Reinforcement learning for cooling rate control during quenching

Elie Hachem, Abhijeet Vishwasrao, Maxime Renault, Jonathan Viquerat, Philippe Méliga

Reinforcement learning for cooling rate control during quenching

Abstract

In the process of quenching heat treatment, it is critical to establish the optimal process parameters producing the least residual stress magnitudes and related distortions and/or cracking, to reduce the cost of manufacturing high-quality components with intricate and durable designs and meet the stringent requirements of a broad range of high-performance industries. Because such effects occur as the result of uneven cooling in different regions of the quenched part, a feasible control objective is thus to enhance the spatial uniformity of heat removal, to prevent spatial gradients of irreversible strains typically originating from heterogeneous plastic deformation. For decades this process has been largely driven by trial and error, intuition and experience. In this study, a single-step Deep Reinforcement Learning (DRL) algorithm is used to provide the best possible cooling rate in industrial quenching processes governed by coupled pseudo-compressible Navier–Stokes and heat equations, along with latent heat formulation. The numerical reward fed to the neural network is computed with an in-house stabilized finite elements environment combining variational multi-scale (VMS) modeling of the governing equations, immerse volume method, and multi-component anisotropic mesh adaptation. A case of Rayleigh–Bénard convection in a high-aspect ratio, closed cavity is used first as testbed for the proposed methodology. In a second phase, we tackle several quenching numerical experiments aiming at improving temperature homogeneity within two-dimensional components in various shapes, whose results showcase the potential of DRL to produce unanticipated solutions by learning the effect of highly unsteady boiling flow physics on the temperature distribution.

Keywords: Deep Reinforcement Learning; Quenching; Computational Fluid Dynamics; Thermal Control, Multiphase Boiling

1. Introduction

Quenching is a heat treatment process used to modify the mechanical properties of steel materials. It consists in heating a part to achieve specific microstructure and material properties (*e.g.*, hardness, strength), after which the part is quickly cooled through the austenitizing temperature of a liquid quenchant that can be water, oil or brine, depending on the material and the desired properties. High cooling rates in the quenching process suppress the diffusion-controlled phase transformations and promote non-diffusional phase transformations to form martensite, one of the desired phases in quenched steel [1]. Although probably the oldest heat treatment process used by man to harden and strengthen steel, quenching remains vital as a safety procedure, to improve product durability and performance in various heavy industries with tight tolerances and high process repeatability requirements. Typical examples include the energy sector (e.g., to manufacture seamless rolled rings), the automotive and aerospace industries (rings, gears, shafts and other transmission parts) or the construction industry (to avoid distortions in rods and bars).

Quenching effectiveness hinges in large measure on the ability of the quenchant to achieve maximal cooling without heating up. Meanwhile, in the absence of a systematic approach to select optimal process parameters, steel materials with varying compositions can respond differently to the hardening treatment, resulting in defects, rejections, reworks and added costs [2]. It is widely accepted that thermal expansions caused by large temperature gradients (which the present study focuses on), volumetric expansions caused by martensitic transformations, or the combination of both produce high residual stresses, contributing to distortions, cracking and fractures that all negatively impact the microstructure of the part and lead to reduced fatigue strength [3]. Despite a large body of literature on this topic contributing to advancements in manufacturing processes, failures persist in the quenching process, with heavy or intricately shaped components having

the highest rejection percentages [4]. In the present context of tackling climate change while ensuring sustainable growth, the capability of optimally controlling the outcome of quenching is more relevant than ever, to deliver durable, high-quality parts at manageable costs while minimizing the many types of waste that can occur during manufacturing [5].

Quenching involves heating the component at temperatures of the order of 1000°C. The initial temperature across any given geometry is constant [1], and the temperature distribution during cooling depends solely upon the outward heat flux from the component, and thereby upon highly localized heat transfer coefficients. The complexity of the quenching process is related to the boiling heat transfer and phase transformations occurring on the surface of the immersed part (mostly the nucleation phase), that makes is very sensitive to small variations in process parameters. For instance, for a part to have uniform heat transfer, any vapor pockets must be minimized for the quenching medium to wrap perfectly around its surface. This requires identify relevant process parameters, a real challenge that requires a perfect balance in the choice of (among many other factors) the austenitizing temperature, the immersion rate and the orientation of the treated component, and the temperature, volume and agitation of the quenching medium. Actually, this must be repeated for each particular geometry, as even the slightest asymmetry in the shapes, combined with tiny variations in process parameters, can add to the lack of uniformity and cause the appearance of unwanted residual stresses associated with diminished mechanical properties [6].

For decades, process parameters have been essentially adjusted through intuition, trial and error, and professional experience. The premise of this research is that this highly complicated task can be rationally and efficiently solved using reinforcement learning (RL), a machine learning method for solving sequential decision-making problems. While RL has long been limited to low dimensional problems, several major obstacles have been lifted using the feature extraction capabilities of deep neural networks and their ability to handle high-dimensional state spaces, giving rise to Deep Reinforcement Learning (Deep RL or DRL). This has yielded unprecedented efficiency after short training spans in many domains such as robotics [7], language processing [8] or games [9, 10], but DRL is also used in many industrial applications, including autonomous cars [11], or data center cooling [12]. The potential application fluid mechanics is also highly promising, for which efforts are ongoing thanks to the sustained commitment from the machine learning community. A few dozen studies provided insight into the performance improvements to be delivered, with particular focus on shape optimization and flow control applied to drag reduction; see [13, 14] for recent reviews. Meanwhile, the literature on thermal control is scarce, with (at time of writing) only a handful of studies applying DRL-based approaches to control natural [15–17] and forced convection [16, 18, 19], although DRL has become a quickly emerging topic in a wide range of thermal applications, from the shape optimization of heat exchangers [20] to the implementation of thermal digital twins [21], including energy efficiency in civil engineering [22] and the estimation of effective statistical properties in complex media [23].

This work aims at introducing DRL into the field of quenching control. It stands as a follow-up of our previous work on DRL-based control of forced convection [16, 18], and combines DRL with advanced immersed methods for the simulation of liquid-vapor phase change, to extend the scope to boiling and increase the complexity of the targeted applications. The proposed framework leverages the capacity of neural networks to accurately approximate the mapping function between input and output spaces, as well as the dynamic programming inherent in the reinforcement learning algorithm. There is no similar study in the literature, to the best of our knowledge, for which a possible explanation is the lack of tried and tested computational solvers capable of reliably simulating the quenching of solid parts. In this context, to act as DRL agent, we use the single-step Proximal Policy Optimization (single-step PPO) algorithm introduced in [24, 25], intended for optimization and open-loop control problems whose optimal policy to be learnt by the agent is state-independent (in which case it suffices to update the neural network parameters only once per episode). The reward function used to train our PPO agent is computed by an in-house, high-fidelity computational framework that predicts accurately the boiling heat transfer behavior of a liquid in the near field of a heated immersed solid, while taking into account the gas-liquid phase changes, the vapor formation and their dynamics, and ultimately, the cooling of the solid [26, 27]. The effectiveness of the DRL-CFD approach is demonstrated by evaluating the homogeneity of the cooling effect on various systems comprising a 2-D part with controlled orientation in a liquid tank. Of note, this is a proof of concept study to lay the groundwork for future research in the field,

not a comparable study demonstrating the competitiveness of DRL-CFD by benchmarking the performance against generic models available off-the-shelf. We therefore focus on positioning the method as an efficient and theoretically well-founded tool to find the best process parameters and achieve the desired final characteristics, and exchange views on the main challenges that should be considered to realize its potential for application in real-world manufacturing environments.

With these considerations in mind, the paper is structured as follows: in section 2, we outline the main features of the finite element CFD environment used to simulate the quenching process and compute the numerical reward fed to the DRL agent.while taking into account liquid-vapor-solid interactions with boiling heat transfer and liquid-vapor transformations. Section 3 breaks down the baseline principles of DRL and PPO, together with the specifics of the single-step PPO algorithm. Section 4 provides the particulars of DRL experiments and implementation, and revisits a Rayleigh–Bénard convection case adapted from [16] for the purpose of validation and assessment part of the method capabilities. In, section 5, DRL is used to improve the temperature homogeneity within various parts immersed in a liquid tank, whose geometrical complexity ranges from the simplest rectangular brick shape to intricate, irregular shapes of engineering interest. Conclusion and strategies for future works in this area are presented in section 6.

2. Computational fluid dynamics and phase change solver

This section breaks down the main ingredients of the adaptive Eulerian framework for the simulation of both boiling and evaporation phenomena occurring at the interface of a heated 2-D solid immersed in a liquid tank. Exhaustive derivation and implementation details are provided in Ref. [27], to which the interested reader is referred for further deepening. Suffice it to say here that a pseudo-compressible model accounting for mass transfer at the liquid-vapor interface is solved with the Immerse Volume Method [28, 29], which allows to compute the heat transfer between the solid and the (liquid) quenchant from the individual material properties on either side of it. This in turn obviates the need to compute a heat transfer coefficient, which would have been a limiting issue from the present numerical experiments where varying the shape and orientation of the solid is integral to the optimization process.

2.1. Interface capturing method

The level set method is employed as an interface capturing technique to identify and monitor the evolution of the liquid-phase interface using the zero iso-value of a smooth level set function. This function distributes the corresponding physical properties in space according to a mixing law. Let Ω represent the entire domain, and Ω_l and Ω_v denote the liquid and vapor domains, respectively. The level set function is a signed distance function from the interface $\Gamma = \Omega_l \cap \Omega_v$, defined at each node X as follows:

$$\phi = \begin{cases} -\mathrm{dist}(X,\Gamma) & \text{if } X \in \Omega_l\,, \\ 0 & \text{if } X \in \Gamma\,, \\ \mathrm{dist}(X,\Gamma) & \text{if } X \in \Omega_v\,, \end{cases} \tag{1}$$

with the convention that $\phi > 0$ in the vapor domain. In the absence of mass transfer between the liquid and vapor phases, the evolution of the level set is governed by the transport equation.

$$\partial_t \phi + \boldsymbol{u} \cdot \nabla \phi = 0\,, \tag{2}$$

with $\boldsymbol{u}$ a velocity. The level set, defined as a distance function, satisfies $||\nabla\phi|| = 1$. However, it can lose this property during the convection process, in which case it requires reinitialization to prevent numerical instabilities. A common method for reinitialization is solving the Hamilton–Jacobi equation

$$\partial_\tau \phi + s(\phi)(||\nabla\phi|| - 1) = 0\,, \tag{3}$$

where τ is a pseudo time-step and $s(\phi)$ is the sign function of ϕ.

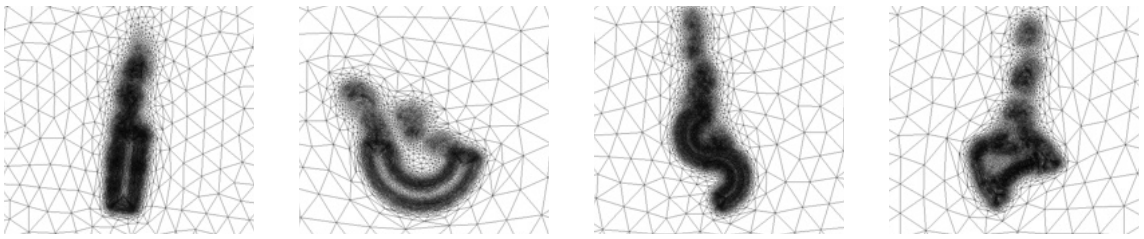

Figure 1: Examples of anisotropic meshes used to simulate the boiling phenomenon.

Once the level set function is computed, the individual physical properties (for example ρ_l and ρ_l, respectively the liquid and vapor densities) are distributed using a mixing law according to

$$\rho = \rho_l(1 - H_\epsilon(\phi)) + \rho_v H_\epsilon(\phi)\,, \tag{4}$$

where H_ϵ is the smoothed Heaviside function defined as

$$H_\epsilon(\phi) = \begin{cases} 0 & \text{if } \phi < -\epsilon\,, \\ \frac{1}{2}(1 + \frac{\phi}{\epsilon} + \frac{1}{\pi}\sin(\pi\frac{\phi}{\epsilon})) & \text{if } |\phi| \leq \epsilon\,, \\ 1 & \text{if } \phi > \epsilon\,, \end{cases} \tag{5}$$

and $\epsilon = 2h_\infty$ is a regularization parameter set here to twice the mesh size in the normal direction to the interface.

2.2. Anisotropic mesh adaptation

Difficulties may arise in an immersed multiphase framework due to the discontinuities in material properties between the solid and fluid regions. These discontinuities are particularly challenging when they intersect the mesh elements arbitrarily, potentially compromising the accuracy of the solution or causing it to fail entirely. To address this, we employ the anisotropic mesh adaptation technique described in [30]. This approach generates highly stretched, well-oriented elements and distributes the fluid properties as accurately and smoothly as possible over a minimal thickness around the interface. This allows for the effective capture of sharp gradients at a low computational cost, ensuring consistency between the solution anisotropy and the mesh. This is achieved by calculating modified distances from a metric, a symmetric positive-definite tensor whose eigenvectors define preferential directions along which mesh sizes are determined based on the corresponding eigenvalues. This metric is isotropic far from the interface, with the mesh size set to h_∞ in all directions. However, near the interface, the metric becomes anisotropic, with the mesh size set to $h_\perp$ in the direction normal to the liquid/vapor interface and to h_∞ in the other directions. For a desired thickness δ, this can be expressed as follows:

$$\boldsymbol{M} = K(\phi)\boldsymbol{n}\otimes\boldsymbol{n} + \frac{1}{h_\infty^2}\boldsymbol{I} \qquad \text{with} \qquad K(\phi) = \begin{cases} 0 & \text{if } |\phi| \geq \delta/2\,, \\ \frac{1}{h_\perp^2} - \frac{1}{h_\infty^2} & \text{if } |\phi| < \delta/2\,, \end{cases} \tag{6}$$

where $\boldsymbol{n} = \nabla\phi/||\nabla\phi||$ is the normal to the interface deduced from the gradient of the level set. An a posteriori anisotropic error estimator is then used to minimize the interpolation error while maintaining a fixed number of edges in the mesh. This is done using multi-component error vectors taking into account the gradients of multiple scalar and/or vector fields [30–33]. In the following numerical experiments, adaptivity combines velocity components and magnitude, temperature and level set, all normalized by their respective global maximum to ensure that a field much larger in magnitude does not dominate the error estimator. Examples of the adapted meshes generated in this study are shown in figure 1. These examples illustrate the appropriate refinement and deformation of mesh elements, which are extremely fine and elongated near the interfaces between the solid, liquid, and vapor, but coarse and uniform away from the interfaces.

2.3. Phase change model

The multiphase framework used to simulate phase change problems relies on pseudo-compressible mass and momentum conservation written in the form of modified Navier–Stokes equations

$$\nabla \cdot \boldsymbol{u} = \dot{m} \left(\frac{1}{\rho_v} - \frac{1}{\rho_l} \right) |\nabla \phi| \delta_\epsilon(\phi) \,, \tag{7}$$

$$\rho(\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u}) - \nabla \cdot (2\mu \boldsymbol{\varepsilon}(\boldsymbol{u})) + \nabla p = \boldsymbol{f}_\gamma + \rho \boldsymbol{g} \,. \tag{8}$$

Here, ρ represents the density, μ the dynamic viscosity, $\boldsymbol{u}$ the velocity, p the pressure, $\epsilon(\boldsymbol{u})$ the rate of deformation tensor, $\boldsymbol{g}$ the gravity acceleration, and we note that the velocity is not divergence-free, since the continuity equation is forced by a surface mass transfer rate, denoted as $\dot{m}$, that measures the exchange of mass occurring at the liquid/vapor interface. Additionally, δ_ϵ is the Dirac function

$$\delta_\epsilon(\phi) = \begin{cases} \dfrac{1}{2\epsilon} \left(1 + \cos\left(\pi \dfrac{\phi}{\epsilon} \right) \right) & \text{if } |\phi| \leq \epsilon \,, \\ 0 & \text{if } |\phi| > \epsilon \,, \end{cases} \tag{9}$$

locating the interface and smoothed with the same regularization parameter ϵ as the Heaviside function (5). Lastly, $\boldsymbol{f}_\gamma$ is a body force representing the impact of surface tension, formulated within the framework of the Continuum Surface Force [34] as

$$\boldsymbol{f}_\gamma = -\gamma \kappa \delta_\epsilon(\phi) \boldsymbol{n} \,, \tag{10}$$

where γ is the surface tension coefficient and $\kappa = \nabla \cdot \boldsymbol{n}$ is the mean interface curvature. Likewise, energy conservation is expressed in the form of a modified heat equation

$$\rho c_p(\partial_t T + \boldsymbol{u} \cdot \nabla T) = \nabla \cdot (\lambda \nabla T) - \left(\mathcal{L} + (c_{p_v} - c_{p_l})(T - T_{sat}) \right) \dot{m} \delta_\epsilon(\phi) |\nabla \phi| \frac{\rho^2}{\rho_l \rho_v} \,. \tag{11}$$

Here, T denotes the temperature, T_{sat} the saturation temperature, $\mathcal{L}$ the liquid latent heat of vaporization, c_p the specific heat, c_{p_l} (resp. c_{p_v}) the specific heat in the liquid (and vapor) phases respectively, and k the thermal conductivity. Due to mass transfer, the level set equation (1) is consequently modified into

$$\partial_t \phi + \left(\boldsymbol{u} - \frac{\rho}{\rho_l \rho_v} \dot{m} \frac{\nabla \phi}{|\nabla \phi|} \right) \cdot \nabla \phi = 0 \,, \tag{12}$$

for the interface to be convected not only by the Navier–Stokes velocity, but also the velocity of the vapor front. The mass transfer rate $\dot{m}$ in the aforementioned general formulation is computed based on the balance of fluxes at the interface, hence

$$\dot{m} = \frac{\int_{\Omega_i} \delta_\epsilon(\phi)(-k_v \nabla T_v + k_l \nabla T_l) \cdot \boldsymbol{n} \mathrm{d}\Omega_i}{\int_{\Omega_i} \delta_\epsilon(\phi) \mathrm{d}\Omega_i} \,, \tag{13}$$

where we sum over all elementary volumes Ω_i intersected by the interface.

2.4. Variational multiscale modeling

All equations are solved using equal-order linear approximations for the velocity and pressure variables, for which we employ stabilized weak forms cast in the Variational Multiscale (VMS) framework. This approach enhances the stability of the Galerkin method by introducing additional integrals over the element interior. It effectively mitigates the node-to-node oscillations that typically arise when discretization schemes violate the Babuska–Brezzi condition. The fundamental concept involves decomposing all quantities into large and small-scale components, representing different levels of resolution. The effect of the unresolved small-scale details, beyond the finite element mesh resolution, is approximated on the large scale through consistently derived residual-based terms. Extensive validation and verification of this numerical framework accuracy and reliability are detailed in [27]. Interested readers are encouraged to consult this reference for comprehensive information on the VMS formulations, stabilization parameters, and discretization schemes applied to the phase-change model.

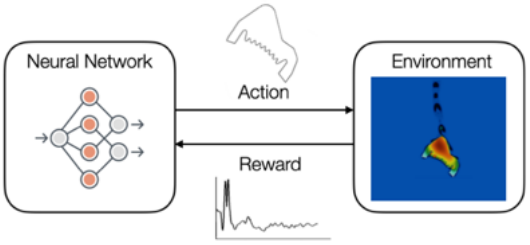

Figure 2: Sketch of the present single-step PPO action-loop.The quenching CFD environment with phase change and the DRL agent are coupled two-way through actions and rewards.At each episode, the same input state $\boldsymbol{s}_0$ is provided to the agent, which in turn provides n actions to n parallel environments. The latter return n rewards, that evaluate the quality of each action taken. Once all the rewards are collected, an update of the agent parameters is made using the PPO loss (14).

3. Deep reinforcement learning and single-step proximal policy optimization

In the following, quenching processes are optimized by solving a decision-making problem with reinforcement learning (RL), a method by which an agent learns to maximize rewards through trial-and-error interactions with its environment. At each step, the agent observes the state s_t of the environment and takes an action a_t, which leads to a reward r_t and a transition to the next state s_{t+1}. This repeats until a termination state is reached, the primary goal of the agent being to learn a sequence of actions that maximizes its cumulative reward over an episode (which is the reference unit for agent update). In the present context, the environment is a CFD simulation with phase change, that uses the stabilized finite element framework described above. The agent is a RL-trained neural network coupled two-way with the environment, as illustrated in Fig. 2: on the one hand, the actions sampled by the agent are used to generate the workpieces meshes immersed in the CFD simulation. On the other hand, the reward function needed by the agent to learn (here, a measure of cooling homogeneity) is obtained by post-processing of the CFD data.

3.1. Single-step deep reinforcement learning

In deep reinforcement learning (DRL), the agent is implemented as a neural network, most often structured into a series of fully connected layers, with information flowing from the input layer to the output layer through hidden layers, each of which acts as a function from $\mathbb{R}^m$ to $\mathbb{R}^n$. The neural network learns the relationship between input (action) and output (reward) data by iteratively adjusting the weights and biases through a process known as back-propagation, which moves from the output layer back through the hidden layers and to the input layer. This training process enables the network to refine its predictions and improve performance over time. In classical DRL, network updates are performed after multi-step episodes for the agent to learn the set of actions $a^\star$ yielding the highest possible reward. The present approach is conversely cast in the single-step deep reinforcement learning framework, an approach that has emerged from the premise that network updates can be performed after one-step episodes (of single-step episodes, hence by extension, single-step DRL) if the optimal behavior to be learned is independent of state. A single-step DRL agent learns instead the optimal mapping $f_{\theta^\star}$ such that $a^\star = f_{\theta^\star}(s_0)$, where s_0 is an input state, usually a constant vector, repeatedly passed to the agent (hence the stateless moniker). A significant advantage of single-step DRL is that it allows to use much smaller networks compared to the typical architectures used in traditional DRL approaches. This is because the agent does not need to learn a complex state-action relationship but only the transformation from a constant input state to a specific action.

3.2. Single-step proximal policy optimization

In the following, a neural network is trained with single-step Proximal Policy Optimization, the single-step variant of the ubiquitous PPO RL algorithm introduced in Ref. [24, 25] and shown in Ref. [16] to hold potential as a reliable, go-to black-box optimizer for natural and forced convection

heat transfer enhancement. In short, one neural network outputs the mean and variance of a d-dimensional multivariate normal distribution (with d the dimension of the action required by the environment). All variables are assumed to have equal variance and to be uncorrelated, meaning that the covariance matrix is identity, thereby establishing an isotropic sampling region for the upcoming episode. Actions drawn in $[-1, 1]^d$ are then mapped into relevant physical ranges, a step deferred to the environment as being problem-specific. Just like PPO, the algorithm computes adaptive learning rates for each policy parameter based on the gradient of the loss function

$$\mathbb{E}_{a\sim\pi_\theta}\left[\min\left(\frac{\pi_\theta(a)}{\pi_{\theta_{old}}(a)}, 1+\epsilon\,\mathrm{sgn}\left(\widehat{A}^{\pi_\theta}(a)\right)\right)\widehat{A}^{\pi_\theta}(a)\right], \tag{14}$$

where $\widehat{A}^{\pi_\theta}$ serves as a biased estimator of the advantage function A^{π_θ}, quantifying the convenience of taking action a compared to the average value (normalized to zero mean and unit variance). The ϵ parameter defines a clipping range that limits how much the new policy can deviate from the old policy. A negative (positive) advantage decreases (increases) the likelihood of taking action a, but always within a proportion smaller than ϵ. If this threshold is exceeded, the policy change is constrained by a ceiling of $1+\epsilon$ (or a floor of $1-\epsilon$), enforced by the minimization operation in Eq. (1) and its corresponding argument. This cautious approach inherited from the parent algorithm [35] ensures that the current and updated policies exhibit similar behavior, which prevents the agent from making abrupt policy changes that could lead to significant performance deterioration. A nuanced distinction between PPO and its single-step variant that is worth mentioning is that PPO operates as an actor-critic framework, where an actor network learns the policy and a critic network estimates advantages. In contrast, single-step PPO does not rely on critic evaluations (thus not following the actor-critic paradigm), as it involves only a single state-action pair trajectory. Setting the discount factor $\gamma = 1$ adjusts the balance between immediate and future rewards, simplifying the advantage to the whitened reward, as further explained in Ref. [25].

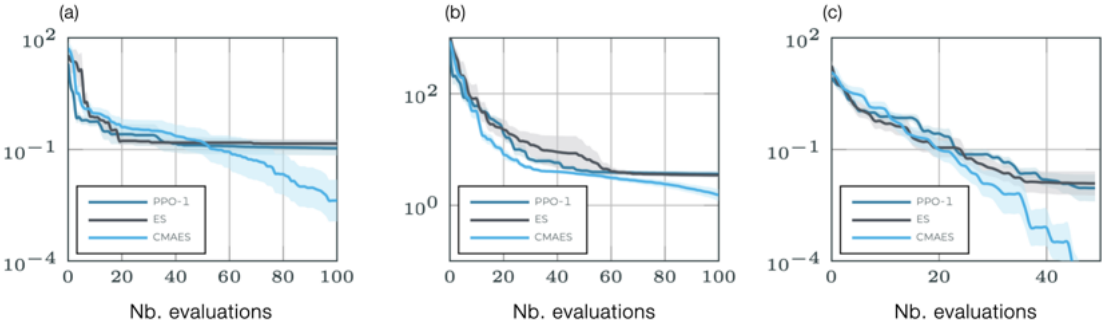

Figure 3: **Benchmark minimization problems** for the (a) two- and (b) five-dimensional Rosenbrock functions, and (c) the two-dimensional Branin function, using the present single-step PPO algorithm and reference (μ-λ)-ES and CMA-ES evolutionary algorithms.

For context, the convergence properties are illustrated in figure 3 for minimization test cases of two- and five-dimensional Rosenbrock functions, whose global minimum is notoriously difficult to catch, and two-dimensional Branin function, that has two identical global minima. Single-step PPO is benchmarked against classical $(\mu - \lambda)$-ES and CMA-ES evolutionary methods, all implemented in in-house production codes. The initial parameters and starting points are identical for all methods to ensure a fair comparison. All runs are afforded the same budget, namely 500 evaluations for Rosenbrock (20 episodes with 5 parallel environments per episode in PPO vs. 20 generations with 5 individuals per generation in ES algorithms) and 50 evaluations for Branin (10 episodes with 5 parallel environments per episode in PPO vs. 10 generations with 5 individuals per generation in ES). A large initial standard deviation is used by default, to ensure a good exploration of the optimization domain. Finally, in order to emphasize flexibility and generalizability, all PPO runs are tackled without fine-tuning of the algorithm, so all runs use the same meta-parameters, namely two steps mini-batches to update the network for 32 epochs, with learning rate set to

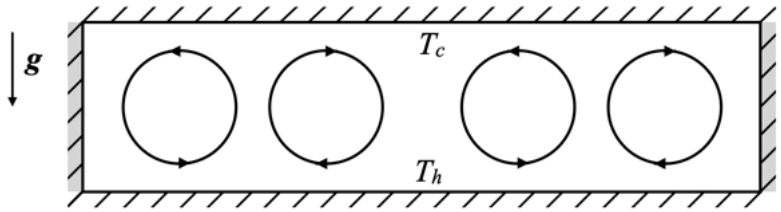

Figure 4: Schematic of the 2-D differentially heated cavity set-up. The gray shade indicates the insulated walls.

0.005 and PPO loss clipping range to $\epsilon = 0.2$. Performances are averaged over 10 runs, with standard deviations shown as the light shade around. Unsurprisingly, CMA-ES performs best, which reflects the improvement of efficiently elongating the research area to suit the shape of the cost function. Among isotropic exploration methods, single-step PPO achieves final cost levels similar to $(\mu - \lambda)$-ES, with faster convergence and better performance at intermediate stages (the final performance level ultimately saturates for the Rosenbrock function because the minimum is in a long, narrow valley, and PPO and $(\mu - \lambda)$-ES use isotropically sampled approximations of the descent direction. The general conclusion is that 1. single-step PPO exhibits strong performance compared to methods relying on similar isotropic search distributions, and 2. it is imperative to utilise anisotropic search distributions to outperform more advanced methods on a consistent basis, an issue that is being addressed in current research efforts by the authors [36].

4. Control of Rayleigh–Bénard convection

4.1. Case description

In order to assess the method's capability compared to data available in the literature, this section tackles the control of Rayleigh–Bénard (natural) convection in the two-dimensional differentially heated cavity sketched in figure 4(a). This is a widely studied benchmark system for thermally-driven flows, relevant in nature and technical applications (*e.g.*, ocean and atmospheric convection, materials processing, metallurgy), that is thus suitable to assess relevance of the numerical framework. The canonical initial condition is a fluid at rest that is being heated from the lower wall and/or cooled from the upper wall, with natural convection ensuing as a result of the induced temperature gradients and fluid-buoyancy effects. A Cartesian coordinate system is used with origin at the lower-left edge, horizontal x-axis, and vertical y-axis. The vertical sidewalls are perfectly insulated from the outside (adiabatic). The horizontal walls are isothermal: the upper wall is kept at a constant "cold" temperature T_c, and the lower wall is entirely controllable via a space-varying "hot" distribution $T_h(x)$ such that $\langle T_h \rangle > T_c$, where the brackets denote the average over space the x-position along the hot wall. Several studies have reported the benefits of similarly using DRL for natural convection heat transfer performance in laterally and bottom-heated square cavities [15, 16]. Here, a laterally extended domain with aspect ratio 4:1 is considered, for which the rationale is twofold: first, the convection cells are closer to the unconstrained cells obtained in wide domains relevant for industrial use. Second, control is more demanding, as it makes more difficult for the DRL agent to use the walls to move around and break the cells.

In this case of no phase change, the governing equations are the classical Navier–Stokes and heat equations written under the Boussinesq approximation as

$$\nabla \cdot \boldsymbol{u} = 0\,, \tag{15}$$

$$\rho(\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u}) - \nabla \cdot (2\mu\boldsymbol{\varepsilon}(\boldsymbol{u})) + \nabla p = -\rho\beta(T - T_c)\boldsymbol{g}\,, \tag{16}$$

$$\rho c_p(\partial_t T + \boldsymbol{u} \cdot \nabla T) = \nabla \cdot (\lambda \nabla T)\,, \tag{17}$$

where β is the thermal expansion coefficient, and we use T_c as Boussinesq reference temperature. The above equations are solved assuming no-slip on the walls and temperature boundary conditions

$$\partial_x T(0, y, t) = \partial_x T(4H, y, t) = 0\,, \qquad T(x, 0, t) = \langle T_h \rangle + \tilde{T}_h(x)\,, \qquad T(x, H, t) = T_c\,, \tag{18}$$

where $\tilde{T}_h$ is a zero-mean (in the sense of the space-average) hot temperature fluctuation, whose magnitude is bounded according to

$$|\tilde{T}_h(x)| \leq \Delta T_{max} \,, \tag{19}$$

to avoid extreme and nonphysical temperature gradients. This system is controlled by the Rayleigh and Prandtl numbers, set here to $\mathrm{Ra} = 10^4$ and $\mathrm{Pr} = 0.71$ (air value at room temperature) using the cavity height, the heat conductivity time, and the (time and space-constant) difference between the mean horizontal temperatures as reference scales.

4.2. Control and reward

Following [15, 16], we seek to optimize the distribution of hot temperature fluctuations $\tilde{T}_h$ by training a DRL agent in selecting piece-wise constant temperatures over $4n_s$ identical segments (labeled from left to right), each of which allows only two pre-determined states referred to as hot or cold. This is intended to reduce the complexity and the computational resources, as large/continuous action spaces are known to be challenging for the convergence of RL methods [37]. All results reported herein are for $\Delta T_{max} = 0.75$, for the hot temperature to vary in the range from 0.25 to 1.75.

Based on the topology of the baseline, uncontrolled solution (more details in the following), symmetric actuation with respect to the vertical centerline is used, in which the network outputs n_s discrete values $\hat{T}_{h,k\in\{1\dots n_s\}} = \pm\Delta T_{max}$, mapped into actual physical fluctuations over the first n_s segments using

$$\tilde{T}_{h,k} = \frac{\hat{T}_{h,k} - \langle \hat{T}_{h,k} \rangle}{\max_{l\in\{1\dots n_s\}} \left(1, \dfrac{|\hat{T}_{h,l} - \langle \hat{T}_{h,l} \rangle|}{\Delta T_{max}}\right)} \,, \tag{20}$$

to fulfill the zero-mean and upper bound constraints. This pattern is mirrored over the n_s following segments, then repeated over the next $2n_s$ segment, according to

$$\tilde{T}_{h,k} = \begin{cases} \tilde{T}_{h,2n_s+1-k} \,, & \text{if} \quad n_s < k \leq 2n_s \,, \\ \tilde{T}_{h,k-2n_s} \,, & \text{if} \quad 2n_s < k \leq 3n_s \,, \\ \tilde{T}_{h,4n_s-k+1} \,, & \text{if} \quad 3n_s < k \leq 4n_s \,. \end{cases} \tag{21}$$

We avoid nonphysical sharp discontinuities across segments using hyperbolic tangent functions to regularize the temperature fluctuation on each segment relatively to its immediate neighbors. This ensures continuously differentiable actuation from one segment to another, and is accounted for in practice to impose the zero-mean condition from (20)-(21). We set here $n_s = 10$, a spatial granularity of 10 segments per vortex of the uncontrolled solution, that has been found to allow suitable controllability.

The agent is incentivized to alleviate convective heat transport by receiving the reward $r_t = -\mathrm{Nu}$, where Nu is the Nusselt number defined as the non-dimensional temperature gradient averaged over the hot bottom wall (hence $r_t = 1$ if heat transfer is by pure conduction, and $r_t > 1$ otherwise). All values are computed from 400 points (100 per vortex of the uncontrolled solution) uniformly distributed along the hot bottom wall. A typical DRL simulation runs on 64 CPU cores, using 8 environments of 8 CPU each. The agent is a fully connected network with two hidden layers holding two neurons. The resolution process uses eight environments and two steps mini-batches to update the network for 32 epochs, with the learning rate set to 0.005 and PPO loss clipping range to $\epsilon = 0.2$. The agent then generates an improved batch of actions for the next generation, and the process repeats until convergence is achieved.

4.3. Results

Using the numerical methods described in the previous sections, the uncontrolled solution is computed starting from an initial condition of zero velocity and uniform temperature equal to T_c. Every five time steps, anisotropic mesh adaptation is performed under the constraint of a fixed number of elements $n_{el} = 80000$, using velocity and temperature as multiple-component adaptation

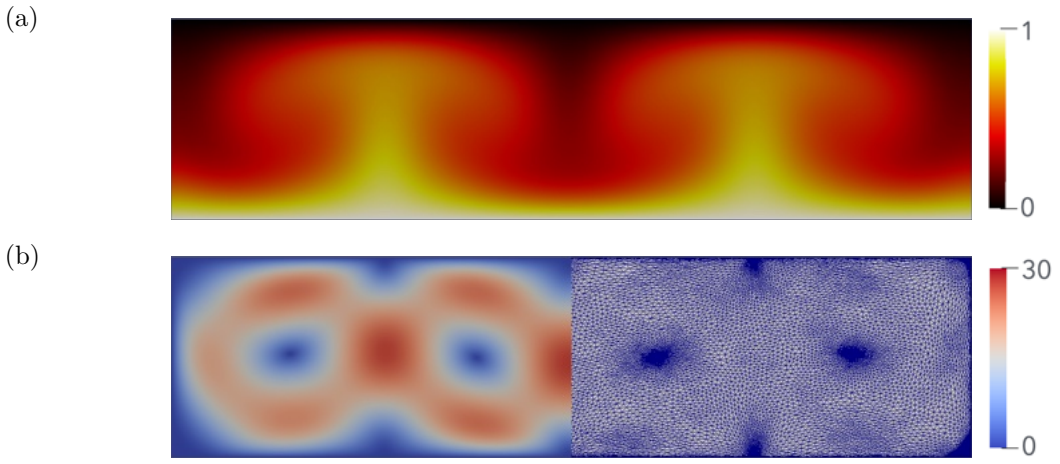

Figure 5: Uncontrolled twin-cell steady state solution. (a) Iso-contours of the temperature between 0 and 1. (b) Iso-contours of the velocity magnitude between 0 and 30, together with the corresponding anisotroic adapted mesh.

criterion (but no level-set, as the solid is solely at the boundary of the computational domain, where either the temperature, or the heat flux is known). As shown in figure 5(a,b), the cold downwelling and hot upwelling fluid organizes into a twin-cell configuration made up of two pairs of counter-rotating vortices, that ultimately becomes stationary. This occurs after approximately 15 time units, after which the Nusselt number converges to a value of $\mathrm{Nu} = 2.56$. The corresponding adapted mesh shown in the right half of figure 5(b) stresses that all boundary layers are sharply captured via extremely stretched elements, and that the adaptation strategy yields refined meshes near high temperature gradients and close to the side walls. Note however, the mesh refinement is not only along the boundary layers but also close to the recirculation regions near the cavity center, while the elements in-between are coarser and essentially isotropic.

Control runs comprise of 100 learning episodes, each of which marches in time the above baseline initial state for a duration of 300 time units using time step $\Delta t = 1$. This represents 800 simulations per run, each of which is performed on 8 cores and lasts 40mn, hence 530h of total CPU cost per run (about 65h in wall time). It is out of the scope of this work to analyze in details the flow patterns that develop when control is applied at the bottom of the cavity. Suffice it to say that the outcome consistently exhibits twin-cell patterns of varying size and magnitude, accompanied by corner eddies at the bottom of the cavity. This is best seen in figure 6 through several iso-contours of the steady-state temperature, each of which corresponds to a different learning episode performed over the course of the DRL optimization, and thus, to a different temperature distribution at the hot bottom wall. For all cases, steady state is achieved within a few ten time units, but the counter-rotating vortices must occasionally exchange place to suit the specifics of the control, which may take up to a few hundred time units. Mesh adaptation is an asset in this regards, as it allows to capture the anisotropy of the transient and asymptotic dynamics, intensified by the sharp (albeit continuous) boundary conditions; see in figure 7 the detailed time-evolution of the temperature field for the one control episode that yields the steady-state in figure 6(d), together with the corresponding adapted meshes. We show in figure 8 the evolution of the reward (Nusselt number), for which performances have bean averaged over 5 independent runs, as is customary in machine learning evaluation. The run-averaged mean Nusselt number during the optimization process is shown as gray line, with standard deviations shown as the light shade around. Finally, the black line shows the moving average Nusselt number, computed from the run-averaged mean as the sliding average over the 50 latest reward values (or the whole sample if it has insufficient size). The latter decreases monotonically and reaches a plateau after about 30 episodes, although we notice that sub-optimal distributions keep being explored occasionally.

The optimal computed by averaging over the 10 latest episodes (hence the 800 latest instant values) is 2.03, which corresponds to an efficiency of about 20% compared to the uncontrolled case. As evidenced by the best temperature distribution over the 5 optimization runs in figure 9, this requires providing a much wider plume by heating on either side of the vortex cores, although convection ultimately remains, consistently with the results in Ref. [16] at the same Rayleigh

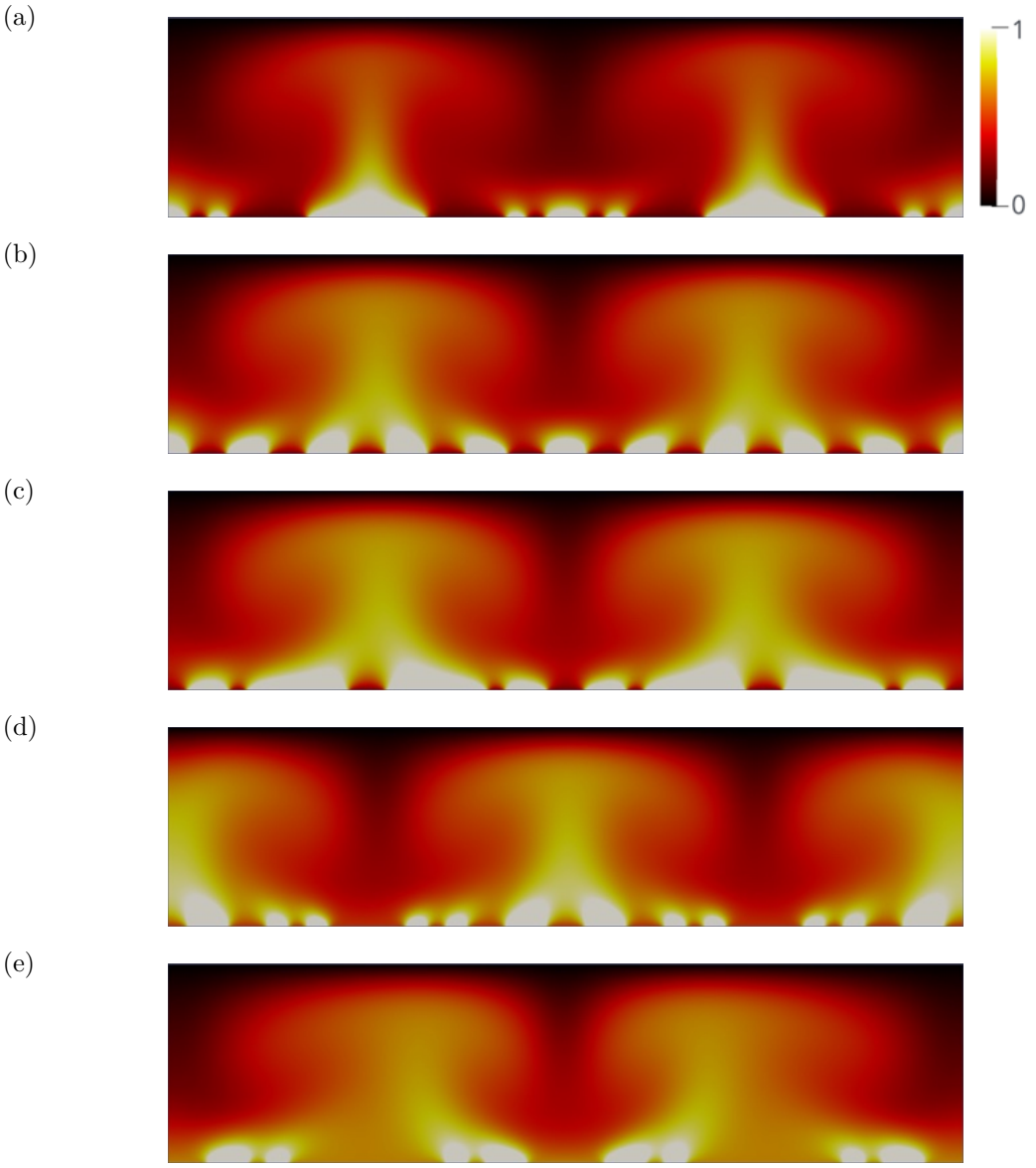

Figure 6: Iso-contours of the steady-state controlled temperature, computed between 0 and 1 under several zero-mean temperature distributed at the hot bottom wall. Each sub-plot illustrates a different learning episode performed over the course of the DRL optimization.

number. The authors in Ref. [15] conversely report complete suppression using a classical multi-step DRL algorithm adjusting dynamically the temperature from appropriate sensing of flow changes, but this only reflects the sub-optimality of operating under an open-loop strategy. For the sake of completeness, we note that a mitigation in similar proportions (with efficiency of about 23%) is reported in Ref. [17] using multi-step DRL, but this is yet another setup in which symmetry is assumed at the lateral ends of the cavity, which makes it difficult to further compare.

5. Control of quenching cooling rate in a 2-D open tank

5.1. General case description

We aim now at controlling quenching in the two-dimensional, rectangular tank described in figure 10, that has width 0.6m and height 0.4m. Four workpiece geometries of increasing complexity are considered, as seen in figure 11, that we refer to as rectangular brick, U-bend, serpentine and teeth, and whose width d (that drives the immersion depth to allow adjusting the insertion angle

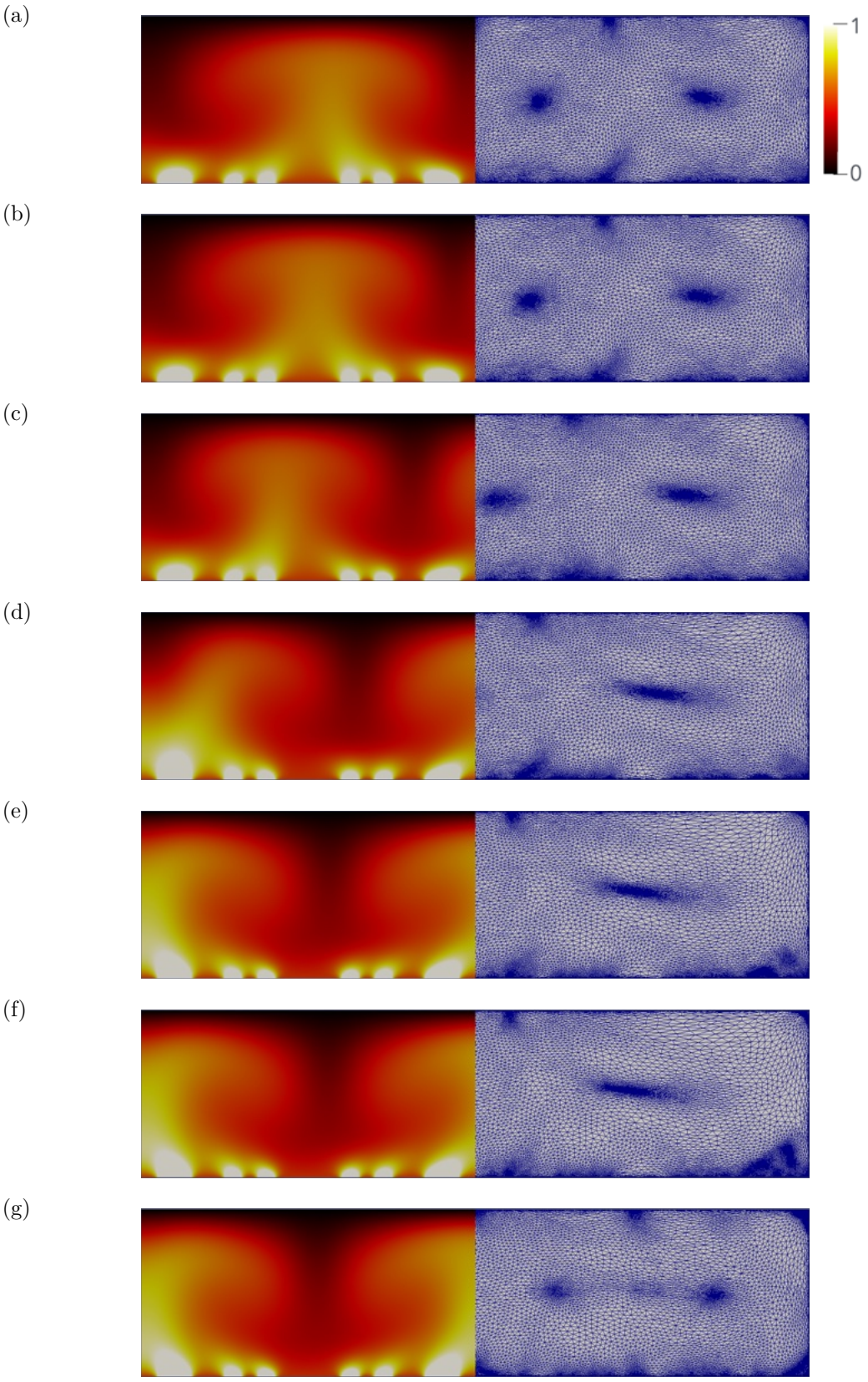

Figure 7: Instantaneous distribution of the controlled temperature, computed between 0 and 1 for the learning episode leading to the steady state in figure 6(d). The snapshots are sampled (from top to bottom) every 20 time units from 0 to 140 time units, after which the steady state from figure 6(d) is recovered in (g).

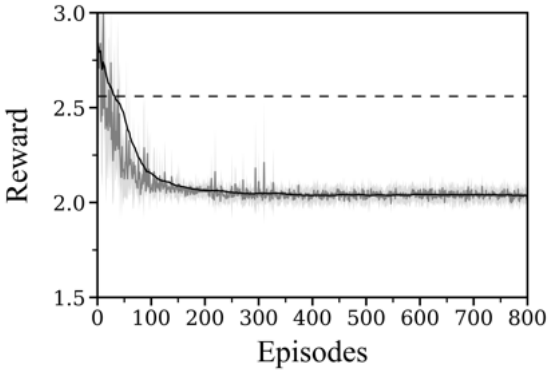

Figure 8: Evolution per learning episode of the instant (in grey) and moving average (in black) Nusselt number, computed by averaging over 20 independent runs. The horizontal dashed line marks the baseline uncontrolled value.

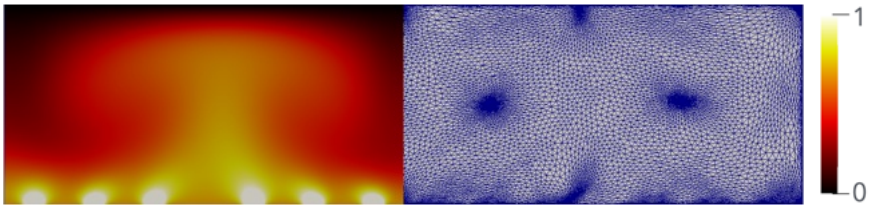

Figure 9: Same as figure 7 for the optimal controlled solution.

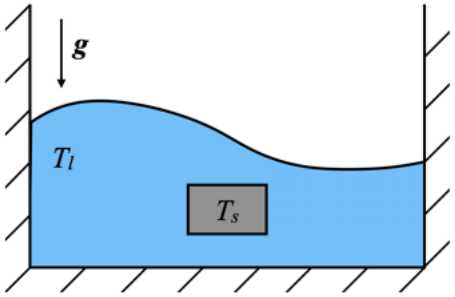

Figure 10: Sketch of the 2-D quenching numerical experiment

without contact with the bottom of the tank) is between 0.08m and 0.11m. A Cartesian coordinate system is used with origin at the center of the tank, horizontal x-axis, and vertical y-axis. The quenchant (here, water) is initially at $T_l = 25°\text{C}$, while the initial temperature of the metal alloy (Inconel) is $T_s = 880°\text{C}$. The saturation temperature of water at normal temperature and pressure conditions is taken to be $T_{sat} = 100°\text{C}$. Dirichlet boundary conditions for velocity and temperature boundary conditions are applied on the boundaries of the tank while the top is kept as a free surface. Each simulation runs for 60s of cooling time, with time steps Δt ranging between 0.005 and 0.1s and number of elements ranging between 20000 and 70000, depending upon the case stiffness. The mesh adaptation algorithm presented in section 2 continuously tracks the evolving vapor phase by remeshing every 5 time steps with minimum mesh size kept at 0.5mm to minimize errors at the interface.

5.2. Control, reward and sensor placement

The quantity being optimized is the orientation of the workpiece measured by its angle θ with the horizontal, with the understanding that θ is positive for a clockwise rotation up to 180°, and $\theta = 0°$ corresponds to the baseline orientation shown in figure 11. It is worth emphasizing that unlike the above Rayleigh-Bénard convection case, the action space here is continuous, although

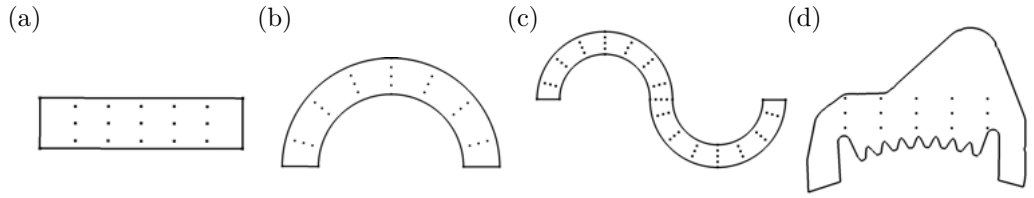

Figure 11: Workpiece geometries for the various test cases in section 5, together with associated sensor placement. (a) Rectangular brick, (b) U-bend, (c) serpentine, and (d) teeth geometry.

γ	T_l	T_s	μ	ρ	λ	c_p			d	n_p
-	-	880	0.001	8000	11.4	435	Solid	Rectangular brick	0.08	15
								U-bend	0.08	21
								Serpentine	0.11	45
								Teeth	0.08	15
0.07	25	-	0.005	1000	0.6	4185	Liquid			
			0.001	1.7	0.025	2010	Vapor			

Table 1: Numerical parameters used in the 2-D quenching numerical experiment. All values in SI units, with the exception of temperatures given in Celsius.

all angles reported in the following are rounded to the nearest integer to ease the reading. The network action output therefore consists of a single value $\hat{x}$ in $[-1; 1]$, mapped into the actual angle according to

$$\theta = \theta_{max}\frac{\hat{x}+1}{2} - \theta_{min}\frac{\hat{x}-1}{2}\,, \tag{22}$$

where we set $\theta_{max} = -\theta_{min} = 180°$. An ideal control would target a very small temperature difference (ideally zero) between any two points of the workpiece at any given time during the entire quenching duration. Since the workpiece in quenching is initially heated to a uniform temperature, one way of mathematically achieving this objective is by forming a reward function that penalizes the agent proportional to the maximum heat flux flowing out of the workpiece at every instant. This forces the DRL agent to homogenize the heat flux out of the workpiece, which achieves indirectly homogeneous temperature distribution. Local temperature and temperature gradient information (as heat flux is directly proportional to temperature gradient) is thus recorded during the simulation at N_p specific sensor locations in the workpiece, after which we compute the reward following the gradient strategy presented in [16], meant to approximate the averaged magnitude of the tangential heat flux from (except that all averages are performed here in space and time to encompass the whole history of phase change occurring of the quenching process). This information can help the agent update its policy so that future actions to minimize the largest gradient in a specific direction.

The probes information and placement for each geometry are documented in Table 1 and figure 11, respectively. For the rectangular brick and the teeth geometries, the probes are arranged in an array of n_x columns and n_y rows with resolutions Δ_x and Δ_y. The following formula gives an estimate of the tangential heat flux by averaging the norm of the temperature gradient in time and across rows and columns in x and y directions, respectively:

$$\langle||\nabla_{\parallel}T||\rangle_i = \frac{2}{n_y-1}|\sum_{j\neq 0}\mathrm{sgn}(j)||\nabla T||_{ij}|\,, \tag{23}$$

$$\langle||\nabla_{\parallel}T||\rangle_j = \frac{2}{n_x-1}|\sum_{i\neq 0}\mathrm{sgn}(i)||\nabla T||_{ij}|\,, \tag{24}$$

where subscripts i, j and ij denote quantities evaluated at $x = i\Delta x$, $y = j\Delta y$ and $(x, y) = (i\Delta x, j\Delta y)$, respectively, and symmetrical numbering is used for the center probe to sit at the

intersection of the zero-th column and row. The reward $r_t = -\langle||\nabla_{\parallel} T||\rangle$ fed to the DRL agent is given by the average of the quantities calculated before as,

$$\langle||\nabla_{\parallel} T||\rangle = \frac{d}{T_{ref}} \frac{1}{n_x + n_y} \sum_{i,j} \langle||\nabla_{\parallel} T||\rangle_i + \langle||\nabla_{\parallel} T||\rangle_j \,, \tag{25}$$

which specially yields $r_t = 0$ for a perfectly homogeneous cooling. For the U-bend and serpentine geometries, a conformal mapping from rectangular to circular sector is used to distribute a similar array of probes on the actual workpiece geometry, in which case the above relations carry over provided the i and j indices are taken to refer to the *unmapped* sensor positions.

5.3. Rectangular brick test case

We start with the most straightforward and popular quenching geometry possible: the rectangular brick. The evolution of the rewards and actions for 320 episodes (40 episodes) is presented in figure 12(a) and (b), respectively. Initially, the agent randomly explores the action space to learn from the rewards. It is rather intuitive that for $\theta = 0°$ configuration shown in figure 13(a), the uneven vapor accumulation will cause differential cooling at the top and bottom surface of the workpiece, in turn leading to differential properties and thermal distortions. Moreover, the associated vapor film evolution pattern shows that a thick vapor film forms on the upper surface compared to the lower surface, that creates thermal insulation. For these reasons, it is standard practice in the industry to avoid this configuration. Nonetheless, it is interesting that the DRL agent learns this aspect without prior knowledge of boiling physics in the first few episodes. As the insertion angle increases to $\theta = 17°$ (rounded to the closest integer), the vapor film insulation accumulates on the top left corner, causing high temperature in this corner up to t=20-30s; see figure 13(b). It is worth mentioning that the workpiece temperature is close to the recrystallization temperature in some zones; hence such skewed temperature distribution will cause worst grain growth (and material properties) than in the previous case. The reward keeps decreasing until the brick is again set to achieve similar states due to the geometrical symmetry, after which the DRL agent settles in a range from 50° to 150° (which is rather fortuitous since the agent does not learn about symmetries under the optimization process [16]), and it takes a dozen episodes for the variance to start diminishing. At this stage, the insertion angle $\theta = 68°$ shown in figure 13(c) yields an overall better temperature evolution throughout the quenching process (compared to the early actions taken by the agent). It is intriguing to see that for this configuration, the vapor film does not initially accumulate on either of the large edges, and the bubble nucleation occurs from both surfaces, producing a homogeneous temperature distribution. However, after t=30s, the bubble nucleation stops on the left side, creating a thin vapor film, after which the temperature distribution is skewed for the rest of the process. It is out of the scope of this work to analyze in details this subtle bubble dynamics, but these observations suffice to highlight the stiffness of boiling physics applied to quenching at different insertion angles, as it is at least challenging, if not impossible, for an experienced professional to anticipate the behaviors discussed herein above just by inspection, even in a simplified 2-D case. The optimal insertion angle for this case is found to be $\theta = 97° \pm 2°$, with the associated reward $r_t = 235 \pm 17$. This is slightly tilted from $\theta = 90°$, which forces the upper surface to convect more vapor mass due to the latter having lower specific density. It is also shown in figure (13(d) to avoid vapor entrapment along the long and short edges while attaining maximum effective length to improve natural convection. It is imperative to note that, in this case, the bubble nucleation occurs throughout the process with equal intensity from both surfaces. This can be one of the indications (along with the uniform temperature profiles) that the current state is arguably the optimum state achieved by the DRL.

5.4. U-bend test case

The U-bend is another common geometries treated in the industry. Similar to the previous case, the test case is set up with a total of 320 simulations (40 episodes), whose rewards and actions evolution is shown in figure 16. One of the most popular configurations for this geometry is the in the quenching industry is the inverted-U shape[38] illustrated in figure 15(a) that corresponds to $\theta = 180°$. In this case, the nucleation of the bubbles mostly occurs at the top of the curved part while the two thongs are covered in a thick vapor film. This leads to a differential temperature

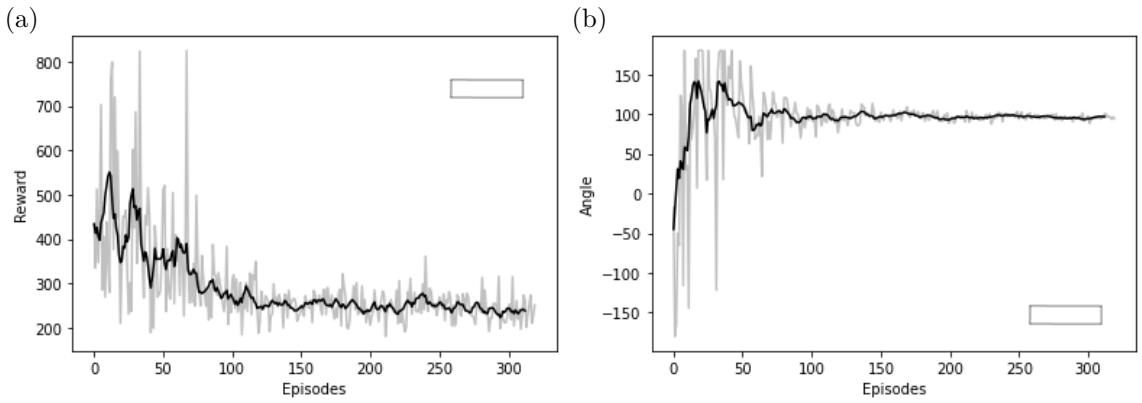

Figure 12: (a) Evolution per learning episode of the instant (in grey) and moving average (in black) reward for the rectangular brick test case. (b) Same as (a) for the workpiece insertion angle.

Figure 13: Temperature evolution for the rectangular brick test case inserted at various angles. (a) $\theta = 0°$, (b-d) random angles selected over the course of optimization, and (e) $\theta = 97°$, the optimal angle selected by the DRL agent. The snapshots are sampled (from left to right) every 10s from 0 to 50s. The iso-contours in the solid (resp. in the fluid) show the temperature field between 0 and 880°C (resp. the density field between 1.7 and 8000 kg/m^3).

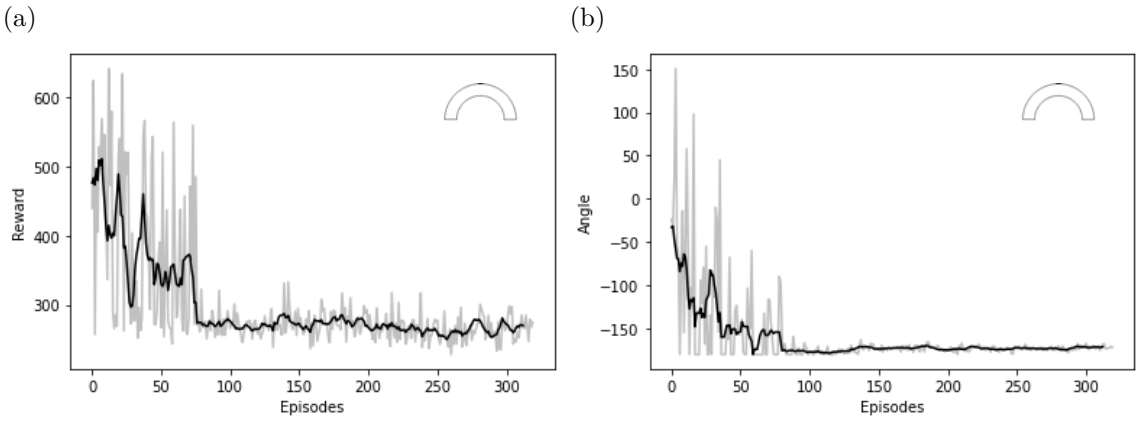

Figure 14: (a) Evolution per learning episode of the instant (in grey) and moving average (in black) reward for the U-bend test case. (b) Same as (a) for the workpiece insertion angle.

distribution within the solid as the mid portion of the workpiece cools faster than two thongs creating three different temperature zones. Another obvious strategy is the U configuration in figure 15(b), that corresponds to $\theta = 0°$. It yields a more uniform temperature distribution than the inverted U configuration, but one that i remains skewed. Meanwhile, it takes DRL agent approximately 50 episodes to find an optimal insertion angle $\theta = -171° \pm 2°$ associated to reward 269 ± 8. The latter corresponds to a beneficial tilt of the U configuration that slightly reduces the skewness in the temperature profile, and achieves a more homogeneous distribution at all profile at all intermediate time steps, as shown in figure 15(d). Again, we only aim here at assessing the ability to output unanticipated quenching solutions using DRL, so explaining the complex underlying physics remains out of scope at this stage, but it should be noted again that such dynamics are complicated to spot with mere observation, even to the shrewd eye.

5.5. Serpentine

The serpentine (or double U-bend) is a more complex geometry for which inferring an optimal insertion angle is a really challenging task. We train here the DRL agent with a total of 640 simulations (80 episodes), with rewards and actions evolution reported in figure 16. For this case, it takes about 25 episodes to converge to an optimal $77° \pm 2°$, which yields a reward 863 ± 20. In this setting, the temperature profile in the two U-bends is almost the same (symmetric) over the whole quenching process, as shown in figure 17(d). This is highly non-trivial, given the difficulty to achieve homogeneous cooling in a single U-bend, discussed in earlier sections. By comparison, at $\theta = 0°$ (figure 17(a)), one half of the geometry loses heat faster than the other one (which can yield incrased residual stresses), while at $\theta = 90°$ (figure 17(b)), the upper and middle parts of the geometry remain hotter compared to the other sections. The other angle reported in figure 17(c) yield differential cooling in the upper and middle section as the vapor gets entrapped in this region, insulating it further.

5.6. Teeth geometry

The teeth geometry is a highly complicated geometry inspired by actual industrial components, one that has no axis of symmetry, which makes inferring an optimal insertion angle completely impossible. The agent is trained with a total of 320 simulations (40 episodes); see figure 18 for the associated rewards and actions. While we do not strictly speaking achieve convergence for this case, as evidenced by the substantial variations in the insertion angles achieved over the last part of training, the agent succeeds in identifying a relevant range of parameters in the order from 100 to 180°. If we compare to the results obtained at empiric angles ($\theta = 0°, 45°, 90°$), it is seen in figure 19 that DRL produces a rather homogeneous temperature profile. As evident from previous cases, the workpiece face where vapor film accumulates and bubble creation occurs is always hotter compared to other regions. At $\theta = 90°$, this vapor accumulation causes skewed temperature profile in both horizontal (due to vapor accumulation) and vertical (due to thickness) directions; see figure 19(a). In other words, the geometry region that is thinner and immersed deeper, always cools first. This

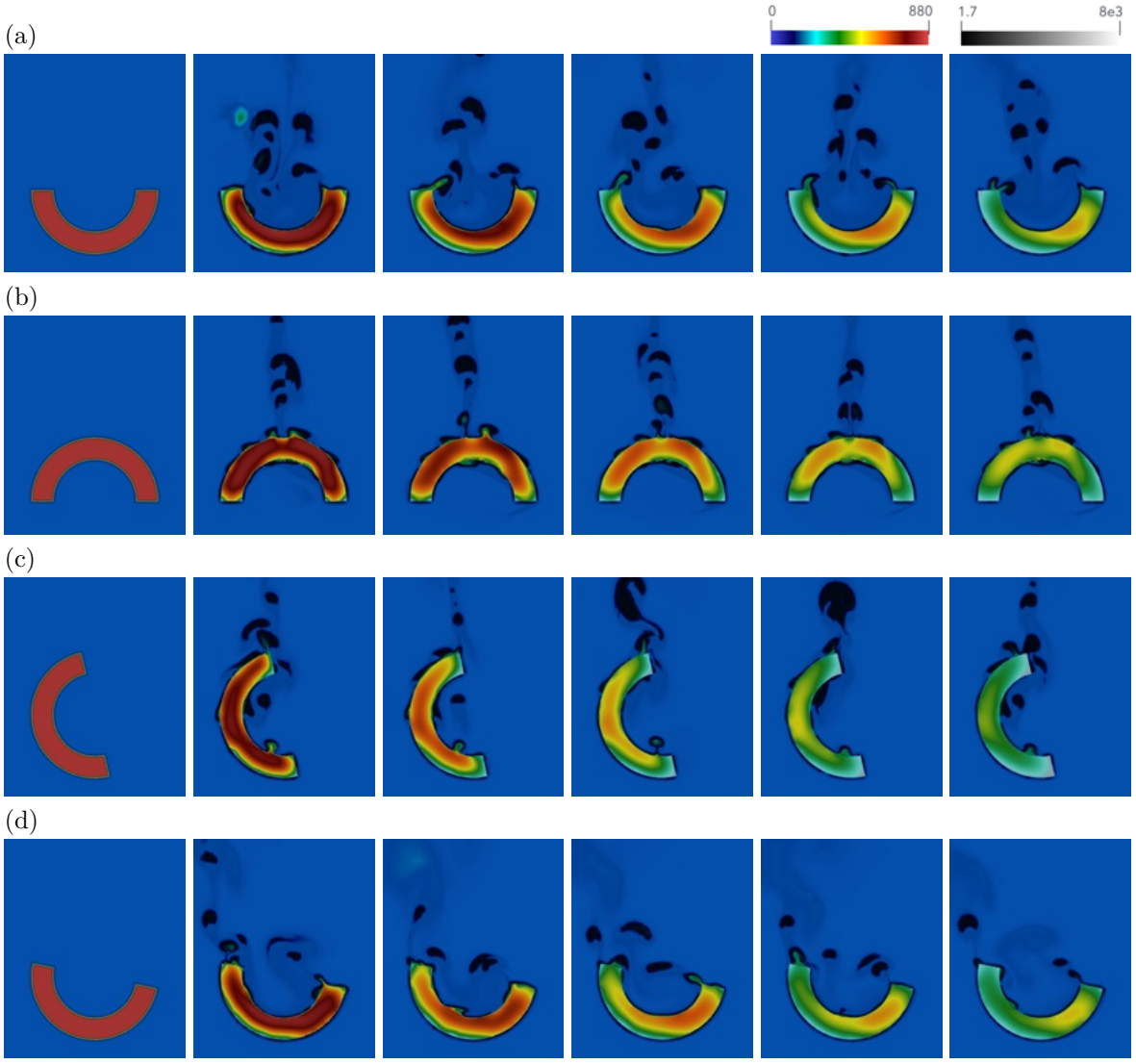

Figure 15: Temperature evolution for the U-bend test case inserted at various angles. (a) $\theta = 0°$, (b) $\theta = 180°$, (c) $\theta = -85°$, and (d) $\theta = -171°$, the optimal angle selected by the DRL agent. The snapshots are sampled (from left to right) every 10s from 0 to 50s. The iso-contours in the solid (resp. in the fluid) show the temperature field between 0 and 880°C (resp. the density field between 1.7 and 8000 kg/m^3).

carryies over to the other empiric setting considered, namely $\theta = 0°$ in figure 19(b) and $\theta = 45°$ in figure 19(c), where the thick region stays hotter compared to the teethes and thongs. Despite the lack of convergence (discussed in the next section), the DRL agent seems to learn these nuances by proposing a solution to insert the workpiece upside down, which increases the heat flux from the thick region (due to natural convection), while the heat flux from the thin teethed decreases due to added insulation coming from the vapor films.

6. Conclusion and recommendations for future research

6.1. Conclusion

In this work, a numerical framework is presented, in which a fully connected network learns to find optimal parameters in the process of quenching heat treatment. The agent is trained with the single-step PPO deep reinforcement algorithm, and gets only one attempt per learning episode at finding the optimal. The numerical reward fed to the network is computed with a

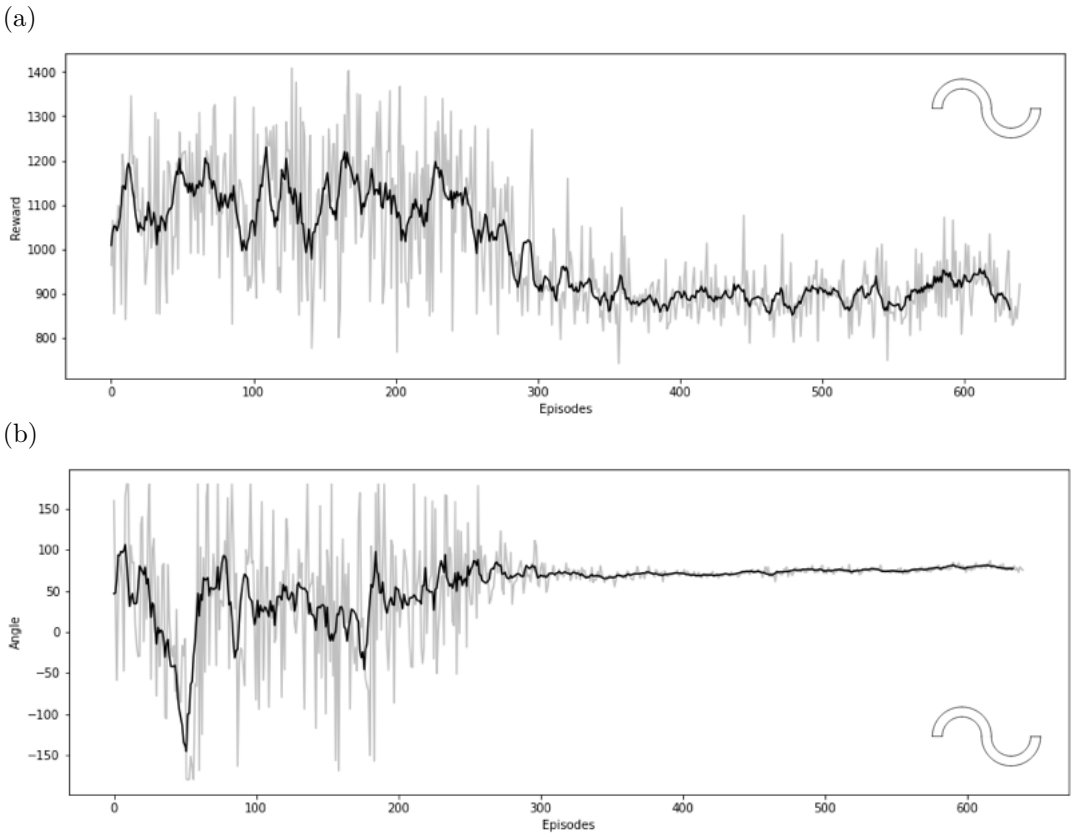

Figure 16: (a) Evolution per learning episode of the instant (in grey) and moving average (in black) reward for the serpentine test case. (b) Same as (a) for the workpiece insertion angle.

stabilized finite elements CFD environment solving a phase change model formulated after pseudo-compressible Navier–Stokes and heat equations, using a combination of variational multi-scale modeling, immerse volume method, and multi-component anisotropic mesh adaptation.

Relevance of the proposed methodology is illustrated by controlling natural convection in a closed cavity with aspect ratio 4:1, for which DRL alleviates the flow-induced enhancement of heat transfer by approximately 20%. Regarding quenching applications, the DRL algorithm succeeds in finding optimal orientations that adequately homogenize the temperature distribution within both simple and complex 2-D part geometries, and improve over simpler trial-and-error configurations classically used in the quenching industry. Such results clearly stress that single-step PPO (and DRL in general) can be effective to explore and discover new solutions from unforeseen parameter combinations in quenching applications.

6.2. Strategies towards practical application

As an exploratory study, the current research provides preliminary evidence of the ability of the proposed DRL-CFD framework in optimizing complex quenching processes by improving cooling uniformity to mitigate thermal residual stresses. The presented results are encouraging, but more work is needed to to confirm and extend our conclusions, and to fully scope out the potential of the approach in real-world scenarios. In concluding the present paper, it is thus proposed to discuss key directions for improvement, all intended to help bridge the gap between the current capabilities and the requirements of practical deployment.

Quenching is a complex thermomecanical process that can be cast as a thermal fluid-structure interaction problem involving the simultaneous resolution of turbulent flows with phase change and conjugate heat transfers between the solid and the fluid subdomains. Overall, the field is ever evolving, and there is a clear need for improved CFD models capable of dealing with this problem in all its complexity. A high-fidelity adaptive, multiphase DRL-CFD framework predicting accurately the phase change at the liquid-vapor interface, but also the phase transformation of the treated

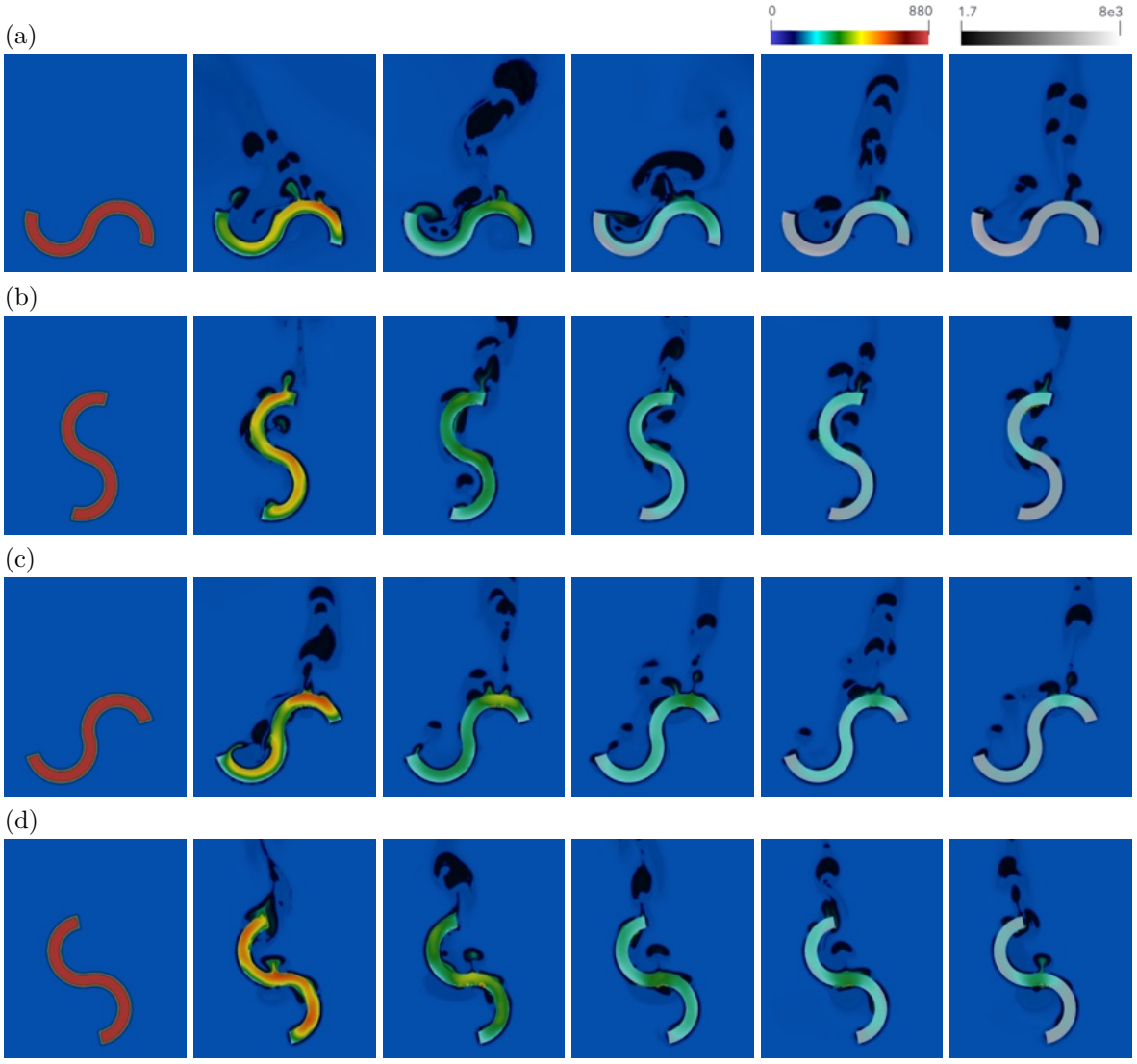

Figure 17: Temperature evolution for the serpentine test case inserted at various angles. (a) $\theta = 0°$, (b) $\theta = 90°$, (c) random angle selected over the course of optimization, and (d) $\theta = 77°$, the optimal angle selected by the DRL agent. The snapshots are sampled (from left to right) every 10s from 0 to 50s. The iso-contours in the solid (resp. in the fluid) show the temperature field between 0 and 880°C (resp. the density field between 1.7 and 8000 kg/m^3).

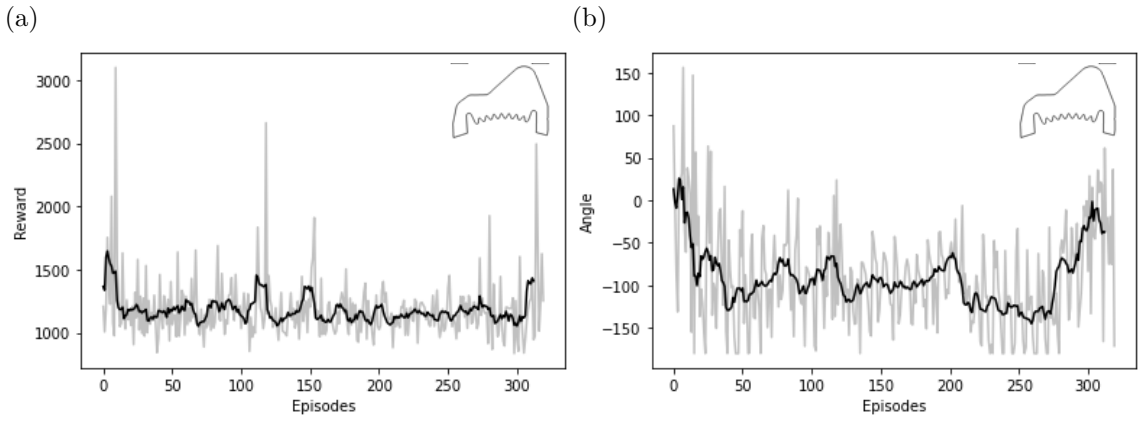

Figure 18: (a) Evolution per learning episode of the instant (in grey) and moving average (in black) reward for the teeth geometry test case. (b) Same as (a) for the workpiece insertion angle.

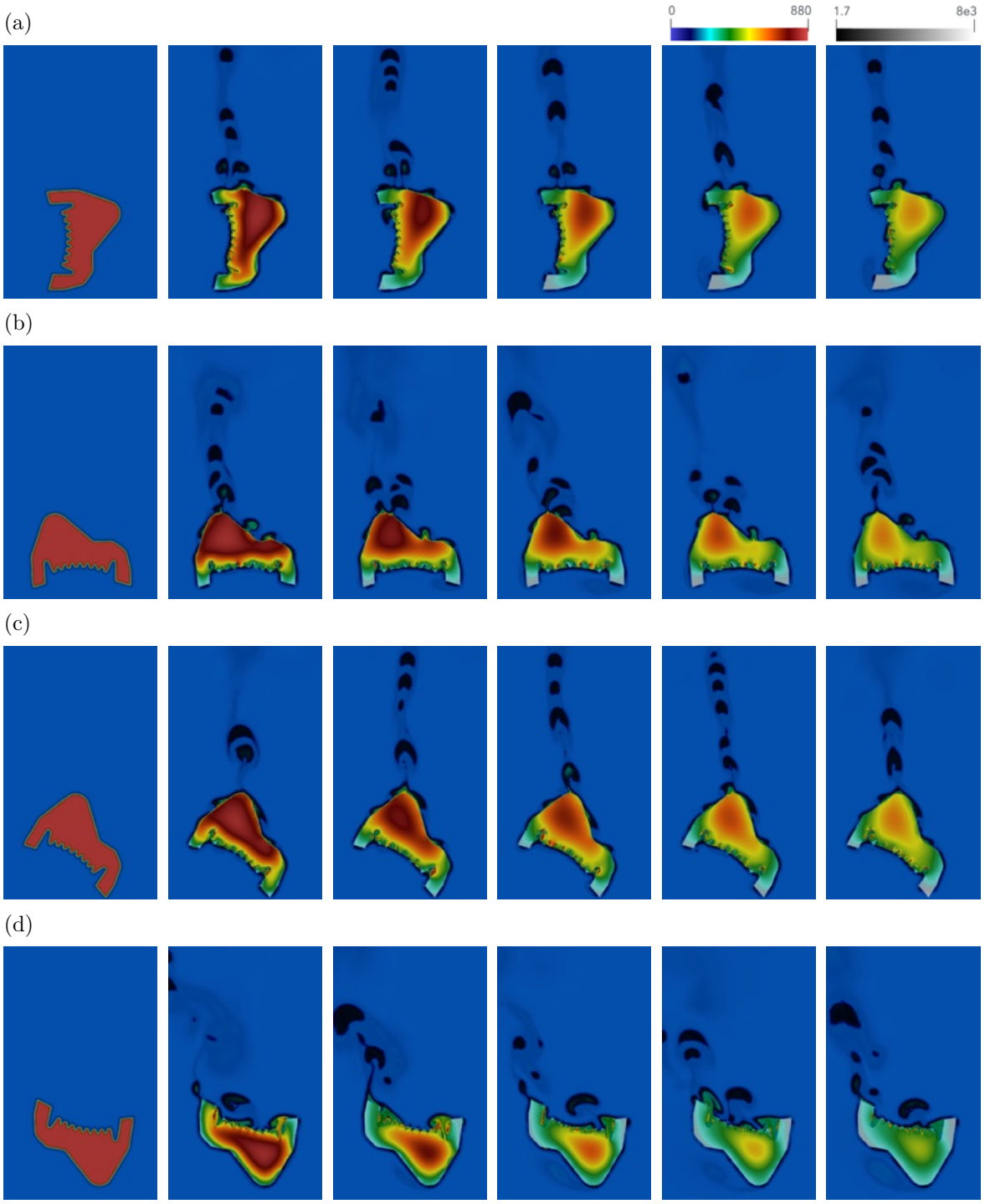

Figure 19: Temperature evolution for the teeth geometry test case inserted at various angles. (a) $\theta = 90°$, (b) $\theta = 0°$, (c) $\theta = 45°$, and (d) $\theta = 180°$, in the optimal range selected by the DRL agent. The snapshots are sampled (from left to right) every 10s from 0 to 50s. The iso-contours in the solid (resp. in the fluid) show the temperature field between 0 and 880°C (resp. the density field between 1.7 and 8000 kg/m^3).

part to predict its metallurgical evolution, will thus be instrumental in providing industrially relevant process parameters encompassing not only the thermal residual stresses caused by large temperature gradients (as has been done here), but also the volumetric residual stresses caused by martensitic transformations. By then, it is reasonable to expect that further developments in the fast-moving field of deep reinforcement learning will allow for faster convergence and lesser execu-

tion load (using, *e.g.*, auto-encoders and systematic state compression, or on-the-fly generation of surrogate models with uncertainty level prediction), and will facilitate application to industrially relevant 3-D configurations. This should set up a framework fast enough to inform process design in a matter of hours rather than days, thereby reliably augmenting industrial applicability. A core feature of the proposed framework in this respect is its high generalizability, that is, the fact that it builds naturally from any improved CFD model or/and DRL training method. For instance, a more elaborated algorithm can easily substitute for the rather simplistic PPO framework, such as Policy-based Optimization [36], another single-step reinforcement algorithm that samples actions from full covariance matrices, and is theoretically better suited to represent higher order logic and to handle complex parameter interactions.

An important limiting factor is the limited availability of process data, often affected to a small number of sensors, which can make it difficult to develop accurate models and control algorithms due to high-variance output. In a broad sense, the agent operates under partially observable environments, so its performance is highly dependent on the quality and relevance of the available data (this is an issue strongly related to data-driven model reduction techniques for large scale dynamical systems, which usually require using measures of observability as an information quality metric). In this regards, we note that the reward construction strategy employed in this work puts constraints on how many sensors can be used and how they can be arranged in the workpiece. This is detrimental to learning, and may explained the overall lack of convergence observed in the teeth geometry, as the agent does not learn the heat flux out of the workpiece in some regions of interest (here, the hood) but has to figure it out from the indirect information it gets from the other sensors. Another explanation is that the reward is approximated from point-wise temperature data (similar to experimental measurements) that has more sensitivity to small numerical errors (*e.g.*, the interpolation error at the probes position) than an integral quantity, and mesh adaptation procedure is not a deterministic process, as the outcome depends on the processors and number of processors used, and any initial difference propagates over the course of the simulation because the meshes keep being adapted dynamically. For these reasons, two control parameters, even close, can yield different rewards on behalf of different interpolation errors at the probes position and different nucleation patterns initiated by slightly different initial conditions, as illustrated in figures 20 for the rectangular brick, the U-bend and the serpentine solution. This also likely explains why the variance in reward is systematically larger (by a factor of almost 5) than that of the action itself for all cases reported herein. Ultimately, it calls for the design of robust reward functions capable of guiding the learning process toward effective and efficient policies even with randomly distributed sensors. This is no small task, in the absence of a best practice on how to design a reward function (this being essentially a trial-and-error process of a practitioner using their knowledge to define a baseline reward intended to provide a consistent feedback to the agent about its performance, observing how the agent performs, then tweaking the reward to achieve greater performance).

Finally, another reason to push DRL forward in this context is the ability of neural networks to transfer knowledge from previous experiences, to quickly adapt to different environments (ıwork-piece geometry and material properties, quenchant) and effectively learn new tasks. For instance, it is easy to compare different settings of design complexity, reflecting different levels of constrained operation when it comes to optimizing a practically meaningful scenarios (*e.g.*, heavily constrained optimization problems relevant to cases where the practitioner has limited freedom to optimize the design, in which case one can seek to optimize the orientation, immersion rate and depth of the solid part and the fluid viscosity, or mildly constrained problems relevant to cases where the practitioner has great freedom to act, in which case additional parameters can include the size of the tank, the number, type and placement of agitators and the agitation rate). We expect that this will be a key feature to reduce learning time and improve neural network performance, as progress are made towards realizing the potential of DRL-CFD for flexible, ready-to-use control of industrial manufacturing processes.

References

[1] Z. Zhao, M. Stuebner, J. Lua, N. Phan, J. Yan, Full-field temperature recovery during water quenching processes via physics-informed machine learning, Journal of Materials Processing Technology 303 (2022) 117534.

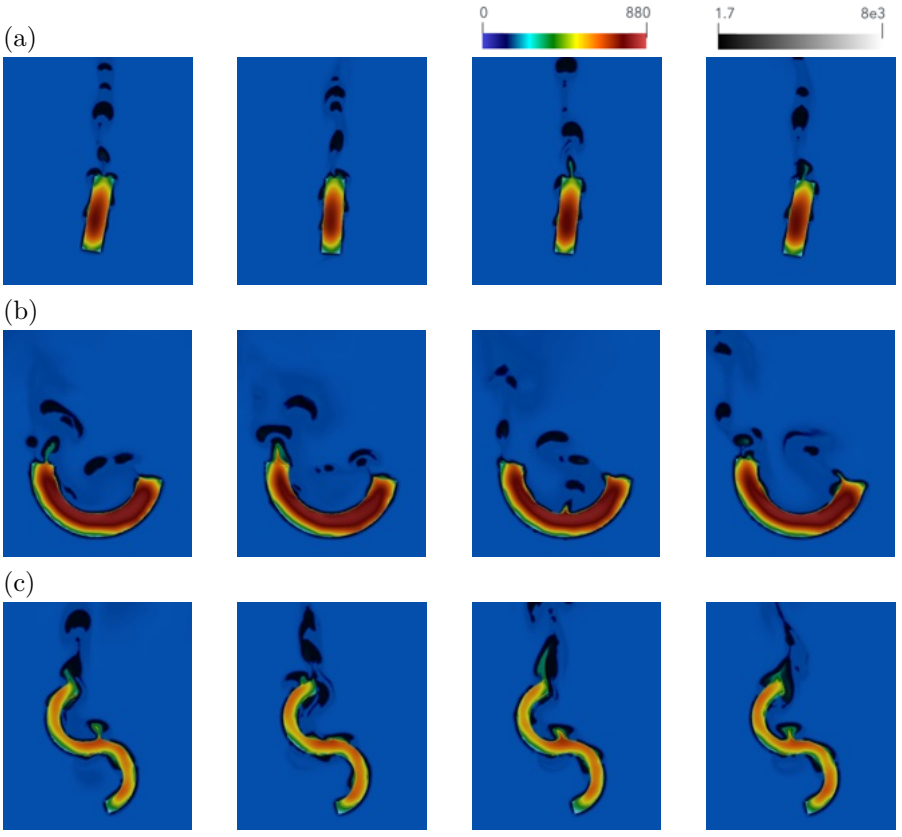

Figure 20: Temperature at $t = 20$s for the (a) rectangular brick, (b) U-bend, and (c) serpentine geometries inserted at several angles close to the DRL optimal.

[2] R. D. Lopez-Garcia, I. Medina-Juárez, A. Maldonado-Reyes, Effect of quenching parameters on distortion phenomena in AISI 4340 steel, Metals 12 (5) (2022).

[3] L. He, H. Li, Fem simulation of quenching residual stress for the plane strain problems, in: 2010 International Conference On Computer Design and Applications, Vol. 3, 2010, pp. V3–119–V3–123.

[4] M. Decroos, M. Seefeldt, The effect of size on the distortion behavior after carburisation and quenching processes of gears, Int. J. Met. Mater. Eng 139 (2017) 1–10.

[5] S. Šolić, B. Podgornik, V. Leskovšek, The occurrence of quenching cracks in high-carbon tool steel depending on the austenitizing temperature, Engineering failure analysis 92 (2018) 140–148.

[6] A. da Silva, T. Pedrosa, J. Gonzalez-Mendez, X. Jiang, P. Cetlin, T. Altan, Distortion in quenching an AISI 4140 C-ring - Predictions and experiments, Materials & Design 42 (2012) 55–61.

[7] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, P. Abbeel, Asymmetric actor critic for image-based robot learning, arXiv preprint arXiv:1710.06542 (2017).

[8] D. Bahdanau, P. Brakel, K. Xu, A. Goyal, R. Lowe, J. Pineau, A. Courville, Y. Bengio, An actor-critic algorithm for sequence prediction, arXiv preprint arXiv:1607.07086 (2016).

[9] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing Atari with Deep Reinforcement Learning, arXiv preprint arXiv:1312.5602 (2013).

[10] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche,

T. Graepel, D. Hassabis, Mastering the game of go without human knowledge, Nature 550 (2017) 354–359.

[11] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, A. Shah, Learning to drive in a day, arXiv preprint arXiv:1807.00412 (2018).

[12] W. Knight, Google just gave control over data center cooling to an AI, MIT Technology Review (2018).

[13] P. Garnier, J. Viquerat, J. Rabault, A. Larcher, A. Kuhnle, E. Hachem, A review on deep reinforcement learning for fluid mechanics, Comp. Fluids 225 (2021) 104973.

[14] J. Viquerat, P. Meliga, A. Larcher, E. Hachem, A review on deep reinforcement learning for fluid mechanics : an update, Phys. Fluids 34 (2022) 111301.

[15] G. Beintema, A. Corbetta, L. Biferale, F. Toschi, Controlling Rayleigh-Bénard convection via reinforcement learning, Journal of Turbulence (2020) 585–605.

[16] E. Hachem, H. Ghraieb, J. Viquerat, A. Larcher, P. Meliga, Deep reinforcement learning for the control of conjugate heat transfer, J. Comput. Phys. 436 (2021) 110317.

[17] C. Vignon, J. Rabault, J. Vasanth, F. Alcántara-Ávila, M. Mortensen, R. Vinuesa, Effective control of two-dimensional Rayleigh–Bénard convection: invariant multi-agent reinforcement learning is all you need, Phys. Fluids 36 (2023) 065146.

[18] M. Renault, J. Viquerat, P. Meliga, G.-A. Grandin, N. Meynet, E. Hachem, Investigating gas furnace control practices with reinforcement learning, Int. J. Heat Mass Transfer 209 (2023) 124147.

[19] Y.-Z. Wang, J.-Z. Peng, N. Aubry, Y.-B. Li, Z.-H. Chen, W.-T. Wu, Control policy transfer of deep reinforcement learning based intelligent forced heat convection control, Int. J. Therm. Sci. 195 (2024) 108618.

[20] H. Keramati, F. Hamdullahpur, M. Barzegari, Deep reinforcement learning for heat exchanger shape optimization, Int. J. Heat Mass Transfer 194 (2022) 123112.

[21] A. di Meglio, N. Massarotti, P. Nithiarasu, A physics-driven and machine learning-based digital twinning approach to transient thermal systems, Int. J. Numer. Methods Heat Fluid Flow ahead-of-print (2024) ahead–of–print.

[22] T. Zhang, J. Luo, P. Chen, J. Liu, Flow rate control in smart district heating systems using deep reinforcement learning, arXiv preprint arXiv:1912.05313 (2019).

[23] M. Kim, J. H. Moon, Deep neural network prediction for effective thermal conductivity and spreading thermal resistance for flat heat pipe, Int. J. Numer. Methods Heat Fluid Flow 33 (2022) 437–455.

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

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

[26] M. Khalloufi, Multiphase flows with phase change and boiling in quenching processes, Ph.D. thesis, PSL Resarch University (2017).

[27] M. Khalloufi, R. Valette, E. Hachem, Adaptive eulerian framework for boiling and evaporation, J. Comput. Phys. 401 (2020) 109030.

[28] E. Hachem, T. Kloczko, H. Digonnet, T. Coupez, Stabilized finite element solution to handle complex heat and fluid flows in industrial furnaces using the immersed volume method, Int. J. Numer. Meth. Eng. 68 (2012) 99–121.

[29] E. Hachem, H. Digonnet, E. Massoni, T. Coupez, Immersed volume method for solving natural convection, conduction and radiation of a hat-shaped disk inside a 3d enclosure, Int. J. Numer. Method H. 22 (2012) 718–741.

[30] C. Gruau, T. Coupez, 3D tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric, Comput. Methods Appl. Mech. Engrg. 194 (2005) 4951–4976.

[31] M. M. Bernitsas, K. Raghavan, Y. Ben-Simon, E. M. H. Garcia, Vivace (vortex induced vibration aquatic clean energy): a new concept in generation of clean and renewable energy from fluid flow, J. Offshore Mech. Arctic Engng. 130 (2008) 041101.

[32] Y. Mesri, H. Digonnet, T. Coupez, Advanced parallel computing in material forming with CIMLib, Eur. J. Comput. Mech. 18 (2009) 669–694.

[33] T. Coupez, Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing, J. Comput. Phys. 230 (2011) 2391–2405.

[34] J. U. Brackbill, D. B. Kothe, C. Zemach, A continuum method for modeling surface tension, Journal of computational physics 100 (2) (1992) 335–354.

[35] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal Policy Optimization Algorithms, arXiv preprint arXiv:1707.06347 (2017).

[36] J. Viquerat, R. Duvigneau, P. Meliga, A. Kuhnle, E. Hachem, Policy-based optimization: single-step policy gradient method seen as an evolution strategy, Neural Comput. Appl. 35 (2023) 449–467.

[37] K. Lee, S. A. Kim, J. Choi, Deep reinforcement learning in continuous action spaces: a case study in the game of simulated curling, in: Procs. of the 35th International Conference on Machine Learning, 2018, pp. 2937–2946.

[38] Y. Nakagawa, K.-I. Mori, S. Yashima, T. Kaido, Springback behaviour and quenchability in hot stamping of thick sheets, Procedia Manufacturing 15 (2018) 1071–1078.