Improving the Robustness of Control of Chaotic Convective Flows with Domain-Informed Reinforcement Learning

Michiel Straat*, Thorben Markmann*, Sebastian Peitz†, Barbara Hammer*
*CITEC, Bielefeld University, Bielefeld, Germany
{mstraat, tmarkmann, bhammer}@techfak.uni-bielefeld.de
†Lamarr Institute, TU Dortmund, Dortmund, Germany
sebastian.peitz@tu-dortmund.de

***Abstract*—Chaotic convective flows arise in many real-world systems, such as microfluidic devices and chemical reactors. Stabilizing these flows is highly desirable but remains challenging, particularly in chaotic regimes where conventional control methods often fail. In this work, we improve the practical feasibility of controlling chaotic flows by Reinforcement Learning (RL), focusing on Rayleigh-Bénard Convection (RBC), a canonical model for convective heat transport. In particular, our goal is to robustly reduce the convective heat transfer. Although RL has shown promise for control in laminar flow settings, its ability to generalize and remain robust under chaotic and turbulent dynamics is not well explored, despite being critical for real-world deployment. Hence, to enhance generalization and sample efficiency, we introduce domain-informed RL agents that are trained using Proximal Policy Optimization across diverse initial conditions and flow regimes. We incorporate domain knowledge in the reward function via a term that encourages Bénard cell merging, as an example of a desirable macroscopic property. In laminar flow regimes, the domain-informed RL agents reduce convective heat transport by up to 33%. In chaotic flow regimes, the agents still achieve a significant 10% reduction, which is markedly better than the conventional controllers used in practice. We compare the domain-informed to uninformed agents: Our results show that the domain-informed reward design results in steady flows, faster convergence during training, and generalization across flow regimes without retraining. Our work demonstrates that elegant domain-informed priors can greatly enhance the robustness of RL-based control of chaotic flows, bringing real-world deployment closer.**

***Index Terms*—Reinforcement Learning, Flow Control, Reward Shaping, Turbulence Control, Domain-Informed Learning, Rayleigh–Bénard Convection**

I. Introduction

AI methods hold great promise for advancing engineering applications in fluid dynamics [1], including those in industry, aviation, energy systems, and climate science. Traditionally, these domains rely on numerical simulations of physical laws, commonly the Navier-Stokes equations, for tasks ranging from airfoil design to weather prediction. With the improvements in sensors and computational capacity, AI now enables data-driven approaches to modeling and control, opening new opportunities to tackle complex flow control problems [2].

Natural convection is the process of heat transport by fluid motion due to temperature differences. It is common in nature and industry, and its controllability plays a crucial role in process reliability and quality. For example, in crystal growth processes or microfluidic gradient generation, uncontrolled convection can introduce instabilities that degrade material quality or disrupt precision measurements [3], [4]. In this work, we study robust RL-based control of Rayleigh-Bénard Convection (RBC), a canonical model for convective heat transport that captures the transition from laminar to chaotic flow as temperature differences increase.

Conventional control methods often struggle to stabilize such chaotic systems. In contrast, Reinforcement Learning (RL) has shown growing promise in flow control tasks such as turbulence suppression, mixing optimization, and drag reduction [5]. However, the robustness and generalization of RL-based control of chaotic flows is largely underexplored. This currently limits real-world deployments, where small variations in initial conditions or system parameters can lead to widely different behaviors. In the context of convective dynamics, it is especially important for control agents to generalize across both initial conditions and different chaotic regimes, ideally without requiring retraining.

In AI for scientific applications, incorporating prior knowledge, such as conservation laws or physical symmetries, has helped improve sample efficiency, generalization from scarce and noisy data, and physical plausibility of learned models [6]. Inspired by this, we explore a domain-informed reward shaping approach, where we embed physically meaningful macroscopic features into the reward function. We show that this addition holds promise to guide the agent toward flow stabilization strategies that generalize across flow regimes and initial conditions, and require fewer rollouts during training.

Initial works have explored the control of convection by RL:

The authors acknowledge financial support by the project "SAIL: SustAInable Life-cycle of Intelligent Socio-Technical Systems" (Grant ID NW21-059A), which is funded by the program "Netzwerke 2021" of the Ministry of Culture and Science of the State of North Rhine Westphalia, Germany. SP acknowledges support by the European Union via the ERC Starting Grant "KoOpeRaDE" (Grant ID 101161457). This research was supported by the Ministry of Culture and Science NRW (Germany) as part of the Lamarr Fellow Network. This publication reflects the views of the authors only.

[7] and [8] proposed a scalable RL approach for a laminar flow regime in 2D and 3D, but the works did not address generalization across initial conditions or system parameters. Reference [9] studied control of convective flows in a narrow domain and identified promising emergent strategies, but did not explore robustness across flow regimes and did not leverage domain knowledge. For a fixed flow regime and a simplified control task, [10] developed a model-based RL setup that learns in a surrogate environment.

Motivated by the open challenge of robust control for turbulent convection, we make three key contributions: **(1)** We introduce a domain-informed reward that reduces training time and promotes fast flow stabilization; **(2)** We demonstrate that the domain-informed reward improves generalization across initial conditions and chaotic flow regimes; and **(3)** We achieve robust control in chaotic convective flow settings where previous methods either fail or have not been tested.

These contributions move RL-based flow control for chaotic flows a step closer to practical deployment. To reproduce the findings of this work, we provide a code repository. Additionally, alongside each section and experiment, the repository contains videos and further details. The links to the code and videos are:

Code — https://github.com/HammerLabML/RBC-Control-SARL
Videos — In file `examples.md` in the code repository

II. Methodology

In this section, we present the necessary details of the dynamical system, its parameters, and measurement quantities. We then introduce the control task and our training setup.

A. Rayleigh-Bénard Convection

Rayleigh-Bénard Convection (RBC) is a widely studied model in fluid dynamics for heat transport between a heated lower plate and a cooled upper plate, see e.g. [11] for the partial differential equation.

We consider RBC on a 2D domain of width $W = 2\pi$ and height $H = 2$, with horizontal coordinate $x \in [0, W]$ and vertical coordinate $y \in [-H/2, H/2]$. The system's state $\boldsymbol{s} = (\boldsymbol{u}, T)$ in 2D consists of the velocity vector field $\boldsymbol{u}(x, y) = (u_x, u_y)$ and the scalar temperature field $T(x, y)$. We use periodic boundaries at the left and right of the domain. For the bottom and top boundaries, we use zero-velocity (i.e. no-slip) conditions and for the temperature we use T_b at the bottom and T_t at the top, where $T_b > T_t$ drives the convective flow. We used a spectral solver from the Shenfun package [12] on a grid of collocation points of size 96×64.[1] The main system parameter is the *Rayleigh number* (Ra). It is a ratio of the timescale of convective to diffusive thermal transport, defined as:

$$\mathrm{Ra} = \frac{ga\Delta T H^3}{\nu\kappa} \,. \tag{1}$$

As seen in the definition, it incorporates several other system parameters, i.e. the gravitational acceleration g, the fluid's thermal expansion coefficient a, the mean difference between the temperature at the bottom and top wall $\Delta T = T_b - T_t$, the height of the fluid layer (i.e. the domain height) H, and both the fluid's kinematic viscosity ν and its thermal diffusivity κ. A higher value of Ra introduces more convection in the system, and interesting dynamics occur in dependence of Ra: for low Ra, the fluid does not move and the temperature field converges to a stable equilibrium through heat conduction only, which is a linear temperature gradient in the vertical direction: $T_{cond}(y) = T_b - (y/H + 0.5)\Delta T$. As Ra increases past a critical value, the system becomes unstable and heat starts to be transported by convective flow, i.e. hot fluid rises, and cold fluid sinks which typically takes place in the form of so-called *Bénard cells* (see Fig. 1b). As Ra increases further, the fluid flow becomes increasingly chaotic and eventually turbulent [13]. We selected a range of Ra that includes unsteady flows and moderately turbulent effects, which makes for an interesting benchmark for robust learning-based control.

1) Measuring Convection: Thermal convection is heat transport through fluid flow that is induced by temperature differences. The strength of the local convective heat transport in the vertical direction is given by the product of the vertical velocity and the temperature:

$$q(x, y, t) = u_y(x, y, t)\theta(x, y, t) \,, \tag{2}$$

where $\theta(x, y, t) = T(x, y, t) - \langle T \rangle_{x,y,t}$ is the temperature difference from the mean temperature over the field. In the system, heat is transported by both convection and conduction. The ratio between transport by convection to conduction is an important measurement quantity known as the *Nusselt number* Nu:

$$\mathrm{Nu}(t) = \frac{\langle q(x, y, t) \rangle_{x,y}}{\kappa(T_b - T_t)/H} \,. \tag{3}$$

The numerator is a measure of the overall heat transport by convection (field average of (2)), whereas the denominator quantifies heat transport by conduction.[2] Note that with increasing Ra, convective flows get stronger, which increases the value of Nu.[3]

B. Flow stabilization by controlling convection

Motivated by interesting chaotic dynamics of the system and industrial relevance to control convective flows, the task is to robustly suppress convective heat transport, as measured by the Nusselt number in (3), by controlling heating elements at the bottom. We illustrate the control setup in Fig. 1a, which bears some similarity to real-world lab set-ups such as in [15], where a simple PD control was applied. In the control task, we divide the bottom boundary into 12 heating elements that each can be set to a temperature a_i. Note that without further constraints, a trivial way to control the system is to set all heaters to the temperature at the top, i.e. $a_i = T_t$.

[1] All other simulation parameters can be found in the repository, see Appendix A.

[2] H: distance between top and lower boundary. κ : thermal diffusivity.

[3] Nu scales like $\mathrm{Nu}_{\mathrm{Base}}(\mathrm{Ra}) \sim \mathrm{Ra}^{1/3}$ up to at least $\mathrm{Ra} = 10^{15}$ [14].

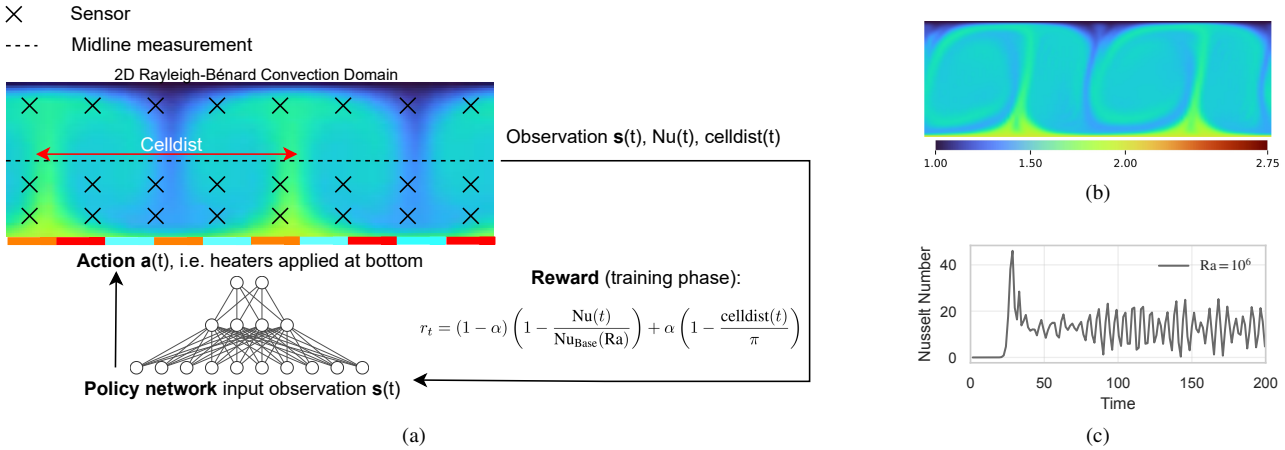


Fig. 1. Fig. (a) shows the schematic of the control setup. The RL agent receives partial observations from an 8x48 grid of sensors and a midline vertical velocity measurement. Based on the observation $\boldsymbol{s}(t)$, the policy network outputs heating actions $\boldsymbol{a}(t)$ for 12 bottom actuators. Note in PPO, there is also a critic network involved that estimates the value function. The reward combines a normalized Nusselt number reduction term with a domain-informed term that encourages cell merging via the measured cell distance. Figures (b) and (c) show a state resulting from Ra $= 10^6$ together with the Nusselt number over time. The colorbar of (b) covers the entire range between T_C and $T_H + C$, with $C = 0.75$, and is valid for the following figures in the paper.

This eliminates the temperature difference between bottom and top, i.e. $\Delta T = 0$, so that heat transport (both conductive and convective) does not occur. Hence, we consider a non-trivial scenario where the bottom heaters are constrained to values $a_i \in [T_b - 0.75, T_b + 0.75]$, which also simulates physical constraints of real-world heaters. In addition, we assume that the mean over all heaters is always fixed to T_b to ensure that the mean temperature difference between the top and bottom plate, ΔT in (1), remains constant.[4] Without this restriction, the controller can resort to the trivial strategy which is to effectively alter the Rayleigh number of the system and thus change the overall thermal forcing and dynamics. Hence, the flow is to be controlled by temperature fluctuations applied instantaneously by the heaters to the bottom boundary.

1) Linear Control: Since our objective is to study the practical feasibility of RL agents in chaotic settings, we consider a comparison to control schemes that are physically realizable and commonly used in practice, most notably proportional-derivative (PD) control, which is the standard approach in experimental setups such as in [15], [16].

The temperature fluctuations at the lower boundary are chosen to oppose the convective flow by increasing heat below downward flows and decreasing heat below upward flows. This is achieved by setting the error to a horizontal midline vertical velocity measurement, i.e. $E(x,t) = u_y(x, y = 0, t)$ and computing the linear control signal:

$$a(x,t) = k_p E(x,t) + k_d E'(x,t)\,, \tag{4}$$

where we found that gains of $k_p = -970$ and $k_d = -2000$ consistently lead to the desired behavior of applying heat between Bénard cells.[5]

[4]We satisfy the constraints by a simple transformation, see Appendix C-A.

[5]To obtain the final $N = 12$ heating values, we downsample the control signal by averaging the grid points within the heater locations and then apply a transformation to satisfy our control limits, see Appendix C-A.

2) Reinforcement Learning: Although the mathematical formulation of the objective in RL bears similarity to that of conventional control, RL has the potential to discover sophisticated control policies through expressive neural representations, which is necessary for control of non-linear chaotic dynamics. To train the agent, we employ the model-free algorithm PPO [17], a successful actor-critic method that improves robustness through a clipped objective. The agent learns a policy $\pi(\boldsymbol{a}|\boldsymbol{s})$, which maps states $\boldsymbol{s}$ to action probabilities $\boldsymbol{a}$. The learning objective is to find a policy π^* that maximizes the expected sum of reward, $\max_\pi \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})\right]$, where actions $\boldsymbol{a}_t$ are chosen according to π and $0 \leq \gamma \leq 1$ is a discount factor that favors short-term over long-term rewards. The state transitions are modeled using a Markov Decision Process (MDP), which is in our case given by the underlying deterministic numerical simulation.

Fig. 1 gives an overview of the RL setup: For the state observations, we assume access to probe sensors that are spread equidistantly on an 8×48 grid over the spatial domain and measure the local temperature and velocity of the fluid. We flatten all measurements into a vector with $3{*}8{*}48 = 1152$ elements as input to the actor and critic network. The policy network outputs vectors $(a_1, a_2, \ldots, a_{12}) \in \mathbb{R}^{12}$ (values for the 12 heaters), where $a_i \in [-1, 1]$, which are transformed to satisfy a mean actuation of T_b and limits of $[T_b - 0.75, T_b + 0.75]$ as explained in the previous section, and then applied to the lower boundary. As the agent's objective is to minimize convective heat transfer, which is measured by the Nusselt number Nu from (3), we incorporate Nu in the reward as follows:

$$R(s_t) = 1 - \frac{\mathrm{Nu}(s_t)}{\mathrm{Nu_{Base}(Ra)}}\,, \tag{5}$$

where we additionally scaled Nu by $\mathrm{Nu_{Base}(Ra)}$, which we obtained as an average over the uncontrolled case, so that approximately $R(s_t) \in [0, 1]$. PPO aims to maximize this

reward over episodes, which corresponds to minimizing the Nusselt number.

C. Reward Shaping: Better stabilization properties through cell merging

Merging of Bénard cells is an effective strategy for stabilizing flow and reducing convection, as was identified in [7] for the laminar flow at $\mathrm{Ra} = 10^4$. Cell merging limits the number of counter-rotating flow structures in favor of a global, slower steady flow. This results in lower Nusselt number with less variation over time. However, cell merging is rather hard to achieve for chaotic flows, because of the existence of competing simple strategies that apply heating between cells. Although the simple strategies reduce the time-average of the Nusselt number, the Nusselt number of single-cell states is lower and also associated with steady flows. Here and in the rest of the paper, by *steady flows*, we mean a flow that hardly changes over time and remains so over long times.

In this work, we are interested in possibilities to facilitate RL for this challenging task and enhance the generalization ability of learned agents across initial conditions and flow scenarios by means of integration of prior knowledge. More specifically, we suggest to extend the reward function with domain-informed or physics-informed macroscopic quantities that can easily be measured: In our case we focus on promoting Bénard cell merging and study its effect on robust flow control. In addition to the $8 * 48$ grid of sensors, we assume a dense horizontal measurement of the vertical velocity field at the middle of the vertical axis, i.e. $u_y(x, y = 0, t)$, inspired by the experiment in [16]. We detect potential cell locations c_i by finding positive peaks in this measurement.[6] In our setup, there are usually two convection cells with horizontal coordinate c_1 and c_2, and we can simply compute:

$$\mathrm{celldist} = \min(|c_1 - c_2|, 2\pi - |c_1 - c_2|)\,, \tag{6}$$

which is the horizontal distance between the cells in the periodic domain $x \in [0, 2\pi]$. In the general case of an arbitrary number of cells, we simply compute the maximum of the pairwise cell distances to summarize the overall degree of cell merging. Next, we modify the reward function to include the cell distance as follows:

$$R(s_t) = (1-\alpha)\left(1 - \frac{\mathrm{Nu}(s_t)}{\mathrm{Nu}_{\mathrm{Base}}(\mathrm{Ra})}\right) + \alpha\left(1 - \frac{\mathrm{celldist}(s_t)}{\pi}\right), \tag{7}$$

where the coefficient $\alpha \in [0, 1]$ balances the importance of cell distance and the Nusselt number in the total reward. The quantity $(1 - \mathrm{celldist}(s_t)/\pi)$ ranges from 0, when the cells are maximally separated,[7] to 1, in case of a single merged cell. Hence, for both terms in the reward, the least and most desirable states are represented by 0 and 1, respectively.

[6] We used find_peaks from scipy.signal with height=0.

[7] Note that π is the maximum possible distance on the periodic domain $x \in [0, 2\pi]$.

III. Experiments and Results

We evaluate the effect of the domain-informed RL agents on generalization performance in three experiments: In Experiment 1, we exclude domain knowledge by using $\alpha = 0$ in (7) and term the resulting agents as *uninformed*. In Experiment 2, we study the effect of including the domain knowledge, i.e. $\alpha > 0$, and we call the resulting agents *domain-informed* (or DI). In Experiment 3, we explicitly study for both cases the generalization performance to other flow regimes by using the agents that were trained on $\mathrm{Ra} = 10^5$ to control the flow at $\mathrm{Ra} = 10^4$ and $\mathrm{Ra} = 10^6$. In all experiments, we evaluate the agent at the full range of Ra (i.e. different levels of chaotic convection) across a variety of initial conditions.

A. Episode Rollouts and PPO training

We trained agents on Rayleigh numbers

$$\mathrm{Ra} \in \{10^4, 10^5, 10^6, 5 \cdot 10^6\}\,.$$

The flows in the regime $10^6 < \mathrm{Ra} < 5 \cdot 10^6$ have a significant chaotic nature. Hence, the regime $\mathrm{Ra} \approx 10^7$ cannot reasonably be controlled in our scenario, and therefore we do not consider this regime. Each simulation starts from a no-motion state with small random perturbations in the vertical temperature gradient. Due to equivariance of the dynamics under horizontal shifts, these small random perturbations lead to random horizontal shifts in the obtained cell configurations. For each Ra, we generated 37 initial conditions in this manner.

During training, the agent is exposed to 20 of these convective states. Furthermore, the agent is validated on 5 unseen states and the best performing agent during the training procedure is saved. Final evaluation is performed on the remaining 12 independent convective states not seen during training or validation. This setup provides a strong test of the agent's generalization capabilities.

We use the PPO implementation from [18] and collect training data from multiple parallel rollouts. A full description of PPO training parameters (number of rollouts, training times, etc) is provided in Appendix C.

B. Experiment 1: Uninformed RL-based flow control

We first set a baseline by evaluating the performance of uninformed RL agents in reducing convective flow using ($\alpha = 0$) in (7). In Fig. 2, three results stand out: (1) The uninformed RL agent ($\alpha = 0$) shows significant promise in reducing the convective flow, as for all considered Ra, it reduces the Nusselt number significantly. (2) The standard deviation is very low, meaning a consistent Nusselt number reduction is achieved across all test checkpoints. (3) PD control, which is used in lab experiments, can only achieve satisfactory performance for the laminar flow at $\mathrm{Ra} = 10^4$.

For the flow at $\mathrm{Ra} = 10^4$, we show in Fig. 3 the temperature field over time when controlled by the uninformed agent. In Fig. 3a and 3b, we see that the uninformed agent starts to merge the left cell with the right cell. After the merge, the agent gradually widens the single cell (Fig. 3c and 3d), which it identified as a strategy to reduce the overall convective

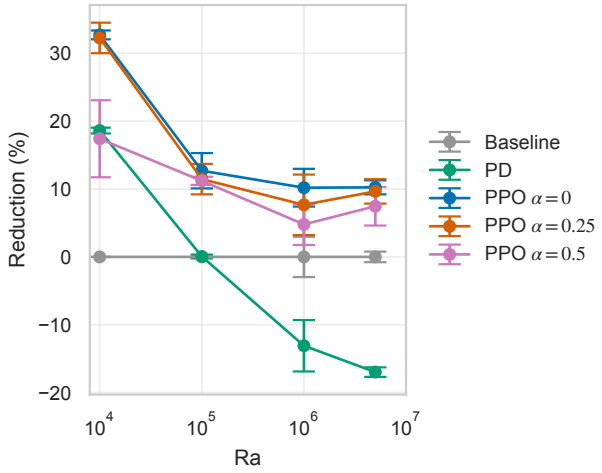
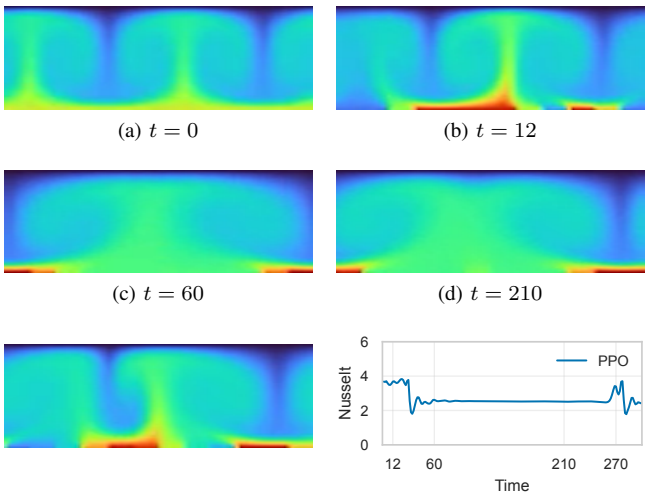


Fig. 2. The relative reduction of the Nusselt number with respect to the uncontrolled baseline for each control method on each Ra. For each test checkpoint we computed the mean Nusselt number over the entire episode and then calculated the percentage change relative to the mean Nusselt number of the uncontrolled baseline. Here, we show the average and standard deviation computed over the 12 test checkpoints. Although the results shown correspond to a single training run per method, preliminary multi-seed experiments indicated stable training behavior, with variations in the achieved Nusselt number reduction typically below 2% across initializations.



(a) $t = 0$ (b) $t = 12$ (c) $t = 60$ (d) $t = 210$ (e) $t = 270$ (f) Nusselt reduction of rollout

Fig. 3. Flow control by an *uninformed agent* for a typical test set episode at $\text{Ra} = 10^4$. Several temperature fields throughout the episode are shown. Red is hottest, dark blue is coldest.

flow. This widening results in a split of the single-cell into the two-cell configuration again (Fig. 3e). Hence, the flow is not stabilized under the control. In Fig. 3f, we marked the time points of the snapshots on the x-axis: The cell merge is associated with a significant decrease in Nu which then becomes stationary. Later, the brief split into two cells at $t = 270$ is associated with an instability in the Nusselt number, indicating that the flow is not stabilized under the control.

For $Ra > 10^4$, successful control with the uninformed agent was almost never achieved, because the uninformed agent remained stuck in simpler strategies that amount to heating between the cells. Although we observed that more exploration in PPO can somewhat alleviate the issue, this requires much more training effort due to an extensive number of rollouts. These baseline results highlight that without domain knowledge, stabilization of chaotic flows is not consistently achieved, motivating the need for domain-informed priors.

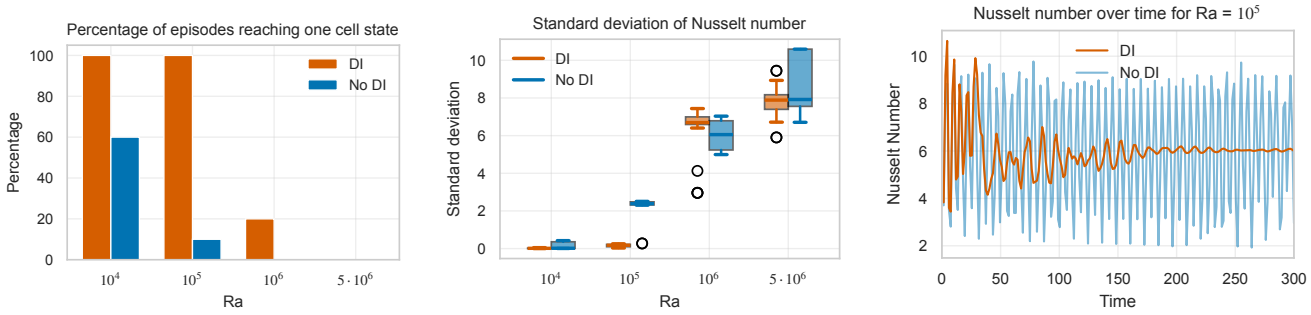
C. Experiment 2: Exploring stabilization properties by adding domain knowledge

We experiment with the inclusion of domain knowledge through promoting Bénard cell merging using the reward function in (7) with $\alpha > 0$. For the coefficient α that determines the importance of this term, we experimented with a range of values in $\alpha \in [0, 1]$. In general, we observed a trade-off between Nusselt number reduction and stabilization through cell merging. Low α gives better time-averaged Nusselt number reduction performance but has difficulty to achieve flow stabilization through cell merging, while high α leads to merged cells with steady flows but slightly worse Nusselt number reduction. As shown in Fig. 2, the value $\alpha = 0.25$ resulted in robust performance in Nusselt number reduction across all Ra and turned out to be the best trade-off between Nusselt number reduction and flow stabilization. Hence, we discuss the results for $\alpha = 0.25$ in the rest of the work.

The key advantage of domain-informed RL is its ability to consistently stabilize initially chaotic flows, such as at $\text{Ra} = 10^5$, into a steady state. We summarize the main differences using key statistics in Fig. 4: Domain-informed training always merged cells in the regime $\text{Ra} = 10^4$ and $\text{Ra} = 10^5$, as shown in Fig. 4a. In addition, this resulted in consistent flow stabilization in this regime (Fig. 4b). This is significant, as for $\text{Ra} = 10^5$ the uncontrolled flow is chaotic and became consistently stable under the domain-informed control.[8] To illustrate the flow stabilization further, Fig. 4c shows examples of the Nusselt number during typical episodes of the Domain-Informed agent (in orange) and the uninformed agent (in blue), where the domain-informed agent achieves a stationary Nusselt number. Fig. 4b confirms that stationary flows were consistently achieved for $\text{Ra} = 10^4$ and $\text{Ra} = 10^5$. For the regime $\text{Ra} > 10^5$, the differences between the domain-informed and the uninformed control are minor: in this highly chaotic regime, both agents resort to the simple control strategy that still reduces the Nusselt number (see Fig. 2). In Fig. 7 of the Appendix, we also see that the domain-informed agents achieve cell merging earlier in the episodes. In summary, in the regime $\text{Ra} = 10^5$, the domain-informed agent consistently transformed a chaotic flow into a stable single-cell configuration with nearly constant Nusselt number over time, a capability absent in uninformed agents.

For $\text{Ra} = 10^4$ and $\text{Ra} = 10^5$, we confirmed that the domain-informed agents achieve flow stabilization early in training. The uninformed agents also tend to overfit to the training

[8]The videos given in the repository (see Appendix A) further illustrate this behavior.



(a) (b) (c)

Fig. 4. The effect of Domain-Informed (DI) training vs. uninformed training (No DI) on flow control, shown by three key statistics computed on the test set: (a) Percentage of episodes where Bénard cells were merged. (b) Standard deviation of the Nusselt number over time, computed over the last 40 actions. (c) Nusselt number during a typical example test episode.

set. In contrast, the domain-informed training objective is less prone to over-fitting and achieves much better generalization to unseen conditions beyond the training set.

D. Experiment 3: Generalization to different flow regimes

We used the domain-informed agent trained on flows at $\mathrm{Ra} = 10^5$ for controlling flows at $\mathrm{Ra} = 10^4$ and $\mathrm{Ra} = 10^6$ and we compared with the same scenario using the uninformed agent. Fig. 5 shows that the domain-informed agent has consistent success in achieving cell merging and associated flow stability, whereas the uninformed agent does not achieve these properties. Counter-intuitively, the domain-informed performance on $\mathrm{Ra} = 10^6$ is better than when training an agent on $\mathrm{Ra} = 10^6$ (cf. Fig. 4). We attribute this to the fact that training of policies in lower chaotic regimes is easier, and such policies may therefore show better generalization ability to more chaotic regimes. The success of the domain-informed agents in cell merging resulted in stationary Nusselt numbers for $\mathrm{Ra} = 10^4$ (Fig. 5c orange line), and a significant reduction in variation for $\mathrm{Ra} = 10^6$, which we show in the Appendix in Fig. 8. This underlines that the learned stabilization strategy generalizes to other flow regimes including higher levels of chaos.

IV. Discussion

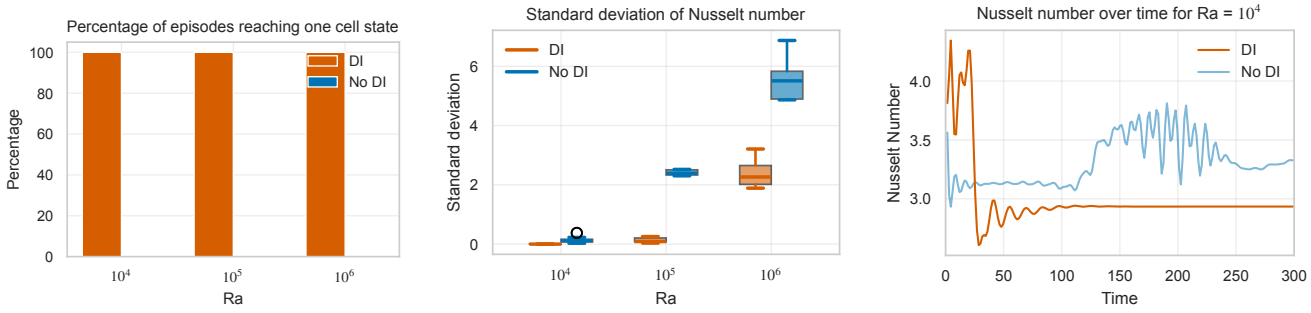
Our results provide clear evidence that the inclusion of domain knowledge is a crucial component to achieve robust control and stabilization of chaotic fluid flows. A key insight is that elegant domain-informed reward design plays a critical role in enabling robust flow control, obtaining steady flows in regimes where the uncontrolled flow is chaotic. Although there are slight differences in the sensor setup between the domain-informed and the uninformed cases, the cell distance can also be well-approximated from the dense sensors in the horizontal direction used for the uninformed case, which would not change the results. Moreover, our results indicated that a balancing value of $\alpha = 0.25$ in the reward in (7) already resulted in robust performance across a large flow regime, making fine-tuning not critical, which is a major advantage for physical implementation.

For the uninformed agent, we only obtained steady flows for $\mathrm{Ra} = 10^4$, but this state was not stable under the control (Fig. 3e). Encouraging Bénard cell merging was an effective way to guide the agent away from the simple two-cell control strategy towards a one-cell setup that has lower convective heat transfer. In addition, those one-cell setups were associated with steady flows: A key result is that for $\mathrm{Ra} = 10^5$, which is a regime with chaotic flow in the uncontrolled setting, the domain-informed agent consistently merged cells and the control transformed the chaotic flow into a stable, steady flow with constant Nusselt number over time (e.g., see Fig. 4c). The domain-informed agent also exhibited generalization ability to flows up to $\mathrm{Ra} = 10^6$, signifantly reducing the Nusselt variation over time.

A. Conclusion

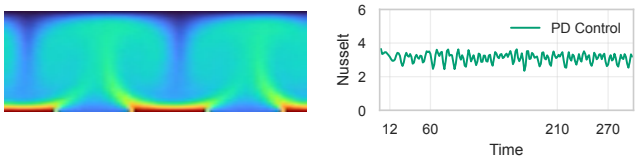
This work bridges artificial intelligence and engineering by demonstrating how domain-informed RL can enable robust control of chaotic flows. Through elegant domain-informed reward design, our agents learned significantly more robust control across initial conditions and exhibited a level of generalization ability across flow regimes. These insights highlight the potential of domain-informed RL for effective robust control in complex flow regimes, especially in settings where conventional control is inadequate.

A challenging next step is the control of 3D convective flows, including turbulent effects. As our experiments in 2D indicated that including domain knowledge becomes increasingly important for complex flows where control is feasible, the inclusion of domain-informed rewards may well prove as a crucial component for enabling flow control in 3D. Our proposed reward is strongly tied to 2D, but similar domain-informed counterparts could be formulated in 3D. It could also be interesting to consider a middle ground between a general formulation of the reward and a reward that is tied to the specific setting (such as our cell-merging term). For instance, one could consider more general desirable fluid flow properties or stability measures, which would be applicable to a wider range of fluid flow applications. We also aim to further reduce sample requirements by learning surrogate models



(a) (b) (c)

Fig. 5. Testing the generalization performance: we trained an agent on flow with Ra $= 10^5$ and evaluated the performance of this agent on Ra $= 10^4$, Ra $= 10^5$ and Ra $= 10^6$. We also compare to uninformed training at Ra $= 10^5$. Shown are (a) the percentage of merged Bénard cells, (b) the standard deviation over time of the Nusselt number, and (c) the Nusselt number during a typical test episode at Ra $= 10^4$.



(a) $t = 30$ (b) Nusselt Reduction of rollout

Fig. 6. Example of linear PD control for an Ra $= 10^4$ rollout of the system.

of the dynamics, which allows for agent training using the surrogate model instead of costly rollouts of the environment. A starting point would be to extend uncontrolled surrogates for RBC, for instance as in [19], [20], with the inclusion of control. Eventually, these works should lead to the exploration of the use of robust RL agents for the control of real-world convective flows, starting with controlled lab experiments.

Appendix A
Code and Videos

For full reproducibility of our experiments, we provide our models and code at https://github.com/HammerLabML/RBC-Control-SARL with detailed README and installation instructions. We also encourage the reader to view the videos at https://github.com/HammerLabML/RBC-Control-SARL/blob/main/examples.md.

Appendix B
PD control

In Fig. 6 we show an example of simple linear PD control on Ra $= 10^4$. This strategy of heating between convection cells can somewhat reduce the Nusselt number, but proves inadequate for higher Ra and does not result in steady flows. For instance, in Fig. 6b, we show that already for Ra $= 10^4$, the Nusselt number over time is not stationary, because the flow is not steady. Hence, although PD control has been used in lab experiments of RBC flow control, it is limited in changing the properties of the flow [15], [16].

Appendix C
Reinforcement Learning training details

The agent always applies an action for 1.5 simulation seconds. For each experiment we performed 100 rollouts in total, each of which consisted of 20 parallel agents performing 200 stochastic actions starting from a random convective state from the training set (20 random states in total). Hence, 4000 (state, action, reward) samples are gathered on which the policy update occurs after each rollout. Further parameters: Discount factor $\gamma = 0.99$, entropy $\beta = 0.01$, two-layer (64 hidden units) neural networks for both the actor and critic. Note that we extensively tested the number of hidden units and 64 was consistently enough for the control problem. 10 training epochs of the networks are performed after each rollout.

A. *Satisfying the constraints on the actuation of the heaters*

The constraints that the actuation value a_i for a heater i should be in $[T_b - C, T_b + C]$, where $C = 0.75$, and in addition that the mean calculated over all heaters is $\langle a_i \rangle_{i=1,2,\ldots,12} = T_b$, are implemented by transforming each of the control outputs a_i as follows:

$$a_i \rightarrow a_i - \frac{\sum_{i=1}^{N} a_i}{N}, \quad a_i \rightarrow \frac{\hat{a}_i C}{\max(1, |a_i|)} + T_b, \tag{8}$$

which is a centering of the heater outputs, followed by re-scaling to the range $[T_b - C, T_b + C]$. The same transformation was presented in previous works, for example [7], where RL was applied to the laminar flow regime at Ra $= 10^4$.

Appendix D
Additional flow statistics of the controlled scenarios

Note that in Fig. 7, because for the uninformed agents we rarely achieve a one-cell state, we only included the timings of the episodes that did achieve a one-cell state. We clearly see that the domain-informed agents always achieve cell merging earlier, and the difference is large for the higher Rayleigh number case.

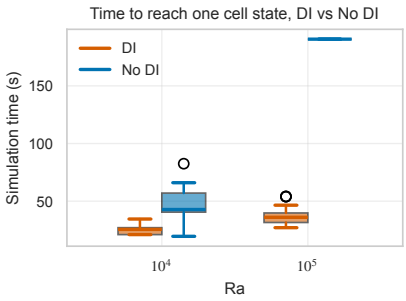


Fig. 7. Domain-informed agents achieve cell merging faster in the episode than the uninformed agents.

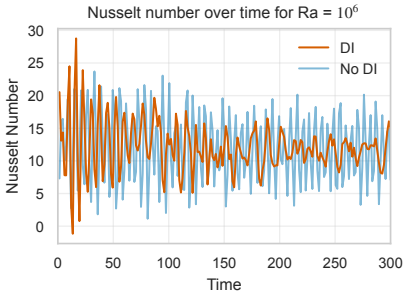


Fig. 8. Generalization to Ra $= 10^6$ for agents trained on flows of Ra $= 10^5$.

Fig. 8 shows agents trained on flows at Ra $= 10^5$ for controlling a flow at Ra $= 10^6$. The domain-informed agent considerably reduces the Nusselt number variation, even though it does not reach a stable one-cell state.

References

[1] H. Wang, Y. Cao, Z. Huang, Y. Liu, P. Hu, X. Luo, Z. Song, W. Zhao, J. Liu, J. Sun, S. Zhang, L. Wei, Y. Wang, T. Wu, Z.-M. Ma, and Y. Sun, "Recent Advances on Machine Learning for Computational Fluid Dynamics: A Survey," Aug. 2024, arXiv:2408.12171 [cs].

[2] R. Vinuesa, S. L. Brunton, and B. J. McKeon, "The transformative potential of machine learning for experiments in fluid mechanics," *Nature Reviews Physics*, vol. 5, no. 9, pp. 536–545, Sep. 2023.

[3] J. Tang and H. H. Bau, "Stabilization of the no-motion state in Rayleigh-Bénard convection through the use of feedback control," *Phys. Rev. Lett.*, vol. 70, pp. 1795–1798, Mar 1993.

[4] Y. Gu, V. Hegde, and K. J. M. Bishop, "Measurement and mitigation of free convection in microfluidic gradient generators," *Lab Chip*, vol. 18, no. 22, pp. 3371–3378, 2018, publisher: The Royal Society of Chemistry.

[5] P. Garnier, J. Viquerat, J. Rabault, A. Larcher, A. Kuhnle, and E. Hachem, "A review on deep reinforcement learning for fluid mechanics," *Computers & Fluids*, vol. 225, p. 104973, 2021.

[6] C. Banerjee, K. Nguyen, C. Fookes, and M. Raissi, "A survey on physics informed reinforcement learning: Review and open problems," *Expert Systems with Applications*, vol. 287, p. 128166, Aug. 2025.

[7] C. Vignon, J. Rabault, J. Vasanth, F. Alcántara-Ávila, M. Mortensen, and R. Vinuesa, "Effective control of two-dimensional Rayleigh–Bénard convection: Invariant multi-agent reinforcement learning is all you need," *Physics of Fluids*, vol. 35, no. 6, p. 065146, 06 2023.

[8] J. Vasanth, J. Rabault, F. Alcántara-Ávila, M. Mortensen, and R. Vinuesa, "Multi-agent Reinforcement Learning for the Control of Three-Dimensional Rayleigh–Bénard Convection," *Flow, Turbulence and Combustion*, pp. 1–37, 2024.

[9] G. Beintema, A. Corbetta, L. Biferale, and F. Toschi, "Controlling Rayleigh–Bénard convection via reinforcement learning," *Journal of Turbulence*, vol. 21, no. 9-10, pp. 585–605, 2020.

[10] Q. Chen and C. R. Constante-Amores, "Stabilizing Rayleigh-Bénard convection with reinforcement learning trained on a reduced-order model," 2025, arXiv:2510.26705 [physics.flu-dyn].

[11] A. Pandey, J. D. Scheel, and J. Schumacher, "Turbulent superstructures in Rayleigh-Bénard convection," *Nature Communications*, vol. 9, no. 2118, 2018.

[12] M. Mortensen, "Shenfun: High performance spectral galerkin computing platform," *Journal of Open Source Software*, vol. 3, no. 31, p. 1071, 2018.

[13] C.-H. Hsia and T. Nishida, "A Route to Chaos in Rayleigh–Bénard Heat Convection," *Journal of Mathematical Fluid Mechanics*, vol. 24, no. 2, p. 38, Mar. 2022.

[14] K. P. Iyer, J. D. Scheel, J. Schumacher, and K. R. Sreenivasan, "Classical 1/3 scaling of convection holds up to $Ra = 10^{15}$," *Proceedings of the National Academy of Sciences*, vol. 117, no. 14, pp. 7594–7598, 2020.

[15] L. E. Howle, "Active control of Rayleigh–Bénard convection," *Physics of Fluids*, vol. 9, p. 1861–1863, 1997.

[16] M. C. Remillieux, H. Zhao, and H. H. Bau, "Suppression of Rayleigh-Bénard convection with proportional-derivative controller," *Physics of Fluids*, vol. 19, no. 1, p. 017102, 01 2007.

[17] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," 2017, arXiv:1707.06347 [cs.LG].

[18] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.

[19] T. Markmann, M. Straat, and B. Hammer, "Koopman-Based Surrogate Modelling of Turbulent Rayleigh-Bénard Convection," in *2024 International Joint Conference on Neural Networks (IJCNN)*, 2024, pp. 1–8.

[20] M. Straat, T. Markmann, and B. Hammer, "Solving Turbulent Rayleigh-Bénard Convection using Fourier Neural Operators," in *Proceedings of the 33rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2025)*. Bruges, Belgium: i6doc.com, Apr. 2025.