \ell = 1, \dots, n_i\}$. For examples of different configurations see Appendix A.1.1. Using the fact that for $i = 1, \dots, N_{\boldsymbol{\vartheta}}$,

$$p(\boldsymbol{y}, \boldsymbol{\varphi} = \boldsymbol{\varphi}_{k,\ell} | \boldsymbol{\vartheta} = \boldsymbol{\vartheta}_i, \boldsymbol{s}) = 0, \quad \text{for } k \neq i\,,$$

and

$$p(\boldsymbol{y} | \boldsymbol{\vartheta} = \boldsymbol{\vartheta}_i, \boldsymbol{\varphi} = \boldsymbol{\varphi}_{i,\ell}, \boldsymbol{s}) = p(\boldsymbol{y} | \boldsymbol{\varphi} = \boldsymbol{\varphi}_{i,\ell}, \boldsymbol{s}), \quad \text{for } \ell = 1, \dots, n_i\,,$$

and the assumption

$$p(\boldsymbol{\varphi} = \boldsymbol{\varphi}_{i,\ell} | \boldsymbol{\vartheta} = \boldsymbol{\vartheta}_i, \boldsymbol{s}) = \frac{1}{n_i}, \quad \text{for } \ell = 1, \dots, n_i\,,$$

the likelihood function can be written as,

$$\begin{aligned} p(\boldsymbol{y} | \boldsymbol{\vartheta} = \boldsymbol{\vartheta}_i, \boldsymbol{s}) &= \sum_{k=1}^{N_{\boldsymbol{\vartheta}}} \sum_{\ell=1}^{n_i} p(\boldsymbol{y}, \boldsymbol{\varphi} = \boldsymbol{\varphi}_{k,\ell} | \boldsymbol{\vartheta} = \boldsymbol{\vartheta}_i, \boldsymbol{s}) \\ &= \sum_{\ell=1}^{n_i} p(\boldsymbol{y}, \boldsymbol{\varphi} = \boldsymbol{\varphi}_{i,\ell} | \boldsymbol{\vartheta} = \boldsymbol{\vartheta}_i, \boldsymbol{s}) \\ &= \sum_{\ell=1}^{n_i} p(\boldsymbol{y} | \boldsymbol{\vartheta} = \boldsymbol{\vartheta}_i, \boldsymbol{\varphi} = \boldsymbol{\varphi}_{i,\ell}, \boldsymbol{s})\, p(\boldsymbol{\varphi} = \boldsymbol{\varphi}_{i,\ell} | \boldsymbol{\vartheta} = \boldsymbol{\vartheta}_i) \\ &= \frac{1}{n_i} \sum_{\ell=1}^{n_i} p(\boldsymbol{y} | \boldsymbol{\varphi} = \boldsymbol{\varphi}_{i,\ell}, \boldsymbol{s})\,. \end{aligned} \tag{95}$$

Notice that the likelihood function for fixed $\boldsymbol{\vartheta}_i$, is a mixture of Gaussian distributions with equal weights and that $p(\boldsymbol{y} | \boldsymbol{\varphi} = \boldsymbol{\varphi}_{i,\ell}, \boldsymbol{s}) = \mathcal{N}(\boldsymbol{y} | \boldsymbol{F}(\boldsymbol{\varphi}_{i,\ell}; \boldsymbol{s}), \Sigma(\boldsymbol{s}))$.

In order to draw a sample from the likelihood, first we draw an integer ℓ^* with equal probability from 1 to n_i and then draw $\boldsymbol{y} \sim p_{\boldsymbol{y}}(\cdot|\boldsymbol{\varphi}_{i,\ell^*}, \boldsymbol{s})$. The final form of the estimator is given by

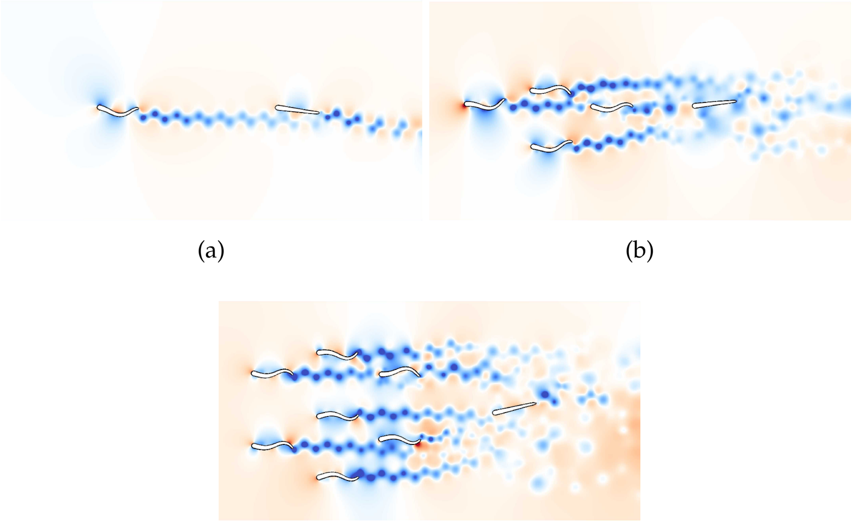
$$\hat{U}(\boldsymbol{s}) = \frac{1}{N_{\boldsymbol{y}}} \sum_{j=1}^{N_{\boldsymbol{y}}} \sum_{i=1}^{N_{\vartheta}} p(\boldsymbol{\vartheta}_i) \Bigg[\ln \frac{1}{n_i} \sum_{\ell=1}^{n_i} p(\boldsymbol{y}^{(i,j)}|\boldsymbol{\varphi}_{i,\ell}, \boldsymbol{s}) \\ - \ln \left(\frac{1}{n_i} \sum_{k=1}^{N_{\vartheta}} p(\boldsymbol{\vartheta}^{(k)}) \sum_{\ell=1}^{n_i} p(\boldsymbol{y}^{(i,j)}|\boldsymbol{\varphi}_{i,\ell}, \boldsymbol{s}) \right) \Bigg] . \tag{96}$$

4.2.5 Results

We examine the optimal arrangement of pressure gradient and shear stress sensors on the surface of a swimmer trailing a school of self-propelled swimmers. We consider two sensing objectives: (a) the size of the leading school and (b) the relative position of the school. The simulations correspond to a Reynolds number $\mathrm{Re} = \frac{L^2}{\nu} = 2000$. In all experiments, we use 4096 points to discretize the horizontal direction $x \in [0, 1]$ and all artificial swimmers have a length of $L = 0.1$.

For the “size of the leading school” experiment, where the aim is to determine the size of the group, we chose the school-sizes to be $\boldsymbol{\vartheta}_i = 1, \dots, 8$. First we consider one configuration per group-size. In this case inferring the configuration is equivalent to inferring the number of swimmer in the group. To increase the difficulty we consider n_i different initial configurations. In each configuration we assign a number $\boldsymbol{\varphi}_{i,\ell}$ for $i = 1, \dots, 8$ and $\ell = 1, \dots, n_i$. In total, we consider $N_{tot} = \sum_i n_i = 61$ distinct configurations each having the same prior probability $1/N_{tot}$. In Appendix A.1.1 we present the initial condition for all configurations. The center of mass of the school is located at $x = 0.3$ and in the y-axis in the middle of the vertical extent of the domain. We use a controller to fix the distance between x and y coordinates of two swimmers to $\Delta x = \Delta y = 0.15$, see section 4.2.1.

For the “relative position” experiment, where the aim is to determine the relative location of the follower to the center of mass of the leading group, we consider three independent experiments with one, four and seven leading swimmers. Snapshots of the pressure field for these simulations are presented in Figure 13. The prior probability for the position of the group is uniform in the domain $[0.6, 0.8] \times [0.1, 0.4]$. The support of the prior probability is discretized with 21×31 gridpoints. Since the experiments are



(a)

(b)

(c)

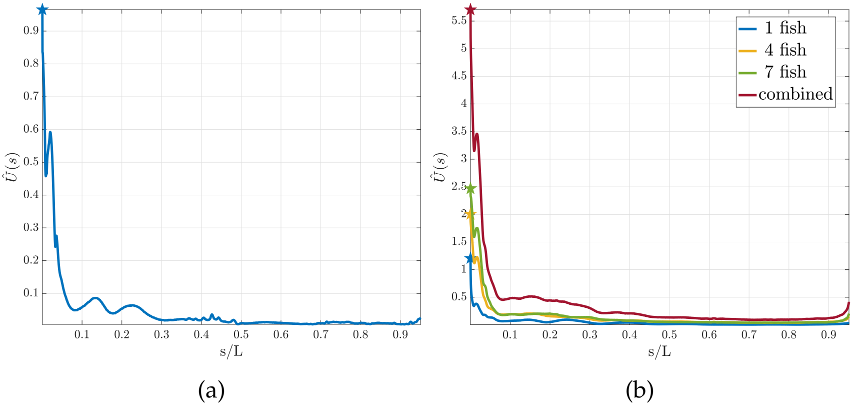
FIGURE 13: Snapshots of the pressure field in the environment of the follower swimmer generated by one (Figure 13a), four (Figure 13b) and seven (Figure 13c) schooling swimmers. The snapshots are taken at the moment the measurement was performed for one particular location of the follower in the prior region. High pressure is shown in red and low pressure in blue.

independent, the total expected utility function for the three cases is the sum of the expected utility of each experiment [181].

For both experiments we record the pressure gradient and shear stress on the surface of the swimmer using the methods discussed in section 4.2.2. The motion of the swimmer introduces disturbances on its own surface. In order to distinguish the self-induced from the environment disturbances we freeze the movement of the following swimmer and set its curvature to zero. The freezing time is selected by evolving the simulation until the wakes of the leading group are sufficiently mixed and passed the following swimmer. We found that this is the case for $T = 22$. The transition from swimming to coasting motion takes place during the time interval $[T, T+1]$. Finally, we record the pressure gradient and the shear stress at time $T+2$. The resulting sensor-signal associated to the midline coordinates $\boldsymbol{s}$ for a given configuration $\boldsymbol{\vartheta}$ is denoted $\boldsymbol{F}(\boldsymbol{\vartheta}; \boldsymbol{s})$, see Equation (81).

4.2.5.1 *Expected Utility for the First Sensor*

In this section we discuss the optimal location of a single pressure gradient sensor using the estimators in Equations (86) and (96). Recall that we estimate the expected KL divergence between the prior and the posterior distribution for different sensor locations $\boldsymbol{s}$. The KL divergence can be understood as a measure of distance between two probability distributions. Thus, higher values of divergence correspond to preferable locations for the sensor, leading to higher information gain. The resulting utilities are plotted in Figure 14. For all experiments we find that the tip of the head ($s = 0$) exhibits the largest utility independent of the number of swimmer in the leading group. At the tip of the head, the two symmetrically placed sensors have the smallest distance. In Equation (93) we have assumed that the two swimmer halves are symmetric and uncorrelated. Due to the small distance of the sensors at the head, spatial correlation between the sensors across the swimmer halves would decrease the utility of this location. In order to test whether the utility for sensors at the head is influenced by this symmetry assumption, we perform experiments where we place a single sensor on one side of the



(a) (b)

FIGURE 14: Utility curves for the first sensor using pressure measurements. In Figure 14a the utility estimator for the “size of the leading school” experiment is presented. Figure 14b corresponds to the utility estimator for the “relative position” experiment. We show the resulting curves for one, three and seven swimmer in the leading group and the total expected utility. We observe that although the form does not drastically change, the total utility increases with increasing size of the leading group.

swimmer. Again, in this case the location at the head is found to have the highest expected utility.

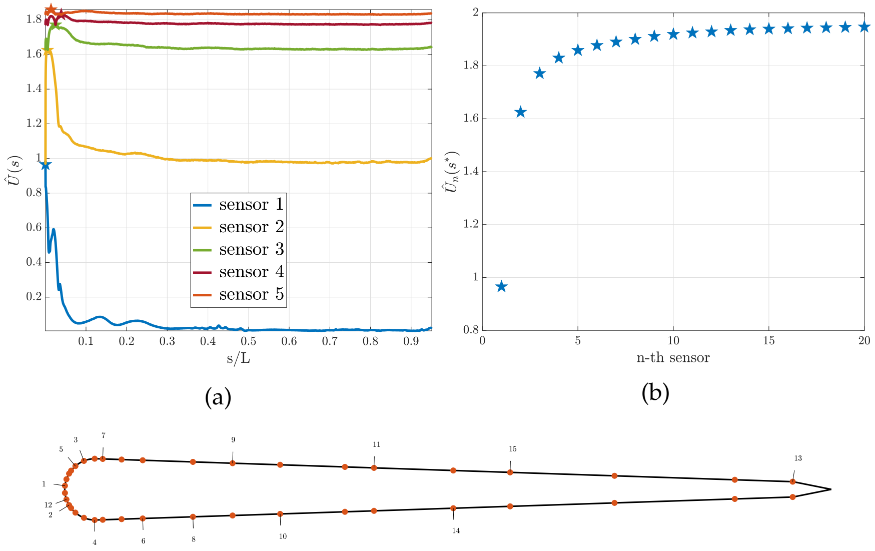
There is evidence that the head experiences the largest variance of pressure gradients $\boldsymbol{F}(\boldsymbol{\vartheta};\boldsymbol{s})$. The same observations can be made for the density of the sub-canal neuromasts, which is also highest in the front of the fish [172]. To check the presence of this correlation in our study, we examine the variance of the values obtained from our numerical solution of the Navier–Stokes equation. We confirm that our simulations are consistent with this experimental observation. We find that independent of the number of swimmers, the variance in the sensor signal $\mathrm{var}_{\boldsymbol{\vartheta}}(\boldsymbol{F}(\boldsymbol{\vartheta};\boldsymbol{s}))$ is largest at $s=0$.

4.2.5.2 Multiple Sensors

In this section we discuss the results of the sequential sensor placement described in Section 4.1.3. For the “size of the leading school” experiment we present the results in Figure 15. In Figure 15a the utility curve for the first five sensors is shown. We observe that the utility curve becomes flatter as the number of sensors increase. Furthermore, we observe that the location where the previous sensor was placed is a minimum for the utility for the next sensor. Figure 15b shows the utility estimator at the optimal sensor for up to 20 sensors and it is evident that the value of the expected utility reaches a plateau. In Figure 15c the found optimal location of the sensors on the skin of the swimmer is presented. The numbers correspond to the iteration in the sequential procedure that the sensor was placed. Note that the sensors are being placed symmetrically.

The optimal sensor placement results for the “relative position” experiment can be found in Figure 16. Similar to the other experiment the utility curves become flatter after every placed sensor and the location for the previous sensor is a minimum for the utility for the next sensor (see Figure 16a). We plot the maximum of the utility for up to 20 sensors (see Figure 16b) and observe a convergence to a constant value. In Figure 16c the found optimal location of the first 20 sensors is presented.

For both experiments, it is evident that the utility of the optimal sensor location approaches a constant value. This fact can be explained by recalling that the expected utility in Equation (85) is a measure of the averaged distance between the prior and the posterior distribution. Increasing the number of sensors leads to an increase in the number of measurements. By the Bayesian central limit theorem, increasing the number of measurements leads to convergence of the posterior to a Dirac distribution. As soon as the



(a)

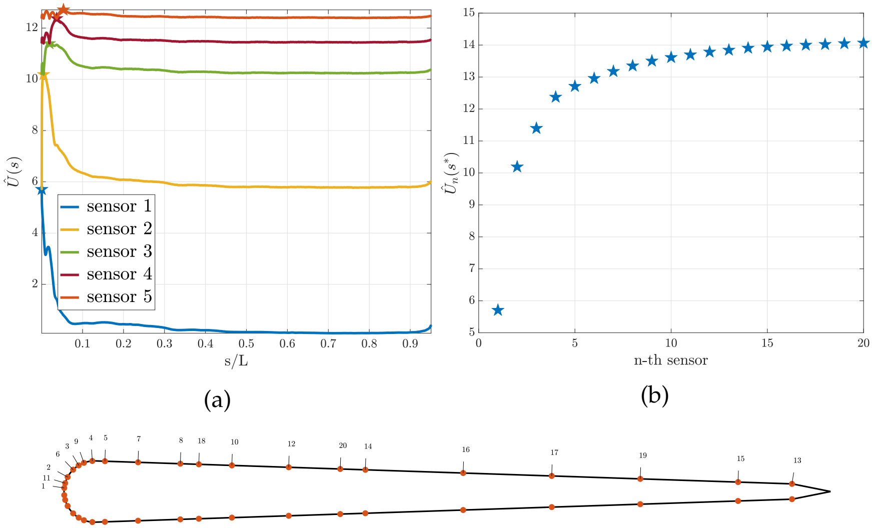
(b)

(c)

FIGURE 15: Optimal sensor placement for the pressure sensors and the "size of the leading school" experiment. In Figure 16a the utility estimator for the first five sensors and in Figure 16b the value of the utility estimator at the optimal sensor location for the first 20 sensors are presented. In Figure 16c, the distribution of the sensors on the swimmer surface is presented. Here, the numbers associated to each sensor indicate that this location is the *i*-th sensor location chosen according to Equation (89).

posterior has converged, the expected distance from the prior, and thus the expected utility, remains constant.

The found sensor distributions for the two objectives are similar, having clusters at the head and uniform distribution along the body. In order to underpin the biological relevance of the observed sensor distribution we compare our results to [172]. Given that the canals display significant 3D branching in the head a direct comparison is difficult. However, the found cluster of sensors at the head agrees qualitatively with the high canal density reported in [172].



(a)

(b)

(c)

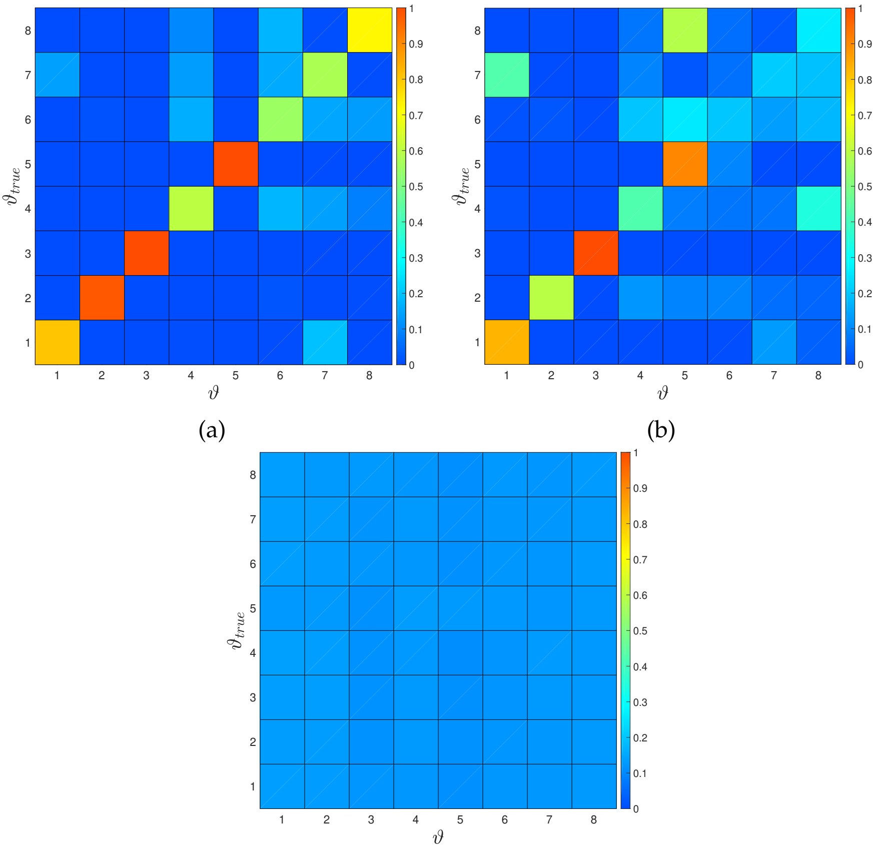
FIGURE 16: Optimal sensor placement for the pressure gradient sensors for the “relative position” experiment. In Figure 16a, the utility estimator for the first five sensors and in Figure 16b the value of the utility estimator at the optimal sensor location for the first 20 sensors are presented. In Figure 16c, the distribution of the sensors on the swimmer surface is presented. Here, the numbers associated to each sensor indicate that this location is the *i*-th sensor location chosen according to Equation (89).

4.2.5.3 Inference of the Environment

In this section we demonstrate the importance of the optimal sensor locations and examine the convergence of the posterior distribution. We compute the posterior distribution via Bayes' theorem given in Equation (82). We set $\boldsymbol{y} = \mathbf{F}(\boldsymbol{\vartheta}, \boldsymbol{s})$ and compute the posterior for different values of $\boldsymbol{\vartheta}$ in the prior region. We consider measurements collected at: (a) the optimal and (b) the worst sensors location.

The posterior probability for the "size of the leading school" experiment is shown in Figure 17. We observe that the worst sensor location implies an almost uniform posterior distribution, reflecting that measurements at this sensor carry no information. On the other hand, the posterior distribution for the optimal sensor is more informative. We observe that for groups with small size the follower is able to identify the size with more confidence, as opposed to larger groups. We compare the posterior for an experiment with only one configuration per group-size to an experiment with multiple configurations. For multiple configurations the posterior is less informative. This indicates that the second case occurs to be a more difficult problem. Finally, notice that the posterior for one configuration is symmetric, where when adding multiple configurations this symmetry is broken. This fact is discussed in Appendix A.1.2.

The posterior density for the "relative position" with one leading swimmer is presented in Figure 18. The posterior for the configuration with three and seven swimmers is similar. We compute the posterior for measurements at the best and the worst location for one and three sensors. For the three sensors the worst location has been selected in all three phases of the sequential placement.The results for the normalized densities are shown in Figure 18. We observe that one sensor at the optimal location gives a very peaked posterior. Three optimal sensors can infer the location with low uncertainty. This is not the case for the worst sensors, where adding more sensors does not immediately lead to uncertainty reduction.

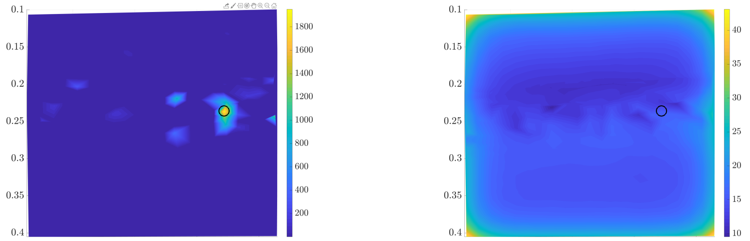


(a)

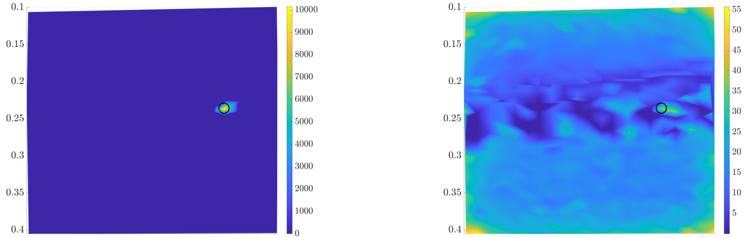
(b)

(c)

FIGURE 17: (**a**) Estimated posterior probability for a single sensor optimally placed and a single configuration per group size. The posterior shows clear peaks at the correct number of swimmer for all cases, leading to perfect inference of the parameter of interest. The posterior probability for (**b**) optimal and (**c**) worst sensor location for multiple configurations per group size. Here, for the optimal sensor location and one, two, three and five swimmer we see a clear peak for the true size of the group. For the worst sensor location the posterior is almost uniform and does not allow to extract any information about the size of group.



(a) One sensor, best location (b) One sensor, worst location



(c) Three sensors, best location (d) Three sensors, worst location

FIGURE 18: Estimated posterior for the final location for the best (left column) and worst (right column) sensor-location for one (upper row) and three sensors (lower row). Light colors correspond to high probability density values. We marked the actual location with a black circle.

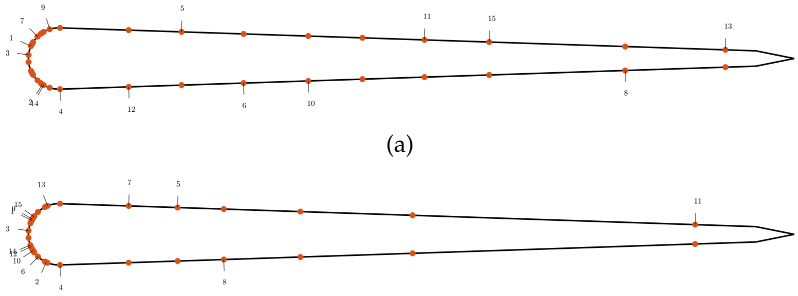
4.2.5.4 Shear Stress Sensors

In this section, we discuss the results for the optimal positioning of shear stress sensors. We follow the same procedure as in Sections 4.2.5.1 and 4.2.5.2. Here, we omit the presentation of all the results and focus on the similarities and differences to the pressure gradient sensors.

The optimal location for a single sensor for the “size of the leading school” experiment is at $s^* = 3.01 \times 10^{-4}$. For the “relative position” experiment we find the optimal location $s^* = 3.84 \times 10^{-4}$. In contrast to the optimal location for one pressure gradient sensor, the found sensor is not at the tip of the head and is at different positions for the two experiments. Examining the variance in the shear signal shows quantitatively the same behaviour as the utility. Comparing the location of the maxima in variance shows that they do not coincide with the found maxima for the expected utility for shear sensors.

We perform sequential placement of 15 sensors. The resulting distribution of sensors is shown in Figure 19. In Section 4.2.5.2 we argue that the expected utility must reach a plateau when placing many sensors using the Bayesian central limit theorem. For shear stress sensors we observe that the convergence is slower compared to the pressure gradient sensors. We conclude that the information gain per shear stress sensor placed is lower as for the pressure gradient sensors.

The posterior density obtained for both experiments is less informative when using the same number of sensors. Also this indicates that shear is a less informative quantity yielding a slower convergence of the posterior. This is in agreement with the observation that the subcanal neuromasts



(a)

(b)

FIGURE 19: Optimal sensor locations for the shear stress measurements for the “size of the leading school” in Figure 19a experiment and “relative position” experiment in Figure 19b.

associated with pressure gradient sensing are more robust to noise [153]. For multiple fish in schools the resulting flow field is disturbed, thus suggesting the use of pressure gradient sensors.

5

CONTROLLING ARTIFICIAL SWIMMERS USING DEEP REINFORCEMENT LEARNING

But in practial affairs, [..], men are needed who combine human experience [..] with a knowledge of science and technology.

— Max Born

In the following we present two applications of the reinforcement learning methods presented in the previous chapters for an algorithmic understanding of natural behavior:

MULTI-TASK REINFORCEMENT LEARNING TO SWIM IN REVERSE AND FORWARD KÁRMÁN WAKES Fish possess a remarkable ability to sense and interact with their aquatic environment. They harvest energy from vortices in the flow during upstream migration and schooling. The present work combines deep reinforcement learning and direct numerical simulations to reproduce the energy harvesting in-silico. We use multi-task learning and adaptive mesh refinement to reduce the computational cost. The experiments focus on forward and reverse Kármán vortex streets in the range of relevant Strouhal numbers. The results allow insights in the flow physics by analyzing the computed flow fields. They showcase the capabilities of the employed methodology to complement the existing experiments in an unified framework. We provide visualizations of the policy and value function to decipher the learned neural network.

TRANSFER LEARNING FOR SCHOOLING SWIMMERS WITH MULTI-AGENT REINFORCEMENT LEARNING The hydrodynamics of schooling fish has remained elusive. One of the reasons for the difficulty is the computational cost of acquiring effective control policies that allow simulations of schools where the swimmers are self-propelled and do not rely on external forcing. In this work we simulate schools up to 100 swimmers with direct numerical simulations and learn their control with reinforcement learning. Varying the number of swimmers allows examining the stability and control law. The study is enabled by adaptive mesh refinement and transfer learning. We refine the grid around the swimmers and in their wake and initialize the learning for larger schools using the policies from smaller schools. We assess the benefits and disadvantages of different training strategies for multi-agent reinforcement learning. We find that while training a single policy per agent gives more robust policies, they are less transferable. In contrast to that, training a single policy for all swimmers allows successful transfer of information from small schools to larger schools. Our study allows unprecedented insight and analysis of the hydrodynamics of fish schools over a large range of sizes.

ACKNOWLEDGMENTS The content is based on [188, 189]. The computational resources were provided by the Swiss National Supercomputing Centre (CSCS) under project s929.

5.1 Reinforcement Learning for Artificial Swimmers

We assume that fish interact with their environment at discrete times $t = 1, \dots, T$. First, the fish observes a sixteen-dimensional state $s^{(t)} \in \mathcal{S} \subset \mathbb{R}^{16}$. The elements correspond to three sources of perceptional cues: sight, flow sensing, and proprioception. The first six components are the displacement, orientation, as well as the linear and angular velocity. The lateral line system allows fish to sense shear stress and pressure gradients via neuromast sensors. We computed the optimal placement of these sensors by using Bayesian optimal experimental design to determine the locations that maximize the information gain from measurements [190, 191]. We include the shear stresses at three of the resulting locations, giving another six components of the state vector. Finally, we model proprioception by adding the time of the previous interaction, the last two signals to modify the curvature, as well as the current phase of the undulation as a state [56, 182]. Given the cues from the environment, the fish decides upon an action $a^{(t)} \in \mathcal{A} \subset \mathbb{R}^2$ that represents the incentive to change the direction and the velocity. After performing an action, the fish observes a new state $s^{(t+1)}$ and a reward $r^{(t)} \in \mathbb{R}$. In the present work, we chose the Froude swimming efficiency as reward [56].

The described interaction loop is visualized in fig. 20 and the implementation of the learning follows the discussion from chapter 3. As we have discussed in chapter 2, we model the deformation via the midline curvature by a sinusoidal wave with period T. The amplitude $K(s)$ is linearly increasing from $K(0) = 0.82/L$ to $K(L) = 5.7/L$. The baseline motion is modified by the actions of the agent at every half swimming period. They control the swim-

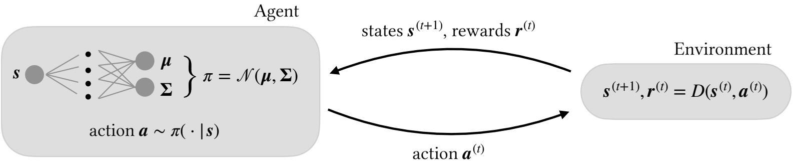

Agent
s
μ
Σ
$\pi = \mathcal{N}(\mu, \Sigma)$
action $a \sim \pi(\cdot\,|s)$
states $s^{(t+1)}$, rewards $r^{(t)}$
Environment
$s^{(t+1)}, r^{(t)} = D(s^{(t)}, a^{(t)})$
action $a^{(t)}$

FIGURE 20: Schematic of the reinforcement learning loop.

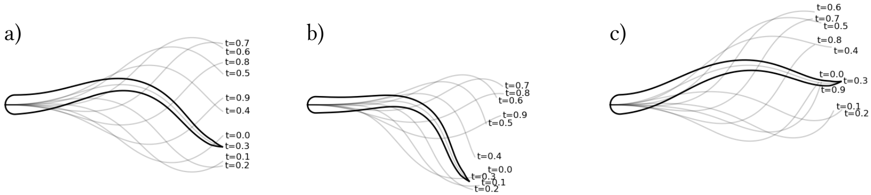


FIGURE 21: Illustration of the swimmer model. In grey we plot the midline over one period ($t \in [0, T]$) of the baseline motion (a), and the modified motion when repetitively taking the minimal- (b), and the maximal action (c). At $t = 0.3$ the shape is plotted together with the midline. Note that b) and c) are symmetric when adding a phase $\phi = \pi$, which explains why the time between actions is half the swimming period.

ming period T and add a wave package $A(s,t)$ to the baseline-curvature (see [56, 182] for details)

$$k_{\mathrm{RL}}(s,t) = K(s)\left[\sin\left(\frac{2\pi t}{T} - \frac{2\pi s}{L} + \phi\right) + A(s,t)\right] . \tag{97}$$

Figure 21 shows the effect of the additional curvature $A(s,t)$ for the minimal and maximal values of the action. As can be seen, the actions allow biasing the motion of the fish towards a certain direction. Using the Frenet–Serret formulas, the midline and the normal vectors can be reconstructed from the curvature [9]. The shape is constructed by adding the width as defined in chapter 2.

During the undulation, drag and thrust production are distributed over the whole body. The total force is computed by integrating the force components over the swimmer surface [192]. Here, we distinguish the negative and positive contributions as drag and thrust respectively [8, 56]. From the forces, the forward power P_{thrust} and the deformation power $P_{\text{deformation}}$ are computed. The Froude swimming [193] efficiency is consequently

$$\eta = \frac{P_{\text{thrust}}}{P_{\text{thrust}} + \max\{0, P_{\text{deformation}}\}} . \tag{98}$$

Clipping of the deformation power is required to avoid negative contributions when the deformation of the obstacle is aligned with the fluid forces [56]. If swimmers collide or a swimmer exits the computational domain, a negative penalty is given.

5.2 Multi-Task Reinforcement Learning to Swim in Reverse and Forward Kármán Wakes

The effectiveness of fish swimming is unprecedented and understanding how can inspire the effective design of underwater robots [194]. In multiple experiments it was shown how fish can benefit from the complex non-linear interactions with the vortices in fluid flows [195, 196]. Reproducing their behaviour in accurate computer simulations allows better understanding of this behaviour and the underlying flow physics. The gained knowledge can then be transferred back to experiment and engineering applications [197]. In the present study we combine *Direct Numerical Simulations* (DNS) and *Reinforcement Learning* (RL) to achieve this task. RL has been successfully applied in fluid mechanics applications [198, 199]. For swimming, RL was used to optimize the motion of artificial swimmers to follow a given path [200, 201], keep station [182], or optimize efficiency [56]. Similarly it was used for goal-reaching and swimming behind a D-shaped cylinder [202, 203], or discovering optimal escape patterns [204].

The existing literature provides insight into the mechanics that govern fish propulsion, collaborative behaviour, and station keeping via case-by-case experimental or reverse engineering studies. In nature, however, fish rarely face a single task, and can quickly adapt to new situations. We thus argue

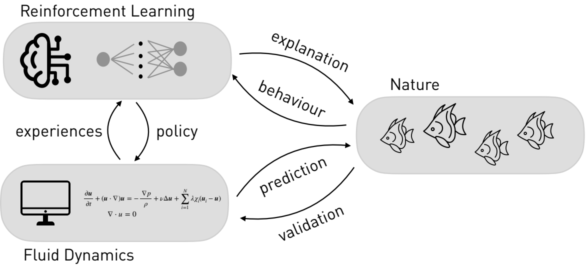

Reinforcement Learning
explanation
behaviour
Nature
experiences
policy
prediction
validation
$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = -\frac{\nabla p}{\rho} + \nu\Delta\boldsymbol{u} + \sum_{i=1}^{N}\lambda\chi_i(\boldsymbol{u}_i - \boldsymbol{u})$
$\nabla\cdot u = 0$
Fluid Dynamics

FIGURE 22: Schematic of the proposed integration of RL in an algorithmic understanding of nature. Computational Fluid Dynamics (CFD) allows predicting nature, where experiments provide data for validating the code. By combining CDF with RL, we generate experiences, that in turn allow training a policy to optimize a reward that is inspired by behaviour observed in nature. Evaluating the resulting policy closes the loop and can be used to explain the targeted behaviour.

that the underlying optimization problem cannot only focus on solving one task at a time. In the present work, we shift the paradigm of algorithmic understanding of nature [205] towards multiple tasks. RL works effectively and we show that it enables the reproduction of the versatile behaviour observed in real fish by optimizing the Froude swimming efficiency. The found policies work in different flow scenarios without retraining or modifying the state. This level of generalization is achieved by randomly sampling the environment during training. Our approach can be transferred to other domains that target the algorithmic understanding of natural behaviour. Furthermore, employing DNS minimizes the modeling error and provides an accurate picture of the governing physics. We visualize the proposed loop in fig. 22: While DNS have to be strictly validated against nature, they allow predicting outcomes in new scenarios. DNS generate experiences using which a policy can be trained via RL. The goal can be inspired from natural behaviour and in turn yields a possible explanation and detailed picture of the underlying physics.

5.2.1 Environment

The velocity $\boldsymbol{u} : \Omega \times [0, T] \to \mathbb{R}^2$ and pressure $p : \Omega \times [0, T] \to \mathbb{R}$ of the fluid with kinematic viscosity $\nu \in \mathbb{R}$ and density $\rho \in \mathbb{R}$ that surrounds the swimmer is computed by solving the incompressible two-dimensional Navier-Stokes equation with Brinkman penalisation as discussed in chapter 2.

For sufficiently high Reynolds numbers, the flow around bluff bodies is characterized by forward Kármán vortex streets. Experimental studies showed, that fish swimming behind D-shaped cylinders reduce their muscle activity [195] and adopt characteristic kinematics, the Kármán gait [206, 207]. Both, the size of the bluff body and the speed of the flow, influence the vortex shedding frequency and correspondingly alter the Kármán gait [208]. Similar studies were performed with dead trout. Although lacking any control, their bodies synchronize with the vortex street and are propelled upstream [209]. In contrast to bluff bodies, self-propelled objects imply reverse Kármán vortex streets. A typical example are flapping hydrofoils, which serve as a model for swimming fish. Also for this type of flow, a recent study found altered swimming gaits for rainbow trout [196]. In earlier studies, flapping hydrofoils enabled a better understanding of the undulation frequency of fish [210–212], interaction of self-propelled bodies with vortices [213, 214], and energy gains in groups [215–217].

We study forward and reverse Kármán wakes by simulating D-shaped cylinders and flapping hydrofoils. Figure 23 illustrates the vorticity field and

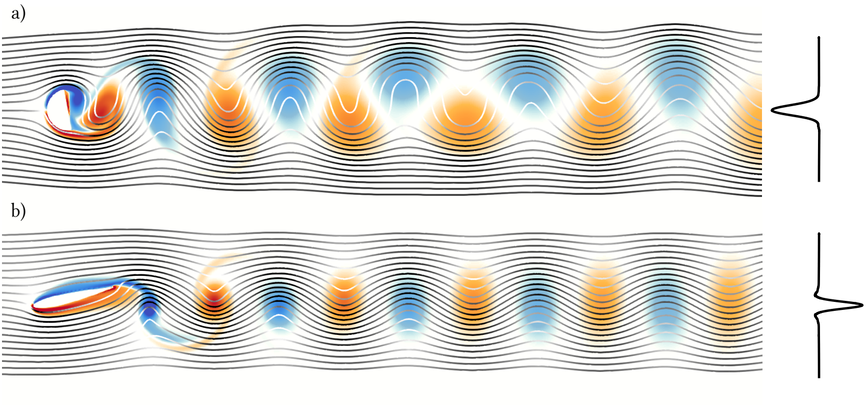


FIGURE 23: The flow behind a D-shaped cylinder with diameter $d = 0.06$ (a), the flow behind a flapping hydrofoil at St $= 0.3$ (b) at Re$=$ $1'000$. Negative vorticity is highlighted in blue, positive vorticity in red. The streamlines display the velocity magnitude (darker lines for larger absolute values). On the right we plot the time-average of the x-component of the flow velocity.

the average velocity profile for such flows. The streamlines are shaded according to velocity magnitude. Light colors indicate lower velocity and thus areas where station keeping is easier. In both cases we can identify light regions where a swimmer can harvest energy from the flow. However, the forward and reverse Kármán wakes are fundamentally different. While the drag inducing D-shaped cylinder shows an overall velocity deficit, the flapping hydrofoil generates a flow excess by generating thrust. The spacing between the vortices is determined by the non-dimensional Strouhal number St $= \frac{f \cdot A}{u}$, where f is the frequency of the vortex shedding for the D-shaped cylinder or the flapping frequency for the hydrofoil; A denotes the characteristic length, that is the diameter of the hydrofoil, or the amplitude of the flapping.

5.2.2 Results

For the simulations, the computational domain is $\Omega = [0, 10] \times [0, 5]$. The reference length scale L equals to one swimmer length and the reference time scale equals one swimming period T. The reference density ρ is set equal to one and the Reynolds number, defined as Re $= \frac{L^2}{T\nu}$, is equal to $1'000$. The moving obstacle's position is fixed at $\boldsymbol{x} = (3, 2.5)$ by moving the frame of reference at constant velocity. The swimmer starts at $\boldsymbol{r} = (4.5, 2.5) + \Delta\boldsymbol{r}$ with $\Delta\boldsymbol{r} \sim \mathcal{U}([-1/3, 1/3]^2)$ downstream from the obstacle at an angle of $\alpha \sim \mathcal{U}([-5, 5])$ degrees. Here, $\mathcal{U}([a, b])$ denotes the uniform distribution over the interval $[a, b] \subset \mathbb{R}$. Given the initial condition, the simulation runs for at most 100 swimming periods, after which the environment is truncated. Numerical difficulties arising from an impulsive start of the undulating motion are avoided by ramping up the amplitude of the curvature over the first swimming period. The swimmer reaches a steady swimming speed after approximately four swimming periods. Similarly, the obstacle velocity is linearly increased from $\mathbf{0}$ to the target velocity $\boldsymbol{v} = (-0.75, 0)$ over five swimming periods. If the swimmer leaves the margins $\Omega_{\text{habitable}} = [0.5, 9.5] \times [0.5, 4.5]$ the simulation is terminated and the swimmer is penalized. We use an effective number of $2'048 \times 1'024$ grid points with seven levels of refinement. The grid is refined around obstacles and according to vorticity magnitude. The resulting grid-spacing at the finest resolution is $\Delta x = \Delta y = 0.005$, i.e. 200 grid points per swimmer length. The time-step is controlled via a Courant number of 0.4. Following the experimental studies on Kármán gaiting [195, 218, 219], we focus on diameter of the D-shaped cylinder in the range $d \in [0.3, 0.7]$ and place it at an angle of 10° to obtain vortex shedding. For the flapping hydrofoil, we fix the amplitude to $A = 13.15^\circ$, the length to $c = 0.6$, and vary the frequency f such that the Strouhal number $\text{St} = \frac{2Acf}{\|\boldsymbol{v}\|} \in [0.1, 0.5]$ is in the natural range for swimming [196, 220].

5.2.2.1 Physical Mechanisms

The physical mechanisms that allow the swimmer to harvest energy while swimming behind the obstacle are examined by observing the flow field (fig. 24, fig. 25, and fig. 26).

As shown in fig. 24, the optimal policy computed with RL adopts the strategy which was observed in experiment [195]. The fish slalom between the vortices, thus avoiding regions with high flow velocity and aligning with the flow field. This behaviour, termed *Kármán gait*, tends to be adopted by fish in nature [221]. In the right of the figure we see how the swimmer aligns with the velocity field, and therefore it reduces the amount of work that has to be brought up against the fluid to deform its shape. This reduces the required muscle contractions. Additionally it benefits from the upward direction of the velocity field to reduce its drag.

In reverse Kármán vortex streets, the mechanism is different. Especially at Strouhal numbers close to the self-propelled swimmer, the wavelength of the vortex-street is smaller. For small wavelengths, slaloming in between

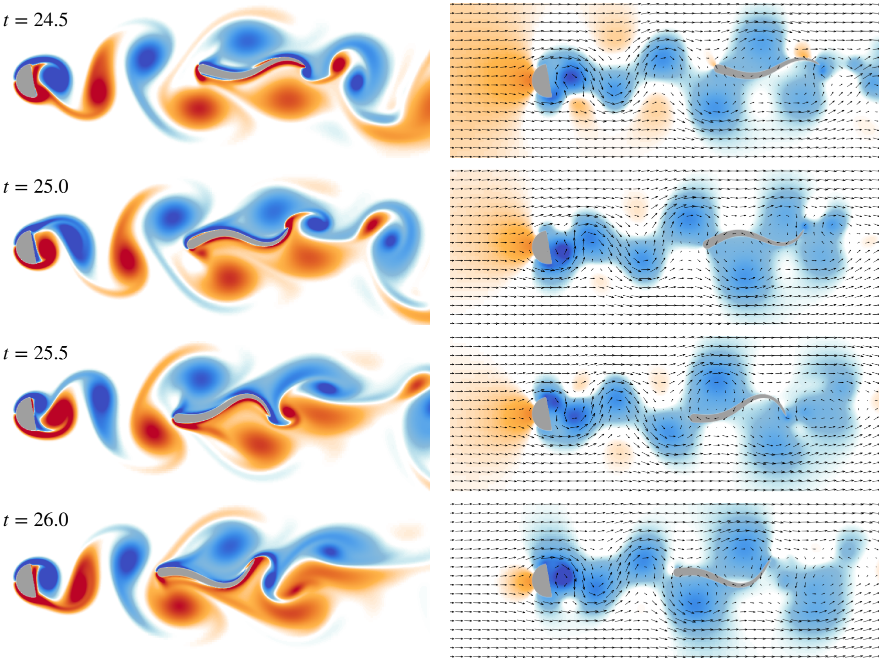


FIGURE 24: Figure showing the flow encountered in the wake of the D-shaped cylinder ($d = 0.3$) during one shedding cycle. The left column shows vorticity, the right the pressure and velocity field. Darker red indicate more positive and blue negative values, respectively.

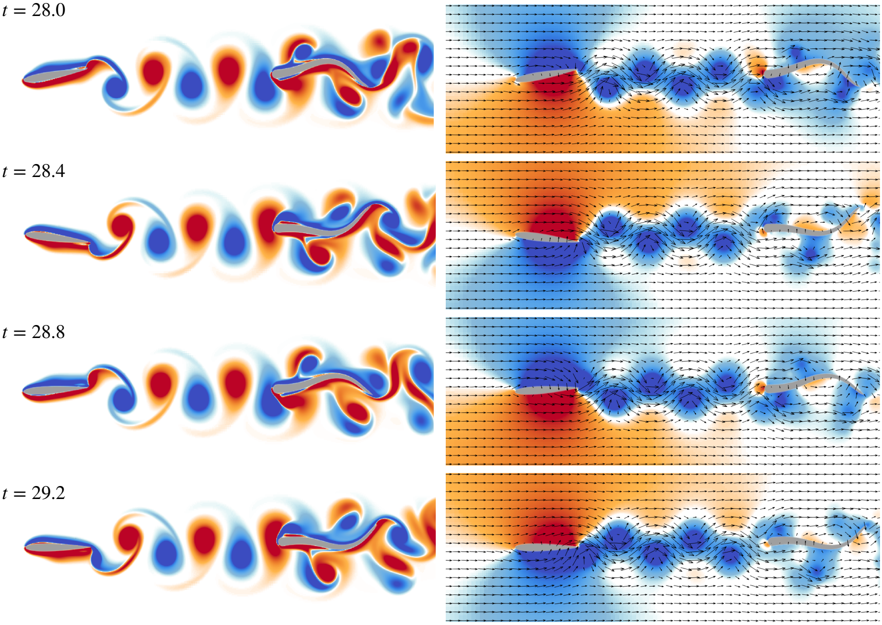


FIGURE 25: Figure showing the flow encountered in the wake of a flapping hydrofoil (St= 0.5) during one swimming cycle. The left column shows vorticity, the right the pressure and velocity field. Darker red indicate more positive and blue negative values, respectively.

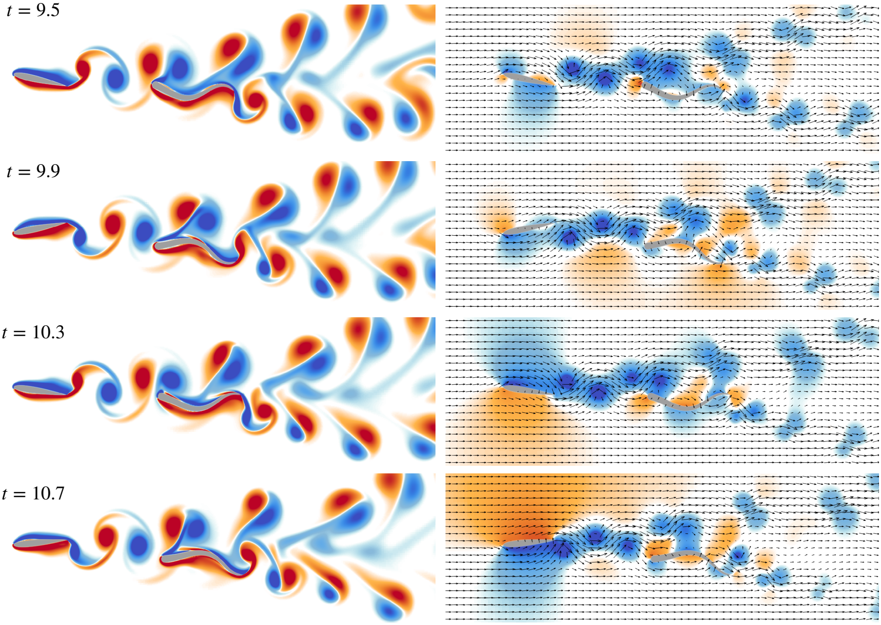


FIGURE 26: Figure showing the flow encountered in the wake of a flapping hydrofoil (St= 0.3) during one swimming cycle. The left column shows vorticity, the right the pressure and velocity field. Darker red indicate more positive and blue negative values, respectively.

vortices becomes infeasible. One strategy employed in tandem swimmers is to intercept vortices [56]. This strategy is also recovered in the present work when using high flapping frequencies (St=0.5) (see fig. 25). In contrast to the previous study, we only see the formation of one secondary vortex on each of the respective sides. The lower Reynolds number used in the present study is 1,000 and was 5,000 in [56]. Reducing the Reynolds number can be seen as a weakening of the inertial forces, which explains the difference.

For intermediate Strouhal numbers, a second strategy is found. As shown in fig. 26, the swimmer positions itself at a slight offset from the central axis. The offset allows the fish to intersect a vortex with its head, thereby transporting negative vorticity along the surface, creating a new vortex dipole. The vorticity generated from the underlying motion is propelled downwards. The resulting fluid momentum points downstream and, according to Newtons third principle, the swimmer is propelled upstream. Interestingly, the swimmer does not align with the flow field. Since from alignment additional energy can be extracted from the flow this yields lower efficiency gains compared to swimming behind the D-shaped cylinder (see fig. 27 b)).

5.2.2.2 Training and Evaluation

We perform three RL studies where we trained the agent for 500'000 interactions with the environment. As shown in fig. 27 a), the returns plateau, indicating that the policy converged. In our first study, we sample D-shaped cylinders with diameter size $d \sim \mathcal{U}([0.3, 0.7])$. In the second study, we trained the swimmer behind hydrofoils with chordlength 0.6 and random Strouhal number St $\sim \mathcal{U}([0.1, 0.5])$. Lastly, we train a multi-task policy where we sample a Bernoulli distribution to decide whether to perform an episode using a cylinder or a hydrofoil. We evaluate the policy's performance with 10 random initial conditions for 10 values of the diameter D-shaped cylinder and frequency for the flapping hydrofoil. The obtained testing returns are shown in fig. 27 b).

The median cumulative rewards and ranges only show slight differences between the environments. Based on this we could prematurely conclude that the returns that can be achieved in these environments are the same. Therefore we could hypothesize, that similar underlying characteristics are shared by both environments.

Multitask training achieves the same returns as training a single task, despite having the same total number of observations. Since the computational cost is mainly determined by the number of interactions, multitask training effectively reduces necessary computational resources. Without multitask

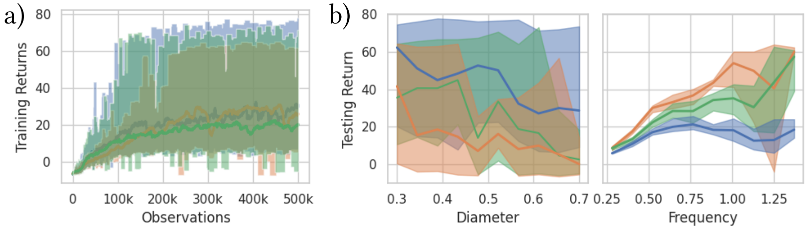


FIGURE 27: Figure illustrating the training and testing process. Figure (a) shows the median of the cumulative sum of rewards over 100 episodes with the range of samples shaded. Figure (b) shows the minimum, mean, and maximal return when varying the radius and flapping frequency for the cylinder and the hydrofoil, respectively. The colors distinguish the returns achieved with the policies of the different reinforcement learning studies: D-shaped cylinders (—), hydrofoils (—), or the multitask training (—).

learning, we would have performed the optimization for each obstacle and also each parameter separately. For the present results we consider 10 different angles and flapping frequencies, resulting in an effective saving of 2×10X. With the presented approach, RL allows optimizing the policy for all environment configurations simultaneously. Thus, the multi-task approach reduces the computational costs of learning a policy that generalizes between environments.

During the evaluation, the differences between the environments and the used training schemes becomes clearer. When regarding fig. 27 b) we find much larger ranges for the returns for the cylinder. They range from 20-80. Although the mean return becomes smaller for larger radii, the maximal attained returns are almost constant (blue shaded area). This indicates that the maximal amount of energy that can be extracted from the flow does not greatly vary when changing the radius of the D-shaped cylinder. However, the required control is more difficult, yielding a lower mean return.

For the agent behind the flapping hydrofoil, the mean return increases from 10 to 60 when increasing the flapping frequency. The range of the returns is much narrower. One exception to the picture is the run at $f = 1.25$. Here a run with a very low value for the return is recorded. This indicates that the swimmer had a fatal encounter with the wake of the hydrofoil. According to our results it is preferable for the swimmer to swim behind a hydrofoil with higher flapping frequency. Furthermore, the found returns is more consistent, indicating a more stable control law.

When evaluating the multitask policy, the dependencies of the testing returns are similar. However, the returns compared to the specialized policies are 20-30% lower. When doing cross-validation, the multi-task learning shows it's benefits. The returns are 40-50% lower when evaluating the policies from one environment in the other. This means a 2X improvement. How each cases can profit from the insights from the respective other cases will become clearer in the analysis in the following section.

5.2.2.3 Policy and Behaviour

In order to understand how these returns are achieved, an evaluation of the collected experiences is performed. Since the policy is converged, the 65'536 samples in memory follow the distribution d of states visited by the optimal policy. We focus on the spatial dependencies $(x, y) \in \mathbb{R}^2$. The marginal is computed for the value function V and mean output μ for the actions

$$V(x, y) = \mathbb{E}_{s\setminus(x,y)\sim d}\left[V(s)\right] , \qquad \mu(x, y) = \mathbb{E}_{s\setminus(x,y)\sim d}\left[\mu(s)\right] , \tag{99}$$

where $s \setminus (x, y)$ indicates that the expectation is over all states s excluding (x, y). The expectation values are computed by discretizing the spatial domain into 5000 squares $[s_b, s_{b+1}] \times [s_{b'}, s_{b'+1}]$ for $b, b' = 0, \dots, 99$ with $s_b = b\Delta$ and $\Delta = 0.1$. We average the entries in the replay memory with $(x, y) \in [s_b, s_{b+1}] \times [s_{b'}, s_{b'+1}]$. The resulting marginals (fig. 28) can then be plotted and analyzed in order to understand the spatial dependency of the value function and the learned policy.

A clear difference between the marginals for the forward Kármán wakes shed by the D-shaped cylinder and the reverse Kármán behind a flapping hydrofoil is the area of the states that are visited. For the hydrofoil, the area is significantly smaller than for the cylinder due to the form of the vortex streets and the paths that allow intersecting the low-velocity regions in the wake (see fig. 23). For the cylinder, it is preferable to slalom between the vortices; for the hydrofoil, it is beneficial to stay close to the center of the vortex street. We discussed these conclusions in section 5.2.2.1.

Notably, the marginal of the value function for the D-shaped cylinder shows a peak, corresponding to regions in space where the attainable efficiency gain is maximal. This finding coincides with the observations from experiments and previous numerical studies [203, 207], where a peak in the occupation probability is found downstream from the cylinder. The effect is less pronounced for the hydrofoil, where any location close enough has

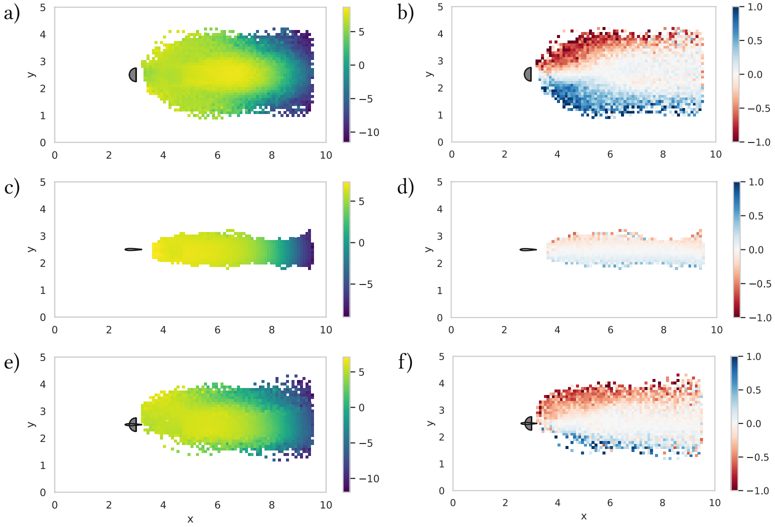


FIGURE 28: Figure illustrating the value function and the policies for the three cases considered. The left column (panels a,c,e) illustrates the marginal of the value function, the right column (panels b,d,f) the marginal of the curvature-controlling component of the actions. The rows correspond to the different training runs. a) and b) are found for the D-shaped cylinder, c) and d) are found for the hydrofoil, and the last row with e) and f) corresponds to the multitask training.

similar value, as observed in experiments [196]. In the multi-policy case, the value function resembles a superposition of the two individual value functions. We note that regions close to the boundary of the habitable zone have a very low value, indicating that a swimmer is in an unrecoverable state.

The mean of the policy provides additional information. When found above the centerline through the obstacle, the fish is likely to perform action `(+1)`; when below the centerline, action `(-1)` is more likely. As depicted in fig. 21, such actions translates into a bias of the undulatory motion that allows the swimmer to stay in the wake of the cylinder. The insight can be used as a prior when designing controllers in engineering applications. A similar analysis of the action controlling the swimming period shows that all agents tend to decrease their frequency. Decreasing the swimming period increases the swimming speed, however it also decreases the swimming efficiency. Since the obstacle moves at a higher velocity than the swimmer can reach during steady swimming, we require harvesting energy from the flow. The magnitude of the action can provide a baseline when designing engineering application.

The differences between for the policies also shed light on the different evaluation results discussed in the previous section. The state visitation for the hydrofoil is confined in a very narrow region with values decreasing when approaching the boundary of the domain. For the D-shaped cylinder, the optimal policy visits much more states that have suboptimal value, indicating higher probability to fail.

5.3 Transfer Learning for Schooling Swimmers with Multi-Agent Reinforcement Learning

Fish school. This observation is fascinating and yields some of the most beautiful patterns in nature. However, it puzzles scientists that seek understanding the reasons. From a physical point of view, schooling means coordination in a fluid dynamic environment. Fluids are described by the Navier-Stokes Equation (NSE), which belongs to the most complex equations known to humanity [222]. For non-trivial cases like fish swimming in a school we can not solve the NSE analytically and require simplified models or Direct Numerical Simulations (DNS).

Early theoretical work on the subject argued that fish school for a hydrodynamic benefit [223, 224]. This results however were controversial [225]. One of the main reasons for the dispute is that in some experiments the predicted structure could be found [226], where for others it was not [227, 228]. An accepted explanation is that energetic benefits are not the only reason for schooling, but is is also a mean of protecting from predators. Correspondingly, the active adoption and change in schooling configuration can be explained by balancing a trade-off between energetic benefit and awareness against predators [229, 230].

Another problem is the proposed static arrangement, that can not be achieved without forcing or active control [9, 200]. The arguably simplest branch of related work ignores any medium and considers particles that interact locally to satisfy three major traits: alignment, avoidance, and attraction [231–233]. Besides neglecting the complex non-linear hydrodynamic interactions these models yield to collisions that are nonphysical. Therefore these models do not allow to understand the physical effects taking place. Including invicid interactions makes this particle models fail and RL was introduced as a means to stabilize the school structure [201]. In later studies, RL was used to minimize the lateral displacement for two fish swimming in tandem [182]. Doing so implied efficiency benefits when intersecting the vortices shed by the leader. A follow up study directly optimized the efficiency, again finding the familiar patterns of schooling fish in nature [56]. More recent studies extended the scope to minimize the deviation from a target swimming velocity [234] and included up to six independent swimmers [57].

Using DNS with adaptive mesh refinement and Reinforcement Learning (RL) we extend the existing studies to 100 swimmers. We train the fish to avoid collisions while maximizing their efficiency. The resulting ability to simulate the school in a DNS allows to analyze the relation between fluid forces and efficient controls. The DNS are performed using a high-fidelity Navier-Stokes solver for incompressible flows around obstacles with distributed block-structured adaptively refined grids. The refinement takes place around the body of the swimmer and in the wake, where vorticity is formed. This allows reducing the computational cost, however the cost is still proportional to the number of swimmers. In order to account for this, we use transfer learning to initialize the learning for larger schools using the policies learned from smaller schools.

Both, the physical insight and the proposed learning strategy can benefit other studies in science and engineering and can help the design of windfarms [235, 236] or robotics [237–239].

5.3.1 Results

We consider different type of schools. The first is an extension of previous works, where leader-follower configurations were regarded [56, 182]. In the second part we regard the applicability of transfer-learning for diamond configurations, the shape considered in the pioneering works in the field [223, 224]. In the last section we discuss the results for simulations of 100 swimmers. The main difference between the first and the later two cases is the choice of policy. While in the first case the control is achieved by using one policy per swimmer, the later requires the use of a shared policy. The reason for this is the emergence of individual policies according to the position in the school. Although we found that individual policies yield more robust control overall [35], this comes at the cost of transferability. When trying to transfer the policies of the 20 swimmers according to their position to a school with 100 swimmers, the control is insufficient and the school breaks up. The emergence of individual roles in the school forbids the transfer of the policy to a new - even related - case. This is the main reason why it is preferable to use a single shared policy when trying to transfer policies learned from one case to another. The simulations are performed using DNS with CUBISMAMR [1, 192]. The Reynolds number is expressed in terms of the swimming period T and swimmer length L and set to Re=$\frac{L^2}{T\nu} = 1'000$. The RL is performed in KORALI [240] where V-RACER with ReF-ER [33] for Multi-Agent RL [35] was implemented and used with the default hyperparameters.

5.3.1.1 Column of Swimmers

We simulated a column of swimmers with $N = 4$ swimmers. Following an earlier study [182], we minimize the lateral displacement with an uncontrolled swimmer that leads the group. Here, it is observed that intersection of vortices lead to spontaneous bursts in efficiency. This motivated a study where we optimize the efficiency for a fixed leader [56]. In the following we revisit this studies with a column of 4, instead of 2 swimmers (see top figure fig. 29). The quantitative results are shown in the lower panels in fig. 29. The achieved mean displacement and efficiency increase for swimmers that are further back in the column. This confirms the intuitive picture that the swimmers in the back are exposed to more disturbances in the flow. While reducing the lateral displacement becomes harder, the ability to harvest energy from the vortices shed by the leader is increased. The mean efficiency for the last agent maximizing efficiency is 10% higher than for the agents minimizing the lateral displacement. For the agents maximizing efficiency, the column structure is not respected, the school breaks up and thus the lateral displacements become large and stabilize at value that are at the edge of the domain. This can be understood in terms of the definition of reward as swimming efficiency. By reducing the swimming speed, the swimmers are able to increase their efficiency. Since the swimmers are penalized if they approach the bounds of the computational domain, they can not fall back forever and form a school near the boundary of the domain. We conjecture, that this is also the reason for the breakup observed in a related study with up to 6 swimmer [57]. Since controlling the swimming speed is considered to be the reason for the division, we perform a study where all swimmers are controlled and the swimming speed can not be controlled. Interestingly in this case neither minimizing lateral displacement, nor maximizing efficiency are successful. This can be understood from the flow physics. By intersecting vortices, the swimmers increase their swimming efficiency. This increase in swimming efficiency implies an increased swimming speed. Since this acceleration only affects the followers, the distance to the leaders is reduce and yields collisions. With the ability to control the swimming speed, the collisions can be readily avoided. In the present case, however, the problem is rendered ill-posed. Without the ability to control the swimming speed, the swimmers can not find an optimal policy. An intriguing picture is found when optimizing both objectives without control over the swimming period. While the lateral displacement is at about the same level as for the case only optimizing displacement, the achieved swimming efficiencies are 41% and 28% higher than for the case optimizing lateral displacement and swimming

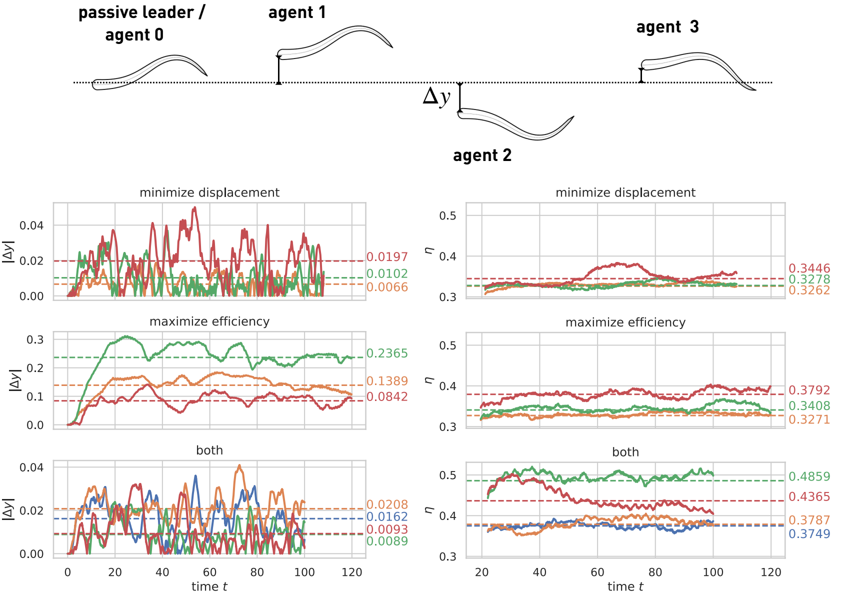


FIGURE 29: Results from the studies with 4 swimmers arranged in a column. The top figure shows the configuration, the lower plots show the results. Here, the first column shows the displacement, the second column the swimming efficiency. The rows the different cases that have been considered. The different colors indicate the different agents. The leader is (▬), and first follower (▬), the second (▬), and the last swimmer in the column (▬). In the first two cases the passive leader is not shown.

efficiency, respectively. The evaluation of the policy shows that the swimmer adopt burst-coast swimming, which is consider one of the most efficient swimming modes [241].

5.3.1.2 Transfer Learning for Diamond Shapes Schools

The ability to transfer knowledge between different sizes of schools is examined by varying the school size in the range $N = 4, 9, 16$ swimmers. The diamond shaped lattice structure is considered for the school of fish [223, 224]. Initially, the lateral displacement of the swimmers is chosen to be one fish length L, where the stream-wise displacement is three fish lengths (see fig. 30). The employed adaptive mesh refinement algorithm refines the grid around the body and in the wake of the swimmers. This in turn implies that the computational cost required to perform a simulation with N swimmers scales linearly with the number of swimmers. Thus, transferring knowledge from small schools to bigger schools results in computational benefits.

The aim of the study is to understand the effect of using the weights of the neural network as an initial guess for the policy, while ignoring any already collected experiences. In order to do so we do a total of six runs for the different number of swimmers $N = 4, 9, 16$. For each school, we train a policy until reaching 500'000 experiences. In the second run we do transfer learning. Here, we start with $N = 4$ swimmer and train for 24 hours. We then initialize the policy for the case with $N = 9$ swimmers and train for 24 hours. The resulting policy is again used to initialize the policy for the case with $N = 16$ swimmers. After that, the training is continued until reaching a total of 500'000 experiences.

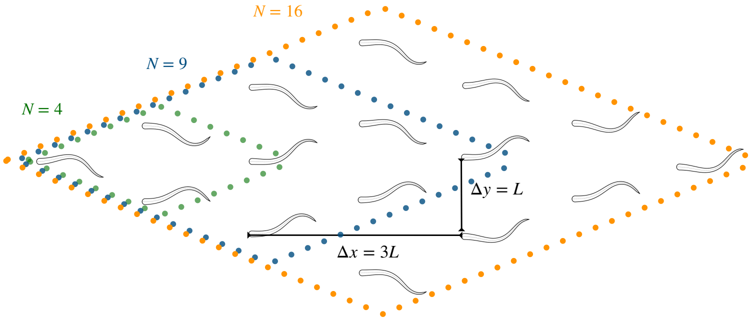


FIGURE 30: Schematic of the diamond shaped schools that were considered.

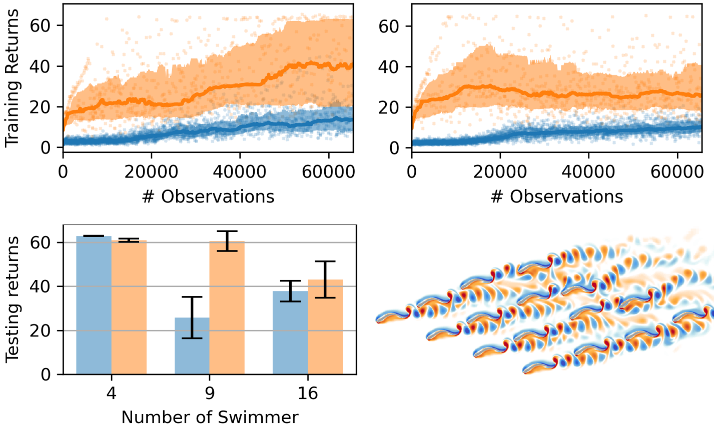


FIGURE 31: Results from the transfer learning. The top row shows the training results for 9 swimmers (left) and 16 swimmers (right). The lower row show the testing returns (left) as well as a snapshot of the flow field for 16 swimmers (right). Orange color corresponds to the results obtained from transfer learning, where blue results are plotted in blue. For the flow field, vorticity is shown, where positive values are red and negative vorticity is shown in blue.

The effect of the transfer learning becomes clearly visible when looking at the training rewards for the first 65’536 experiences, which are the experiences required to initially fill the replay memory (fig. 31). While the cold start mostly results in bad returns in the initial episodes, the transferred policy shows successful from the start. In order to compare the wealth of the final policies from the two strategies, we perform evaluation runs with the mean configuration and 10 configurations with a random displacement from the initial configurations. For the resulting 11 runs, we compute the mean and standard deviation of the returns. The results are shown in in fig. 31. As can be seen, the transfer learning performs better. This indicates that transferring the weights from a run with few fish to a run with more fish can be an effective initialization strategy. Moreover, we regard the computational cost. While during the first 24 hours 4 swimmer produce 220’086 experiences. The runs with 9 fish only reach 86’521 experiences. For the school with 16 swimmers we find 68’978 experiences. The resulting speedups are consistent with the linear scaling with respect to the number of swimmers. We consider this

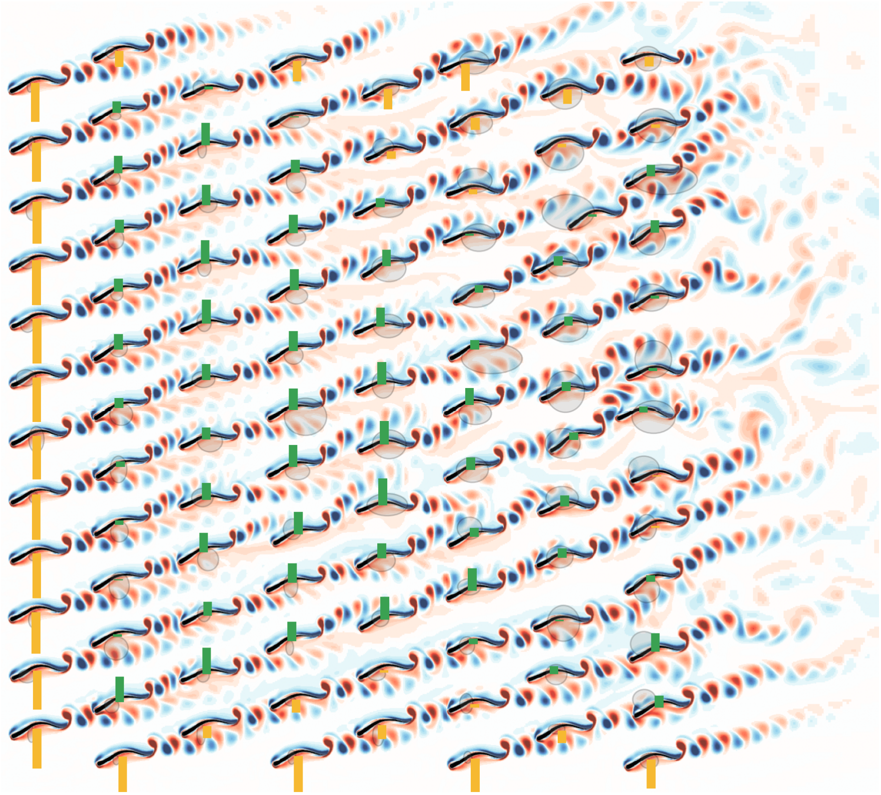


FIGURE 32: Results from the 100 swimmers run.

strong evidence, that the proposed transfer learning can effectively reduce computational burdens in reinforcement learning studies where complexity is systematically increased.

5.3.2 Schooling Hydrodynamics

Using the proposed transfer learning allows learning the optimal policy for a school with 100 swimmers. The results are shown in fig. 32. The vorticity is plotted together with the envelope of the motion of the swimmers and the swimming efficiency. We note that the motion is biased since the swimmer all start swimming in the same direction. The optimal policy that is found yields swimming on the diagonal, which is consistent with the observations in the previous section: swimmers in the back of a column are exposed to more disturbances that they can harvest in order to increase their efficiency. The found efficiency distribution and the envelopes reflect this. While the leading swimmers, and the swimmers at the top and bottom edge have below-average efficiency, the swimmers in the center benefit from the arrangement in the school. Interestingly, this does not apply to all swimmers, and the ones in the top right corner, that have many fish in front show lower efficiency and also a larger envelope. This indicates, that the increased disturbance implies higher probability of collisions. Thus the controller is mainly focusing on avoiding collisions and can not benefit from increased swimming efficiency.

An important difference to the situation in 3D is the form of the wake. Where in 2D, the swimming motion implies a reverse Von Karman vortex street, in 3D, the vortices are shed in a V-shaped pattern. In order to achieve the intersection with the eddies of the leader, the swimmer would prefer swimming straight. Although RL demonstrates success in the simplified 2D model, the transfer of the knowledge to 3D is not straightforward and requires control to maintain the positions that were already discussed in previous publications [56].

6

DISCUSSION AND OUTLOOK

The belief that there is only one truth and that oneself is in possession of it seems to me the root of all the evil that is in the world.

— Max Born

Fluid-Structure Interaction in Incompressible Flows

In the first chapter, we presented a two-dimensional incompressible flow solver on an adaptively refined mesh. Validation and verification results were shown for the flow past an impulsively started cylinder at several Reynolds numbers that are in excellent agreement with previous simulations and theoretical work. The solver is open source and available under `https://github.com/cselab/CUP2D`. It provides a reliable tool for making distributed, large-scale AMR simulations for machine learning applications.

Bayesian Optimal Experimental Design

In the second chapter, we presented a Bayesian approach to optimal experimental design, which is implemented open source in the high-performance korali framework under `https://github.com/cselab/korali`. After presenting the underlying formalism, we applied this method in two scenarios: The optimal sensor placement for schooling swimmers and the optimal allocation of limited test resources. These applications demonstrate the potential of Bayesian optimal experimental design to guide measurement execution in order to improve the collection of data to calibrate model parameters.

Optimal Sensor Placement for Schooling Swimmers

We presented a study of the optimal sensor locations on a self-propelled swimmer for detecting the size and location of a leading group of swimmers. This optimization combined Bayesian experimental design with large scale simulations of the two dimensional Navier–Stokes equations. Mimicking the function of sensory organs in real fish, we used the shear stress and pressure gradient on the surface of the swimmers to determine the sensor feedback generated by a disturbance in the flow field. The optimization was performed for different configurations of swimmers, ranging from a simple leader-follower configuration with two swimmers, to a group of up to eight swimmers leading a single follower. We considered two types of information: the number of swimmers in the leading group and the relative location of the leading group. We find that, although the general shape of the utility function varies between the two objectives, the preferred location of the first sensor on the head of the swimmer is consistent. Furthermore, we find that the objective is only weakly influenced when varying the number of members in the leading group. We performed a sequential sensor placement and found that the utility converges to a constant value and thus concluded that few sensors suffice to infer the quantities of the surrounding flow. Indeed, we found that the optimal sensor locations correspond to a posterior distribution that is strongly peaked around the true value of the quantity of interest. In summary, we found, that for the group sizes under examination, changing the number of swimmers in the leading group does not influence the follower's ability to infer the mean school location. Furthermore, we were able to show that choosing the locations for the measurements in a systematic way we are able to infer the number of swimmer in the leading group and the location of our agent to high accuracy. We envision that the presented methodology can provide guidance in developing autonomous systems of schooling artificial swimmers. While biological organisms have distinct flow fields from those examined in the present two-dimensional simulations, we believe that the algorithms presented here in can be extended to 3D flows. Moreover, while we draw a distinction between fish and the studied artificial swimmers, we note the capability of identifying neighboring swimmers using shear and pressure information on the body of the swimmers, indicating sufficiency of such type of information for flow sensing.

Optimal Allocation of Limited Test Resources

We introduced a systematic approach to identify optimal times and locations for epidemiological surveys to quantify infectious individuals in a country's population during the COVID-19 epidemic. The proposed OPALITS methodology exploits prior information and available data to maximise the expected information gain in quantities of interest and to minimise uncertainties in the forecasts of epidemiological models. The study addressed the need for an accurate assessment of COVID-19 infections [242] and it is shown to be far more accurate than the currently applied random testing. The proposed methodology was, to the best of our knowledge, the first method to propose an optimal spatiotemporal allocation of limited test kit resources. A first study of the estimation of unobserved COVID-19 infections [243] in the USA indicated that early testing would have decreased the surveillance gap during a critical phase of the epidemic. After that, a number of studies have emerged that address the optimal allocation of resources. The “Test and Contain” process suggested in [244] addresses an idealised population of 10,000 and solves an allocation problem using predictions of the SIR model. They assume isolation of the positively identified individuals and showed that just one test a day can reduce the peak of infected individuals by 27%. This study is similar to ours in casting the test allocation problem in an optimisation framework, using linear programming in contrast to information maximisation that we propose. However, their approach is not data informed and does not address a realistic country scenario. Another study [245] focused on test kit allocation in the Philippines. They use a statistical approach and non-linear programming to determine the optimal percentage allocation of COVID-19 test kits among accredited testing centres in the Philippines, aiming for an equitable chance for all infected individuals to be tested. Their goal of optimal percentage allocation differs from ours, which is optimal space and time allocation of test kits. The proposed method is demonstrated by focusing on the outbreak of the epidemic in Switzerland. We compare OPALITS with random testing and demonstrate its advantages in producing forecasts with far reduced uncertainties. We note that the existing testing capacity of 1500 tests per million people in Switzerland can be better allocated than the ongoing random testing. Moreover we show that the present methodology will be of particular importance to countries with testing capacity that is far lower than that of Switzerland [246]. The methodology relies on Bayesian experimental design using prior information and available data of reported infections along with forecasts from the SEI^rI^uR model. We compute the optimal testing strategy for three phases of the epidemic. At the onset of the epidemic the method identifies the most crucial dates and locations for randomised tests in the country’s population. The deployment of OPALITS at this phase would have allowed authorities to perform randomised testing in a period of high uncertainty, well in advance of the disease outbreak. Moreover, the presented approach is applicable to any newly arising epidemic and can be used to identify important surveying locations and a general protocol of action, whenever an unknown disease starts to spread. In the case of COVID-19, such course of action would limit early inaccurate estimates of metrics such as the virus mortality rate, estimated around 3% in early March 2020 by the World Health Organization [247] and currently believed to be lower than 1% [248]. During the period of nonpharmaceutical interventions, the proposed strategy would help quantify their effectiveness, assisting decision-making for further interventions or retraction of measures that may be harmful to the economy. In this study, available data for the daily reported infections prior to any interventions, combined with the proposed methodology, indicated that conducting two surveys after measures are imposed is sufficient. This can help to identify the new virus dynamics quickly and adjust interventions accordingly. Similarly, the OPALITS can assist monitoring for a recurrence of the disease after preventive measures have been relaxed and help guide further planning of interventions. Since massive testing for a new disease might not be a possibility during its first outbreak and cheap individual tests might become available only later, applying the proposed methodology at this point provides a useful guideline on how to use the individual tests to conduct large-scale surveys. For instance, in Switzerland it was not before mid-April 2020 that rapid COVID-19 tests were released on the market [249]. Collecting data for the reported cases before that and using it to inform the proposed approach to find an OPALITS (after cheap individual tests become available) that will be applied during a possible lockdown would be the suggested course of action in this case. There are a number of issues that the model should be able to accommodate in the future. These include accounting for virological test sensitivity, delays in the reporting of the test results and bias in the estimate of the unreported infected individuals (Cochran's formula). Further developments may include models that account for different transmission dynamics in cantons, and the classical Bayesian inference methods may be replaced with Hierarchical Bayesian Method to account for heterogeneous data. We remark that the proposed OPALITS does not depend on a particular type of data/model or to the country of Switzerland. The open source code is modular, scalable and readily adaptable to different scenarios for the epidemic and countries around the world. We believe that the present work can be a valuable tool for decision makers to allocate resources efficiently for testing the population, providing a reliable quantification of the spread of the disease and designing effective interventions. Finally the accurate estimation of the spread of the disease can guide the timely distribution of vaccines.

Deep Reinforcement Learning

In the third chapter we presented some advancements in deep reinforcement learning. In particular we extended the ReF-ER algorithm to multiple agents and introduced a Baysian approach to DRL. The implementation of this methods is available open source in the high-performance korali framework under `https://github.com/cselab/korali`.

Remember and Forget Experience Replay for Multi-Agent Reinforcement Learning

We present ReF-ER MARL, the multi-agent generalization of the V-RACER algorithm with ReF-ER [33]. The combination of V-RACER with ReF-ER has shown significant promise in several benchmark problems and challenging fluid dynamics applications. In the proposed ReF-ER MARL the actions of the agents are independent while the probability distribution of an action is conditioned on the respective agent's state. We examine the effects of different relations between the agents and vary the strength of their interactions. This is achieved by modifying the value estimator and the importance weight. For the value estimator we distinguish an individual and a cooperative setting by either using the individual reward or the average of the rewards from all agents. The strength of the agents' interaction is controlled via the importance weight. We benchmark ReF-ER MARL on the Stanford Intelligent Systems Laboratory (SISL) Environments. We compare the value-estimates, and importance weights for different assumptions. In these collaborative environments one may expect that assuming a strong interaction and cooperative estimates for the state-value is beneficial. However, we find that the preferred approach is to estimate the value using individual rewards, and to consider a local dynamics model. We find that ReF-ER MARL using a single feed forward neural network outperforms the state of the art algorithms, that often relies on complex network architectures. Finally, we test the proposed algorithms when training multiple policies. The median returns are lower during training. During testing, one controller per agent outperforms a shared controller. Besides robustness, we show that introducing stricter dependencies between the agents via the full dynamics (FDCo) cures the deadly combination of non-stationarity and reward averaging. Furthermore, the LDI with multiple policies is applied to enforce coordinated swimming on 20 swimmers experiencing hydrodynamic interactions in high-fidelity simulations of the Navier-Stokes equation. It enables stable schooling formations for up to 100 tail-beat periods (the formation brakes up after 20 tail beats without control).

A Bayesian Perspective on Uncertainties in Deep Reinforcement Learning

We incorporated Bayesian inference for model-free actor-critic off-policy reinforcement learning algorithms. We have first analyzed the testing and training performance of different samplers. We then compare the uncertainties that were found, and tested the detection of out-of-distribution samples. One of the main caveats is the larger computational cost of the method, which makes it challenging on benchmark environments that are traditionally employed to test new methods. However, we note that computationally challenging applications from science and engineering can be a valuable tool to assess the uncertainties after training the model. In future work, it would be interesting to expand the scope of the present work from ReF-ER to a broader benchmark with different RL algorithms and also to benchmark the Gaussian approximations. Furthermore, one could include more elaborate sampling algorithms to ensure that the samples are a good representation of the posterior. With appropriate samples of the posterior at hand, the present method enables model comparisons and select states, actions, and rewards in a systematic way. We also envision developing a new method by taking inspiration from CMA-ES: Following the ideas of SWAG, we could approximate the posterior distribution from samples along the SGD trajectory, however instead of recomputing the approximation it could be adopted along the lines of CMA-ES.

Controlling Artificial Swimmers using Deep Reinforcement Learning

In the last chapter of this doctoral thesis we accumulate the work presented in the previous chapters and apply the method in order to advance the understanding of natural behaviour of swimmers. In a first part we examined the swimming behaviour behind bluff bodies by leveraging multi-task reinforcement learning to swim in reverse and forward Kármán wakes. The second part uses transfer learning for schooling swimmers with multi-agent reinforcement learning.

Multi-Task Reinforcement Learning to Swim in Reverse and Forward Kármán Wakes

In the present work, we successfully train artificial swimmers to maximize the Froude swimming efficiency. The training is performed in a multi-task setting. For a given obstacle, a continuous parameter that determines the flow is varied. We furthermore change the shape of the obstacles. Our results generalizes previous studies that used RL to understand the behaviour of swimmers. We find that the multi-task approach allows saving computational resources compared to state-of-the-art approaches that usually sample discrete realizations or learn only one task at a time. The underlying direct numerical simulations allow studying the flow field in detail and assess the mechanisms that propel the fish which result in the reduced energy expenditure that was observed in experiment [195]. Using multi-task RL, we can also compare the mechanisms across obstacles, giving an unifying picture on the flow physics exploited by fish in nature. The found policies show intriguing similarities to past experimental studies on the behaviour of swimmers behind obstacles. This shows that RL is suitable to understand natural behavior [205]. It can readily be combined with experiments to uncover rewards that are compatible with the observed behaviour for a multitude of scenarios [197, 238] by employing inverse reinforcement learning [250].

Transfer Learning for Schooling Swimmers with Multi-Agent Reinforcement Learning

In this study, we comprehensively analyzed the impact of reward function choice on a column of swimmers, comparing minimizing lateral displacement and maximizing swimming efficiency with and without control over the leader swimmer. Our findings revealed valuable insights into multi-agent swimming dynamics. Minimizing lateral displacement resulted in higher mean displacement for rear swimmers, supporting previous studies on the influence of flow disturbances. Maximizing swimming efficiency led to higher efficiency for rear swimmers, indicating their ability to harness vortices shed by the leader. However, pursuing maximum efficiency disrupted the column structure, causing larger lateral displacements that stabilized at the domain edges. This was attributed to swimmers reducing their speed to optimize efficiency, making a smaller school near the boundary beneficial. To explore sensitivity to potential actions, we conducted two additional studies. Controlling all swimmers' speed led to collisions, underscoring the importance of speed control for avoiding collisions and achieving optimal policies. Optimizing both objectives without controlling the swimming period showed similar lateral displacement levels but significantly higher swimming efficiencies, suggesting the adoption of burst-coast swimming. Transfer learning demonstrated its effectiveness in transferring knowledge across different school sizes. Initializing policies with fewer swimmers and progressively increasing the number improved training efficiency and performance, surpassing cold start initialization. Transfer learning facilitated learning the optimal policy for a 100-swimmer school, resulting in line formation and increased efficiency for center swimmers. In conclusion, our study provides valuable insights into multi-agent swimming dynamics, highlighting the influence of reward function choice, swimming speed control, and burst-coast swimming strategies on behavior optimization. Transfer learning proves to be a powerful technique for efficiently transferring knowledge, enabling reinforcement learning in complex scenarios.

A

APPENDIX FOR CHAPTER BAYESIAN OPTIMAL EXPERIMENTAL DESIGN

A.1 Optimal Sensor Placement for Schooling Swimmers

A.1.1 Configurations

The configuration used for the "size of the leading school" experiment. For the configurations with three rows the vertical extent $y \in [0, 0.5]$ was discretized using 2048 gridpoints, for the ones with four rows it was extended to $y \in [0, 0.75]$ and discretized using 3072 gridpoints.

A.1.2 The posterior Is Not Symmetric

The estimated posterior in Figure 17b is not symmetric with respect to the $\vartheta_{\text{true}} = \vartheta$ diagonal. This observation indicates that the posterior is not symmetric with respect to an exchange of ϑ and ϑ_{true}, the parameter we try to infer and the one used in the simulation. Here, we want to show that this observation is true in general. In order to lighten the notation, we neglect the dependence of the distributions on the sensor location $\boldsymbol{s}$.

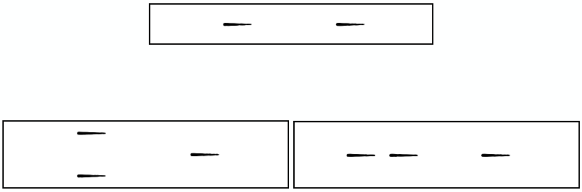


FIGURE 33: Configurations for two leading swimmers.

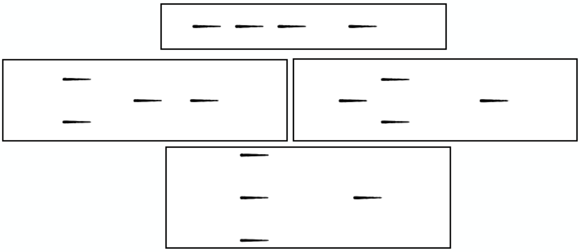


FIGURE 34: Configurations for three leading swimmers.

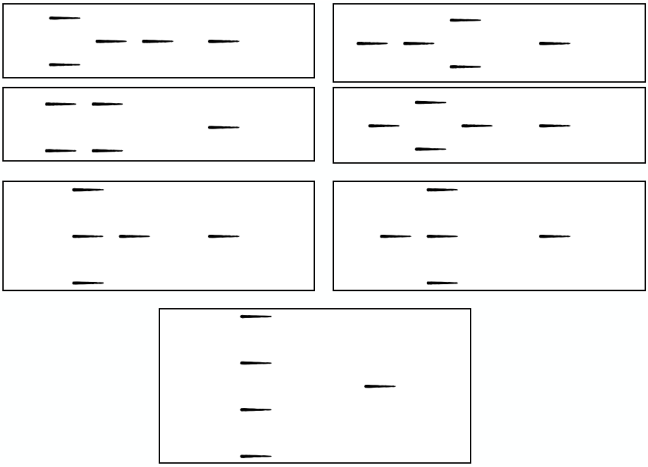


FIGURE 35: Configurations for four leading swimmers.

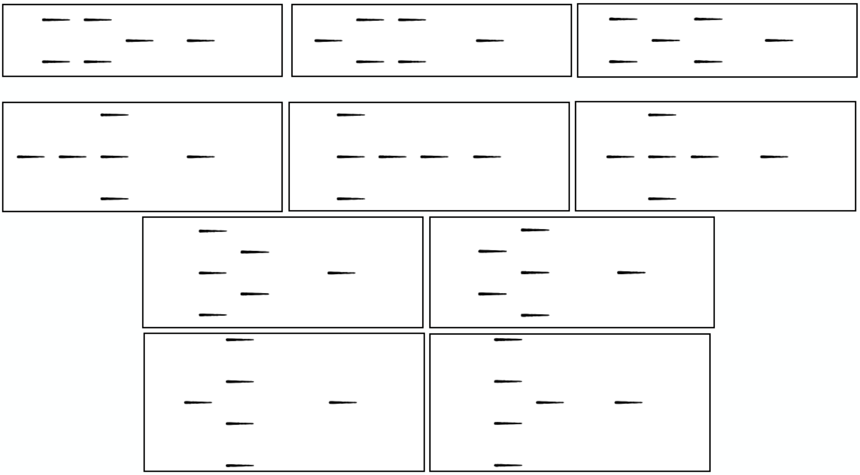


FIGURE 36: Configurations for five leading swimmers.

In section 4.1.2 we showed that the distribution of $\boldsymbol{\vartheta}_i$ conditioned on measurements $\boldsymbol{y}$, under the assumption of uniform prior, is proportional to

$$p(\boldsymbol{\vartheta}_i|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{\vartheta}_i)\, p(\boldsymbol{\vartheta}_i) = \frac{1}{8}\frac{1}{n_i}\sum_{\ell=1}^{n_i}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(\boldsymbol{y}-\mathbf{F}(\boldsymbol{\varphi}_{i,\ell}))^2}{2\sigma^2}\right). \tag{100}$$

We want to show that for any $i \neq j$,

$$p(\boldsymbol{\vartheta}_i|\boldsymbol{\vartheta}_j) = p(\boldsymbol{\vartheta}_i|\boldsymbol{y} = F(\boldsymbol{\varphi}_{j,k})) \neq p(\boldsymbol{\vartheta}_j|\boldsymbol{y} = F(\boldsymbol{\varphi}_{i,\ell})) = p(\boldsymbol{\vartheta}_j|\boldsymbol{\vartheta}_i)\,, \tag{101}$$

for any configurations $\boldsymbol{\varphi}_{j,k}$ and $\boldsymbol{\varphi}_{i,\ell}$ corresponding to a school of size $\boldsymbol{\vartheta}_j$ and $\boldsymbol{\vartheta}_i$, respectively. From Equation (100) it is easy to see that (101) is true due to the fact that

$$\frac{1}{n_i}\sum_{\ell'=1}^{n_i}\exp\left(-\frac{(\mathbf{F}(\boldsymbol{\varphi}_{j,k})-\mathbf{F}(\boldsymbol{\varphi}_{i,\ell'}))^2}{2\sigma^2}\right) \neq \frac{1}{n_j}\sum_{k'=1}^{n_j}\exp\left(-\frac{(\mathbf{F}(\boldsymbol{\varphi}_{i,\ell})-\mathbf{F}(\boldsymbol{\varphi}_{j,k'}))^2}{2\sigma^2}\right). \tag{102}$$

Finally, we note that in the case where we have only one configuration per group size, i.e., $n_i = 1$ for all i, the statement in (102) is not true and the posterior is symmetric.

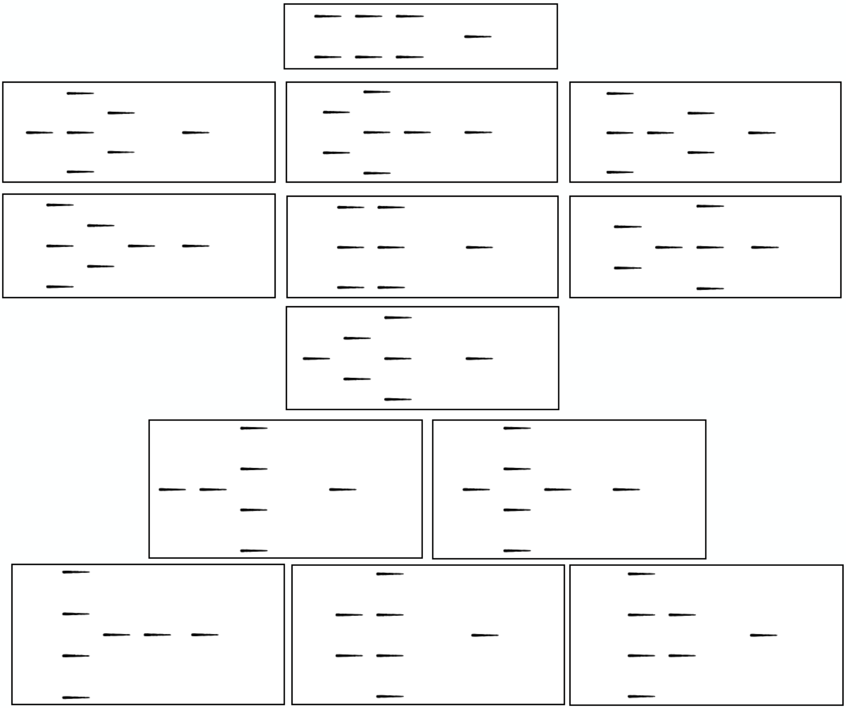


FIGURE 37: Configurations for six leading swimmers.

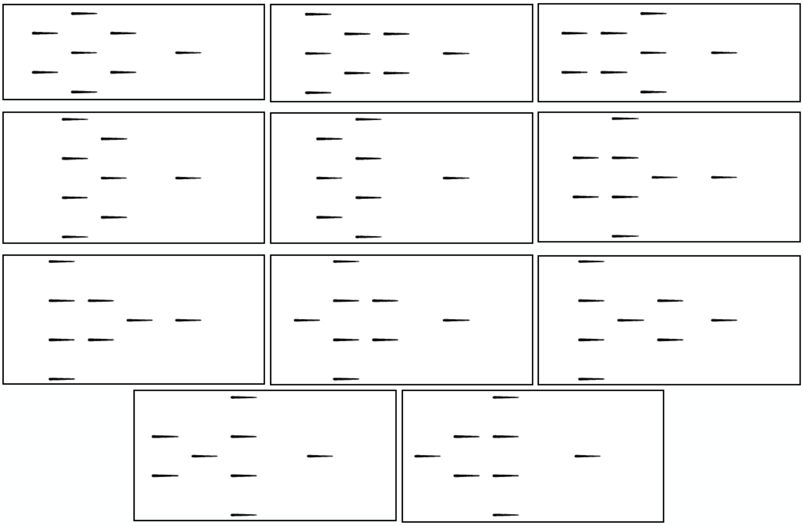


FIGURE 38: Configurations for seven leading swimmers.

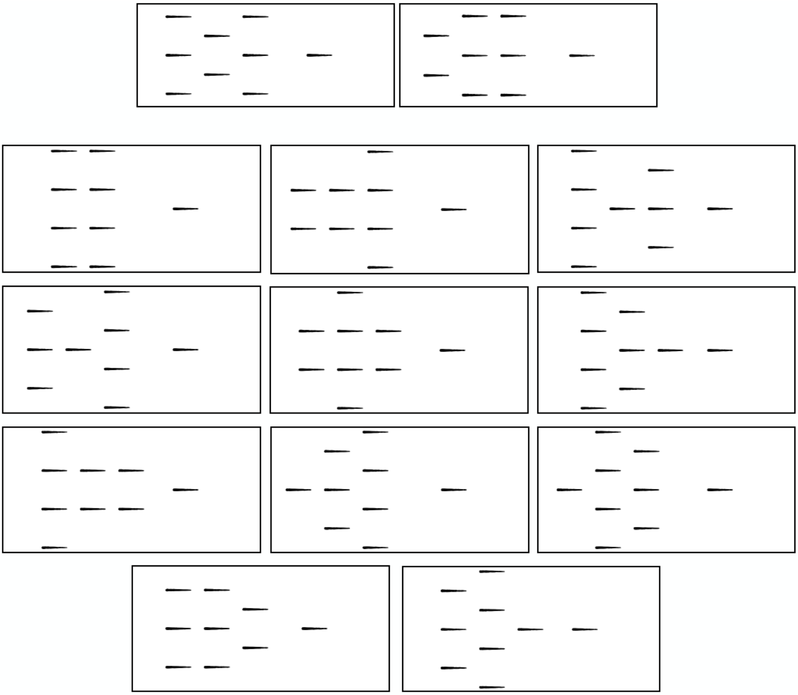


FIGURE 39: Configurations for eight leading swimmers.

A.2 Optimal Allocation of Limited Test Ressources

The systematic identification of infected individuals is critical for the containment of the COVID-19 pandemic. Presently, the spread of the disease is mostly quantified by the reported numbers of infections, hospitalizations, recoveries and deaths; these quantities inform epidemiology models that provide forecasts for the spread of the epidemic and guide policy making. The veracity of these forecasts depends on the discrepancy between the numbers of reported and unreported, yet infectious, individuals. We combine Bayesian experimental design with an epidemiology model and propose a methodology for the optimal allocation of limited testing resources in space and time, which maximizes the information gain for such unreported infections. The proposed approach is applicable at the onset and spreading of the epidemic and can forewarn for a possible recurrence of the disease after relaxation of interventions. We examine its application in Switzerland; the open source software is, however, readily adaptable to countries around the world. We find that following the proposed methodology can lead to vastly less uncertain predictions for the spread of the disease. Estimates of the effective reproduction number and of the future number of unreported infections are improved, which in turn can provide timely and systematic guidance for the effective identification of infectious individuals and for decision-making.

A.2.1 Introduction

The identification of unreported individuals infected by SARS-CoV-2 was critical for the quantification, forecasting and planning of interventions during the COVID-19 pandemic [251]. The spread of the disease is mostly quantified by the reported numbers of infections, hospitalisations, recoveries and deaths [252]. These quantities inform epidemiology models that provide short term forecasts for the spread of the epidemic, help quantify the role of possible interventions and guide policy making. The veracity of these forecasts depends on the discrepancy between the numbers of reported, and unreported yet infectious, individuals.

During the COVID-pandemic, the estimation of unreported infections has been the subject of several testing campaigns [253, 254]. Although there is valuable information being gathered, their estimates rely on testing individuals who are either already symptomatic or have been selected based on certain criteria (hospital visits, airport arrivals, geographic vicinity to researchers, etc.). Generic, randomised tests of the population are broadly applied, but they have been hampered either by delays [243] or by insufficient numbers of test kits [255]. There is broad recognition that efficient testing strategies are critical for the timely identification of infectious individuals and the optimal allocation of resources [244]. However, targeted testing entails bias and random tests require access to a high percentage of the population with commensurate high costs. The quality of the data, as well as the ways they are incorporated in the epidemiology models, is critical for their predictions and for estimating their uncertainties [256]. A way to minimise these uncertainties by suitably distributing in space and time a given number of test kits is the subject of this work. This optimal allocation of testing resources and the respective increase in the fidelity of forecasting models are essential to effective policy making throughout the pandemic.

Here, we present a methodology for the OPtimal Allocation of LImited Testing resourceS (OPALITS) that maximises the information gain over any prior knowledge regarding infections. The method relies on forecasts by epidemiological models with parameters adjusted through Bayesian inference as data become available through suitable surveys [257]. The forecasts are combined with Bayesian experimental design [123, 258, 259] to determine the optimal test allocation in space and time for various objectives (minimise prediction uncertainty, maximise information gain of unreported infections). We emphasise that the proposed OPALITS is applicable in all stages of the pandemic, regardless of the availability of data.

We employ the SEI^rI^uR model [260], which quantifies the spread of a disease in a country's population distributed in a number of communities that are interacting through mobility networks. The SEI^rI^uR model predicts the number of susceptible (S), exposed (E), infectious reported (I^r), unreported (I^u), and removed (R) individuals from the population. Here we focus on Switzerland and consider its cantons as the respective communities. The model parameters are: the relative transmission rate between reported and unreported infectious individuals (μ), the virus latency period (Z), the infectious period (D) and the reporting rate (α). The transmission rate (β) and the mobility factor (θ) are considered to be time dependent in order to account for government interventions. For all stages of the epidemic, the uncertainties of the model parameters are quantified and propagated using Bayesian inference. At the onset of the epidemic, the uncertainty was quantified through prior probability distributions. As data on daily infections become available, the uncertainty in model parameters was updated through Bayesian inference. The parameter probability distributions are used to propagate uncertainties in the model forecasts and can assist decision makers in quantifying risks associated with the progression of the disease. The proper quantification of uncertainty bounds in the model parameters has a profound effect on predictions of the disease dynamics [256]. Large uncertainty bounds around the most probable parameter values hinder the decision process for identifying effective interventions.

The OPALITS aims to assign limited test-kit resources to acquire data that would reduce the model prediction uncertainties. Minimising the uncertainty of the model parameters leads to more reliable predictions for quantities such as the reproduction number [261]. Moreover, the reduced model uncertainties help minimise risks associated with the decision-making process including timing, extent of interventions and probability of exceeding hospital capacity.

We quantify the information gain from these tests using a utility function [259, 262] based on the Kullback-Leibler divergence between the inferred posterior distribution and the current prior distribution of the model parameters. The prior can be formulated using the posterior distribution estimated from daily data of the infectious reported individuals up to the current date (see Materials and methods). Hence, at any stage of the epidemic, the OPALITS provides guidance on the time and location/community where testing needs to be carried out to maximise the expected information gain regarding infections in a population.

We demonstrate the simplicity and applicability of the present method in estimating the spread of the coronavirus disease in the cantons of Switzerland. We find that the OPALITS methodology outperforms non-specific, randomised testing of sub-populations throughout the COVID-19 pandemic. The proposed strategy is readily applicable to other countries and the employed open source software can readily accommodate different epidemiological models.

A.2.2 Epidemiological Model

Here we employ the SEI^rI^uR epidemiological model [260] to forecast the dynamics of the coronavirus outbreak in Switzerland

$$\begin{aligned}\frac{\mathrm{d}S_k}{\mathrm{d}t} = -\frac{\beta S_k I_k^r}{N_k} - \frac{\mu\beta S_k I_k^u}{N_k} \\ + \theta \sum_{l=1}^{K} \left(\frac{M_{kl} S_l}{N_l - I_l^r} - \frac{M_{lk} S_k}{N_k - I_k^r} \right)\end{aligned} \tag{103}$$

$$
\begin{aligned}
\frac{\mathrm{d}E_k}{\mathrm{d}t} &= \frac{\beta S_k I_k^r}{N_k} + \frac{\mu\beta S_k I_k^u}{N_k} - \frac{E_k}{Z} \\
&\qquad + \theta \sum_{l=1}^{K} \left(\frac{M_{kl} E_l}{N_l - I_l^r} - \frac{M_{lk} E_k}{N_k - I_k^r} \right)
\end{aligned}
\tag{104}
$$

$$
\frac{\mathrm{d}I_k^r}{\mathrm{d}t} = \alpha \frac{E_k}{Z} - \frac{I_k^r}{D} \tag{105}
$$

$$
\frac{\mathrm{d}I_k^u}{\mathrm{d}t} = (1-\alpha)\frac{E_k}{Z} - \frac{I_k^u}{D} + \theta \sum_{l=1}^{K} \left(\frac{M_{kl} I_l^u}{N_l - I_l^r} - \frac{M_{lk} I_k^u}{N_k - I_k^r} \right) \tag{106}
$$

$$
\frac{\mathrm{d}N_k}{\mathrm{d}t} = \theta \sum_{l=1}^{K} (M_{kl} - M_{lk}), \tag{107}
$$

where S_k, E_k, I_k^r and I_k^u denote the number of individuals in canton $k = \{1, \dots, K\}$ that are susceptible, exposed, reported infectious and unreported infectious, respectively. We denote by K the number of cantons (26 in Switzerland), by N_k the total population of the canton k, while the population mobility between cantons k and l is denoted by M_{kl} with values obtained from the Swiss Federal Statistical Office [263]. The model parameters are the transmission rate (β), the relative transmission rate between reported and unreported infectious individuals (μ), the virus latency period (Z), the infectious period (D), the reporting rate (α) and the mobility factor (θ).

We employ different time-dependent expressions for the transmission rate and the mobility factor for each stage of the epidemic. Constants are chosen for the start of an epidemic while in the cases of monitoring of interventions, the following expressions are used:

$$
\beta(t) = \begin{cases} b_0, & t \le \delta_1 \\ b_1, & \delta_1 < t \end{cases}, \quad \theta(t) = \begin{cases} \theta_0, & t \le \delta_1 \\ \theta_1, & \delta_1 < t \end{cases}, \tag{108}
$$

where b_0, b_1, θ_0 and θ_1 are the transmission rates and mobility factors before and after the intervention. Time $t = 0$ corresponds to the 25th of February 2020, and $\delta_1 = 21$ to the 17th of March 2020, when the lockdown was announced in Switzerland [264]. Finally, for the third case (monitoring of a second outbreak) we assume that

$$\beta(t) = \begin{cases} b_0, & 0 \leq t \leq \delta_1 \\ b_1, & \delta_1 < t \leq \delta_2 \\ b_2, & \delta_2 < t \leq \delta_3 \\ b_3(t), & \delta_3 < t \end{cases}, \quad \theta(t) = \begin{cases} \theta_0, & 0 \leq t \leq \delta_1 \\ \theta_1, & \delta_1 < t \leq \delta_2 \\ \theta_2, & \delta_2 < t \leq \delta_3 \\ \theta_0, & \delta_3 < t \end{cases}. \tag{109}$$

As in eq. (125), b_0 is the transmission rate before the intervention while $b_1 = c_1\, b_0$ and $b_2 = c_2\, b_0$ with $c_1, c_2 \in [0, 1]$ are the transmission rates after the two interventions. Similarly, θ_0 is the mobility factor before any interventions took place, while $\theta_1 = c_3\, \theta_0$ and $\theta_2 = c_4\, \theta_0$ with $c_3, c_4 \in [0, 1]$ are the mobility factors after the two interventions. Moreover, δ_1 and δ_2 correspond to the days of the interventions. The day when the measures are loosened is denoted by δ_3. After that day, the transmission rate is gradually increasing

$$b_3(t) = \min(b_2 + \lambda(t - \delta_3), b_0)\,, \tag{110}$$

with $\lambda \in [0, 0.03]$, while the mobility factor regains its initial value of θ_0. In the present work, the assumed nuisance parameters are the correlation time τ and the initial condition of the unreported infections in the cantons of Aargau, Bern, Basel-Landschaft, Basel-Stadt, Fribourg, Geneva, Grisons, St.Gallen, Ticino, Vaud, Valais and Zurich $\boldsymbol{I}^u_{\mathrm{IC}} = (I^u_{\mathrm{AR}}, I^u_{\mathrm{BE}}, I^u_{\mathrm{BL}}, I^u_{\mathrm{BS}}, I^u_{\mathrm{FR}}, I^u_{\mathrm{GE}}, I^u_{\mathrm{GR}}, I^u_{\mathrm{SG}}, I^u_{\mathrm{TI}}, I^u_{\mathrm{VD}}, I^u_{\mathrm{VS}}, I^u_{\mathrm{ZH}})$, with prior distributions $\boldsymbol{I}^u_{\mathrm{IC}} \sim \mathcal{U}([0, 50]^{12})$ and $\tau \sim \mathcal{U}([0.5, 3.5])$.

A.2.3 Optimal Testing

We consider a testing campaign including a set ($\boldsymbol{s}$) of surveys $s_i = (k_i, t_i), i = 1, \ldots M_y$ performed in location $k_i \in \mathcal{C}$ and on day $t_i \in \mathcal{T}$. These surveys measure a quantity of interest (QoI), that is denoted by $\boldsymbol{y}(\boldsymbol{s}) = (y_1, \ldots, y_{M_y})$. Here, y_i is the number of unreported infectious individuals, measured through survey s_i. The QoI can be predicted by a model $\boldsymbol{g}(\boldsymbol{s}, \boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}})$ (here the SEI^rI^uR epidemiological model) that depends on parameters of interest $\boldsymbol{\vartheta} \in \mathbb{R}^N$ and nuisance parameters $\widetilde{\boldsymbol{\vartheta}} \in \mathbb{R}^{\tilde{N}}$. We note that both sets of parameters are uncertain and the proposed method aims to reduce the uncertainty only in the parameters of interest. In the present study, the QoI measured by a survey is the number of unreported infectious individuals in a particular canton on a particular date. This implicitly assumes that there no restrictions on when the survey can be conducted and that there are no observational delays, which means the the QoI is instantaneously obtained. Both assumptions are not restrictive however. Restrictions on the possible survey dates can be accounted for by simply excluding those dates from the dates on which the utility function is evaluated. Also, a delay of one day (meaning that two days are needed to survey a canton k, starting from day t) would mean that $\boldsymbol{y} = (I_k^u(t) + I_k^u(t+1))/2$ is measured. In other words, when there is a delay the measured quantity can still be mapped to a model quantity, which allows us to perform Bayesian inference. There are several types of measurements (Rapid testing [265], PCR [266], Schwabs [267]) being proposed for testing asymptomatic individuals. We emphasize that our methodology is compatible with any of these types. Data related issues such as uncertainties, test sensitivities and delays in processing can be accommodated in the Bayesian inference framework and in the input to the SEIR model.

A.2.4 Error Model

According to eq. (81) the QoI is linked to the model prediction via an Gaussian error term. The elements of the covariance matrix $(\Sigma_{s,s'})$ correspond to surveys taken at $s = (k, t)$ and $s' = (k', t')$ and are given by

$$\Sigma_{s,s'} = \sigma_t \, \sigma_{t'} \; \exp\left(-\frac{|t - t'|}{\tau} \right) \; \delta_{kk'} \, , \tag{111}$$

where $\delta_{kk'}$ is the Kronecker delta, which is 1 for $k = k'$ and 0 otherwise. The correlation time $\tau \in [0.5, 3.5]$ is considered a nuisance parameter. These assumptions about the covariance imply that surveys in different locations are not correlated, while those in the same location have an exponentially decaying temporal correlation. The latter avoids clustering of surveys in small time intervals [268, 269]. The factor $\sigma_t \in \mathbb{R}$ is assumed proportional to the expectation of the QoI, taken over all possible survey locations and over the range of model and nuisance parameters

$$\sigma_t = c \, \frac{1}{K} \sum_{i=1}^{K} \mathbb{E}_{\boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}}} \left[g(s_i, \boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}}) \right] , \tag{112}$$

where $s_i = (i, t)$. The parameter $c \in [0, 0.25]$ is considered a model parameter. The expectation $\mathbb{E}_{\boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}}}[\,\cdot\,]$ is taken with respect to all parameters $\boldsymbol{\vartheta}$ and $\widetilde{\boldsymbol{\vartheta}}$ that follow the prior probability distribution with density $p(\boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}}) = p(\boldsymbol{\vartheta}) p(\widetilde{\boldsymbol{\vartheta}})$.

A.2.5 Informed Priors

A data informed prior $p(\boldsymbol{\vartheta}|\boldsymbol{d})$ of the model parameters $\boldsymbol{\vartheta}$ can be computed from available data $\boldsymbol{d} = (d_1, \dots, d_{M_d})$, collected at M_d locations and days. Here, available data $\boldsymbol{d}$ refer to the daily number of reported infectious individuals and they are contrasted from the data $\boldsymbol{y}$ of the number of unreported infectious individuals. The latter are obtained from testing strategies at selected populations using optimal experimental design. The data is mapped via a distinct model output $f(s, \boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}})$ through the following error model

$$p(d_i|\boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}}, \nu) = \mathcal{NB}\left(d_i \mid f(s_i, \boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}}), \nu\right) \tag{113}$$

where $\mathcal{NB}$ is the negative binomial distribution with mean f and dispersion ν. Also, $s_i = (k_i, t_i)$ is the location and time the data d_i for $i = 1, \dots, M_d$ was collected. The choice of a different error model, compared to equation 81, is based on the assumption that the data are independent and identically distributed. Such an assumption would not be acceptable in the measurement model in equation 81, as it may result in uncorrelated measurements that can become clustered in small time intervals [268, 269].

The data $\boldsymbol{d} = (d_1, \dots, d_{M_d})$ are the daily number of reported infections per canton in Switzerland [270] which corresponds to the following model quantity

$$f(s_i, \boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}}) := \int_{t_i-0.5}^{t_i+0.5} \frac{\alpha}{Z} E_{k_i}(\tau) d\tau \approx \frac{\alpha}{Z} E_{k_i}(t_i) \,. \tag{114}$$

The posterior distribution that will be used subsequently as a data informed prior is obtained using Bayes' theorem

$$p(\boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}}|\boldsymbol{d}) = \frac{p(\boldsymbol{d}|\boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}})\, p(\boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}})}{p(\boldsymbol{d})} \,, \tag{115}$$

and is sampled with a nested sampling algorithm [261]. Note the difference to equation 121 and the optimal testing methodology, where we are interested to reduce the uncertainty in $p(\boldsymbol{\vartheta}|\boldsymbol{y}, \widetilde{\boldsymbol{\vartheta}}, \boldsymbol{s})$, which excludes the nuisance parameters $\widetilde{\boldsymbol{\vartheta}}$. For the dispersion parameter in equation 133, it is assumed that $\nu = r\, f(s_i, \boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}})$. The coefficient r is unknown and included in the parameter set, where $r \sim \mathcal{U}([0, 2])$.

The three inferences performed are summarized in table S5, which shows the involved model parameters in each case. The histograms for the found samples are shown in figures S1, S2, and S3.

We remark that, using the present methodology, the inferred date for the beginning of the intervention is $\delta_1 = 22.5$, which is the 18th of March 2020,

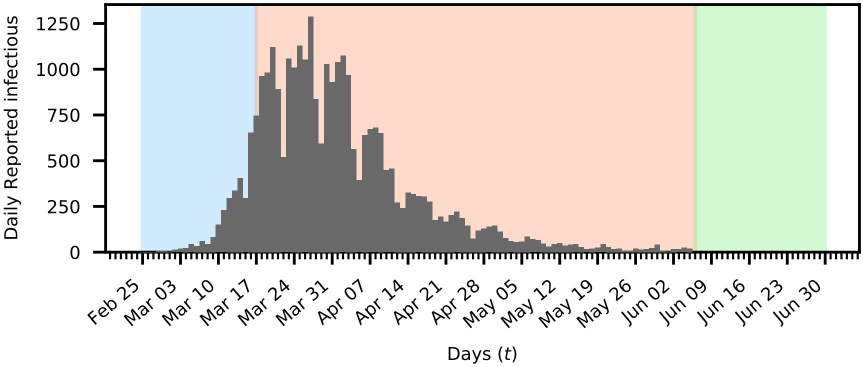


FIGURE 40: **Testing scenarios for the COVID19 outbreak in Switzerland.** Daily reported Coronavirus cases in Switzerland are plotted as gray bars. The period before (blue), during (red) and after (green) imposing non-pharmaceutical interventions are marked with color.

corresponding well with the 17th of March 2020 on which the lockdown was introduced in Switzerland [264]. Moreover, we infer a significant reduction in the mobility factor, which indicates that traffic between cantons was also minimized. For the inference III we plot the fit using the inferred parameters in figure S4. The daily reported cases per canton are shown, together with the data used for the inference.

A.2.6 Results

We present the optimal test-kit allocation strategy for three stages of the epidemic: (i) starting phase (blue), (ii) containment after enforcement of interventions (red) and (iii) relaxing of interventions and monitoring for a possible second outbreak (green) (fig.40). The strategy relies on Bayesian experimental design and can operate when no data are available (as in the start of the epidemic), as well as when data have been accumulated, as in the last two stages of the epidemic. Testing campaigns rely on acquiring randomized samples from a population. The collected data, together with epidemiological models, help determine quantities of interest, such as the basic reproduction number of the disease [261]. By suitably adapting the testing campaign, the data can help to reduce the model uncertainty, thus enabling improved estimates regarding the severity of the epidemic.

A testing campaign consists of a set $\boldsymbol{s}$ of surveys $s_i = (k_i, t_i)$ which are labeled by $i = 1, ... M_y$ and performed in locations $k_i \in \mathcal{C}$ and on days $t_i \in \mathcal{T}$, where $\mathcal{C}$ and $\mathcal{T}$ are the set of all available locations and days, respectively. In this work, a survey aims to determine the number of unreported infectious individuals in a particular location on a particular day. In the following we assume limited testing resources, where N test kits are available and each test kit corresponds to testing one person. The goal is to allocate these test kits in different times and locations so that we maximise the information gain regarding forecasts of the epidemiology model. The locations are the different Swiss cantons, and $\mathcal{C} := \{\text{ZH}, \text{BE}, \text{LU}, ...\}$ is the set of the strings with canton name abbreviations.

The results of the survey in a canton enable the estimation of a desired quantity of interest, such as the size of the unreported infected population (I^u). The number of samples needed to estimate population proportions within a given confidence interval, error tolerance, and probability of proportion is given by Cochran's formula [257] corrected for a finite population size. Using Cochran's formula with confidence level 99%, error tolerance 1% and probability of infection 0.1 we find that the samples that would be required to survey the largest Swiss canton (Zurich) are approximately 5950. All the other cantons need up to 14% fewer samples, with the exception of the smallest canton that needs 27% fewer samples (fig. 53). Hence we assume the minimum sample size is the same for all cantons. Assuming random sampling of a population with higher probability (up to 0.9) of infection or requiring tighter error bounds would have implied even more samples according to Cochran's formula. We note that as of October 2020, 1500 tests per one million people are performed on a daily basis in Switzerland [246]. This amounts to approximately 460 individual tests per canton, which is about an order of magnitude less than what would be required from Cochran's formula for informative random sampling. In turn, by using the proposed OPALITS, we can compensate for this lack of test kits with an optimal and systematic process.

We outline the application of the proposed approach to a country with distinct administrative units (cantons in the case of Switzerland) (see figure 41). First, we determine how many cantons will be surveyed, given the number of available test-kits N. Then, the sequential optimisation of the expected utility function is performed (see Materials and methods) to identify optimal survey locations (cantons). We then distribute the test kits to the identified cantons and test a random subset of their population on the suggested day. After collecting the results from all the surveys we update the prior distributions

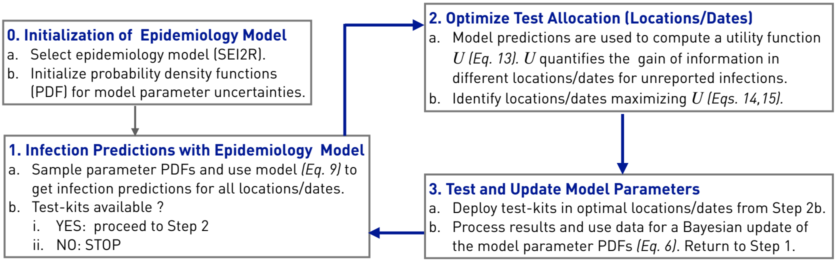

0. Initialization of Epidemiology Model
a. Select epidemiology model (SEI2R).
b. Initialize probability density functions (PDF) for model parameter uncertainties.
1. Infection Predictions with Epidemiology Model
a. Sample parameter PDFs and use model (Eq. 9) to get infection predictions for all locations/dates.
b. Test-kits available ?
i. YES: proceed to Step 2
ii. NO: STOP
2. Optimize Test Allocation (Locations/Dates)
a. Model predictions are used to compute a utility function U (Eq. 13). U quantifies the gain of information in different locations/dates for unreported infections.
b. Identify locations/dates maximizing U (Eqs. 14,15).
3. Test and Update Model Parameters
a. Deploy test-kits in optimal locations/dates from Step 2b.
b. Process results and use data for a Bayesian update of the model parameter PDFs (Eq. 6). Return to Step 1.

FIGURE 41: Schematic for the deployment of the Optimal Allocation of Limited Testing Resources (OPALITS) methodology

of the model parameters. The collected data lead the maximal information gain in the model parameters. This in turn translates into minimal uncertainty in predictions made with the model for quantities, such as the number of unreported infections.

The expected information gain of a particular strategy for selecting the survey locations/times $\boldsymbol{s}$ is quantified by a utility function $\hat{U}(\boldsymbol{s})$ [262]. The maximum of this function corresponds to an optimal strategy that yields the most information about the quantities of interest. The expected utility function can be understood as a measure of the difference between prior knowledge of the model parameters and the posterior knowledge, after surveys have been conducted in a set of locations and dates. Given such a set, the utility function estimates the expected difference, the equivalent information gain, by taking the expectation over all possible survey results.

The OPALITS relies on forecasts by suitable epidemiological models. In turn, these forecasts rely on prior information and their predictions are further adjusted as data become available in a Bayesian inference framework [271]. The set of ordinary differential equations (ODEs) describing the $SEI^{r}I^{u}R$ model [260] are integrated to produce the model output. The uncertainty of the model output and its discrepancy from the available data is quantified through a parametrised error model. The resulting stochastic model and its quantified uncertainties are then used to identify the optimal spatiotemporal allocation of limited test resources.

A.2.6.1 Case 1: Beginning of the epidemic – optimal testing without data

At the start of an epidemic, there are no data and we assume no other prior information regarding the spread of the pathogen in a country. The initial conditions for the number of unreported infections ($\boldsymbol{I}^{u}_{\text{IC}}$) were selected with non-zero values for the cantons of Aargau, Bern, Basel-Landschaft, Basel-Stadt, Fribourg, Geneva, Grisons, St Gallen, Ticino, Vaud, Valais and Zurich based on their populations and their large number of interconnections. Because of the lack of any prior information and relevant data, all the parameters are assumed to follow uniform prior distributions (see table S5 in appendix 1 for details).

The first infectious person in Switzerland was reported on 25 February in the canton of Ticino ($I^{r}_{\text{TI}} = 1$) with no initial reported infections in any other canton. The initial number of exposed individuals is set proportional to the number of unreported infections $E_k = 3I^{u}_{k}$ in accordance with the value of $R_0 \approx 3$ reported in [272] in the initial stage of the disease. The rest of the population is assumed to be susceptible. The methodology involves parameters of interest ($\boldsymbol{\vartheta} = (\beta, \mu, \alpha, Z, D, \theta, c)$) and nuisance parameters ($\widetilde{\boldsymbol{\vartheta}} = (\boldsymbol{I}^{u}_{\text{IC}}, \tau)$) that the testing strategy does not aim to determine (see Materials and methods section for definitions).

The estimated expected utility functions $\hat{U}(\boldsymbol{s})$ for up to four surveys in the cantons of Switzerland for a time horizon of 8 days is shown in Figure 42, $\mathcal{T} = \{\text{Feb } 25, \dots, \text{Mar } 3\}$. Higher values for expected utility are estimated in cantons with larger population, reflecting the larger relative uncertainty for cantons with only few reported cases. This implies that smaller cantons, with lower mobility rates, are less preferred for performing tests since their contribution to the information gain is not significant. This reflects the fact that the assumed covariance matrix is shared among cantons (see Materials and methods). This implies a smaller relative error when surveying larger cantons with consequently higher number of infections. The Bayesian analysis allows the inference of the particular cantons and days on which a survey should be performed in order to maximise the information gain. Accordingly, the most informative survey should have been made in Zurich on 2 March. The optimal location and time for the second survey is determined to be canton of Vaud on the 27 February. As expected, the information gained from tests in the canton of Vaud is less than the information gained from the canton of Zurich. The information that would have been gained by surveying the next two selected cantons of Vaud and Basel-Landschaft on 3 March and 28 February, respectively, is progressively reduced to a small level that, given the testing costs, does not justify carrying out surveys in more than four cantons. The values of the optimal times are listed in table 5.

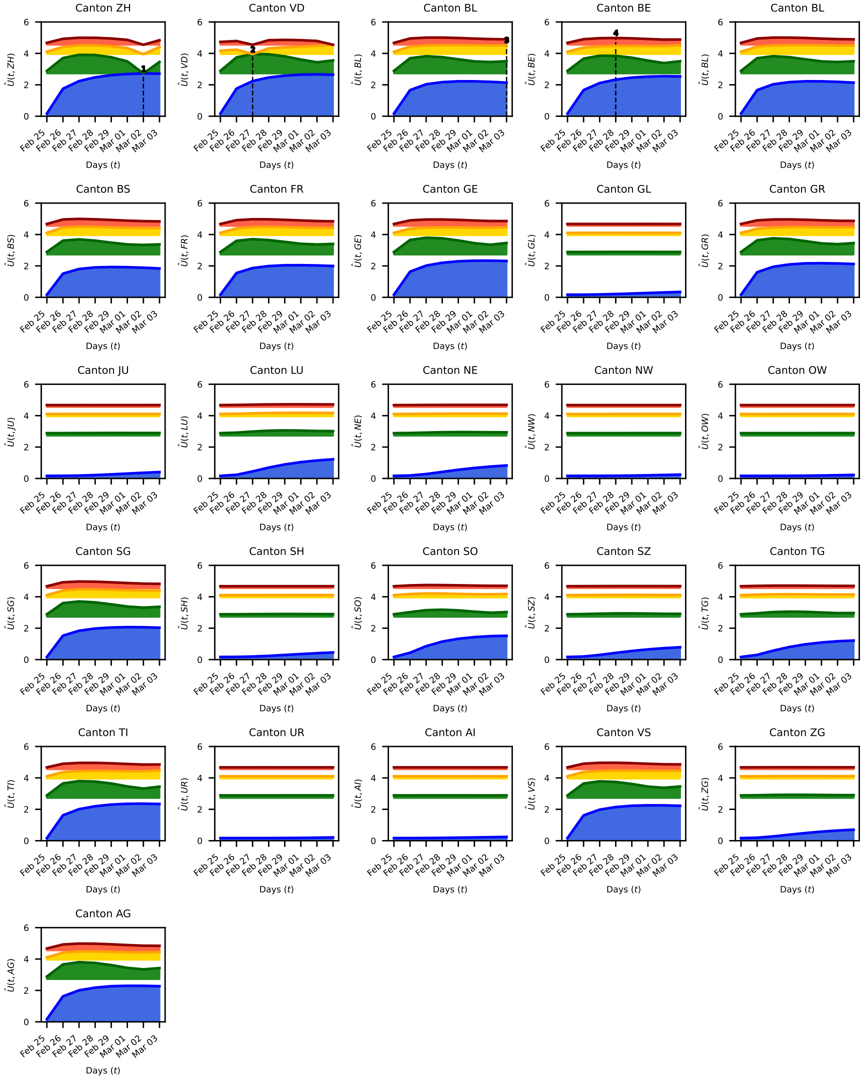


FIGURE 42: **Expected information gain during start of epidemic.** The blue, green, blue, yellow, and red curve corresponds to the utility for one, two, three, four surveys, respectively. The fixed dates and location of each survey are plotted with black dashed lines. The shaded areas indicate the difference from the expected information gain of the previous survey, which becomes thinner as additional surveys do not yield a further significant information gain.

The results indicate that the proposed OPALITS methodology selects certain populous and well interconnected cantons at specific times to acquire the most information for estimating the model parameters.

A.2.6.2 Case 2: Exponential spreading and optimal testing strategy during nonpharmaceutical interventions

When the spreading of the coronavirus entered an exponential growth stage, several governments (including the Swiss) decided to make nonpharmaceutical interventions, such as requesting social distancing, closing schools and restaurants, or ordering a complete lockdown in order to contain the epidemic. Here, the goal of the OPALITS is to propose surveys that would help to better assess the effectiveness of these interventions.

In this case, probability distributions of model parameters are informed using data from the existing spread of the COVID-19. The daily reported infections in Switzerland [270] from the 25 February up to the 17March 2020 are used to update the distributions specified in the previous phase by using Bayesian inference. The marginal posteriors are plotted in fig. 47. The SEI^rI^uR models the nonpharmaceutical interventions with a time-dependent transmission rate β and mobility factor θ. These parameters are calibrated by the data and provide an estimate of the timing and effectiveness of the interventions [256].

Figure 43 shows the maximum values of the information gain for each survey for $\mathcal{T} = \{\text{Mar 17}, \dots, \text{Mar 30}\}$. For cantons with a small population and low connectivity to other cantons, a low information gain is found. The opposite can be observed for cantons with large population and strong connections to other cantons. The values for the maximum utility in time for the measurements are listed in table 6. If only a single canton were to be selected (due to limited availability of test kits in the country), then a survey in the canton of Vaud carried out on the 30 March would be preferred over surveys in the cantons of Zurich, Bern or Geneva (blue in fig. 43). If two surveys could be afforded, the OPALITS methodology proposes them in the same canton (Vaud) on the 17 and on the 30 March (blue and green in fig. 43). Note that the canton of Zurich, ranked as the next preferred canton for a single survey (blue in fig. 43), is not selected by the methodology since part of the information that would be gained from testing is already contained in surveys performed in Vaud. If more test kits were available, in addition to the two tests in Vaud, the optimal location and time for a third survey would have been the canton of Grisons on the 30 March (yellow in fig. 43). The canton of Zurich is proposed as the fourth location to be surveyed also on the 30

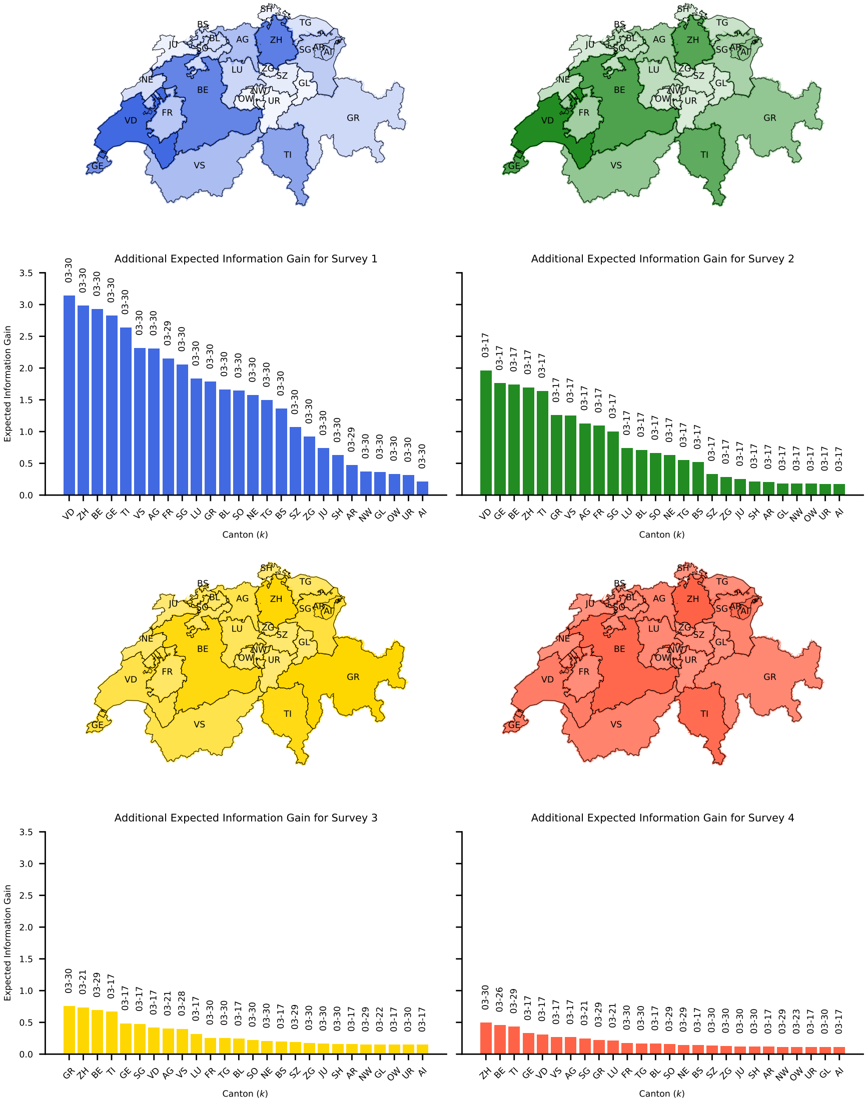


FIGURE 43: **Optimal testing strategy for effect of non-pharmaceutical interventions.** The maximum gain of information is plotted on the map of Switzerland using an exponential colour map. Here blue corresponds to taking one survey, green to adding a second, yellow to a third and red to a fourth. Below the map we plot the magnitude of the expected information gain of each survey, along with the optimal measurement dates per canton.

March. However, the information gain from the fourth survey in the canton of Zurich is approximately 10% of the total information gained from the surveys carried optimally in the first three cantons.

The results suggest that surveys at two locations/times provide significant information for assessing the effectiveness of interventions. Further tests on more locations/times did not add substantial information. It is evident that a trade-off between the required information gain and cost of testing are decisive for the number of necessary surveys and test kits.

A.2.6.3 Case 3: Optimal monitoring for a second outbreak

After the relaxation of measures that assisted in mitigating the initial spread of the disease, it is critical to monitor the population for a possible second outbreak. The OPALITS methodology supports such monitoring with surveys of the population based on data up to and after the release of the measures.

First, Bayesian inference is performed with data available from 25 February to 6 June, to update the uniform priors the resulting marginal posteriors are shown in fig. 48. This date is in accordance with the first stage of the major release of measures in Switzerland [264]. The effects of interventions are modelled by a parametrised time-dependent transmission rate and mobility factor (see Materials and methods). The inferred probability distributions of these additional parameters are taken into account as the OPALITS maximises the information gain. Note that $\mathcal{T} = \{\text{Jun } 7, \dots, \text{Jun } 14\}$ in this case.

Subsequently, data from 25 February to 9 July are included, repeating the Bayesian inference and estimating the marginal distributions and predictions shown in supplementary figs. 49 and 50, $\mathcal{T} = \{\text{Jul } 10, \dots, \text{Jul } 17\}$. The results indicate that the relaxation of measures correlates with an increase in the number of reported infections (fig. 44). The information gain for each canton indicates that the most informative surveys should be performed a week after performing the inference. The provided information could then assist in estimating the severity of a second outbreak, as indicated by the maximum of the utility in time (tables 7 and 8). Given that tests should be carried out in four locations and times, the methodology promotes optimal surveys for two different times, within a week, in the cantons of Zurich and Vaud. First, surveys should be performed in Zurich, providing high information gain for both considered cases. The next two surveys are to be performed in Zurich and Vaud, with a rank that depends on the considered case, and the fourth test should be performed in Vaud. We find that the information gain from the last test is approximately 10% of the cumulative information gain from the

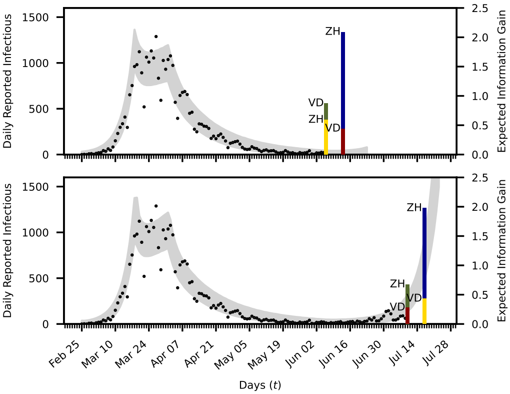


FIGURE 44: **Optimal testing strategy to monitor a second outbreak.** Bayesian inference determines the parameters of the first infection wave using the data (black dots) of the daily newly reported infections up to the 6 June (upper plot) and to the 9 July (lower plot). The 99% confidence intervals are plotted in grey. The proposed testing strategy is plotted with vertical bars at the optimal days found. Here blue indicated the utilities for the first survey. The green bars correspond to the gain in utility when adding a second survey assuming the first was chosen in the optimal location, where the yellow and red correspond to adding a third and fourth survey.

first three surveys. The number of surveys can be then selected according to the available test-kits N.

A.2.6.4 Case 4: Effectiveness of optimal testing

We demonstrate the importance of following the OPALITS by comparing it with a non-specific testing campaign that is based on heuristics. We first re-examine the situation at the start of an epidemic and assume that the available resources allow for two surveys. Surveys are simulated by evaluating the epidemiological model with the maximum a-posteriori estimate (MPE) of the parameters obtained from the inference in phase II (exponential growth) of the epidemic. We used data for the first 21 days of the infection spread in Switzerland [270] (25 February to 17 March). After evaluating the model, artificial surveys are obtained by adding a stochastic error term.

For the optimal strategy, data are collected by consulting figure 42. Thus, the two surveys are performed in the cantons of Zurich and Vaud, on the 2nd of March and the 27th of February, respectively. For a non-specific strategy, the cantons of Ticino and Bern were selected, on the 28th of28 February. We remark that this isthese are the canton where the first infection was reported and the capital of the country, respectively. These artificial data, obtained for the two strategies, are added to the real data of the daily reported cases from the first 8 days after the outbreak in Ticino. For the expanded data-set $\mathcal{D}$ the posterior distributions $p(\boldsymbol{\vartheta}|\widetilde{\boldsymbol{\vartheta}}_{\mathsf{MPE}}, \mathcal{D})$ are found by sampling the model parameters using nested sampling [261]. Note that the value of c is also inferred. For simplicity, the value of the correlation time τ is assumed to be known as this does not influence the results as long as the surveys are carried out in different cantons.

The resulting one- and two-dimensional marginalised posterior distributions for both strategies are shown in figure 45. We note that the dispersion coefficient r (defined in the Materials and methods) in the error model for the real data (the reported infections) and the correlation parameter are almost the same for both strategies. However, the model parameters show significant differences even when only two new data-points are added to a set of 208 data-points. The posterior distributions of the parameters of interest are propagated through the epidemiology model to provide the uncertainties in the number of unreported infectious individuals. In figure 46 the model output for the total number of unreported infections is plotted together with a 99% confidence interval along with the true value of the unreported cases obtained by using the selected parameters. The predictions from the OPALITS have a much higher certainty with a confidence interval that is up to four

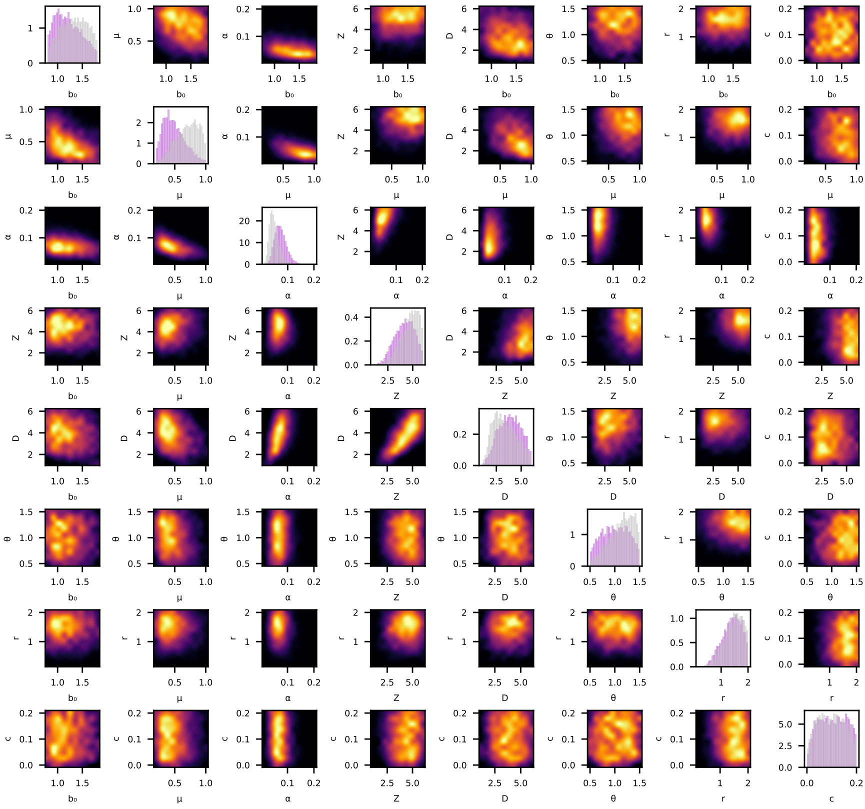


FIGURE 45: **Marginal posterior distributions for two strategies.** The diagonal shows the histogram for the marginal distribution for every parameter. Purple indicates posterior for the survey following the optimal testing strategy, gray the one for the non-specific strategy. The lower half and upper half show the samples of the joint distribution of two parameters for the optimal and the non-specific strategy respectively. Here black indicates low density and yellow high density.

times narrower than the one from a non-specific strategy. The same figure also shows the relative histogram plots for the effective reproduction number, which for the employed model is given from $R_t = \beta D\alpha + \beta D\mu(1-\alpha)$ [260]. Not only is the histogram more peaked, when data are optimally collected, but also the mean value of the two histograms is different. When data are optimally collected, the found mean value for the effective reproduction number is 2.1, whereas when the non-specific strategy is followed the average value is 3.2. A mean value of 3.2 could lead to more strict non-pharmaceutical interventions, which might prove unnecessary and harmful for the economy.

Further comparisons, demonstrating the value of the OPALITS, include model predictions with higher certainty, as indicated by confidence intervals that are narrower than the ones obtained from a non-specific strategy (figs. 51 and 52). Narrower uncertainty bounds provide higher confidence for decisions related to possible interventions to contain the epidemic. Further comparisons, demonstrating the value of the OpST over non-specific testing, are included in the Supplementary Material. First, fig. 50 shows the predictions based on the computational model output in every canton. The predictions per canton are obtained by evaluating the SEI^rI^uR model for all samples of the posterior distributions of the model parameters. The exact values and the data-points used in the inference are plotted as well. Also there, the predictions that correspond to an optimal testing strategy have higher certainty, as their confidence intervals are narrower than the ones from a non-specific strategy. Second, in fig. 51 the predictions that include the computational model output and the error model are shown. Once again, the predictions that correspond to an optimal testing strategy display smaller uncertainty.

A.2.7 Bayesian Inference from randomized testing

We consider a testing campaign including a set ($\boldsymbol{s}$) of surveys $s_i = (k_i, t_i), i = 1, \dots M_y$ performed in location $k_i \in \mathcal{C}$ and on day $t_i \in \mathcal{T}$. These surveys measure a quantity of interest (QoI), that is denoted by $\boldsymbol{y}(\boldsymbol{s}) = (y_1, \dots, y_{M_y})$. Here, y_i is the number of unreported infectious individuals, measured through survey s_i. The QoI can be predicted by a model $\boldsymbol{g}(\boldsymbol{s}, \boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}})$ (here the SEI^rI^uR epidemiological model) that depends on parameters of interest $\boldsymbol{\vartheta} \in \mathbb{R}^N$ and nuisance parameters $\widetilde{\boldsymbol{\vartheta}} \in \mathbb{R}^{\tilde{N}}$. The distinction between model and nuisance parameters is discussed in later sections. We note that both sets of parameters are uncertain and the proposed method aims to reduce the uncertainty only in the parameters of interest.

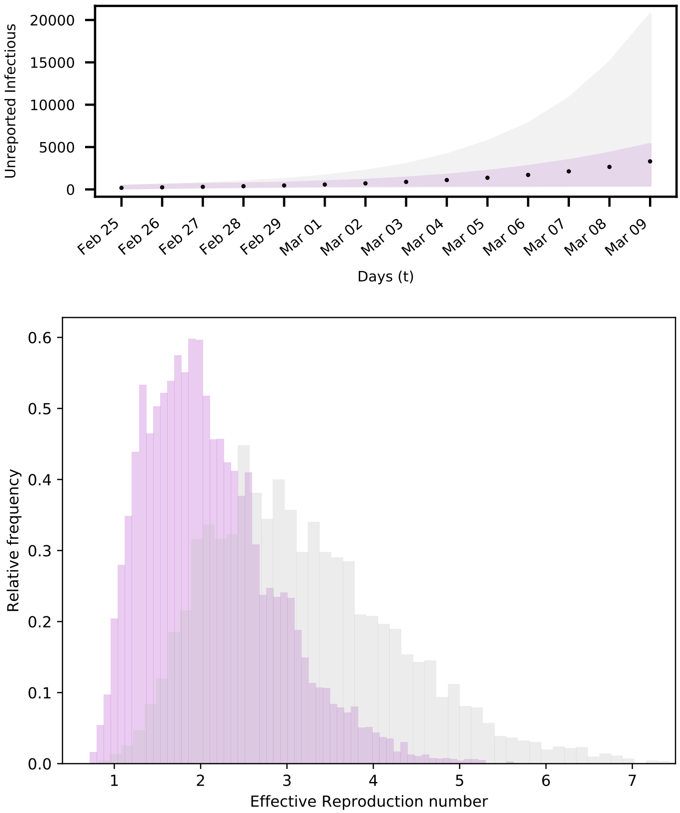


FIGURE 46: **Prediction uncertainty for different testing strategies.** Up: The black dots show the actual unreported infectious for an artificial spread in Switzerland. The error bounds show the 99% confidence intervals of the model output for samples of the parameters with data obtained by optimal (purple) and non-specific testing (gray). Down: Relative frequency histograms for effective reproduction number, predicted with data obtained by optimal (purple) and non-specific testing (gray).

A stochastic error term $\varepsilon(\boldsymbol{s})$ links the model prediction with the QoI

$$\boldsymbol{y}(\boldsymbol{s}) = \boldsymbol{g}(\boldsymbol{s}, \boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}}) + \varepsilon(\boldsymbol{s})\,. \tag{116}$$

The error $\varepsilon(\boldsymbol{s})$ is assumed to follow a zero-mean multivariate normal distribution $\mathcal{N}(0, \Sigma)$ with covariance matrix $\Sigma \in \mathbb{R}^{M_y \times M_y}$. The elements of the covariance matrix $(\Sigma_{s,s'})$ correspond to surveys taken at $s = (k, t)$ and $s' = (k', t')$ and are given by

$$\Sigma_{s,s'} = \sigma_t \, \sigma_{t'} \, \exp\left(-\frac{|t - t'|}{\tau}\right) \, \delta_{kk'}\,, \tag{117}$$

where $\delta_{kk'}$ is the Kronecker delta, which is 1 for $k = k'$ and 0 otherwise. The correlation time $\tau \in [0.5, 3.5]$ is considered a nuisance parameter. These assumptions about the covariance imply that surveys in different locations are not correlated, while those in the same location have an exponentially decaying temporal correlation. The latter avoids clustering of surveys in small time intervals [268]. The factor $\sigma_t \in \mathbb{R}$ is assumed proportional to the expectation of the QoI, taken over all possible survey locations and over the range of model and nuisance parameters

$$\sigma_t = c \, \frac{1}{K} \sum_{i=1}^{K} \mathbb{E}_{\boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}}} \left[g(s_i, \boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}}) \right]\,, \tag{118}$$

where $s_i = (i, t)$. The parameter $c \in [0, 0.25]$ is considered a model parameter. The expectation $\mathbb{E}_{\boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}}}[\,\cdot\,]$ is taken with respect to all parameters $\boldsymbol{\vartheta}$ and $\widetilde{\boldsymbol{\vartheta}}$ that follow the prior probability distribution with density $p(\boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}}) = p(\boldsymbol{\vartheta}) p(\widetilde{\boldsymbol{\vartheta}})$.

Under these assumptions, the conditional probability of $\boldsymbol{y}$ on $\boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}}$ and $\boldsymbol{s}$ is given by

$$p(\boldsymbol{y} | \boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}}, \boldsymbol{s}) = \frac{1}{\sqrt{(2\pi)^{M_y} |\Sigma(\boldsymbol{s})|}} \exp\left(-\frac{1}{2} \mathbf{z}^\top \Sigma(\boldsymbol{s})^{-1} \mathbf{z}\right)\,, \tag{119}$$

where $|\Sigma(\boldsymbol{s})|$ is the determinant of the covariance matrix and $\mathbf{z} = \boldsymbol{y}(\boldsymbol{s}) - \boldsymbol{g}(\boldsymbol{s}, \boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}})$.

In the present study, the QoI measured by a survey is the number of unreported infectious individuals in a particular canton on a particular date. This implicitly assumes that there no restrictions on when the survey can be conducted and that there are no observational delays, which means the the QoI is instantaneously obtained. Both assumptions are not restrictive however. Restrictions on the possible survey dates can be accounted for by simply excluding those dates from the dates on which the utility function is evaluated. Also, a delay of one day (meaning that two days are needed to survey a canton k, starting from day t) would mean that $\boldsymbol{y} = (I_k^u(t) + I_k^u(t+1))/2$ is measured. In other words, when there is a delay the measured quantity can still be mapped to a model quantity, which allows us to perform Bayesian inference. There are several types of measurements (Rapid testing [265], PCR [266], Schwabs [267]) being proposed for testing asymptomatic individuals. We emphasize that our methodology is compatible with any of these types. Data related issues such as uncertainties, test sensitivities and delays in processing can be accommodated in the Bayesian inference framework and in the input to the SEIR model.

A.2.8 Expected Information Gain

The most informative surveys $\boldsymbol{y}$ provide the least uncertainty in the estimates of the model parameters $\boldsymbol{\vartheta}$. Starting with a user-postulated prior distribution $p(\boldsymbol{\vartheta})$, Bayesian learning is used to update the uncertainties in the model parameters leading to a posterior distribution $p(\boldsymbol{\vartheta}|\boldsymbol{y}, \widetilde{\boldsymbol{\vartheta}}, \boldsymbol{s})$, based on the information contained in the test data $\boldsymbol{y}$. The Kullback–Leibler (KL) divergence between the posterior $p(\boldsymbol{\vartheta}|\boldsymbol{y}, \widetilde{\boldsymbol{\vartheta}}, \boldsymbol{s})$ and the prior distributions $p(\boldsymbol{\vartheta})$ of the model parameters measures the distance between the two distributions. Informative data produce posterior distributions that differ from the prior; greater differences lead to higher information gain. Therefore, the most informative data $\boldsymbol{y}$ correspond to the testing strategy (measurement locations and times) with the highest information gain [262, 273].

The OPALITS is identified by maximizing a utility function [259]. One choice is the KL divergence $u(\boldsymbol{y}, \widetilde{\boldsymbol{\vartheta}}, \boldsymbol{s}) = D_{\mathrm{KL}}\big(p(\boldsymbol{\vartheta}|\boldsymbol{y}, \widetilde{\boldsymbol{\vartheta}}, \boldsymbol{s}) \| p(\boldsymbol{\vartheta})\big)$ quantifying the information gain from the data [259]. However, since data are not available in the experimental design phase, the utility function is selected here to be the expected KL divergence $\mathbb{E}_{\boldsymbol{y}|\widetilde{\boldsymbol{\vartheta}},\boldsymbol{s}}\big[u(\boldsymbol{y}, \widetilde{\boldsymbol{\vartheta}}, \boldsymbol{s})\big]$ over all data generated by the model prediction error equation 116. Also, to account for the uncertainty in nuisance parameters $\widetilde{\boldsymbol{\vartheta}}$, encoded in the prior distribution $p(\widetilde{\boldsymbol{\vartheta}})$, the expectation is also taken with respect to $\widetilde{\boldsymbol{\vartheta}}$, which results in the utility function [259]

$$\begin{aligned} U(\boldsymbol{s}) &= \mathbb{E}_{\widetilde{\boldsymbol{\vartheta}}}\left[\mathbb{E}_{\boldsymbol{y}|\widetilde{\boldsymbol{\vartheta}},\boldsymbol{s}}\big[u(\boldsymbol{y}, \widetilde{\boldsymbol{\vartheta}}, \boldsymbol{s})\big]\right] = \\ &\iiint \log\left(\frac{p(\boldsymbol{\vartheta}|\boldsymbol{y}, \widetilde{\boldsymbol{\vartheta}}, \boldsymbol{s})}{p(\boldsymbol{\vartheta})}\right) p(\boldsymbol{\vartheta}|\boldsymbol{y}, \widetilde{\boldsymbol{\vartheta}}, \boldsymbol{s})\, \mathrm{d}\boldsymbol{\vartheta}\, p(\boldsymbol{y}|\widetilde{\boldsymbol{\vartheta}}, \boldsymbol{s})\, \mathrm{d}\boldsymbol{y}\, p(\widetilde{\boldsymbol{\vartheta}})\mathrm{d}\widetilde{\boldsymbol{\vartheta}}\,. \end{aligned} \tag{120}$$

By using Bayes' theorem

$$p(\boldsymbol{\vartheta}|\boldsymbol{y},\widetilde{\boldsymbol{\vartheta}},\boldsymbol{s}) = \frac{p(\boldsymbol{y}|\boldsymbol{\vartheta},\widetilde{\boldsymbol{\vartheta}},\boldsymbol{s})\,p(\boldsymbol{\vartheta})}{p(\boldsymbol{y}|\widetilde{\boldsymbol{\vartheta}},\boldsymbol{s})}\,, \tag{121}$$

the utility function can be simplified to

$$U(\boldsymbol{s}) = \iiint \log\left(\frac{p(\boldsymbol{y}|\boldsymbol{\vartheta},\widetilde{\boldsymbol{\vartheta}},\boldsymbol{s})}{p(\boldsymbol{y}|\widetilde{\boldsymbol{\vartheta}},\boldsymbol{s})}\right) p(\boldsymbol{y}|\boldsymbol{\vartheta},\widetilde{\boldsymbol{\vartheta}},\boldsymbol{s})\,p(\boldsymbol{\vartheta})\,p(\widetilde{\boldsymbol{\vartheta}})\,\mathrm{d}\boldsymbol{y}\,\mathrm{d}\boldsymbol{\vartheta}\,\mathrm{d}\widetilde{\boldsymbol{\vartheta}}\,. \tag{122}$$

Note that the expected utility only depends on the locations and times of the measurements via $\boldsymbol{s}$. The term $p(\boldsymbol{y}|\widetilde{\boldsymbol{\vartheta}},\boldsymbol{s})$ is the model evidence given by

$$p(\boldsymbol{y}|\widetilde{\boldsymbol{\vartheta}},\boldsymbol{s}) = \int p(\boldsymbol{y}|\boldsymbol{\vartheta},\widetilde{\boldsymbol{\vartheta}},\boldsymbol{s})\,p(\boldsymbol{\vartheta})\,\mathrm{d}\boldsymbol{\vartheta}\,. \tag{123}$$

The choice of the prior distribution $p(\boldsymbol{\vartheta})$ for the parameters allows to incorporate prior knowledge from epidemiology. If no information is available from data, a case encountered in the beginning of the infection, a uniform prior distribution can be assumed. Table S5 summarizes our choice of prior distributions for all the involved uncertain quantities. If data $\boldsymbol{d}$ of the daily number of reported infectious individuals is available, Bayesian inference can be used to inform the prior distribution, as described later on. In this case, the prior $p(\boldsymbol{\vartheta})$ in equation 122 is replaced by the distribution $p(\boldsymbol{\vartheta}|\boldsymbol{d})$ informed from the data $\boldsymbol{d}$.

In the present work, the assumed nuisance parameters are the correlation time τ and the initial condition of the unreported infections in the cantons of Aargau, Bern, Basel-Landschaft, Basel-Stadt, Fribourg, Geneva, Grisons, St.Gallen, Ticino, Vaud, Valais and Zurich

$$\boldsymbol{I}^u_{\mathrm{IC}} = (I^u_{\mathrm{AR}}, I^u_{\mathrm{BE}}, I^u_{\mathrm{BL}}, I^u_{\mathrm{BS}}, I^u_{\mathrm{FR}}, I^u_{\mathrm{GE}}, I^u_{\mathrm{GR}}, I^u_{\mathrm{SG}}, I^u_{\mathrm{TI}}, I^u_{\mathrm{VD}}, I^u_{\mathrm{VS}}, I^u_{\mathrm{ZH}})$$

with prior distributions $\boldsymbol{I}^u_{\mathrm{IC}} \sim \mathcal{U}([0,50]^{12})$ and $\tau \sim \mathcal{U}([0.5,3.5])$.

EPIDEMIOLOGICAL MODEL Here we employ the SEI^rI^uR epidemiological model [260] to forecast the dynamics of the coronavirus outbreak in Switzerland

$$\begin{aligned}
\frac{\mathrm{d}S_k}{\mathrm{d}t} &= -\frac{\beta S_k I_k^r}{N_k} - \frac{\mu\beta S_k I_k^u}{N_k} + \theta\sum_{l=1}^{K}\frac{M_{kl}S_l}{N_l - I_l^r} - \theta\sum_{l=1}^{K}\frac{M_{lk}S_k}{N_k - I_k^r}\\
\frac{\mathrm{d}E_k}{\mathrm{d}t} &= \frac{\beta S_k I_k^r}{N_k} + \frac{\mu\beta S_k I_k^u}{N_k} - \frac{E_k}{Z} + \theta\sum_{l=1}^{K}\frac{M_{kl}E_l}{N_l - I_l^r} - \theta\sum_{l=1}^{K}\frac{M_{lk}E_k}{N_k - I_k^r}\\
\frac{\mathrm{d}I_k^r}{\mathrm{d}t} &= \alpha\frac{E_k}{Z} - \frac{I_k^r}{D}\\
\frac{\mathrm{d}I_k^u}{\mathrm{d}t} &= (1-\alpha)\frac{E_k}{Z} - \frac{I_k^u}{D} + \theta\sum_{l=1}^{K}\frac{M_{kl}I_l^u}{N_l - I_l^r} - \theta\sum_{l=1}^{K}\frac{M_{lk}I_k^u}{N_k - I_k^r}\\
\frac{\mathrm{d}N_k}{\mathrm{d}t} &= \theta\sum_{l=1}^{K}(M_{kl} - M_{lk}),
\end{aligned} \tag{124}$$

where S_k, E_k, I_k^r and I_k^u denote the number of individuals in canton $k = \{1, \dots, K\}$ that are susceptible, exposed, reported infectious and unreported infectious, respectively. We denote by K the number of cantons (26 in Switzerland), by N_k the total population of the canton k, while the population mobility between cantons k and l is denoted by M_{kl} with values obtained from the Swiss Federal Statistical Office [263]. The model parameters are the transmission rate (β), the relative transmission rate between reported and unreported infectious individuals (μ), the virus latency period (Z), the infectious period (D), the reporting rate (α) and the mobility factor (θ).

We employ different time-dependent expressions for the transmission rate and the mobility factor for each stage of the epidemic. Constants are chosen for the start of an epidemic while in the cases of monitoring of interventions, the following expressions are used:

$$\beta(t) = \begin{cases} b_0, & 0 \le t \le \delta_1 \\ b_1, & \delta_1 < t \end{cases}, \qquad \theta(t) = \begin{cases} \theta_0, & 0 \le t \le \delta_1 \\ \theta_1, & \delta_1 < t \end{cases}, \tag{125}$$

where b_0, b_1, θ_0 and θ_1 are the transmission rates and mobility factors before and after the intervention. Time $t = 0$ corresponds to the 25th of February 2020, and $\delta_1 = 21$ to the 17th of March 2020, when the lockdown was announced in Switzerland [264]. Finally, for the third case (monitoring of a second outbreak) we assume that

$$\beta(t) = \begin{cases} b_0, & 0 \leq t \leq \delta_1 \\ b_1, & \delta_1 < t \leq \delta_2 \\ b_2, & \delta_2 < t \leq \delta_3 \\ b_3(t), & \delta_3 < t \end{cases}, \ \theta(t) = \begin{cases} \theta_0, & 0 \leq t \leq \delta_1 \\ \theta_1, & \delta_1 < t \leq \delta_2 \\ \theta_2, & \delta_2 < t \leq \delta_3 \\ \theta_0, & \delta_3 < t \end{cases}. \tag{126}$$

As in equation 125, b_0 is the transmission rate before the intervention while $b_1 = c_1\, b_0$ and $b_2 = c_2\, b_0$ with $c_1, c_2 \in [0, 1]$ are the transmission rates after the two interventions. Similarly, θ_0 is the mobility factor before any interventions took place, while $\theta_1 = c_3\, \theta_0$ and $\theta_2 = c_4\, \theta_0$ with $c_3, c_4 \in [0, 1]$ are the mobility factors after the two interventions. Moreover, δ_1 and δ_2 correspond to the days of the interventions. The day when the measures are loosened is denoted by δ_3. After that day, the transmission rate is gradually increasing

$$b_3(t) = \min(b_2 + \lambda(t - \delta_3), b_0)\,, \tag{127}$$

with $\lambda \in [0, 0.03]$, while the mobility factor regains its initial value of θ_0.

A.2.9 Estimation of the Expected Information Gain

The calculation of the expected utility from equation 122 is performed with Monte-Carlo integration. Samples from the prior distribution are denoted by $\boldsymbol{\vartheta}^{(i)} \sim p(\boldsymbol{\vartheta})$ and by $\widetilde{\boldsymbol{\vartheta}}^{(i)} \sim p(\widetilde{\boldsymbol{\vartheta}})$, while samples on the measurement space are denoted by $\boldsymbol{y}^{(i,j)} \sim p(\boldsymbol{y}|\boldsymbol{\vartheta}^{(i)}, \widetilde{\boldsymbol{\vartheta}}^{(i)}, \boldsymbol{s})$, where $i \in \{1, \dots, N_\vartheta\}$ and $j \in \{1, \dots, N_y\}$. With these samples, an estimate of the expected utility is computed as

$$\begin{aligned} \hat{U}(\boldsymbol{s}) &= \frac{1}{N_\vartheta N_{\boldsymbol{y}}} \sum_{i=1}^{N_\vartheta} \sum_{j=1}^{N_y} \left[\log \left(\frac{p\big(\boldsymbol{y}^{(i,j)}|\boldsymbol{\vartheta}^{(i)}, \widetilde{\boldsymbol{\vartheta}}^{(i)}, \boldsymbol{s}\big)}{p\big(\boldsymbol{y}^{(i,j)}|\widetilde{\boldsymbol{\vartheta}}^{(i)}, \boldsymbol{s}\big)} \right) \right], \\ p\big(\boldsymbol{y}^{(i,j)}|\widetilde{\boldsymbol{\vartheta}}^{(i)}, \boldsymbol{s}\big) &:= \frac{1}{N_\vartheta} \sum_{n=1}^{N_\vartheta} p\big(\boldsymbol{y}^{(i,j)}|\boldsymbol{\vartheta}^{(n)}, \widetilde{\boldsymbol{\vartheta}}^{(i)}, \boldsymbol{s}\big)\,. \end{aligned} \tag{128}$$

In our implementation the samples $\boldsymbol{\vartheta}^{(i)}$ and $\widetilde{\boldsymbol{\vartheta}}^{(i)}$, $(i = 1, \dots, N_\theta)$, remain the same for different values of $\boldsymbol{s}$. Thus, the model evaluations $\boldsymbol{g}(\boldsymbol{s}, \boldsymbol{\vartheta}^{(i)}, \widetilde{\boldsymbol{\vartheta}}^{(i)})$ are only carried out once and are stored and used in the iteration process involved in the optimization. This allows to separate the computational cost of the model evaluation from the cost of computing the utility, which scales as $\mathcal{O}(N_{\vartheta}^2 N_y)$.

A.2.10 Optimal Location and Time of Testing

We define the optimal survey times and locations as

$$\boldsymbol{s}^* = \underset{s_1,\dots,s_{M_y}}{\arg\max}\, \hat{U}(\boldsymbol{s}) \,, \tag{129}$$

where $\boldsymbol{s}^* = (s_1^*, \dots, s_{M_y}^*)$ with $s_i^* = (k_i^*, t_i^*)$ denote the locations k_i^* and times t_i^* for the optimal surveys with $i \in \{1, \dots, M_y\}$. For a grid search, the associated computational cost is $\mathcal{O}((KT)^{M_y})$ and thus grows exponentially with the number of surveys. This curse of dimensionality is avoided by using a sequential optimization method [125] to approximate the global optimum by iteratively solving

$$s_n^* = \underset{s}{\arg\max}\, \hat{U}_n(s) \,, \quad \forall n = 1, \dots, M_y \,, \tag{130}$$

where $s = (k, t)$ is the location and time to be estimated sequentially starting with $n = 1$ and

$$\hat{U}_n(s) = \hat{U}(\boldsymbol{s}) \,, \qquad \boldsymbol{s} = (s_1^*, \dots, s_{n-1}^*, s) \,. \tag{131}$$

Following this, we define the expected information gain for survey n as

$$\Delta\hat{U}_n(s) = \begin{cases} \hat{U}_1(s), & n = 1 \\ \hat{U}_n(s) - \hat{U}_{n-1}(s_{n-1}^*), & n > 1. \end{cases} \tag{132}$$

A.2.11 Quantification of Uncertainty

A data informed prior $p(\boldsymbol{\vartheta}|\boldsymbol{d})$ of the model parameters $\boldsymbol{\vartheta}$ can be computed from available data $\boldsymbol{d} = (d_1, \dots, d_{M_d})$, collected at M_d locations and days. Here, available data $\boldsymbol{d}$ refer to the daily number of reported infectious individuals and they are contrasted from the data $\boldsymbol{y}$ of the number of unreported infectious individuals. The latter are obtained from testing strategies at selected populations using optimal experimental design. The data is mapped via a distinct model output $f(s, \boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}})$ through the following error model

$$p(d_i|\boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}}, \nu) = \mathcal{NB}\left(d_i \,|\, f(s_i, \boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}}), \nu\right) , \quad i = 1, \dots, M_d \,. \tag{133}$$

where $\mathcal{NB}$ is the negative binomial distribution with mean f and dispersion ν. Also, $s_i = (k_i, t_i)$ is the location and time the data d_i was collected. The choice of a different error model, compared to equation 116, is based on the assumption that the data are independent and identically distributed. Such an assumption would not be acceptable in the measurement model in equation 116, as it may result in uncorrelated measurements that can become clustered in small time intervals [268].

The data $\boldsymbol{d} = (d_1, \dots, d_{M_d})$ are the daily number of reported infections per canton in Switzerland [270] which corresponds to the following model quantity

$$f(s_i, \boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}}) := \int_{t_i-0.5}^{t_i+0.5} \frac{\alpha}{Z} E_{k_i}(\tau)d\tau \approx \frac{\alpha}{Z} E_{k_i}(t_i)\,. \tag{134}$$

The posterior distribution that will be used subsequently as a data informed prior is obtained using Bayes' theorem

$$p(\boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}}|\boldsymbol{d}) = \frac{p(\boldsymbol{d}|\boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}})\, p(\boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}})}{p(\boldsymbol{d})}\,, \tag{135}$$

and is sampled with a nested sampling algorithm [261]. Note the difference to equation 121 and the optimal testing methodology, where we are interested to reduce the uncertainty in $p(\boldsymbol{\vartheta}|\boldsymbol{y}, \widetilde{\boldsymbol{\vartheta}}, \boldsymbol{s})$, which excludes the nuisance parameters $\widetilde{\boldsymbol{\vartheta}}$. For the dispersion parameter in equation 133, it is assumed that $\nu = r\, f(s_i, \boldsymbol{\vartheta}, \widetilde{\boldsymbol{\vartheta}})$. The coefficient r is unknown and included in the parameter set, where $r \sim \mathcal{U}([0, 2])$.

The three inferences performed are summarized in table S5, which shows the involved model parameters in each case. The histograms for the found samples are shown in figures S1, S2, and S3.

We remark that, using the present methodology, the inferred date for the beginning of the intervention is $\delta_1 = 22.5$, which is the 18th of March 2020, corresponding well with the 17th of March 2020 on which the lockdown was introduced in Switzerland [264]. Moreover, we infer a significant reduction in the mobility factor, which indicates that traffic between cantons was also minimized. For the inference III we plot the fit using the inferred parameters in figure S4. The daily reported cases per canton are shown, together with the data used for the inference.

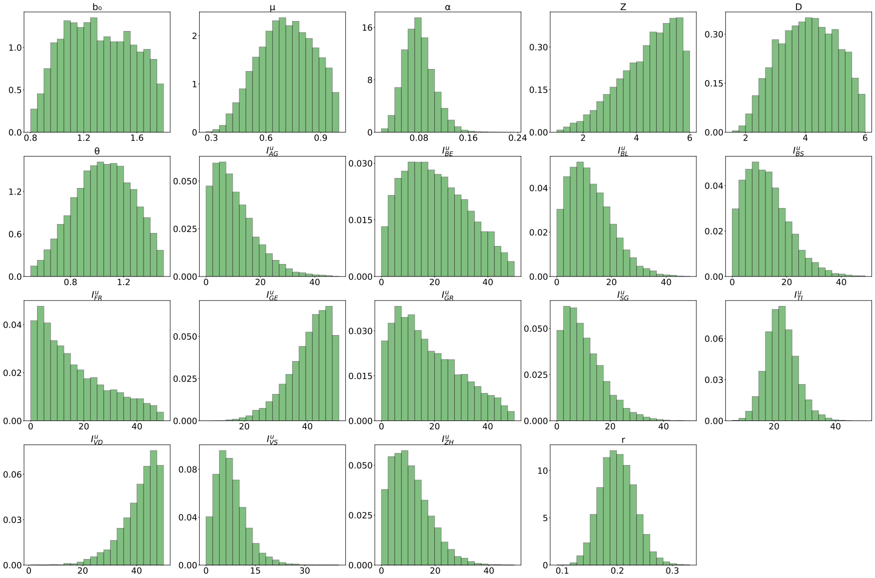


FIGURE 47: **Marginal posterior distributions with data up to 17th of March 2020.** The used data correspond to the daily reported infectious persons in the cantons of Switzerland. The marginals with a canton label XY correspond to the initial condition $I^u_{\mathrm{XY}}(t = 0)$ for the unreported cases in that canton.

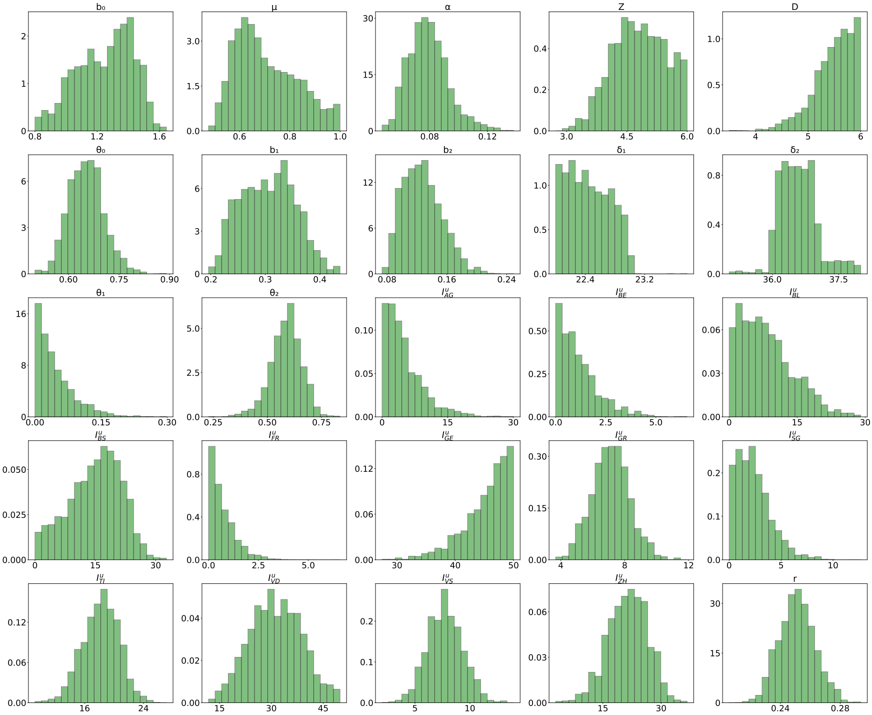


FIGURE 48: **Marginal posterior distributions with data up to 6th of June 2020.** The used data correspond to the daily reported infectious persons in the cantons of Switzerland. The marginals with a canton label XY correspond to the initial condition $I^u_{\mathrm{XY}}(t=0)$ for the unreported cases in that canton.

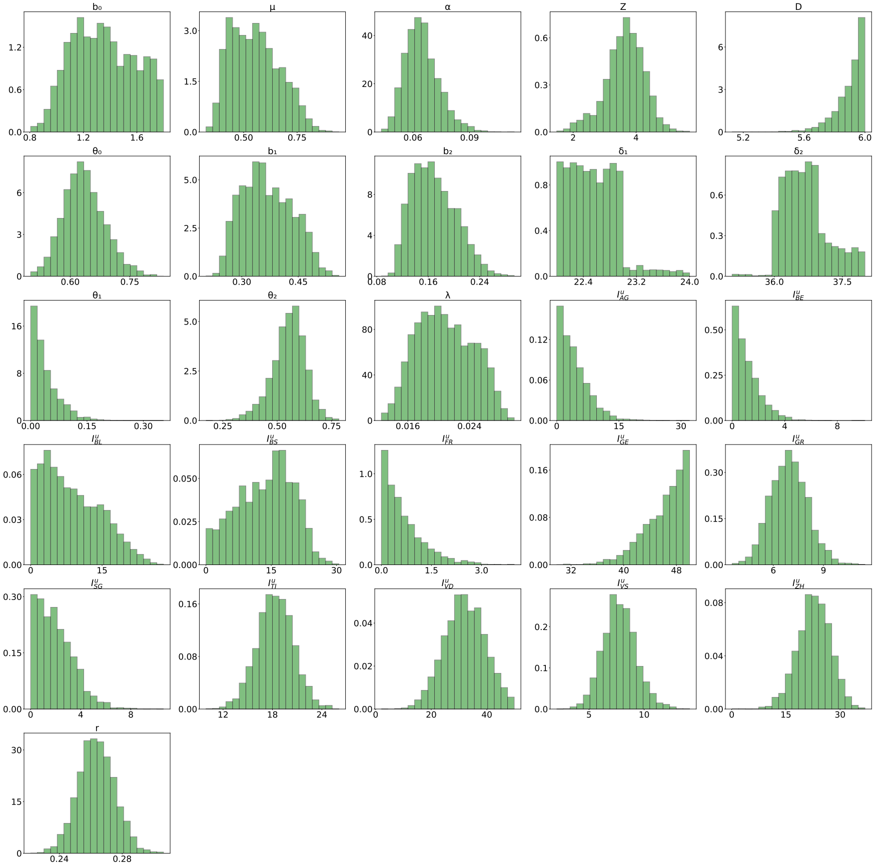


FIGURE 49: **Marginal posterior distributions with data up to 9th of July 2020.** The used data correspond to the daily reported infectious persons in the cantons of Switzerland. The marginals with a canton label XY correspond to the initial condition $I^u_{\mathrm{XY}}(t=0)$ for the unreported cases in that canton.

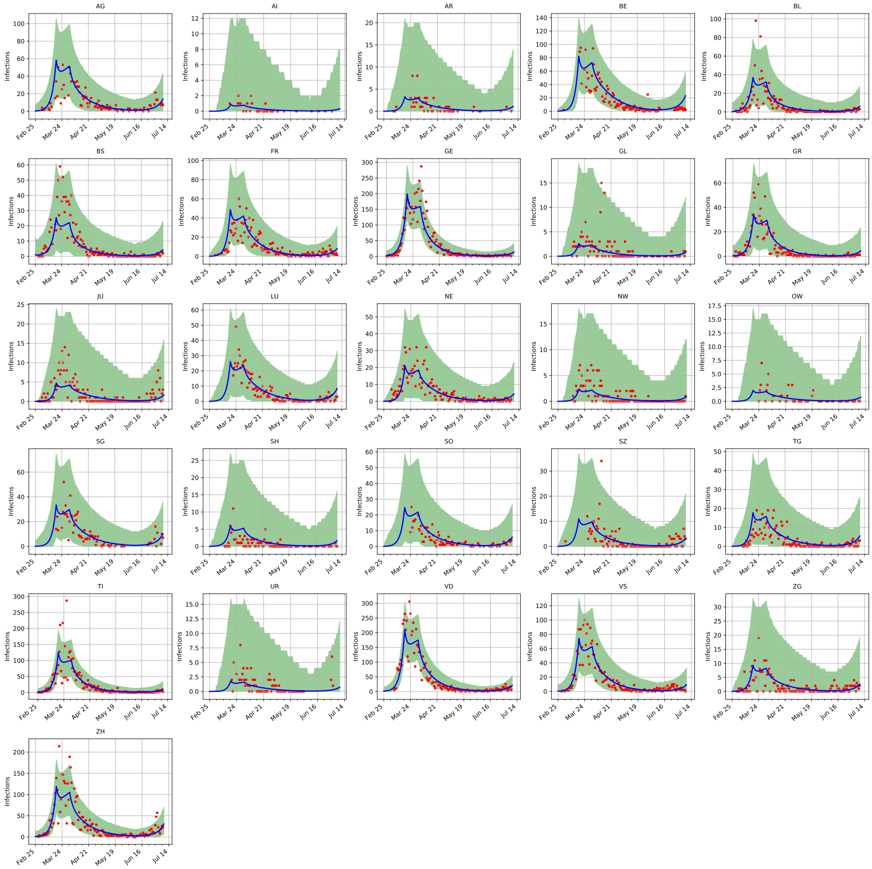


FIGURE 50: **Maximum a-posteriori prediction with data up to 9th of July 2020.** The red points correspond to the daily reported cases per cantons and the blue curve shows the maximum a-posteriori prediction. The 99% confidence interval is plotted in green and based on the sample shown in figure S3.

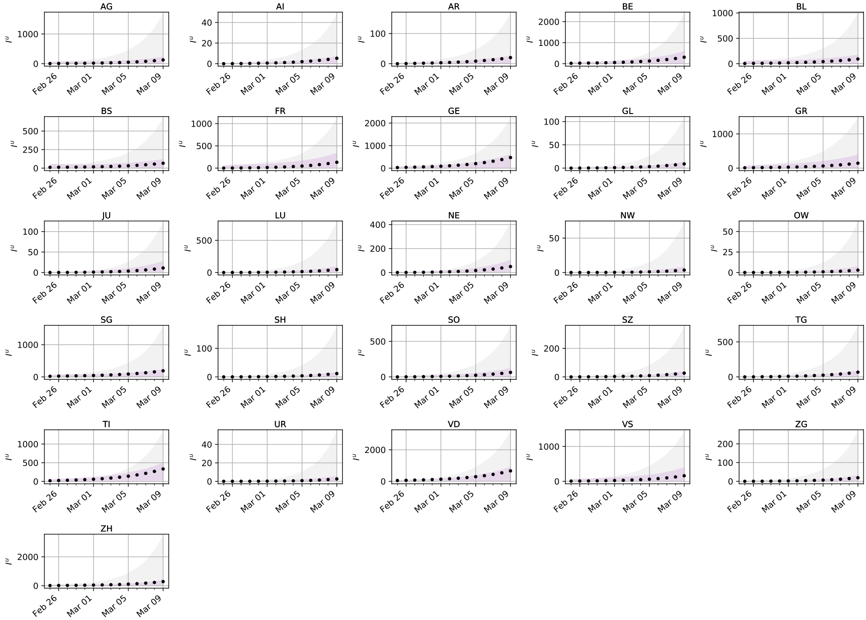


FIGURE 51: **Comparison of prediction uncertainty per canton.** The predictions are based on optimal strategies and non-specific testing for collection of data. They are also based on the SEI^rI^uR model output. The error bounds show the 99% confidence intervals of the unreported infectious model output for samples of the parameters with data obtained by optimal (purple) and standard testing (gray). The black dots show the actual unreported infectious for an artificial spread in Switzerland.

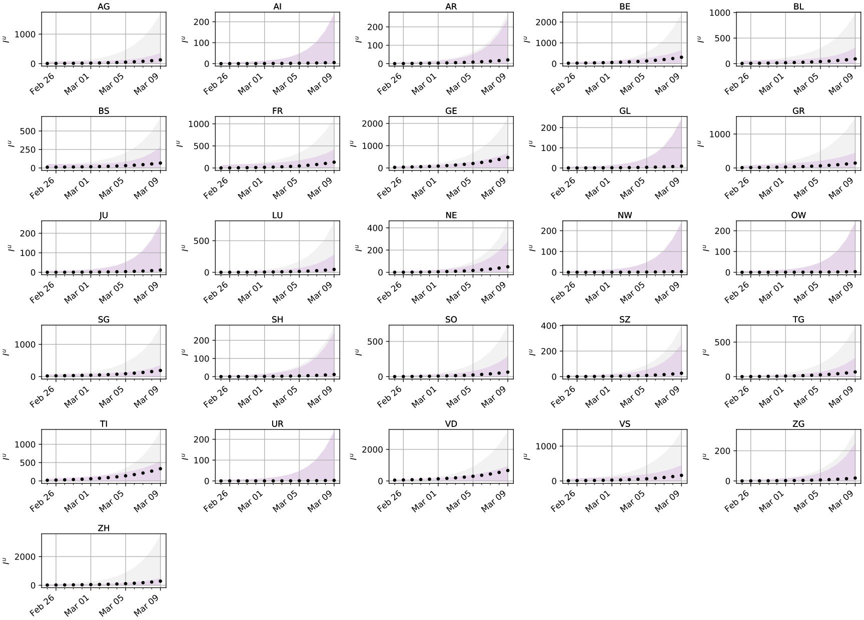


FIGURE 52: **Comparison of propagated uncertainty per canton.** The predictions are based on optimal strategies and non-specific testing. The SEI^rI^uR model output with added model error for the unreported infectious is shown. The error bounds show the 99% confidence intervals of the model output with added model error for samples of the parameters with data obtained by optimal (purple) and standard testing (gray). The black dots show the actual unreported infectious for an artificial spread in Switzerland.

	Maximum of Expected Information Gain			
Canton	1st measurement	2nd measurement	3rd measurement	4th measurement
AG	2.297 (01-03)	1.080 (27-02)	0.530 (28-02)	0.436 (27-02)
AI	0.240 (03-03)	0.167 (28-02)	0.152 (27-02)	0.123 (28-02)
AR	0.538 (03-03)	0.189 (28-02)	0.157 (03-03)	0.127 (26-02)
BE	2.547 (02-03)	1.130 (27-02)	0.558 (03-03)	0.432 (28-02)
BL	2.224 (29-02)	1.099 (27-02)	0.567 (28-02)	0.458 (28-02)
BS	1.930 (29-02)	0.969 (27-02)	0.545 (27-02)	0.445 (27-02)
FR	2.046 (01-03)	0.983 (27-02)	0.533 (28-02)	0.424 (27-02)
GE	2.338 (02-03)	1.074 (27-02)	0.512 (03-03)	0.410 (28-02)
GL	0.344 (03-03)	0.171 (28-02)	0.152 (27-02)	0.124 (01-03)
GR	2.174 (01-03)	1.039 (27-02)	0.517 (29-02)	0.416 (28-02)
JU	0.408 (03-03)	0.176 (29-02)	0.154 (01-03)	0.125 (03-03)
LU	1.220 (03-03)	0.340 (29-02)	0.225 (01-03)	0.173 (01-03)
NE	0.828 (03-03)	0.234 (29-02)	0.176 (01-03)	0.137 (01-03)
NW	0.246 (03-03)	0.168 (01-03)	0.152 (02-03)	0.123 (03-03)
OW	0.225 (03-03)	0.168 (03-03)	0.152 (02-03)	0.124 (03-03)
SG	2.067 (01-03)	0.981 (27-02)	0.519 (28-02)	0.425 (27-02)
SH	0.456 (03-03)	0.182 (29-02)	0.156 (01-03)	0.126 (01-03)
SO	1.515 (03-03)	0.455 (28-02)	0.256 (27-02)	0.199 (27-02)
SZ	0.785 (03-03)	0.221 (29-02)	0.167 (29-02)	0.134 (28-02)
TG	1.214 (03-03)	0.334 (28-02)	0.209 (28-02)	0.163 (28-02)
TI	2.362 (02-03)	1.077 (27-02)	0.516 (03-03)	0.409 (28-02)
UR	0.210 (03-03)	0.167 (03-03)	0.152 (02-03)	0.124 (28-02)
VD	2.666 (02-03)	1.233 (27-02)	0.594 (03-03)	0.314 (29-02)
VS	2.254 (01-03)	1.061 (27-02)	0.514 (28-02)	0.417 (28-02)
ZG	0.701 (03-03)	0.206 (28-02)	0.161 (29-02)	0.130 (28-02)
ZH	2.721 (02-03)	1.187 (27-02)	0.556 (28-02)	0.449 (27-02)

TABLE 5: **Maximum expected information gain for outbreak of a new disease.** The corresponding optimal dates are shown in parenthesis.

	Maximum of Expected Information Gain			
Canton	1st measurement	2nd measurement	3rd measurement	4th measurement
AG	2.307 (30-03)	1.128 (17-03)	0.399 (30-03)	0.264 (30-03)
AI	0.211 (30-03)	0.169 (17-03)	0.146 (21-03)	0.112 (26-03)
AR	0.474 (30-03)	0.205 (17-03)	0.154 (29-03)	0.115 (29-03)
BE	2.927 (30-03)	1.738 (17-03)	0.691 (17-03)	0.459 (17-03)
BL	1.663 (30-03)	0.710 (17-03)	0.241 (17-03)	0.167 (17-03)
BS	1.359 (30-03)	0.518 (17-03)	0.196 (17-03)	0.140 (17-03)
FR	2.149 (30-03)	1.093 (17-03)	0.256 (17-03)	0.176 (17-03)
GE	2.825 (29-03)	1.760 (17-03)	0.483 (21-03)	0.327 (21-03)
GL	0.364 (30-03)	0.183 (17-03)	0.149 (28-03)	0.114 (29-03)
GR	1.783 (30-03)	1.256 (17-03)	0.754 (17-03)	0.222 (21-03)
JU	0.736 (30-03)	0.251 (17-03)	0.163 (30-03)	0.121 (30-03)
LU	1.830 (30-03)	0.738 (17-03)	0.316 (30-03)	0.210 (30-03)
NE	1.573 (30-03)	0.626 (17-03)	0.205 (17-03)	0.145 (17-03)
NW	0.369 (30-03)	0.181 (17-03)	0.149 (30-03)	0.114 (29-03)
OW	0.332 (30-03)	0.176 (17-03)	0.148 (30-03)	0.114 (29-03)
SG	2.056 (30-03)	1.003 (17-03)	0.471 (17-03)	0.247 (17-03)
SH	0.625 (30-03)	0.215 (17-03)	0.158 (29-03)	0.119 (30-03)
SO	1.642 (30-03)	0.663 (17-03)	0.224 (30-03)	0.155 (30-03)
SZ	1.066 (30-03)	0.330 (17-03)	0.186 (30-03)	0.135 (30-03)
TG	1.495 (30-03)	0.548 (17-03)	0.250 (30-03)	0.168 (30-03)
TI	2.640 (29-03)	1.639 (17-03)	0.668 (17-03)	0.436 (17-03)
UR	0.315 (30-03)	0.175 (17-03)	0.148 (29-03)	0.114 (29-03)
VD	3.139 (30-03)	1.961 (17-03)	0.415 (22-03)	0.311 (23-03)
VS	2.313 (30-03)	1.251 (17-03)	0.397 (17-03)	0.264 (17-03)
ZG	0.920 (30-03)	0.285 (17-03)	0.172 (30-03)	0.127 (30-03)
ZH	2.980 (30-03)	1.695 (17-03)	0.735 (17-03)	0.494 (17-03)

TABLE 6: **Maximum expected information gain of non-pharmaceutical interventions.** The corresponding optimal dates are shown in parenthesis.

	Maximum of Expected Information Gain			
Canton	1^{st} measurement	2^{nd} measurement	3^{rd} measurement	4^{th} measurement
AG	1.356 (13-06)	0.434 (06-06)	0.292 (06-06)	0.210 (13-06)
AI	0.171 (13-06)	0.167 (09-06)	0.159 (09-06)	0.142 (08-06)
AR	0.207 (13-06)	0.169 (08-06)	0.159 (08-06)	0.142 (11-06)
BE	1.877 (13-06)	0.746 (06-06)	0.435 (06-06)	0.339 (13-06)
BL	0.712 (13-06)	0.236 (07-06)	0.186 (06-06)	0.156 (13-06)
BS	0.512 (13-06)	0.202 (06-06)	0.172 (06-06)	0.148 (13-06)
FR	0.973 (13-06)	0.330 (06-06)	0.208 (13-06)	0.184 (13-06)
GE	1.490 (13-06)	0.644 (06-06)	0.381 (13-06)	0.328 (13-06)
GL	0.189 (13-06)	0.168 (08-06)	0.159 (12-06)	0.142 (08-06)
GR	0.567 (13-06)	0.219 (06-06)	0.173 (06-06)	0.152 (13-06)
JU	0.255 (13-06)	0.173 (07-06)	0.161 (06-06)	0.143 (13-06)
LU	0.936 (13-06)	0.286 (07-06)	0.213 (06-06)	0.168 (13-06)
NE	0.576 (13-06)	0.222 (06-06)	0.172 (13-06)	0.153 (13-06)
NW	0.193 (13-06)	0.168 (07-06)	0.159 (06-06)	0.142 (08-06)
OW	0.187 (13-06)	0.167 (06-06)	0.159 (07-06)	0.142 (11-06)
SG	1.084 (13-06)	0.330 (06-06)	0.238 (06-06)	0.179 (13-06)
SH	0.247 (13-06)	0.172 (07-06)	0.161 (06-06)	0.142 (07-06)
SO	0.701 (13-06)	0.235 (06-06)	0.184 (06-06)	0.154 (13-06)
SZ	0.403 (13-06)	0.187 (07-06)	0.167 (06-06)	0.145 (13-06)
TG	0.651 (13-06)	0.223 (06-06)	0.183 (06-06)	0.153 (13-06)
TI	1.241 (13-06)	0.539 (06-06)	0.367 (13-06)	0.322 (13-06)
UR	0.186 (13-06)	0.167 (09-06)	0.158 (09-06)	0.142 (08-06)
VD	1.881 (13-06)	0.873 (06-06)	0.505 (13-06)	0.443 (13-06)
VS	1.138 (13-06)	0.420 (06-06)	0.256 (13-06)	0.223 (13-06)
ZG	0.337 (13-06)	0.180 (07-06)	0.163 (07-06)	0.144 (13-06)
ZH	2.092 (13-06)	0.862 (06-06)	0.592 (06-06)	0.276 (09-06)

TABLE 7: **Maximum expected information gain for monitoring of a second outbreak with uninformed b_3.** The corresponding optimal dates are shown in parenthesis.

Assume we want to estimate the proportion of a population with some margin of error d and a small risk α, i.e., we want $\Pr(|P - p| \geq d) = \alpha$. Here, the proportion corresponds to the proportion of unreported infected population. The minimum number of samples to achieve this is given by Cochran's formula [257],

$$n_0 = \frac{z_\alpha^2}{d^2} p(1-p)\,,$$

where z_α is the inverse of the standard normal cumulative distribution function evaluated at $1 - \alpha/2$. In this formula, we have assumed that the population is of infinite size. In order to correct for a finite size population N, we compute

$$n = \frac{n_0}{1 + n_0/N}\,.$$

In the next figure we present the minimum number of samples needed to sample the cantons of Switzerland for $d = 0.01$ and $\alpha = 0.01$. Notice that $\alpha = 0.01$ corresponds to a 99% confidence interval.

If the available test-kits are more than $26 \times 5950 = 154700$ then the maximum information gain will be achieved by deploying all tests uniformly in all cantons. However, when it is not realistic to conduct over 154700 tests, we consider testing with limited resources. For example assuming 30000 available tests, will be enough to test 5 cantons 5×5950. The question we answer then is which 5 cantons (from the 26) should we test given that we must test a minimum population of 5950 per canton?.

Distributing less than a particular number of tests (5950) in a canton will not provide a statistically reliable estimate for the number of unreported infections there. Thus, in such a case, the measured unreported infections should not be used to estimate the expected information gain.

Finally, we note that in this work we ignore the bias in the estimate of I^u. This means that the estimates of unreported infected enter the Bayesian framework without explicitly accounting for this known error.

	Maximum of Expected Information Gain			
Canton	1st measurement	2nd measurement	3rd measurement	4th measurement
AG	1.233 (17-07)	0.328 (10-07)	0.220 (17-07)	0.194 (10-07)
AI	0.170 (17-07)	0.167 (17-07)	0.154 (13-07)	0.136 (17-07)
AR	0.198 (17-07)	0.169 (10-07)	0.154 (14-07)	0.136 (16-07)
BE	1.596 (17-07)	0.460 (10-07)	0.347 (17-07)	0.267 (10-07)
BL	0.616 (17-07)	0.202 (10-07)	0.166 (17-07)	0.147 (10-07)
BS	0.441 (17-07)	0.184 (10-07)	0.159 (17-07)	0.141 (10-07)
FR	0.636 (17-07)	0.209 (10-07)	0.193 (17-07)	0.156 (17-07)
GE	0.896 (17-07)	0.326 (17-07)	0.308 (17-07)	0.225 (17-07)
GL	0.184 (17-07)	0.168 (13-07)	0.154 (17-07)	0.136 (10-07)
GR	0.418 (17-07)	0.182 (10-07)	0.162 (17-07)	0.141 (10-07)
JU	0.219 (17-07)	0.169 (10-07)	0.155 (17-07)	0.136 (17-07)
LU	0.834 (17-07)	0.234 (10-07)	0.178 (17-07)	0.157 (10-07)
NE	0.378 (17-07)	0.180 (10-07)	0.165 (17-07)	0.142 (17-07)
NW	0.187 (17-07)	0.168 (10-07)	0.154 (15-07)	0.135 (10-07)
OW	0.183 (17-07)	0.168 (10-07)	0.154 (10-07)	0.136 (12-07)
SG	0.994 (17-07)	0.267 (10-07)	0.191 (17-07)	0.169 (10-07)
SH	0.232 (17-07)	0.170 (10-07)	0.154 (15-07)	0.136 (11-07)
SO	0.581 (17-07)	0.197 (10-07)	0.166 (17-07)	0.145 (10-07)
SZ	0.362 (17-07)	0.178 (10-07)	0.157 (10-07)	0.139 (10-07)
TG	0.591 (17-07)	0.200 (10-07)	0.164 (17-07)	0.145 (10-07)
TI	0.556 (17-07)	0.318 (17-07)	0.296 (17-07)	0.224 (17-07)
UR	0.181 (17-07)	0.168 (11-07)	0.154 (16-07)	0.135 (16-07)
VD	1.297 (17-07)	0.452 (17-07)	0.433 (17-07)	0.281 (10-07)
VS	0.655 (17-07)	0.245 (17-07)	0.228 (17-07)	0.176 (17-07)
ZG	0.303 (17-07)	0.174 (10-07)	0.155 (15-07)	0.137 (11-07)
ZH	1.976 (17-07)	0.675 (10-07)	0.276 (14-07)	0.250 (13-07)

TABLE 8: **Maximum expected information gain to monitor a second outbreak with informed b_3.** The corresponding optimal dates are shown in parenthesis.

Parameter	Prior distribution / fixed value	Inference I	Inference II	Inference III
b_0	$\mathcal{U}([0.8, 1.8])$	yes	yes	yes
α	$\mathcal{U}([0.01, 1])$	yes	yes	yes
μ	$\mathcal{U}([0.2, 1])$	yes	yes	yes
Z	$\mathcal{U}([1, 6])$	yes	yes	yes
D	$\mathcal{U}([1, 6])$	yes	yes	yes
θ_0	$\mathcal{U}([0.5, 1.5])$	yes	yes	yes
c_1	$\mathcal{U}([0, 1])$	no	yes	yes
c_2	$\mathcal{U}([0, 1])$	no	yes	yes
c_3	$\mathcal{U}([0, 1])$	no	yes	yes
c_4	$\mathcal{U}([0, 1])$	no	yes	yes
δ_1	$\mathcal{U}([20, 30])$	no	yes	yes
δ_2	$\mathcal{U}([30, 40])$	no	yes	yes
δ_3	102	no	yes	yes
λ	$\mathcal{U}([0, 0.03])$	no	no	yes
r	$\mathcal{U}([0, 2])$	yes	yes	yes
$\boldsymbol{I}^u_{\mathrm{IC}}$	$\mathcal{U}([0, 50]^{12})$	yes	yes	yes

TABLE 9: **Parameters and prior distributions used in Bayesian inference.** Here the data corresponds to the daily reported infections. In all cases, data are used from the 25th of February 2020, when the first reported case was found in the canton of Ticino. Inference I uses data up to the day non-pharmaceutical interventions were announced (17th of March 2020). Inference II uses data up to the day measures were relaxed (6th of June 2020). Inference III uses data up to the 9th of July 2020. The choice of prior distributions is consistent with the choice found in [260]; the ranges used in our study are slightly extended.

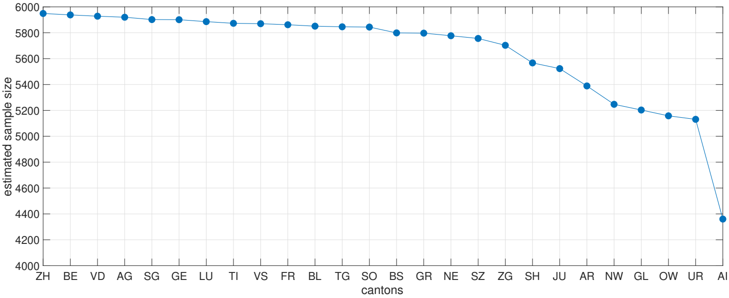


FIGURE 53: Estimated sample size using Cochran's [257] formula for every canton for confidence level 99%, margin of error 1% and probability of infection 0.1. The cantons are sorted in descending order of their population. The maximum sample size is estimated for Zurich and is equal to 5950. All the other cantons need up to 14% less samples with the exception of the smallest canton that needs 27% less samples.

B

APPENDIX DEEP REINFORCEMENT LEARNING

B.1 V-RACER for discrete action space

The V-RACER paper [33] presents the continuous action version of V-RACER. For discrete action environments we employ a neural network which takes as an input the state and outputs the state-value estimate, the energies of each action ϵ_i for $i = 1, \dots, |\mathcal{A}|$ and an inverse temperature β parameter. The probability p_i of sampling action a_i is calculated according to the Boltzmann distribution

$$p_i = \frac{\exp\left(-\epsilon_i\right)\beta}{\sum_{k=1}^{|\mathcal{A}|} \exp\left(-\epsilon_k\right)\beta}\,. \tag{136}$$

Here we omit the derivation of the importance weight and the gradient thereof, as well as the gradient of the Kullback-Leibler divergence of the current policy from the past policy.

B.2 Clipped Normal Distribution

We use the clipped normal distribution to enforce action bounds during the sampling of the actions in a truncated domain $[a, b] \subset \mathbb{R}$. The probability density function of the clipped normal distribution [274] is given by

$$\begin{aligned} f(x;\mu,\sigma) = {} & \mathbb{I}_{a<x<b} f_{\mathcal{N}}(x;\mu,\sigma) \\ & + \delta(x-a) F_{\mathcal{N}}(a;\mu,\sigma) + \delta(x-b)\left[1 - F_{\mathcal{N}}(b;\mu,\sigma)\right], \end{aligned} \tag{137}$$

where $\mathbb{I}_{a\leq x<b}$ is the indicator function that is 1 inside (a, b) and 0 outside, δ denotes the Dirac delta distribution, and $f_{\mathcal{N}}(x;\mu,\sigma)$ is the density function of the normal distribution with mean μ and standard deviation σ. In contrast to the squashed normal distribution [275], the clipped normal distribution retains a higher probability mass towards the action bounds. We found that clipping performs superior to applying a squashing function on the samples. In the following, we derive the gradient of the KL divergence, the importance weight, and the gradient thereof.

B.2.1 Importance Weight

The gradient of the importance weight

$$\mathrm{IW}(x;\mu_q,\sigma_q,\mu_p,\sigma_p) = \begin{cases} \frac{F_{\mathcal{N}}(a;\mu_q,\sigma_q)}{F_{\mathcal{N}}(a;\mu_p,\sigma_p)}\,, & \text{if } x = a\,, \\ \frac{f_{\mathcal{N}}(x;\mu_q,\sigma_q)}{f_{\mathcal{N}}(x;\mu_p,\sigma_p)}\,, & \text{if } a < x < b\,, \\ \frac{1-F_{\mathcal{N}}(b;\mu_q,\sigma_q)}{1-F_{\mathcal{N}}(b;\mu_p,\sigma_p)}\,, & \text{if } x = b\,, \end{cases} \tag{138}$$

with respect to the parameters μ_q and σ_q can be computed using

$$\begin{aligned} \frac{\partial f_{\mathcal{N}}(x;\mu,\sigma)}{\partial \mu_q} &= \frac{x-\mu}{\sigma^2} f_{\mathcal{N}}(x;\mu,\sigma)\,, \\ \frac{\partial f_{\mathcal{N}}(x;\mu,\sigma)}{\partial \sigma_q} &= \Big(-\frac{1}{\sigma} + \frac{(x-\mu)^2}{\sigma^3}\Big) f_{\mathcal{N}}(x;\mu,\sigma) \\ &= \frac{\mu^2-\sigma^2-2\mu x+x^2}{\sigma^3} f_{\mathcal{N}}(x;\mu,\sigma)\,, \end{aligned} \tag{139}$$

and eq. (149) as

$$\frac{\partial\,\mathrm{IW}}{\partial \mu_q} = \begin{cases} \frac{-f_{\mathcal{N}}(a;\mu_q,\sigma_q)}{F_{\mathcal{N}}(a;\mu_p,\sigma_p)}\,, & \text{if } x = a\,, \\ \frac{(x-\mu_q) f_{\mathcal{N}}(x;\mu_q,\sigma_q)}{\sigma_q^2 f_{\mathcal{N}}(x;\mu_p,\sigma_p)}\,, & \text{if } a < x < b\,, \\ \frac{f_{\mathcal{N}}(b;\mu_q,\sigma_q)}{1-F_{\mathcal{N}}(b;\mu_p,\sigma_p)}\,, & \text{if } x = b\,. \end{cases} \tag{140}$$

$$\frac{\partial\,\mathrm{IW}}{\partial \sigma_q} = \begin{cases} -\frac{a-\mu_q}{\sigma_q}\frac{f_{\mathcal{N}}(a;\mu_q,\sigma_q)}{F_{\mathcal{N}}(a;\mu_p,\sigma_p)}\,, & \text{if } x = a\,, \\ \frac{((x-\mu_q)^2-\sigma_q^2) f_{\mathcal{N}}(x;\mu_q,\sigma_q)}{\sigma_q^3 f_{\mathcal{N}}(x;\mu_p,\sigma_p)}\,, & \text{if } a < x < b\,, \\ \frac{b-\mu_q}{\sigma_q}\frac{f_{\mathcal{N}}(b;\mu_q,\sigma_q)}{1-F_{\mathcal{N}}(b;\mu_p,\sigma_p)}\,, & \text{if } x = b\,. \end{cases} \tag{141}$$

B.2.2 Kullback-Leibler Divergence

The definition of the KL divergence is given by

$$D_{\mathrm{KL}}(p\|q) = \int_{-\infty}^{\infty} \log\left[\frac{p(x;\mu_p,\sigma_p)}{q(x;\mu_q,\sigma_q)}\right] p(x;\mu_p,\sigma_p)\mathrm{d}x\,. \tag{142}$$

Plugging in the expression from eq. (137) we find

$$
\begin{aligned}
D_{\mathrm{KL}}(p\|q) = {} & \log\left[\frac{F_{\mathcal{N}}(a;\mu_p,\sigma_p)}{F_{\mathcal{N}}(a;\mu_q,\sigma_q)}\right] F_{\mathcal{N}}(a;\mu_p,\sigma_p) \\
& \underbrace{+\int\limits_a^b \log\left[\frac{f_{\mathcal{N}}(x;\mu_p,\sigma_p)}{f_{\mathcal{N}}(x;\mu_q,\sigma_q)}\right] f_{\mathcal{N}}(x;\mu_p,\sigma_p)\mathrm{d}x}_{I} \\
& + \log\left[\frac{1-F_{\mathcal{N}}(b;\mu_p,\sigma_p)}{1-F_{\mathcal{N}}(b;\mu_q,\sigma_q)}\right]\left[1-F_{\mathcal{N}}(b;\mu_p,\sigma_p)\right] .
\end{aligned}
\tag{143}
$$

Plugging in the expression for the normal distribution we can write the integral expression for $x \in (a,b)$ as

$$
\begin{aligned}
I = \frac{1}{\sqrt{2\pi}\sigma_p}\int_a^b & \left\{\log\left(\frac{\sigma_q}{\sigma_p}\right) - \frac{1}{2}\left[\frac{(x-\mu_p)^2}{\sigma_p^2} - \frac{(x-\mu_q)^2}{\sigma_q^2}\right]\right\} \\
& \cdot \exp\left[-\frac{(x-\mu_p)^2}{2\sigma_p^2}\right]\mathrm{d}x\,.
\end{aligned}
\tag{144}
$$

Using a substitution of variables $x' = \frac{x-\mu_p}{\sqrt{2}\sigma_p}$, the integral reads

$$
\begin{aligned}
I = \frac{1}{\sqrt{\pi}}\int_{\frac{a-\mu_p}{\sqrt{2}\sigma_p}}^{\frac{b-\mu_p}{\sqrt{2}\sigma_p}} & \underbrace{\left[\log\left(\frac{\sigma_q}{\sigma_p}\right) - x'^2 + \frac{1}{2}\frac{(\sqrt{2}\sigma_p x' + \mu_p - \mu_q)^2}{\sigma_q^2}\right]}_{Q} \\
& \cdot e^{-x'^2}\mathrm{d}x'\,.
\end{aligned}
\tag{145}
$$

We expand Q giving

$$
\begin{aligned}
Q = {} & \underbrace{\log\left(\frac{\sigma_q}{\sigma_p}\right) + \frac{(\mu_p-\mu_q)^2}{2\sigma_q^2}}_{C_1} \\
& + \underbrace{\sqrt{2}\sigma_p\frac{(\mu_p-\mu_q)}{\sigma_q^2}}_{C_2} x' - \underbrace{\left(1-\frac{\sigma_p^2}{\sigma_q^2}\right)}_{C_3} x'^2\,.
\end{aligned}
\tag{146}
$$

Using the identities

$$\begin{aligned}
\int_a^b e^{-x^2}\mathrm{d}x &= \frac{\sqrt{\pi}}{2}\left[\mathrm{erf}\left(b\right) - \mathrm{erf}\left(a\right)\right] \, , \\
\int_a^b x e^{-x^2}\mathrm{d}x &= -\frac{1}{2}\left[\exp\left(b^2\right) - \exp\left(a^2\right)\right] \, , \\
\int_a^b x^2 e^{-x^2}\mathrm{d}x &= \frac{\sqrt{\pi}}{4}\left[\mathrm{erf}(b) - \mathrm{erf}(a)\right] \\
&\quad - \frac{1}{2}\left[b\exp(-b^2) - a\exp(-a^2)\right] \, ,
\end{aligned} \tag{147}$$

the integration can be performed

$$\begin{aligned}
D_{KL}(p||q) = \log&\left[\frac{F_{\mathcal{N}}(a;\mu_p,\sigma_p)}{F_{\mathcal{N}}(a;\mu_q,\sigma_q)}\right] F_{\mathcal{N}}(a;\mu_p,\sigma_p) \\
&+ \frac{1}{2}\left[\log\left(\frac{\sigma_q}{\sigma_p}\right) + \frac{(\mu_p-\mu_q)^2}{2\sigma_q^2} - \frac{1}{2}\left(1 - \frac{\sigma_p^2}{\sigma_q^2}\right)\right] \\
&\cdot \left[\mathrm{erf}\left(\frac{b-\mu_p}{\sigma_p\sqrt{2}}\right) - \mathrm{erf}\left(\frac{a-\mu_p}{\sigma_p\sqrt{2}}\right)\right] \\
&+ \frac{1}{\sqrt{2\pi}}\left[\frac{1}{2}\left(1-\frac{\sigma_p^2}{\sigma_q^2}\right)\left(\frac{b-\mu_p}{\sigma_p}\right) - \frac{\sigma_p(\mu_p-\mu_q)}{\sigma_q^2}\right] \\
&\cdot \exp\left(-\frac{(b-\mu_p)^2}{2\sigma_p^2}\right) \\
&- \frac{1}{\sqrt{2\pi}}\left[\frac{1}{2}\left(1-\frac{\sigma_p^2}{\sigma_q^2}\right)\left(\frac{a-\mu_p}{\sigma_p}\right) - \frac{\sigma_p(\mu_p-\mu_q)}{\sigma_q^2}\right] \\
&\cdot \exp\left(-\frac{(a-\mu_p)^2}{2\sigma_p^2}\right) \\
&+ \log\left[\frac{1-F_{\mathcal{N}}(b;\mu_p,\sigma_p)}{1-F_{\mathcal{N}}(b;\mu_q,\sigma_q)}\right]\left[1-F_{\mathcal{N}}(b;\mu_p,\sigma_p)\right] \, .
\end{aligned} \tag{148}$$

Using the derivatives of the cumulative distribution function with respect to the mean and standard deviation

$$\begin{aligned}
\frac{F_{\mathcal{N}}(x;\mu,\sigma)}{\partial\mu_q} &= -f_{\mathcal{N}}(x;\mu,\sigma)\, , \\
\frac{F_{\mathcal{N}}(x;\mu,\sigma)}{\partial\sigma_q} &= -\frac{x-\mu}{\sigma} f_{\mathcal{N}}(x;\mu,\sigma)\, ,
\end{aligned} \tag{149}$$

and the gradient of the error function

$$\frac{\partial}{\partial z}\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} e^{-z^2}\,, \tag{150}$$

the derivative of the KL divergence with respect to μ_q

$$\begin{aligned}\frac{\partial D_{\mathrm{KL}}(p\|q)}{\partial \mu_q} &= \frac{F_{\mathcal{N}}(a;\mu_p,\sigma_p)}{F_{\mathcal{N}}(a;\mu_q,\sigma_q)} f_{\mathcal{N}}(a;\mu_q,\sigma_q) \\ &- \frac{1}{2}\frac{\mu_p-\mu_q}{\sigma_q^2}\Big[\operatorname{erf}\left(\frac{b-\mu_p}{\sqrt{2}\sigma_p}\right) \\ &- \operatorname{erf}\left(\frac{a-\mu_p}{\sqrt{2}\sigma_p}\right)\Big] \\ &+ \frac{1}{\sqrt{2\pi}}\frac{\sigma_p}{\sigma_q^2}\Big[\exp\left(-\frac{(b-\mu_p)^2}{2\sigma_p^2}\right) \\ &- \exp\left(-\frac{(a-\mu_p)^2}{2\sigma_p^2}\right)\Big] \\ &- \frac{1-F_{\mathcal{N}}(b;\mu_p,\sigma_p)}{1-F_{\mathcal{N}}(b;\mu_q,\sigma_q)} f_{\mathcal{N}}(b;\mu_q,\sigma_q)\end{aligned} \tag{151}$$

and σ_q is given by

$$\begin{aligned}
\frac{\partial D_{\mathrm{KL}}(p\|q)}{\partial \sigma_q} = {} & \frac{a-\mu_q}{\sigma_q}\frac{F_{\mathcal{N}}(a;\mu_p,\sigma_p)}{F_{\mathcal{N}}(a;\mu_q,\sigma_q)} f_{\mathcal{N}}(a;\mu_q,\sigma_q) \\
& + \frac{1}{2}\left[\frac{1}{\sigma_q} - \frac{(\mu_p-\mu_q)^2}{\sigma_q^3} - \frac{\sigma_p^2}{\sigma_q^3}\right] \\
& \cdot \left[\operatorname{erf}\left(\frac{b-\mu_p}{\sigma_p\sqrt{2}}\right) - \operatorname{erf}\left(\frac{a-\mu_p}{\sigma_p\sqrt{2}}\right)\right] \\
& + \frac{1}{\sqrt{2\pi}}\left[\frac{\sigma_p^2}{\sigma_q^3}\left(\frac{b-\mu_p}{\sigma_p}\right) + \frac{2\sigma_p(\mu_p-\mu_q)}{\sigma_q^3}\right] \\
& \cdot \exp\left(-\frac{(b-\mu_p)^2}{2\sigma_p^2}\right) \\
& - \frac{1}{\sqrt{2\pi}}\left[\frac{\sigma_p^2}{\sigma_q^3}\left(\frac{a-\mu_p}{\sigma_p}\right) + \frac{2\sigma_p(\mu_p-\mu_q)}{\sigma_q^3}\right] \\
& \cdot \exp\left(-\frac{(a-\mu_p)^2}{2\sigma_p^2}\right) \\
& - \frac{b-\mu_q}{\sigma_q}\frac{1-F_{\mathcal{N}}(b;\mu_p,\sigma_p)}{1-F_{\mathcal{N}}(b;\mu_q,\sigma_q)} f_{\mathcal{N}}(b;\mu_q,\sigma_q)\,.
\end{aligned} \tag{152}$$

B.3 Algorithms

Algorithm 1 Multi-Agent Reinforcement Learning (Synchronous Variant)

```
Input : Environment function $D(\boldsymbol{s}, \boldsymbol{a})$, Replay Memory $\mathcal{RM}$, Neural Net-
        work(s) $NN$ for agents $i = 1, \dots, N$, Termination Criteria $\mathcal{T}$
while not $\mathcal{T}$ do
    // COLLECTING EXPERIENCES
    Sample Initial States $s_0^{(i)}$
    for $t \in 1, \dots, T$ do
        Forward Neural Networks $V^\pi(s_{t-1}^{(i)}), \pi_{t-1}(\cdot|s_{t-1}^{(i)}) = NN(s_{t-1}^{(i)}; \boldsymbol{\vartheta}, \boldsymbol{\omega})$
        Sample Actions $a_{t-1}^{(i)} \sim \pi_{t-1}(\cdot|s_{t-1}^{(i)})$ for $i = 1, \dots, N$
        Run Environment Function $\boldsymbol{r}_{t-1}, \boldsymbol{s}_t = D(\boldsymbol{a}_{t-1}, \boldsymbol{s}_{t-1})$
    end
    // POSTPROCESS EPISODE
    Save Episode $\mathcal{E}_k = \{(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{r}_t, \boldsymbol{V}_t, \boldsymbol{\pi}_t)\}_{t=0}^{T_k}$ in Replay Memory $\mathcal{RM}$
    Set $\hat{V}_T^{\text{tbc},(i)} = V^\pi(s_T^{(i)})$
    for $t \in T-1, \dots, 1$ do
        Compute $\hat{V}_t^{\text{tbc},(i)} = V^\pi(s_t^{(i)}) + \bar{\rho}_t \left[ f(\boldsymbol{r}_t) + \gamma \hat{V}_{t+1}^{\text{tbc},(i)} - V^\pi(s_t^{(i)}) \right]$
        Add $\hat{V}_t^{\text{tbc},(i)}$ to Replay Memory $\mathcal{RM}$
    end
    // TRAINING NEURAL NETWORK
    for $i \in 1, \dots, T$ do
        miniBatch = generateMinibatch()
        trainPolicy( miniBatch )
        updateReFERparams()
    end
end
```

Here f implements the relation between the agents. The function UPDATEREFERPARAMS updates the ReF-ER parameters, GENERATEMINIBATCH samples a minibatch of experiences from the replay memory. TRAINPOLICY implements the learning algorithm 2. The functions and COMPUTEIMPORTANCEWEIGHT and COMPUTEPOLICYGRADIENT implement the respective variants described in the main text. The function isOnPolicy classifies the experience as on- or off-policy.

Algorithm 2 trainPolicy

```
Input: miniBatch
$\hat{\boldsymbol{g}}^V(\boldsymbol{\vartheta}) = \hat{\boldsymbol{g}}^\pi(\boldsymbol{\omega}) = \mathbf{0}$
for $(\boldsymbol{s}_b, \boldsymbol{a}_b, \boldsymbol{r}_b, \boldsymbol{\pi}_b) \in miniBatch$ do
    // Forward Neural Network
    $V^\pi_{\boldsymbol{\vartheta}}(s_b^{(i)}), \pi_\omega(s_b^{(i)}) = NN(s_b^{(i)}; \boldsymbol{\vartheta}, \boldsymbol{\omega})$
    // Backwards Update of Value Estimator
    $b_{\text{first}} \leftarrow$ find first experience for episode containing $b$
    for $t \in b, \dots, b_{first}$ do
        $V_f^\pi(s_t^{(i)}) \leftarrow$ scalarize value from replay memory
        $\hat{V}_t^{\text{tbc},(i)} = V_f^\pi(s_t^{(i)}) + \bar{\rho}_t \left[ f(\boldsymbol{r}_t) + \gamma \hat{V}_{t+1}^{\text{tbc},(i)} - V_f^\pi(s_t^{(i)}) \right]$
    end
    // Compute Importance Weight
    importanceWeight = computeImportanceWeight($\boldsymbol{a}_b, \boldsymbol{\pi}_b, \boldsymbol{\pi}_{\boldsymbol{\vartheta}}$)
    // Compute Value Gradient
    $V_f^\pi(s_b^{(i)}) \leftarrow$ scalarize value $V_{\boldsymbol{\vartheta}}^\pi(s_b^{(i)})$
    $\hat{\boldsymbol{g}}_b^{V,(i)}(\boldsymbol{\vartheta}) = \frac{1}{N} \left[ V_f^\pi(s_b^{(i)}) - \hat{V}_b^{\text{tbc},(i)} \right] \nabla_{\boldsymbol{\vartheta}} V_\omega^\pi(s_b^{(i)})$
    // Compute Policy Gradient
    $\hat{\boldsymbol{g}}_b^{\text{KL},(i)}(\omega) = \nabla_\omega D_{\text{KL}} \left( \pi_b \| \pi_\omega \right) (s_b^{(i)})$
    if isOnPolicy(importanceWeight) then
        $\hat{\boldsymbol{g}}_b^{(i)}(\omega)$ =computePolicyGradient(importanceWeight, $V_{\boldsymbol{\vartheta}}^\pi(s_b^{(i)}), \pi_\omega(s_b^{(i)})$)
    else
        $\hat{\boldsymbol{g}}_b^{(i)}(\omega) = \mathbf{0}$
    end
    // Accumulate Gradient for Value Function
    $\hat{\boldsymbol{g}}^V(\boldsymbol{\vartheta}) = \hat{\boldsymbol{g}}^V(\boldsymbol{\vartheta}) + \frac{1}{N \cdot |miniBatch|} \hat{\boldsymbol{g}}_b^{V,(i)}(\boldsymbol{\vartheta})$
    // Accumulate Gradient for Policy
    $\hat{\boldsymbol{g}}^\pi(\omega) = \hat{\boldsymbol{g}}^\pi(\omega) + \frac{1}{N \cdot |miniBatch|} (\beta \hat{\boldsymbol{g}}_b^{(i)}(\omega) + (1 - \beta) \hat{\boldsymbol{g}}_b^{\text{KL},(i)}(\omega))$
```

```
end
// UPDATE HYPERPARAMETERS
$\boldsymbol{\vartheta} \leftarrow \boldsymbol{\vartheta} - \eta \hat{\boldsymbol{g}}^{V}(\boldsymbol{\vartheta})$
$\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} - \eta \hat{\boldsymbol{g}}^{\pi}(\boldsymbol{\omega})$
```

BIBLIOGRAPHY

1. Chatzimanolakis, M., Weber, P., Wermelinger, F. & Koumoutsakos, P. *CubismAMR – A C++ library for Distributed Block-Structured Adaptive Mesh Refinement* 2022.
2. Chatzimanolakis, M., Weber, P. & Koumoutsakos, P. Vortex separation cascades in simulations of the planar flow past an impulsively started cylinder up to $\boldsymbol{Re} = \mathbf{100\,000}$. *Journal of Fluid Mechanics* **953**, R2 (2022).
3. Angot, P., Bruneau, C. H. & Fabrie, P. A penalization method to take into account obstacles in incompressible viscous flows. *Numerische Mathematik* **81**, 497 (1999).
4. Coquerelle, M. & Cottet, G.-H. A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies. *Journal of Computational Physics* **227**. Special Issue Celebrating Tony Leonard's 70th Birthday, 9121 (2008).
5. Bost, C., Cottet, G.-H. & Maitre, E. Convergence Analysis of a Penalization Method for the Three-Dimensional Motion of a Rigid Body in an Incompressible Viscous Fluid. *SIAM Journal on Numerical Analysis* **48**, 1313 (2010).
6. Verma, S., Abbati, G., Novati, G. & Koumoutsakos, P. Computing the force distribution on the surface of complex, deforming geometries using vortex methods and Brinkman penalization. *International Journal for Numerical Methods in Fluids* **85**, 484 (2017).
7. Carling, J., Williams, T. L. & Bowtell, G. Self-propelled anguilliform swimming: simultaneous solution of the two-dimensional navier-stokes equations and Newton's laws of motion. *Journal of Experimental Biology* **201**, 3143 (1998).
8. Kern, S. & Koumoutsakos, P. Simulations of optimized anguilliform swimming. *Journal of Experimental Biology* **209**, 4841 (2006).
9. Gazzola, M., Chatelain, P., van Rees, W. M. & Koumoutsakos, P. Simulations of single and multiple swimmers with non-divergence free deforming geometries. *Journal of Computational Physics* **230**, 7093 (2011).

10. Chorin, A. J. Numerical Solution of the Navier-Stokes Equations. *Mathematics of Computation* **22**, 745 (1968).
11. Towers, J. D. Finite difference methods for approximating Heaviside functions. *Journal of Computational Physics* **228**, 3478 (2009).
12. Shu, C.-W. in *High-Order Methods for Computational Physics* 439 (Springer Berlin Heidelberg, Berlin, Heidelberg, 1999).
13. Ueda, Y. & Kida, T. Asymptotic analysis of initial flow around an impulsively started circular cylinder using a Brinkman penalization method. *Journal of Fluid Mechanics* **929**, A31 (2021).
14. Martin, D., Colella, P. & Graves, D. A Cell-Centered Adaptive Projection Method for the IncompressibleNavier-Stokes Equations in Three Dimensions. *Journal of Computational Physics* **227**, 1863 (2008).
15. Van der Vorst, H. A. Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems. *SIAM Journal on Scientific and Statistical Computing* **13**, 631 (1992).
16. Hejazialhosseini, B., Rossinelli, D., Conti, C. & Koumoutsakos, P. *High throughput software for direct numerical simulations of compressible two-phase flows* in *SC '12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis* (2012), 1.
17. Rossinelli, D., Hejazi, B., Hadjidoukas, P., Bekas, C., Curioni, A., Bertsch, A., Futral, S., Schmidt, S., Adams, N. & Koumoutsakos, P. *11 PFLOP/s simulations of cloud cavitation collapse* in (2013).
18. Hadjidoukas, P. E., Rossinelli, D., Hejazialhosseini, B. & Koumoutsakos, P. *From 11 to 14.4 PFLOPs: Performance Optimization for Finite Volume Flow Solver* in *Proceedings of the 3rd International Conference on Exascale Applications and Software* (University of Edinburgh, Edinburgh, UK, 2015), 7.
19. Rossinelli, D., Hejazialhosseini, B., Spampinato, D. G. & Koumoutsakos, P. Multicore/Multi-GPU Accelerated Simulations of Multiphase Compressible Flows Using Wavelet Adapted Grids. *SIAM J. Sci. Comput.* **33**, 512 (2011).
20. Vasilyev, O. V. & Kevlahan, N. K.-R. An adaptive multilevel wavelet collocation method for elliptic problems. *Journal of Computational Physics* **206**, 412 (2005).
21. Harten, A. Adaptive Multiresolution Schemes for Shock Computations. *Journal of Computational Physics* **115**, 319 (1994).

22. Harten, A. Multiresolution Representation of Data: A General Framework. *SIAM Journal on Numerical Analysis* **33**, 1205 (1996).
23. Rossinelli, D., Hejazialhosseini, B., van Rees, W., Gazzola, M., Bergdorf, M. & Koumoutsakos, P. MRAG-I2D: Multi-resolution adapted grids for remeshed vortex methods on multicore architectures. *J. Comput. Phys.* **288**, 1 (2015).
24. Adams, M., Colella, P., Graves, D. T., Johnson, J., Keen, N., McCorquodale, T. J. L. D. F. M. P., Schwartz, D. M. P., Sternberg, T. & Straalen, B. V. Chombo Software Package for AMR Applications - Design Document. *Lawrence Berkeley National Laboratory Technical Report* **LBNL-6616E** (2019).
25. Almgren, A. S., Bell, J. B., Colella, P., Howell, L. H. & Welcome, M. L. A Conservative Adaptive Projection Method for the Variable Density Incompressible Navier–Stokes Equations. *Journal of Computational Physics* **142**, 1 (1998).
26. Berger, M. J. & Oliger, J. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics* **53**, 484 (1984).
27. Berger, M. & Colella, P. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics* **82**, 64 (1989).
28. Hilbert, D. in *Dritter Band: Analysis · Grundlagen der Mathematik · Physik Verschiedenes: Nebst Einer Lebensgeschichte* 1 (Springer Berlin Heidelberg, Berlin, Heidelberg, 1935).
29. Cybenko, G. Dynamic load balancing for distributed memory multiprocessors. *Journal of Parallel and Distributed Computing* **7**, 279 (1989).
30. Collins, W. M. & Dennis, S. C. R. Flow past an impulsively started circular cylinder. *Journal of Fluid Mechanics* **60**, 105 (1973).
31. Koumoutsakos, P. & Leonard, A. High-resolution simulations of the flow around an impulsively started cylinder using vortex methods. *Journal of Fluid Mechanics* **296**, 1 (1995).
32. Bergdorf, M. & Koumoutsakos, P. A Lagrangian Particle-Wavelet Method. *mms* **5** (2006).
33. Novati, G. & Koumoutsakos, P. *Remember and Forget for Experience Replay* in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA* (eds Chaudhuri, K. & Salakhutdinov, R.) **97** (PMLR, 2019), 4851.

34. Martin, S. M., Wälchli, D., Arampatzis, G., Economides, A. E., Karnakov, P. & Koumoutsakos, P. Korali: Efficient and scalable software framework for Bayesian uncertainty quantification and stochastic optimization. *Comput. Method. Appl. M.*, 114264 (2021).
35. Weber, P., Wälchli, D., Zeqiri, M. & Koumoutsakos, P. *Remember and Forget Experience Replay for Multi-Agent Reinforcement Learning* 2022.
36. Weber, P., Wälchli, D. & Koumoutsakos, P. *Quantification of Uncertainties in Deep Reinforcement Learning for Scientific Machine Learning* 2023.
37. Munos, R., Stepleton, T., Harutyunyan, A. & Bellemare, M. G. *Safe and Efficient Off-Policy Reinforcement Learning* in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain* (eds Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I. & Garnett, R.) (2016), 1046.
38. Wang, Z., Bapst, V., Heess, N., Mnih, V., Munos, R., Kavukcuoglu, K. & de Freitas, N. Sample Efficient Actor-Critic with Experience Replay. *CoRR* **abs/1611.01224** (2016).
39. Degris, T., White, M. & Sutton, R. S. Off-Policy Actor-Critic. *CoRR* **abs/1205.4839** (2012).
40. Schulman, J., Levine, S., Abbeel, P., Jordan, M. I. & Moritz, P. *Trust Region Policy Optimization* in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015* (eds Bach, F. R. & Blei, D. M.) **37** (JMLR.org, 2015), 1889.
41. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M. A., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. & Hassabis, D. Human-level control through deep reinforcement learning. *Nat.* **518**, 529 (2015).
42. Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T. P., Simonyan, K. & Hassabis, D. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *CoRR* **abs/1712.01815** (2017).
43. Brown, N. & Sandholm, T. Superhuman AI for multiplayer poker. *Science* **365**, 885 (2019).

44. OpenAI, : Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., Józefowicz, R., Gray, S., Olsson, C., Pachocki, J., Petrov, M., Pinto, H. P. d. O., Raiman, J., Salimans, T., Schlatter, J., Schneider, J., Sidor, S., Sutskever, I., Tang, J., Wolski, F. & Zhang, S. Dota 2 with Large Scale Deep Reinforcement Learning. *arXiv e-prints*, arXiv:1912.06680 (2019).
45. Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., Vezhnevets, A. S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T. L., Gulcehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Kavukcuoglu, K., Hassabis, D., Apps, C. & Silver, D. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* **575**, 350 (2019).
46. Novati, G., de Laroussilhe, H. L. & Koumoutsakos, P. Automating turbulence modelling by multi-agent reinforcement learning. *Nat. Mach. Intell.* **3**, 87 (2021).
47. Bae, H. J. & Koumoutsakos, P. Scientific multi-agent reinforcement learning for wall-models of turbulent flows. *Nature Communications* **13**, 1 (2022).
48. Gupta, J. K., Egorov, M. & Kochenderfer, M. J. *Cooperative Multi-agent Control Using Deep Reinforcement Learning* in *Autonomous Agents and Multiagent Systems - AAMAS 2017 Workshops, Best Papers, São Paulo, Brazil, May 8-12, 2017, Revised Selected Papers* (eds Sukthankar, G. & Rodríguez-Aguilar, J. A.) **10642** (Springer, 2017), 66.
49. Terry, J. K., Grammel, N., Hari, A., Santos, L. & Black, B. Revisiting parameter sharing in multi-agent deep reinforcement learning. *arXiv preprint arXiv:2005.13625* (2020).
50. Oliehoek, F. A., Spaan, M. T. J. & Vlassis, N. Optimal and Approximate Q-value Functions for Decentralized POMDPs. *J. Artif. Intell. Res.* **32**, 289 (2008).
51. Kraemer, L. & Banerjee, B. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing* **190**, 82 (2016).

52. Kingma, D. P. & Ba, J. *Adam: A Method for Stochastic Optimization* in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (eds Bengio, Y. & LeCun, Y.) (2015).
53. Terry, J. K., Black, B., Grammel, N., Jayakumar, M., Hari, A., Sulivan, R., Santos, L., Perez, R., Horsch, C., Dieffendahl, C., Williams, N. L., Lokesh, Y., Sullivan, R. & Ravi, P. PettingZoo: Gym for Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:2009.14471* (2020).
54. Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. Proximal Policy Optimization Algorithms. *CoRR* **abs/1707.06347** (2017).
55. Horgan, D., Quan, J., Budden, D., Barth-Maron, G., Hessel, M., van Hasselt, H. & Silver, D. Distributed Prioritized Experience Replay. *CoRR* **abs/1803.00933** (2018).
56. Verma, S., Novati, G. & Koumoutsakos, P. Efficient collective swimming by harnessing vortices through deep reinforcement learning. *Proceedings of the National Academy of Sciences* **115**, 5849 (2018).
57. Yu, H., Liu, B., Wang, C., Liu, X., Lu, X.-Y. & Huang, H. Deep-reinforcement-learning-based self-organization of freely undulatory swimmers. *Phys. Rev. E* **105**, 045105 (4 2022).
58. Gal, Y., Koumoutsakos, P., Lanusse, F., Louppe, G. & Papadimitriou, C. Bayesian uncertainty quantification for machine-learned models in physics. *Nature Reviews Physics* **4**, 573 (2022).
59. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *nature* **521**, 436 (2015).
60. Baldi, P. *Deep Learning in Science* (Cambridge University Press, 2021).
61. Hüllermeier, E. & Waegeman, W. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning* **110**, 457 (2021).
62. Jospin, L. V., Laga, H., Boussaid, F., Buntine, W. & Bennamoun, M. Hands-On Bayesian Neural Networks—A Tutorial for Deep Learning Users. *IEEE Computational Intelligence Magazine* **17**, 29 (2022).
63. Magris, M. & Iosifidis, A. Bayesian Learning for Neural Networks: an algorithmic survey. *arXiv e-prints*, arXiv:2211.11865 (2022).

64. Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., Makarenkov, V. & Nahavandi, S. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion* **76**, 243 (2021).
65. Osband, I., Aslanides, J. & Cassirer, A. *Randomized Prior Functions for Deep Reinforcement Learning* in *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Curran Associates Inc., Montréal, Canada, 2018), 8626.
66. Dulac-Arnold, G., Mankowitz, D. & Hester, T. Challenges of Real-World Reinforcement Learning. *arXiv e-prints*, arXiv:1904.12901 (2019).
67. Wiering, M. A. & van Hasselt, H. Ensemble Algorithms in Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **38**, 930 (2008).
68. Fausser, S. & Schwenker, F. Neural Network Ensembles in Reinforcement Learning. *Neural Process. Lett.* **41**, 55 (2015).
69. Osband, I., Blundell, C., Pritzel, A. & Roy, B. V. Deep Exploration via Bootstrapped DQN. *CoRR* **abs/1602.04621** (2016).
70. Xiliang, C., Cao, L., Li, C.-x., Xu, Z.-x. & Lai, J. Ensemble Network Architecture for Deep Reinforcement Learning. *Mathematical Problems in Engineering* **2018**, 1 (2018).
71. Lee, K., Laskin, M., Srinivas, A. & Abbeel, P. SUNRISE: A Simple Unified Framework for Ensemble Learning in Deep Reinforcement Learning. *arXiv e-prints*, arXiv:2007.04938 (2020).
72. An, G., Moon, S., Kim, J.-H. & Song, H. O. *Uncertainty-Based Offline Reinforcement Learning with Diversified Q-Ensemble* in *NeurIPS* (2021), 7436.
73. Saphal, R., Ravindran, B., Mudigere, D., Avancha, S. & Kaul, B. *SEERL: Sample Efficient Ensemble Reinforcement Learning* in *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems* (International Foundation for Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, 2021), 1100.

74. Sheikh, H., Frisbee, K. & Phielipp, M. *DNS: Determinantal Point Process Based Neural Network Sampler for Ensemble Reinforcement Learning* in *Proceedings of the 39th International Conference on Machine Learning* (eds Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G. & Sabato, S.) **162** (PMLR, 2022), 19731.
75. Chen, L., Zhou, X., Chang, C., Yang, R. & Yu, K. *Agent-Aware Dropout DQN for Safe and Efficient On-line Dialogue Policy Learning* in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Copenhagen, Denmark, 2017), 2454.
76. Lütjens, B., Everett, M. & How, J. P. Safe Reinforcement Learning with Model Uncertainty Estimates. *arXiv e-prints*, arXiv:1810.08700 (2018).
77. Hiraoka, T., Imagawa, T., Hashimoto, T., Onishi, T. & Tsuruoka, Y. *Dropout Q-Functions for Doubly Efficient Reinforcement Learning* in *International Conference on Learning Representations* (2022).
78. Hausknecht, M. & Wagener, N. Consistent Dropout for Policy Gradient Reinforcement Learning. *arXiv e-prints*, arXiv:2202.11818 (2022).
79. Kamalaruban, P., Huang, Y., Hsieh, Y., Rolland, P., Shi, C. & Cevher, V. Robust Reinforcement Learning via Adversarial training with Langevin Dynamics. *CoRR* **abs/2002.06063** (2020).
80. Madhushani, U., Dey, B., Leonard, N. E. & Chakraborty, A. Hamiltonian Q-Learning: Leveraging Importance-sampling for Data Efficient RL. *CoRR* **abs/2011.05927** (2020).
81. Xu, D. & Fekri, F. Improving Actor-Critic Reinforcement Learning via Hamiltonian Policy. *CoRR* **abs/2103.12020** (2021).
82. Clements, W. R., Van Delft, B., Robaglia, B.-M., Bahi Slaoui, R. & Toth, S. Estimating Risk and Uncertainty in Deep Reinforcement Learning. *arXiv e-prints*, arXiv:1905.09638 (2019).
83. Sedlmeier, A., Gabor, T., Phan, T., Belzner, L. & Linnhoff-Popien, C. Uncertainty-Based Out-of-Distribution Detection in Deep Reinforcement Learning. *arXiv e-prints*, arXiv:1901.02219 (2019).
84. Parvez Mohammed, A. & Valdenegro-Toro, M. Benchmark for Out-of-Distribution Detection in Deep Reinforcement Learning. *arXiv e-prints*, arXiv:2112.02694 (2021).

85. Charpentier, B., Senanayake, R., Kochenderfer, M. & Günnemann, S. Disentangling Epistemic and Aleatoric Uncertainty in Reinforcement Learning. *arXiv e-prints*, arXiv:2206.01558 (2022).

86. Bykovets, E., Metz, Y., El-Assady, M., Keim, D. A. & Buhmann, J. M. How to Enable Uncertainty Estimation in Proximal Policy Optimization. *arXiv e-prints*, arXiv:2210.03649 (2022).

87. Todorov, E., Erez, T. & Tassa, Y. *MuJoCo: A physics engine for model-based control* in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2012), 5026.

88. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J. & Zaremba, W. *OpenAI Gym* 2016.

89. Sutton, R. S. & Barto, A. G. *Reinforcement learning: An introduction* (MIT press, 2018).

90. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* **15**, 1929 (2014).

91. Wan, L., Zeiler, M., Zhang, S., Le Cun, Y. & Fergus, R. *Regularization of Neural Networks using DropConnect* in *Proceedings of the 30th International Conference on Machine Learning* (eds Dasgupta, S. & McAllester, D.) **28** (PMLR, Atlanta, Georgia, USA, 2013), 1058.

92. Gal, Y. & Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. *arXiv e-prints*, arXiv:1506.02142 (2015).

93. Mobiny, A., Yuan, P., Moulik, S. K., Garg, N., Wu, C. C. & Van Nguyen, H. Dropconnect is effective in modeling uncertainty of bayesian deep networks. *Scientific reports* **11**, 1 (2021).

94. Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D. & Wilson, A. G. Averaging Weights Leads to Wider Optima and Better Generalization. *arXiv e-prints*, arXiv:1803.05407 (2018).

95. Maddox, W., Garipov, T., Izmailov, P., Vetrov, D. & Wilson, A. G. A Simple Baseline for Bayesian Uncertainty in Deep Learning. *arXiv e-prints*, arXiv:1902.02476 (2019).

96. Lakshminarayanan, B., Pritzel, A. & Blundell, C. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. *arXiv e-prints*, arXiv:1612.01474 (2016).

97. Wilson, A. G. & Izmailov, P. Bayesian Deep Learning and a Probabilistic Perspective of Generalization. *arXiv e-prints*, arXiv:2002.08791 (2020).

98. Durasov, N., Bagautdinov, T., Baque, P. & Fua, P. Masksembles for Uncertainty Estimation. *arXiv e-prints*, arXiv:2012.08334 (2020).

99. Mandt, S., Hoffman, M. D. & Blei, D. M. Stochastic Gradient Descent as Approximate Bayesian Inference. *arXiv e-prints*, arXiv:1704.04289 (2017).

100. Kingma, D. P. & Ba, J. *Adam: A Method for Stochastic Optimization* in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (eds Bengio, Y. & LeCun, Y.) (2015).

101. Welling, M. & Teh, Y. W. *Bayesian Learning via Stochastic Gradient Langevin Dynamics* in *Proceedings of the 28th International Conference on International Conference on Machine Learning* (Omnipress, Bellevue, Washington, USA, 2011), 681.

102. Whye Teh, Y., Thiéry, A. & Vollmer, S. Consistency and fluctuations for stochastic gradient Langevin dynamics. *arXiv e-prints*, arXiv:1409.0578 (2014).

103. Girolami, M. & Calderhead, B. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **73**, 123 (2011).

104. Ahn, S., Korattikara, A. & Welling, M. Bayesian Posterior Sampling via Stochastic Gradient Fisher Scoring. *arXiv e-prints*, arXiv:1206.6380 (2012).

105. Li, C., Chen, C., Carlson, D. & Carin, L. Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks. *arXiv e-prints*, arXiv:1512.07666 (2015).

106. Kim, S., Song, Q. & Liang, F. Stochastic gradient Langevin dynamics with adaptive drifts. *Journal of Statistical Computation and Simulation* **92**, 318 (2022).

107. Duane, S., Kennedy, A., Pendleton, B. J. & Roweth, D. Hybrid Monte Carlo. *Physics Letters B* **195**, 216 (1987).

108. Neal, R. M. *Bayesian Learning for Neural Networks* (Springer-Verlag, Berlin, Heidelberg, 1996).

109. Neal, R. in *Handbook of Markov Chain Monte Carlo* 113 (Chapman and Hall/CRC, 2011).
110. Chen, T., Fox, E. B. & Guestrin, C. Stochastic Gradient Hamiltonian Monte Carlo. *arXiv e-prints*, arXiv:1402.4102 (2014).
111. Ma, Y.-A., Chen, T. & Fox, E. *A Complete Recipe for Stochastic Gradient MCMC* in *Advances in Neural Information Processing Systems* (eds Cortes, C., Lawrence, N., Lee, D., Sugiyama, M. & Garnett, R.) **28** (Curran Associates, Inc., 2015).
112. Lu, X., Perrone, V., Hasenclever, L., Whye Teh, Y. & Vollmer, S. J. Relativistic Monte Carlo. *arXiv e-prints*, arXiv:1609.04388 (2016).
113. Ray, S. & Lindsay, B. G. The topography of multivariate normal mixtures. *arXiv Mathematics e-prints*, math/0602238 (2006).
114. Hershey, J. R. & Olsen, P. A. *Approximating the Kullback Leibler Divergence Between Gaussian Mixture Models* in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07* **4** (2007), IV-317-IV.
115. Durrieu, J.-L., Thiran, J.-P. & Kelly, F. *Lower and upper bounds for approximation of the Kullback-Leibler divergence between Gaussian Mixture Models* in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2012), 4833.
116. Queeney, J., Paschalidis, Y. & Cassandras, C. G. *Generalized Proximal Policy Optimization with Sample Reuse* in *Advances in Neural Information Processing Systems* (eds Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P. & Vaughan, J. W.) **34** (Curran Associates, Inc., 2021), 11909.
117. Rabault, J. & Kuhnle, A. Accelerating deep reinforcement learning strategies of flow control through a multi-environment approach. *Physics of Fluids* **31**, 094105 (2019).
118. Kurz, M., Offenhäuser, P., Viola, D., Shcherbakov, O., Resch, M. & Beck, A. Deep Reinforcement Learning for Computational Fluid Dynamics on HPC Systems. *arXiv e-prints*, arXiv:2205.06502 (2022).
119. Weber, P., Arampatzis, G., Novati, G., Verma, S., Papadimitriou, C. & Koumoutsakos, P. Optimal Flow Sensing for Schooling Swimmers. *Biomimetics* **5** (2020).

120. Chatzimanolakis, M., Weber, P., Arampatzis, G., Wälchli, D., Kičić, I., Karnakov, P., Papadimitriou, C. & Koumoutsakos, P. Optimal allocation of limited test resources for the quantification of COVID-19 infections. *Swiss Med. Wkly.* (2020).
121. Papadimitriou, C. & Lombaert, G. The effect of prediction error correlation on optimal sensor placement in structural dynamics. *Mech. Syst. Signal Process.* **28**, 105 (2012).
122. Ryan, K. J. Estimating Expected Information Gains for Experimental Designs With Application to the Random Fatigue-Limit Model. *J. Comput. Graph. Stat.* **12**, 585 (2003).
123. Huan, X. & Marzouk, Y. M. Simulation-based optimal Bayesian experimental design for nonlinear systems. *Journal of Computational Physics* **232**, 288 (2013).
124. Papadimitriou, D. I. & Papadimitriou, C. Optimal sensor placement for the estimation of turbulence model parameters in CFD. *International Journal for Uncertainty Quantification* **5**, 545 (2015).
125. Papadimitriou, C. Optimal sensor placement methodology for parametric identification of structural systems. *Journal of Sound and Vibration* **278**, 923 (2004).
126. Morrow, J. E. Schooling Behavior in Fishes. *The Quarterly Review of Biology* **23**. PMID: 18904898, 27 (1948).
127. Partridge, B. L. & j. Pitcher, T. The sensory basis of fish schools: Relative roles of lateral line and vision. *Journal of comparative physiology* **135**, 315 (1980).
128. Triantafyllou, M. S., Weymouth, G. D. & Miao, J. Biomimetic Survival Hydrodynamics and Flow Sensing. *Annual Review of Fluid Mechanics* **48**, 1 (2016).
129. Ward, A. J. W., Sumpter, D. J. T., Couzin, I. D., Hart, P. J. B. & Krause, J. Quorum decision-making facilitates information transfer in fish shoals. *Proceedings of the National Academy of Sciences* **105**, 6948 (2008).
130. Puckett, J. G., Pokhrel, A. R. & Giannini, J. A. Collective gradient sensing in fish schools. *Scientific Reports* **8**, 7587 (2018).
131. Dykgraaf, S. Untersuchungen über die Funktion der Seitenorgane an Fischen. *Zeitschrift für vergleichende Physiologie* **20**, 162 (1933).
132. Dykgraaf, S. The functioning and significance of the lateral-line organs. *Biological reviews of the Cambridge Philosophical Society* **38**, 51 (1963).

133. Bleckmann, H., Przybilla, A., Klein, A., Schmitz, A., Kunze, S. & Brücker, C. in *Nature-Inspired Fluid Mechanics: Results of the DFG Priority Programme 1207 ”Nature-inspired Fluid Mechanics” 2006-2012* 161 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2012).

134. Sutterlin, A. & Waddy, S. Possible Role of the Posterior Lateral Line in Obstacle Entrainment by Brook Trout (Salvelinus fontinalis). *Journal of the Fisheries Research Board of Canada* **32**, 2441 (2011).

135. Akanyeti, O., Venturelli, R., Visentin, F., Chambers, L., Megill, W. M. & Fiorini, P. What information do Kármán streets offer to flow sensing? *Bioinspiration & Biomimetics* **6**, 036001 (2011).

136. Chambers, L. D., Akanyeti, O., Venturelli, R., Ježov, J., Brown, J., Kruusmaa, M., Fiorini, P. & Megill, W. M. A fish perspective: detecting flow features while moving using an artificial lateral line in steady and unsteady flow. *J. R. Soc. Interface* **11** (2014).

137. Von Campenhausen, C., Riess, I. & Weissert, R. Detection of stationary objects by the blind Cave FishAnoptichthys jordani (Characidae). *Journal of comparative physiology* **143**, 369 (1981).

138. Hassan, E. S. *Hydrodynamic Imaging of the Surroundings by the Lateral Line of the Blind Cave Fish Anoptichthys jordani* in *The Mechanosensory Lateral Line* (eds Coombs, S., Görner, P. & Münz, H.) (Springer New York, New York, NY, 1989), 217.

139. Windsor, S. P., Norris, S. E., Cameron, S. M., Mallinson, G. D. & Montgomery, J. C. The flow fields involved in hydrodynamic imaging by blind Mexican cave fish (Astyanax fasciatus). Part I: open water and heading towards a wall. *Journal of Experimental Biology* **213**, 3819 (2010).

140. Windsor, S. P., Norris, S. E., Cameron, S. M., Mallinson, G. D. & Montgomery, J. C. The flow fields involved in hydrodynamic imaging by blind Mexican cave fish (Astyanax fasciatus). Part II: gliding parallel to a wall. *Journal of Experimental Biology* **213**, 3832 (2010).

141. Hoekstra, D. & Janssen, J. Non-visual feeding behavior of the mottled sculpin, Cottus bairdi, in Lake Michigan. *Environmental Biology of Fishes* **12**, 111 (1985).

142. Pitcher, T., Partridge, B. & Wardle, C. A blind fish can school. *Science* **194**, 963 (1976).

143. Satou, M., Takeuchi, H.-A., Nishii, J., Tanabe, M., Kitamura, S., Okumoto, N. & Iwata, M. Behavioral and electrophysiological evidences that the lateral line is involved in the inter-sexual vibrational communication of the himé salmon (landlocked red salmon, Oncorhynchus nerka). *Journal of Comparative Physiology A* **174**, 539 (1994).

144. Huijbers, C. M., Nagelkerken, I., Lössbroek, P. A. C., Schulten, I. E., Siegenthaler, A., Holderied, M. W. & Simpson, S. D. A test of the senses: Fish select novel habitats by responding to multiple cues. *Ecology* **93**, 46 (2012).

145. Montgomery, J. C., Baker, C. F. & Carton, A. G. The lateral line can mediate rheotaxis in fish. *Nature* **389**, 960 (1997).

146. Coombs, S., Janssen, J. & Webb, J. F. in *Sensory biology of aquatic animals* 553 (Springer, 1988).

147. Coombs, S., Görner, P. & Münz, H. *A Brief Overview of the Mechanosensory Lateral Line System and the Contributions to This Volume* in *The Mechanosensory Lateral Line* (eds Coombs, S., Görner, P. & Münz, H.) (Springer New York, New York, NY, 1989), 3.

148. Denton, E. J. & Gray, J. A. B. *Some Observations on the Forces Acting on Neuromasts in Fish Lateral Line Canals* in *The Mechanosensory Lateral Line* (eds Coombs, S., Görner, P. & Münz, H.) (Springer New York, New York, NY, 1989), 229.

149. Coombs, S. & Braun, C. B. in *Sensory Processing in Aquatic Environments* 122 (Springer New York, New York, NY, 2003).

150. Coombs, S. & Netten, S. V. in *Fish Biomechanics* 103 (Academic Press, 2005).

151. Bleckmann, H. Peripheral and central processing of lateral line information. *Journal of Comparative Physiology A* **194**, 145 (2008).

152. Jiang, Y., Ma, Z. & Zhang, D. Flow field perception based on the fish lateral line system. *Bioinspiration & Biomimetics* **14**, 041001 (2019).

153. Engelmann, J., Hanke, W., Mogdans, J. & Bleckmann, H. Hydrodynamic stimuli and the fish lateral line. *Nature* **408**, 1476 (2000).

154. Kottapalli, A. G. P., Asadnia, M., Miao, J. M., Barbastathis, G. & Triantafyllou, M. S. A flexible liquid crystal polymer MEMS pressure sensor array for fish-like underwater sensing. *Smart Mater. Struct.* **21**, 115030 (2012).

155. Tao, J. & Yu, X. Hair flow sensors: from bio-inspiration to bio-mimicking - A review. *Smart Mater. Struct.* **21**, 113001 (2012).
156. Asadnia, M., Kottapalli, A. G. P., Miao, J., Warkiani, M. E. & Triantafyllou, M. S. Artificial fish skin of self-powered micro-electromechanical systems hair cells for sensing hydrodynamic flow phenomena. *J. R. Soc. Interface* **12** (2015).
157. Kottapalli, A. G. P., Bora, M., Sengupta, D., Miao, J. & Triantafyllou, M. S. *Hydrogel-CNT Biomimetic Cilia for Flow Sensing* in *2018 IEEE SENSORS* (2018), 1.
158. Wolf, B. J., Morton, J. A. S., MacPherson, W. N. & van Netten, S. M. Bio-inspired all-optical artificial neuromast for 2D flow sensing. *Bioinspiration & Biomimetics* **13**, 026013 (2018).
159. Yang, Y., Chen, J., Engel, J., Pandya, S., Chen, N., Tucker, C., Coombs, S., Jones, D. L. & Liu, C. Distant touch hydrodynamic imaging with an artificial lateral line. *Proceedings of the National Academy of Sciences* **103**, 18891 (2006).
160. Yang, Y., Nguyen, N., Chen, N., Lockwood, M., Tucker, C., Hu, H., Bleckmann, H., Liu, C. & Jones, D. L. Artificial lateral line with biomimetic neuromasts to emulate fish sensing. *Bioinspir. Biomim.* **5**, 016001 (2010).
161. Strokina, N., Kämäräinen, J., Tuhtan, J. A., Fuentes-Pérez, J. F. & Kruusmaa, M. Joint Estimation of Bulk Flow Velocity and Angle Using a Lateral Line Probe. *IEEE Trans. Instrum. Meas.* **65**, 601 (2016).
162. Xu, Y. & Mohseni, K. A Pressure Sensory System Inspired by the Fish Lateral Line: Hydrodynamic Force Estimation and Wall Detection. *IEEE Journal of Oceanic Engineering* **42**, 532 (2017).
163. Sengupta, D., Chen, S.-H. & Kottapalli, A. G. P. in *Self-Powered and Soft Polymer MEMS/NEMS Devices* 61 (Springer International Publishing, Cham, 2019).
164. Zhang, X., Shan, X., Shen, Z., Xie, T. & Miao, J. A New Self-Powered Sensor Using the Radial Field Piezoelectric Diaphragm in d33 Mode for Detecting Underwater Disturbances. *Sensors* **19**, 962 (2019).
165. Ježov, J., Akanyeti, O., Chambers, L. D. & Kruusmaa, M. *Sensing oscillations in unsteady flow for better robotic swimming efficiency* in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (2012), 91.

166. Kruusmaa, M., Fiorini, P., Megill, W., de Vittorio, M., Akanyeti, O., Visentin, F., Chambers, L., El Daou, H., Fiazza, M., Ježov, J., Listak, M., Rossi, L., Salumae, T., Toming, G., Venturelli, R., Jung, D. S., Brown, J., Rizzi, F., Qualtieri, A., Maud, J. L. & Liszewski, A. FILOSE for Svenning: A Flow Sensing Bioinspired Robot. *IEEE Robot. Autom. Mag.* **21**, 51 (2014).
167. DeVries, L., Lagor, F. D., Lei, H., Tan, X. & Paley, D. A. Distributed flow estimation and closed-loop control of an underwater vehicle with a multi-modal artificial lateral line. *Bioinspir. Biomim.* **10**, 025002 (2015).
168. Yen, W., Sierra, D. M. & Guo, J. Controlling a Robotic Fish to Swim Along a Wall Using Hydrodynamic Pressure Feedback. *IEEE Journal of Oceanic Engineering* **43**, 369 (2018).
169. Krieg, M., Nelson, K. & Mohseni, K. Distributed sensing for fluid disturbance compensation and motion control of intelligent robots. *Nature machine intelligence* **1** (2019).
170. Zheng, X., Wang, M., Zheng, J., Tian, R., Xiong, M. & Xie, G. *Artificial lateral line based longitudinal separation sensing for two swimming robotic fish with leader-follower formation* in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2019), 2539.
171. Ćurčić-Blake, B. & van Netten, S. M. Source location encoding in the fish lateral line canal. *Journal of Experimental Biology* **209**, 1548 (2006).
172. Ristroph, L., Liao, J. C. & Zhang, J. Lateral Line Layout Correlates with the Differential Hydrodynamic Pressure on Swimming Fish. *Phys. Rev. Lett.* **114**, 018102 (1 2015).
173. Zhang, F., Lagor, F., Yeo, D., Washington, P. & Paley, D. *Distributed Flow Sensing Using Bayesian Estimation for a Flexible Fish Robot* in *Proceedings of the ASME 2015 Dynamic Systems and Control Conference* (2015).
174. Ahrari, A., Lei, H., Sharif, M. A., Deb, K. & Tan, X. *Design optimization of artificial lateral line system under uncertain conditions* in *2015 IEEE Congress on Evolutionary Computation (CEC)* (2015), 1807.
175. Ahrari, A., Lei, H., Sharif, M. A., Deb, K. & Tan, X. Reliable underwater dipole source characterization in 3D space by an optimally designed artificial lateral line system. *Bioinspir. Biomim.* **12**, 036010 (2017).

176. Boulogne, L. H., Wolf, B. J., Wiering, M. A. & van Netten, S. M. Performance of neural networks for localizing moving objects with an artificial lateral line. *Bioinspiration & Biomimetics* **12**, 056009 (2017).
177. Colvert, B., Alsalman, M. & Kanso, E. Classifying vortex wakes using neural networks. *Bioinspiration & Biomimetics* **13**, 025003 (2018).
178. Wolf, B. J., Pirih, P., Kruusmaa, M. & Van Netten, S. M. Shape Classification Using Hydrodynamic Detection via a Sparse Large-Scale 2D-Sensitive Artificial Lateral Line. *IEEE Access* **8**, 11393 (2020).
179. Wolf, B., van de Wolfshaar, J. & van Netten, S. Three-dimensional multi-source localization of underwater objects using convolutional neural networks for artificial lateral lines. *Journal of the Royal Society Interface* **17**, 20190616 (2020).
180. Xu, D., Lv, Z., Zeng, H., Bessaih, H. & Sun, B. Sensor placement optimization in the artificial lateral line using optimal weight analysis combining feature distance and variance evaluation. *ISA Transactions* **86**, 110 (2019).
181. Verma, S., Papadimitriou, C., Luethen, N., Arampatzis, G. & Koumoutsakos, P. Optimal sensor placement for artificial swimmers. *J. Fluid Mech.* **884** (2019).
182. Novati, G., Verma, S., Alexeev, D., Rossinelli, D., Van Rees, W. M. & Koumoutsakos, P. Synchronisation through learning for two self-propelled swimmers. *Bioinspiration and Biomimetics* **12**, aa6311 (2017).
183. Kroese, A. B. & Schellart, N. A. Velocity- and acceleration-sensitive units in the trunk lateral line of the trout. *J. Neurophysiol.* **68**, 2212 (1992).
184. Bleckmann, H. & Zelick, R. Lateral line system of fish. *Integr. Zool.* **4**, 13 (2009).
185. Jones, E., Oliphant, T., Peterson, P., *et al. SciPy: Open source scientific tools for Python* [Online; accessed <today>]. 2001.
186. Dierckx, P. An algorithm for smoothing, differentiation and integration of experimental data using spline functions. *Journal of Computational and Applied Mathematics* **1**, 165 (1975).
187. Simoen, E., Papadimitriou, C. & Lombaert, G. On prediction error correlation in Bayesian model updating. *J. Sound Vib.* **332**, 4136 (2013).

188. Weber, P., Wälchli, D., Zeqiri, M. & Koumoutsakos, P. *Remember and Forget Experience Replay for Multi-Agent Reinforcement Learning* 2022.

189. Weber, P., Chatzimanolakis, M. & Koumoutsakos, P. *Control and Reinforcement Learning for Schooling Hydrodynamics* 2023.

190. Verma, S., Papadimitriou, C., Lüthen, N., Arampatzis, G. & Koumoutsakos, P. Optimal sensor placement for artificial swimmers. *Journal of Fluid Mechanics* **884**, A24 (2020).

191. Weber, P., Arampatzis, G., Novati, G., Verma, S., Papadimitriou, C. & Koumoutsakos, P. Optimal Flow Sensing for Schooling Swimmers. *Biomimetics* **5**, 10 (2020).

192. Chatzimanolakis, M., Weber, P. & Koumoutsakos, P. Vortex separation cascades in simulations of the planar flow past an impulsively started cylinder up to $\boldsymbol{Re} = \mathbf{100\,000}$. *Journal of Fluid Mechanics* **953**, R2 (2022).

193. Tytell, E. D. & Lauder, G. V. The hydrodynamics of eel swimming: I. Wake structure. *Journal of Experimental Biology* **207**, 1825 (2004).

194. Liao, J. C. Fish swimming efficiency. *Current Biology* **32**, R666 (2022).

195. Liao, J. C., Beal, D. N., Lauder, G. V. & Triantafyllou, M. S. Fish Exploiting Vortices Decrease Muscle Activity. *Science* **302**, 1566 (2003).

196. Harvey, S. T., Muhawenimana, V., Müller, S., Wilson, C. A. M. E. & Denissenko, P. An inertial mechanism behind dynamic station holding by fish swinging in a vortex street. *Scientific Reports* **12**, 12660 (2022).

197. Zheng, J., Zhang, T., Wang, C., Xiong, M. & Xie, G. Learning for Attitude Holding of a Robotic Fish: An End-to-End Approach With Sim-to-Real Transfer. *IEEE Transactions on Robotics* **38**, 1287 (2022).

198. Brunton, S. L., Noack, B. R. & Koumoutsakos, P. Machine Learning for Fluid Mechanics. *Annu. Rev. Fluid Mech.* **52**, 477 (2020).

199. Garnier, P., Viquerat, J., Rabault, J., Larcher, A., Kuhnle, A. & Hachem, E. A review on deep reinforcement learning for fluid mechanics. *Computers & Fluids* **225**, 104973 (2021).

200. Gazzola, M., Hejazialhosseini, B. & Koumoutsakos, P. Reinforcement Learning and Wavelet Adapted Vortex Methods for Simulations of Self-propelled Swimmers. *SIAM Journal on Scientific Computing* **36**, B622 (2014).

201. Gazzola, M., Tchieu, A. A., Alexeev, D., de Brauer, A. & Koumoutsakos, P. Learning to school in the presence of hydrodynamic interactions. *Journal of Fluid Mechanics* **789**, 726 (2016).

202. Koumoutsakos, P. *Deep Reinforcement Learning for Flow Control* in *APS Division of Fluid Dynamics Meeting Abstracts* (2019), H17.

203. Zhu, Y., Tian, F.-B., Young, J., Liao, J. & Lai, J. A numerical study of fish adaption behaviors in complex environments with a deep reinforcement learning and immersed boundary–lattice Boltzmann method. *Scientific Reports* **11**, 1691 (2021).

204. Mandralis, I., Weber, P., Novati, G. & Koumoutsakos, P. Learning swimming escape patterns under energy constraints. *arXiv e-prints*, arXiv:2105.00771 (2021).

205. Hein, A. M., Altshuler, D. L., Cade, D. E., Liao, J. C., Martin, B. T. & Taylor, G. K. An Algorithmic Approach to Natural Behavior. *Current Biology* **30**, R663 (2020).

206. Liao, J. C., Beal, D. N., Lauder, G. V. & Triantafyllou, M. S. The Kármán gait: novel body kinematics of rainbow trout swimming in a vortex street. *Journal of Experimental Biology* **206**, 1059 (2003).

207. Liao, J. & Akanyeti, O. Fish Swimming in a Kármán Vortex Street: Kinematics, Sensory Biology and Energetics. *Marine Technology Society Journal* **51**, 48 (2017).

208. Akanyeti, O. & Liao, J. C. The effect of flow speed and body size on Kármán gait kinematics in rainbow trout. *Journal of Experimental Biology* **216**, 3442 (2013).

209. Beal, D. N., Hover, F. S., Triantafyllou, M. S., Liao, J. C. & Lauder, G. V. Passive propulsion in vortex wakes. *Journal of Fluid Mechanics* **549**, 385 (2006).

210. Triantafyllou, G., Triantafyllou, M. & Grosenbaugh, M. Optimal Thrust Development in Oscillating Foils with Application to Fish Propulsion. *Journal of Fluids and Structures* **7**, 205 (1993).

211. Moored, K. W., Dewey, P. A., Smits, A. J. & Haj-Hariri, H. Hydrodynamic wake resonance as an underlying principle of efficient unsteady propulsion. *Journal of Fluid Mechanics* **708**, 329 (2012).

212. Eloy, C. Optimal Strouhal number for swimming animals. *Journal of Fluids and Structures* **30**, 205 (2012).

213. Gopalkrishnan, R., Triantafyllou, M. S., Triantafyllou, G. S. & Barrett, D. Active vorticity control in a shear flow using a flapping foil. *Journal of Fluid Mechanics* **274**, 1 (1994).

214. Streitlien, K., Triantafyllou, G. S. & Triantafyllou, M. S. Efficient foil propulsion through vortex control. *AIAA Journal* **34**, 2315 (1996).

215. Pavlov, D. & Kasumyan, A. Patterns and mechanisms of schooling behavior in fish: A review. *Journal of Ichthyology* **40**, S163 (2000).

216. Boschitsch, B. M., Dewey, P. A. & Smits, A. J. Propulsive performance of unsteady tandem hydrofoils in an in-line configuration. *Physics of Fluids* **26**, 051901 (2014).

217. Maertens, A. P., Gao, A. & Triantafyllou, M. S. Optimal undulatory swimming for a single fish-like body and for a pair of interacting swimmers. *Journal of Fluid Mechanics* **813**, 301 (2017).

218. Sutterlin, A. M. & Waddy, S. Possible Role of the Posterior Lateral Line in Obstacle Entrainment by Brook Trout (Salvelinus fontinalis). *Journal of the Fisheries Research Board of Canada* **32**, 2441 (1975).

219. Webb, P. Entrainment by river chub nocomis micropogon and smallmouth bass micropterus dolomieu on cylinders. *Journal of Experimental Biology* **201**, 2403 (1998).

220. Triantafyllou, M. S., Triantafyllou, G. S. & Gopalkrishnan, R. Wake mechanics for thrust generation in oscillating foils. *Physics of Fluids A: Fluid Dynamics* **3**, 2835 (1991).

221. Liao, J. C. A review of fish swimming mechanics and behaviour in altered flows. *Philosophical Transactions of the Royal Society B: Biological Sciences* **362**, 1973 (2007).

222. Fefferman, C. Existence and smoothness of the Navier-Stokes equation. *The Millennium Prize Problems* (2006).

223. Weihs, D. Hydromechanics of Fish Schooling. *Nature* **241**, 290 (1973).

224. Weihs, D. in *Swimming and Flying in Nature: Volume 2* 703 (Springer US, Boston, MA, 1975).

225. Abrahams, M. V. & Colgan, P. W. Fish schools and their hydrodynamic function: a reanalysis. *Environmental Biology of Fishes* **20**, 79 (1987).

226. Partridge, B. L., Johansson, J. & Kalish, J. The structure of schools of giant bluefin tuna in Cape Cod Bay. *Environmental biology of fishes* **9**, 253 (1983).

227. Partridge, B. & Pitcher, T. Evidence against a hydrodynamic function for fish schools. *Nature* **279**, 418 (1979).

228. Pitcher, T. J. in *The Behaviour of Teleost Fishes* 294 (Springer US, Boston, MA, 1986).

229. Partridge, B. L. The structure and function of fish schools. *Scientific american* **246**, 114 (1982).

230. Abrahams, M. V. & Colgan, P. W. Risk of predation, hydrodynamic efficiency and their influence on school structure. *Environmental Biology of Fishes* **13**, 195 (1985).

231. Breder, C. M. Equations Descriptive of Fish Schools and Other Animal Aggregations. *Ecology* **35**, 361 (1954).

232. Aoki, I. A Simulation Study on the Schooling Mechanism in Fish. *Nippon Suisan Gakkaishi* **48**, 1081 (1982).

233. Reynolds, C. W. *Flocks, Herds and Schools: A Distributed Behavioral Model* in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques* (Association for Computing Machinery, New York, NY, USA, 1987), 25.

234. Zhu, Y., Pang, J.-H. & Tian, F.-B. Stable Schooling Formations Emerge from the Combined Effect of the Active Control and Passive Self-Organization. *Fluids* **7** (2022).

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

236. Nakhchi, M., Win Naung, S. & Rahmati, M. A novel hybrid control strategy of wind turbine wakes in tandem configuration to improve power production. *Energy Conversion and Management* **260**, 115575 (2022).

237. Gravish, N. & Lauder, G. V. Robotics-inspired biology. *Journal of Experimental Biology* **221**. jeb138438 (2018).

238. Fan, D., Jodin, G., Consi, T. R., Bonfiglio, L., Ma, Y., Keyes, L. R., Karniadakis, G. E. & Triantafyllou, M. S. A robotic Intelligent Towing Tank for learning complex fluid-structure dynamics. *Science Robotics* **4** (2019).

239. Berlinger, F., Gauci, M. & Nagpal, R. Implicit coordination for 3D underwater collective behaviors in a fish-inspired robot swarm. *Science Robotics* **6**, eabd8668 (2021).

240. Martin, S. M., Wälchli, D., Arampatzis, G., Economides, A. E., Karnakov, P. & Koumoutsakos, P. Korali: Efficient and scalable software framework for Bayesian uncertainty quantification and stochastic optimization. *Computer Methods in Applied Mechanics and Engineering* **389**, 114264 (2022).

241. Verma, S., Hadjidoukas, P., Wirth, P. & Koumoutsakos, P. *Multi-objective optimization of artificial swimmers* in *2017 IEEE Congress on Evolutionary Computation (CEC)* (2017), 1037.

242. Angulo, F. J., Finelli, L. & Swerdlow, D. L. Reopening Society and the Need for Real-Time Assessment of COVID-19 at the Community Level. *JAMA* **323**, 2247 (2020).

243. Perkins, T. A., Cavany, S. M., Moore, S. M., Oidtman, R. J., Lerch, A. & Poterek, M. Estimating unobserved SARS-CoV-2 infections in the United States. *Proceedings of the National Academy of Sciences* **117**, 22597 (2020).

244. Jonnerby, J., Lazos, P., Lock, E., Marmolejo-Cossío, F., Ramsey, C. B. & Sridhar, D. *Test and Contain: A Resource-Optimal Testing Strategy for COVID-19* in *AI for Social Good Workshop* (2020).

245. Buhat, C. A. H., Duero, J. C. C., Felix, E. F. O., Rabajante, J. F. & Mamplata, J. B. Optimal Allocation of COVID-19 Test Kits Among Accredited Testing Centers in the Philippines. *medRxiv* **0** (2020).

246. Hasell, J., Mathieu, E., Beltekian, D., Macdonald, B., Giattino, C., Ortiz-Ospina, E., Roser, M. & Ritchie, H. A cross-country database of COVID-19 testing. *Sci Data* **7** (2020).

247. World Health Organization. *Q&A: Influenza and COVID-19 - similarities and differences* `https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub`. online. 2020.

248. Stringhini, S., Wisniak, A., Piumatti, G., Azman, A. S., Lauer, S. A., Baysson, H., De Ridder, D., Petrovic, D., Schrempft, S., Marcus, K., Yerly, S., Arm Vernez, I., Keiser, O., Hurst, S., Posfay-Barbe, K. M., Trono, D., Pittet, D., Gétaz, L., Chappuis, F., Eckerle, I., Vuilleumier, N., Meyer, B., Flahault, A., Kaiser, L. & Guessous, I. Seroprevalence of anti-SARS-CoV-2 IgG antibodies in Geneva, Switzerland (SEROCoV-POP): a population-based study. *The Lancet* **396**, 313 (2020).

249. *Launch of the first rapid test for COVID-19 on the Swiss market* `https://www.startupticker.ch/en/news/april-2020/launch-of-the-first-rapid-test-for-covid-19\protect\penalty\z@-on-the-swiss-market`. online. 2020.

250. Ng, A. Y. & Russell, S. J. *Algorithms for Inverse Reinforcement Learning* in *Proceedings of the Seventeenth International Conference on Machine Learning* (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000), 663.

251. Du Toit, A. Outbreak of a novel coronavirus. *Nature Reviews Microbiology* **18**, 123 (2020).

252. Verity, R. e. a. Estimates of the severity of coronavirus disease 2019: a model-based analysis. *The Lancet Infectious Diseases* **20** (6 2020).

253. Department of Health and Social Care, UK. *Real-time Assessment of Community Transmission findings* `https://www.imperial.ac.uk/medicine/research-and-impact/groups/react-study/real-time-assessment-of-community\protect\penalty\z@-transmission-findings/`. online, accessed 2020-10-18. 2020.

254. Lohse, S., Pfuhl, T., Berko-Göttel, B., Rissland, J., Geißler, T., Gärtner, B., Becker, S. L., Schneitler, S. & Smola, S. Pooling of samples for testing for SARS-CoV-2 in asymptomatic people. *The Lancet Infectious Diseases* **20**, 1231 (2020).

255. Abdalhamid, B., Bilder, C. R., McCutchen, E. L., Hinrichs, S. H., Koepsell, S. A. & Iwen, P. C. Assessment of Specimen Pooling to Conserve SARS CoV-2 Testing Resources. *American Journal of Clinical Pathology* **153**, 715 (2020).

256. Karnakov, P., Arampatzis, G., Kičić, I., Wermelinger, F., Wälchli, D., Papadimitriou, C. & Koumoutsakos, P. Data-driven inference of the reproduction number for COVID-19 before and after interventions for 51 European countries. *Swiss Medical Weekly* **150** (2020).

257. Cochran, W. *Sampling Techniques* (Wiley, New York, 1963).

258. Chaloner, K. & Verdinelli, I. Bayesian Experimental Design: A Review. *Statistical Science* **10**, 273 (1995).

259. Ryan, E. G., Drovandi, C. C., McGree, J. M. & Pettitt, A. N. A Review of Modern Computational Algorithms for Bayesian Optimal Design. *International Statistical Review* **84**, 128 (2016).

260. Li, R., Pei, S., Chen, B., Song, Y., Zhang, T., Yang, W. & Shaman, J. Substantial undocumented infection facilitates the rapid dissemination of novel coronavirus (SARS-CoV-2). *Science* **368**, 489 (2020).

261. Speagle, J. S. Dynesty: a dynamic nested sampling package for estimating Bayesian posteriors and evidences. *Monthly Notices of the Royal Astronomical Society* **493**, 3132 (2020).

262. Lindley, D. On a measure of the information provided by an experiment. *Annals of Mathematical Statistics* **27**, 986 (1956).

263. Bundesamtes für Statistik. *Commuters for work purposes* `https://www.bfs.admin.ch/bfsstatic/dam/assets/8507281/master`. online. 2020.

264. Bundesamtes für Gesundheit. *Coronavirus: Informationen vom Bundesamtes für Gesundheit* `https://www.bag.admin.ch/bag/de/home/krankheiten/ausbrueche-epidemien-pandemien/aktuelle-ausbrueche-epidemien/novel-cov.html`. online. 2020.

265. Chau, C. H., Strope, J. D. & Figg, W. D. COVID-19 Clinical Diagnostics and Testing Technology. *Pharmacotherapy: The Journal of Human Pharmacology and Drug Therapy* **40**, 857 (2020).

266. Long, C., Xu, H., Shen, Q., Zhang, X., Fan, B., Wang, C., Zeng, B., Li, Z., Li, X. & Li, H. Diagnosis of the Coronavirus disease (COVID-19): rRT-PCR or CT? *European Journal of Radiology* **126**, 108961 (2020).

267. Heller, L., Mota, C. R. & Greco, D. B. COVID-19 faecal-oral transmission: Are we asking the right questions? *Science of The Total Environment* **729**, 138919 (2020).

268. Papadimitriou, C. & Lombaert, G. The effect of prediction error correlation on optimal sensor placement in structural dynamics. *Mechanical Systems and Signal Processing* **28**. Interdisciplinary and Integration Aspects in Structural Health Monitoring, 105 (2012).

269. Simoen, E., Papadimitriou, C. & Lombaert, G. On prediction error correlation in Bayesian model updating. *Journal of Sound and Vibration* **332**, 4136 (2013).

270. Kanton Zürich, Statistisches Amt. *SARS-CoV-2 Cases communicated by Swiss Cantons and Principality of Liechtenstein (FL)* `https://github.com/openZH/covid_19`. online. 2020.

271. Dehning, J., Zierenberg, J., Spitzner, F. P., Wibral, M., Neto, J. P., Wilczek, M. & Priesemann, V. Inferring change points in the spread of COVID-19 reveals the effectiveness of interventions. *Science* **369** (2020).
272. Liu, Y., Gayle, A. A., Wilder-Smith, A. & Rocklöv, J. The reproductive number of COVID-19 is higher compared to SARS coronavirus. *Journal of Travel Medicine* **27** (2020).
273. Karni, E. & Schmeidler, D. in *Handbook of Mathematical Economics* 1763 (Elsevier, 1991).
274. Fujita, Y. & Maeda, S.-i. *Clipped Action Policy Gradient* in *Proceedings of the 35th International Conference on Machine Learning* **80** (2018), 1597.
275. Haarnoja, T., Zhou, A., Abbeel, P. & Levine, S. *Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor* in *Proceedings of the 35th International Conference on Machine Learning* (eds Dy, J. & Krause, A.) **80** (PMLR, 2018), 1861.

CURRICULUM VITAE

Personal data

Name	Pascal Weber
Date of Birth	December 18, 1990
Place of Birth	Zürich, Switzerland
Citizen of	Zug and Zürich, Switzerland

Education

September 2016 – October 2018	Master of Science ETH Physics ETH Zürich
September 2012 – September 2016	Bachelor of Science ETH Interdisciplinary Science ETH Zürich
March 2012	Swiss Matura, focus Economy and Law

Employment

October 2018 – December 2023	Scientific Assistant *ETH Zürich*
January 2022 – May 2022	Teaching Assistant *Harvard University*
October 2010 – December 2012	Teleconsultant *Digicomp Academy AG*
October 2009 – October 2010	Telemarketing Outbound *rbc Solutions AG*

PUBLICATIONS

Articles in peer-reviewed journals:

1. Chatzimanolakis, M., Weber, P. & Koumoutsakos, P. Vortex separation cascades in simulations of the planar flow past an impulsively started cylinder up to Re=100'000. *Journal of Fluid Mechanics* **953** (2022).
2. Mandralis, I., Weber, P., Novati, G. & Koumoutsakos, P. Learning swimming escape patterns for larval fish under energy constraints. *Phys. Rev. Fluids* **6**, 093101 (9 2021).
3. Chatzimanolakis, M., Weber, P., Arampatzis, G., Wälchli, D., Kičić, I., Karnakov, P., Papadimitriou, C. & Koumoutsakos, P. Optimal allocation of limited test resources for the quantification of COVID-19 infections. *Swiss Med. Wkly.* (2020).
4. Weber, P., Arampatzis, G., Novati, G., Verma, S., Papadimitriou, C. & Koumoutsakos, P. Optimal Flow Sensing for Schooling Swimmers. *Biomimetics* **5** (2020).

Conference contributions:

5. Arampatzis, G., Wälchli, D., Weber, P., Rästas, H. & Koumoutsakos, P. (μ, λ)*-CCMA-ES for Constrained Optimization with an Application in Pharmacodynamics* in *Proceedings of the Platform for Advanced Scientific Computing Conference - PASC'19* (ACM Press, 2019).

Preprints / Pending for Submission:

6. Weber, P., Wälchli, D., Zeqiri, M. & Koumoutsakos, P. *Remember and Forget Experience Replay for Multi-Agent Reinforcement Learning* 2022.
7. Chatzimanolakis, M., Weber, P., Wermelinger, F. & Koumoutsakos, P. *CubismAMR – A C++ library for Distributed Block-Structured Adaptive Mesh Refinement* 2022.
8. Waelchli, D., Weber, P. & Koumoutsakos, P. *Discovering Individual Rewards in Collective Behavior through Inverse Multi-Agent Reinforcement Learning* 2023.

9. Weber, P., Chatzimanolakis, M., Wälchli, D. & Koumoutsakos, P. *Multi-Task Reinforcement Learning to Swim in Reverse and Forward Kármán Wakes* 2023.
10. Weber, P., Chatzimanolakis, M. & Koumoutsakos, P. *Control and Reinforcement Learning for Schooling Hydrodynamics* 2023.
11. Chatzimanolakis, M., Weber, P. & Koumoutsakos, P. *Discovery of Drag Reduction Actions in Flow Past a Cylinder Through Reinforcement Learning* 2023.
12. Weber, P., Wälchli, D. & Koumoutsakos, P. *Quantification of Uncertainties in Deep Reinforcement Learning for Scientific Machine Learning* 2023.