Sandia National Laboratories

Exceptional service in the national interest

Reinforcement Learning for Adaptive Control of PDE-Constrained Environments

Nick Winovich[†], Bart van Bloemen Waanders[†]
Deepanshu Verma[‡], Lars Ruthotto[‡]

[†] Sandia National Laboratories
[‡] Emory University

Sandia National Laboratories

U.S. Department of Energy NNSA

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Secu DE-NA0003525.

SAND2023-04412C

Problem Formulation

$$\min_{s,a} J(s,a) \;=\; \int_0^T L(t,s,a)\,dt \;+\; \Psi(s(T)) \qquad \begin{pmatrix} L(t,s,a) \;=\; \frac{1}{2}\|a(t)\|^2 \\ \Psi(s) \;=\; \int_{\Omega_T} s^+\,dx \end{pmatrix} \tag{1}$$

$$\text{where } \; s(t) \; \text{ solves } \quad \begin{cases} F(t,s,\dot{s},a) \;=\; 0 \\ G(s(0),a) \;=\; 0 \end{cases} \tag{2}$$

where F corresponds to the implicit equations associated with semi-discretization of the PDE:

$$\begin{cases} \frac{\partial u}{\partial t} \;+\; \mathcal{F}_{\xi,\omega,\phi}(u,\nabla u,\Delta u) \;=\; \zeta(a) \\ u \;=\; 0 \quad \text{on } \partial\Omega \end{cases} \tag{3}$$

and G defines the initial conditions of the system. More precisely, after discretizing in space:

$$\begin{cases} F \;=\; \frac{\partial u}{\partial t} - \left[\mathbf{A}u + \mathbf{Q}_{\xi,\omega} + \mathbf{V}_{\phi}u + \mathbf{P}(\alpha)\beta\right] \;=\; 0 \\ G \;=\; u(0) - u_0 \;=\; 0 \end{cases} \quad \text{with} \quad s = \begin{bmatrix} | \\ u \\ | \end{bmatrix} \quad \text{and} \quad a = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{4}$$

Gradient-Based Optimization

a local approach

Gradient-Based Optimization: Motivation for Lagrangian

To find an optimal control a^* for the objective function $J(s,a)$, it is natural to consider the gradient:

$$\nabla_a J \;=\; \int_0^T \nabla_a L(t,s,a)\,dt \;+\; \nabla_a \Psi(s(T)) \;=\; \int_0^T \tfrac{\partial L}{\partial s}\nabla_a s + \tfrac{\partial L}{\partial a}\,dt \;+\; \tfrac{\partial \Psi}{\partial s}\nabla_a s\Big|_{t=T} \tag{5}$$

However, computing this directly is typically infeasible due to the appearance of $\nabla_a s$ on the right-hand-side.

Fortunately, a workaround exists which is based on the following observation:

$$\nabla_a J \;=\; \nabla_a \mathcal{L} \quad \text{for all } \lambda(t), \mu(t) \quad \text{when we set} \tag{6}$$

$$\mathcal{L} \;\equiv\; \int_0^T L(t,s,a) + \lambda^T F(t,s,\dot{s},a)\,dt \;+\; \mu^T G(s(0),a) + \Psi(s(T)) \tag{7}$$

This allows us to avoid calculating $\nabla_a s$ by choosing λ, μ so that these problematic terms cancel out.

Gradient-Based Optimization: Motivation for Lagrangian

To find an optimal control a^* for the objective function $J(s, a)$, it is natural to consider the gradient:

$$\nabla_a J \;=\; \int_0^T \nabla_a L(t, s, a)\, dt \;+\; \nabla_a \Psi(s(T)) \;=\; \int_0^T \tfrac{\partial L}{\partial s} \nabla_a s + \tfrac{\partial L}{\partial a}\, dt \;+\; \tfrac{\partial \Psi}{\partial s} \nabla_a s\Big|_{t=T} \tag{5}$$

However, computing this directly is typically infeasible due to the appearance of $\nabla_a s$ on the right-hand-side.

Fortunately, a workaround exists which is based on the following observation:

$$\nabla_a J \;=\; \nabla_a \mathcal{L} \quad \text{for all } \lambda(t), \mu(t) \quad \text{when we set} \tag{6}$$

$$\mathcal{L} \;\equiv\; \int_0^T L(t, s, a) + \lambda^T F(t, s, \dot{s}, a)\, dt \;+\; \mu^T G(s(0), a) + \Psi(s(T)) \tag{7}$$

This allows us to avoid calculating $\nabla_a s$ by choosing λ, μ so that these problematic terms cancel out.

Numerical Algorithm

I.) Integrate $\begin{cases} F(t, s, \dot{s}, a) = 0 \\ s(0) = s_0 \end{cases}$ for s forward in time

II.) Integrate $\begin{cases} \frac{\partial L}{\partial s} + \lambda^T \left(\frac{\partial F}{\partial s} - \frac{\partial}{\partial t} \frac{\partial F}{\partial \dot{s}} \right) - \dot{\lambda}^T \frac{\partial F}{\partial \dot{s}} = 0 \\ \lambda(T) = -\frac{\partial \Psi}{\partial s} \left[\frac{\partial F}{\partial \dot{s}} \right]^{-1} \end{cases}$ for λ backward in time

III.) Compute $\nabla_a J$ using the reduced expression:

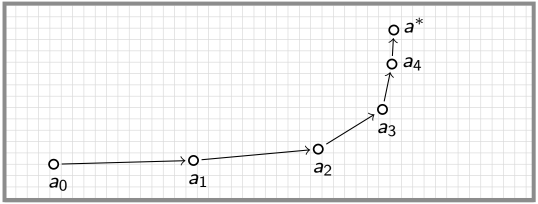
$$\nabla_a J = \int_0^T \frac{\partial L}{\partial a} + \lambda^T \frac{\partial F}{\partial a} \, dt + \lambda^T \frac{\partial F}{\partial \dot{s}} \bigg|_{t=0} \left[G_{s(0)} \right]^{-1} \frac{\partial G}{\partial a} \tag{8}$$

Now the optimal control a^* can be found using gradient-based search algorithms with only one additional PDE solve required for each gradient calculation (i.e. integrating backward in time for λ).

Local Nature of Solutions

Recall that the dynamics in the original problem formulation were actually parameterized by (ξ, ω, ϕ):

$$\begin{cases} \frac{\partial u}{\partial t} + \mathcal{F}_{\xi,\omega,\phi}(u, \nabla u, \Delta u) = \zeta(a) \\ u = 0 \quad \text{on } \partial\Omega \end{cases} \tag{9}$$

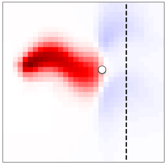


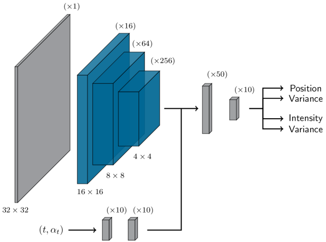
Since the gradient-based search is designed to converge to one specific action sequence a^*, the entire process must be repeated whenever new parameters (ξ', ω', ϕ') are encountered.

Reinforcement Learning

`gradient-free search`

Reinforcement Learning: Markov Policy Approach





$$a(t_n) = \begin{bmatrix} \alpha(t_n) \\ \beta(t_n) \end{bmatrix}$$

Replace direct search for optimal $a^* = \{a(0), a(t_1), \ldots, a(T)\}$ with one-step-ahead policy model; i.e. propose the next action $a(t_n)$ based on the current system state $s(t_n)$.

→ does not require gradients from PDE system, only needs a differentiable policy π_θ

→ uses intermediate observations of the system and loss to propose controls sequentially

→ in principle, can be applied to several distinct problem scenarios at the same time

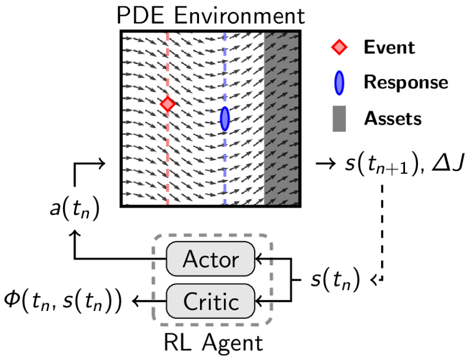
Reinforcement Learning: Trial-and-Error Strategy

I.) Sample collection of configurations $\{(\xi_i, \omega_i, \phi_i)\}$

II.) Apply policy to each system from $t = 0$ to $t = T$

III.) Evaluate outcomes $\{J(s_i, a_i)\}$ for current policy

IV.) Update the policy based on outcomes and repeat

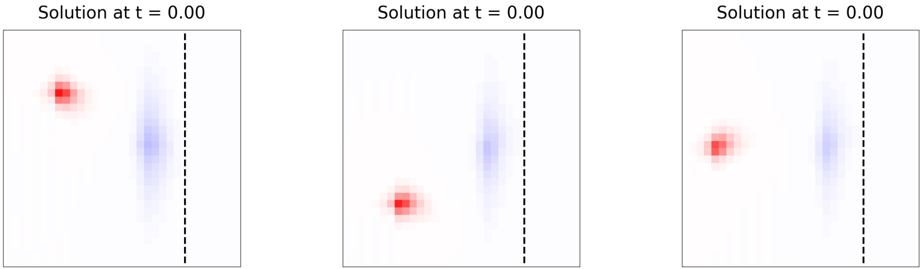


- Performance is gauged w.r.t. state “value” estimates:

$$\Phi(t, s(t)) = \mathbb{E}_{\pi_\theta}\left[\int_t^T L(t, s, a)dt + \Psi(s(T))\right] \quad (10)$$

Reinforcement Learning Policies for Adaptive Control

Trained RL policy provides approximate solutions for a *complete family of problems* $\{(\xi_i, \omega_i, \phi_i)\}$:



`But without leveraging PDE gradients/structure we find:`
`i.) poor data efficiency`
`ii.) suboptimal convergence`

Approximate HJB

`semi-global solutions`

Dynamic Programming Perspective

$$\min_a J(s,a) \;=\; \int_0^T L(t,s,a)\,dt \;+\; \Psi(s(T)) \quad \text{subject to} \quad \begin{cases} \frac{\partial s}{\partial t} \;=\; f(t,s,a) \\ s(0) \;=\; s_0 \end{cases} \tag{11}$$

where $f(t,s,a) \;=\; \zeta(a) - \mathcal{F}_{\xi,\omega,\phi}(u, \nabla u, \Delta u) \;\sim$ dynamics for a given configuration (ξ, ω, ϕ).

- Aiming for a procedure *applicable to multiple configurations*, we first augment the state variable:

$$s^T \;=\; \Big[\, \text{—}\; u \;\text{—} \;,\; \xi \;,\; \omega,\; \phi \,\Big] \tag{12}$$

Core Mathematical Elements

Value Function: $$\Phi(t,s) = \inf_a J(t,s,a) \quad \text{where} \quad J(t,s,a) = \int_t^T L(t,s,a)\,dt + \Psi(s(T)) \tag{13}$$

→ extends analysis to handle multiple problem configurations (as in RL approach)

Hamiltonian: $$\mathcal{H}(t,s,a,p) = p(t)^T f(t,s,a) - L(t,s,a) \tag{14}$$

→ incorporates mathematical structure of system dynamics (similar to local approach)

Adjoint Equation: $$\begin{cases} \frac{\partial}{\partial t} p(t) = \nabla_s \mathcal{H}(t,s^*,a^*,p) \\ p(T) = \nabla_s \Psi(s^*(T)) \end{cases} \quad \text{where } s^*,\ a^* \sim \text{optimal state/control} \tag{15}$$

Core Mathematical Elements

Value Function: $$\Phi(t,s) \;=\; \inf_a J(t,s,a) \quad \text{where} \quad J(t,s,a) \;=\; \int_t^T L(t,s,a)\,dt \;+\; \Psi(s(T)) \tag{13}$$

→ extends analysis to handle multiple problem configurations (as in RL approach)

Hamiltonian: $$\mathcal{H}(t,s,a,p) \;=\; p(t)^T f(t,s,a) - L(t,s,a) \tag{14}$$

→ incorporates mathematical structure of system dynamics (similar to local approach)

Adjoint Equation: $$\begin{cases} \frac{\partial}{\partial t} p(t) \;=\; \nabla_s \mathcal{H}(t,s^*,a^*,p) \\ p(T) \;=\; \nabla_s \Psi(s^*(T)) \end{cases} \quad \text{where} \quad s^*\,,\; a^* \;\sim\; \text{optimal state/control} \tag{15}$$

Core Mathematical Elements

Value Function: $\Phi(t, s) = \inf_a J(t, s, a) \quad \text{where} \quad J(t, s, a) = \int_t^T L(t, s, a)\, dt + \Psi(s(T))$ (13)

→ extends analysis to handle multiple problem configurations (as in RL approach)

Hamiltonian: $\mathcal{H}(t, s, a, p) = p(t)^T f(t, s, a) - L(t, s, a)$ (14)

→ incorporates mathematical structure of system dynamics (similar to local approach)

Adjoint Equation: $\begin{cases} \frac{\partial}{\partial t} p(t) = \nabla_s \mathcal{H}(t, s^*, a^*, p) \\ p(T) = \nabla_s \Psi(s^*(T)) \end{cases} \quad \text{where} \quad s^*,\ a^* \sim \text{optimal state/control}$ (15)

Pontryagin Maximum Principle: Feedback Form

For any optimal state/costate pair (s^*, p^*), the *Pontryagin Maximum Principle* states that:

$$a^*(t) \in \arg\sup_a \mathcal{H}(t, s^*, a, p^*) \tag{16}$$

In practice, this simple property can provide remarkably precise information about the optimal control $a^*(t)$.

In particular, when the dynamics f and running cost L are differentiable with respect to a:

$$\nabla_a \mathcal{H}(t, s, a, p) = p^T \nabla_a f - \nabla_a L = 0 \tag{17}$$

$\Rightarrow$ express optimal control in terms of state s and adjoint variable p

$$\textbf{Feedback Form:} \quad a^*(t) = \Lambda(t, s^*(t), p^*(t)) \tag{18}$$

Hamilton-Jacobi-Bellman Equations

If Φ is twice differentiable at $(t, s^*(t))$ for $t < T$, then an optimal costate/adjoint variable is given by:

$$p^*(t) \;=\; -\nabla_s \Phi(t, s^*(t)) \tag{19}$$

Moreover, the following *Hamilton-Jacobi-Bellman equations* (HJB) hold at every state s and time $t < T$:

$$\frac{\partial}{\partial t}\Phi(t, s) \;=\; \sup_a \mathcal{H}(t, s^*, a, -\nabla_s \Phi(t, s)) \tag{20}$$

$$\text{w/}\;\; a^*(t) \;=\; \Lambda(t, s^*(t), -\nabla_s \Phi(t, s^*(t))) \tag{21}$$

Technical Approach for Approximate HJB Solutions

- The feedback form gives an expression for optimal control in terms of optimal state and costate.
- The optimal costate is given by the gradient of the value function.

$\Longrightarrow$ The control problem can be reduced to approximating the value function.

Goal: Construct a differentiable surrogate $\Phi_\theta(t, s)$ for the true value function which satisfies:

$$\begin{cases} \frac{\partial}{\partial t}\Phi_\theta(t,s) = \mathcal{H}(t, s^*, \Lambda(t, s, -\nabla_s \Phi_\theta(t,s)), -\nabla_s \Phi_\theta(t,s)) \\ \Phi_\theta(T, s(T)) = \Psi(s(T)) \\ \nabla_s \Phi_\theta(T, s(T)) = \nabla_s \Psi(s(T)) \end{cases} \tag{22}$$

Technical Approach for Approximate HJB Solutions

- The feedback form gives an expression for optimal control in terms of optimal state and costate.

- The optimal costate is given by the gradient of the value function.

$\Longrightarrow$ The control problem can be reduced to approximating the value function.

Goal: Construct a differentiable surrogate $\Phi_\theta(t, s)$ for the true value function which satisfies:

$$\begin{cases} \frac{\partial}{\partial t}\Phi_\theta(t,s) = \mathcal{H}(t, s^*, \Lambda(t, s, -\nabla_s\Phi_\theta(t,s)), -\nabla_s\Phi_\theta(t,s)) \\ \Phi_\theta(T, s(T)) = \Psi(s(T)) \\ \nabla_s\Phi_\theta(T, s(T)) = \nabla_s\Psi(s(T)) \end{cases} \tag{22}$$

Feedback Form Calculation

$$\mathcal{H}(t,s,a,p) \;=\; p(t)^T f(t,s,a) - L(t,s,a) \tag{23}$$

$$f(t,s,a) \;=\; \left[\mathbf{A}u + \mathbf{Q}_{\xi,\omega} + \mathbf{V}_\phi u + \mathbf{P}(\alpha)\beta\right] \quad \text{and} \quad L(t,s,a) \;=\; \tfrac{1}{2}\|a\|^2 \tag{24}$$

$$\nabla_a \mathcal{H} \;=\; p^T \nabla_a f - \nabla_a L \;=\; 0 \tag{25}$$

$$\frac{\partial}{\partial\beta}\mathcal{H} \;=\; p^T \frac{\partial}{\partial\beta} f - \frac{\partial}{\partial\beta} L \;=\; 0 \tag{26}$$

$$\Rightarrow \; p^T\mathbf{P} - \beta^T \;=\; 0 \tag{27}$$

$$\Rightarrow \; \beta \;=\; \mathbf{P}^T p \tag{28}$$

$$\beta(t) \;=\; -\mathbf{P}^T \nabla_u \Phi_\theta(t,s) \tag{29}$$

Value Function Dynamics

$$\mathcal{H}(t,s,a,p) \;=\; p(t)^T f(t,s,a) \,-\, L(t,s,a) \tag{30}$$

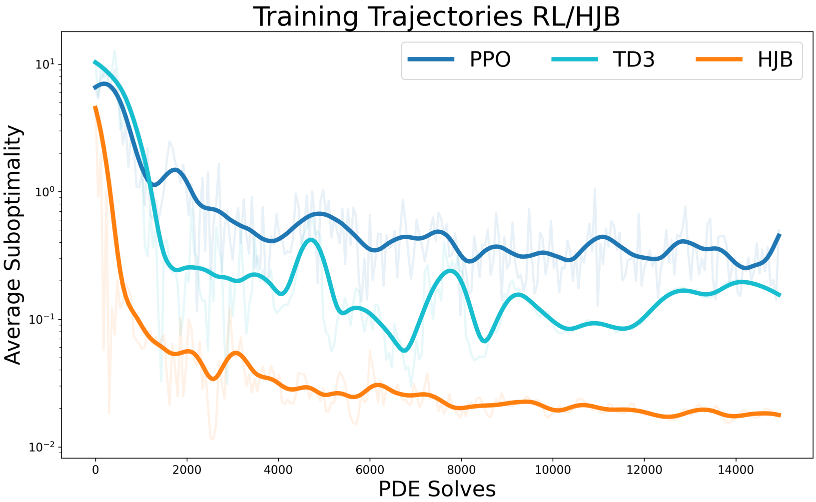
$$f(t,s,a) \;=\; \left[\,\mathbf{A}u + \mathbf{Q}_{\xi,\omega} + \mathbf{V}_\phi u + \mathbf{P}(\alpha)\beta\,\right] \quad \text{and} \quad L(t,s,a) \;=\; \tfrac{1}{2}\|a\|^2 \tag{31}$$

$$\tfrac{\partial}{\partial t}\Phi_\theta(t,s) \;=\; \mathcal{H}(t,s^*,\Lambda(t,s,-\nabla_s\Phi_\theta(t,s))\,,\,-\nabla_s\Phi_\theta(t,s)) \tag{32}$$

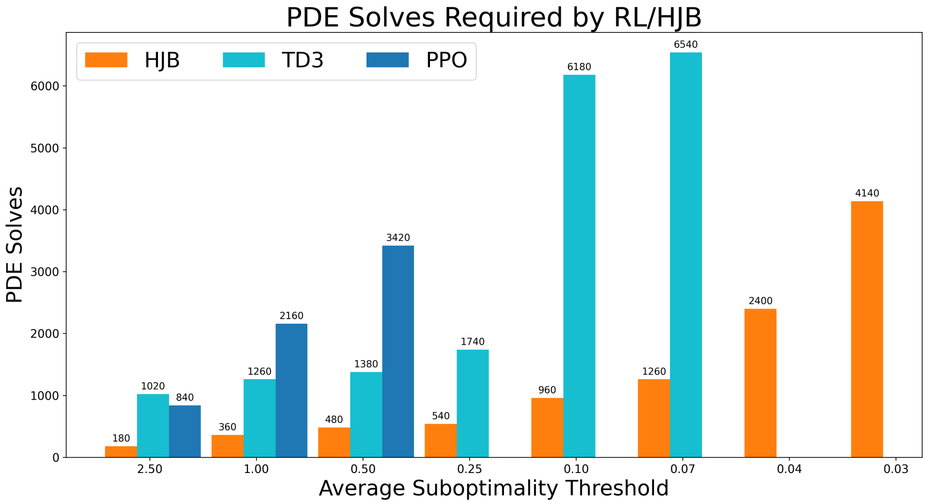
$$\tfrac{\partial}{\partial t}\Phi_\theta \;=\; -\left[\nabla_s\Phi_\theta\right]^T\left[\,\mathbf{A}u + \mathbf{Q}_{\xi,\omega} + \mathbf{V}_\phi u + \mathbf{P}(\alpha)\beta\,\right] \,-\, \tfrac{1}{2}(\|\alpha\|^2 + \|\beta\|^2) \tag{33}$$

where α and β are computed using the feedback form expressions and $\nabla_s\Phi(t,s)$.

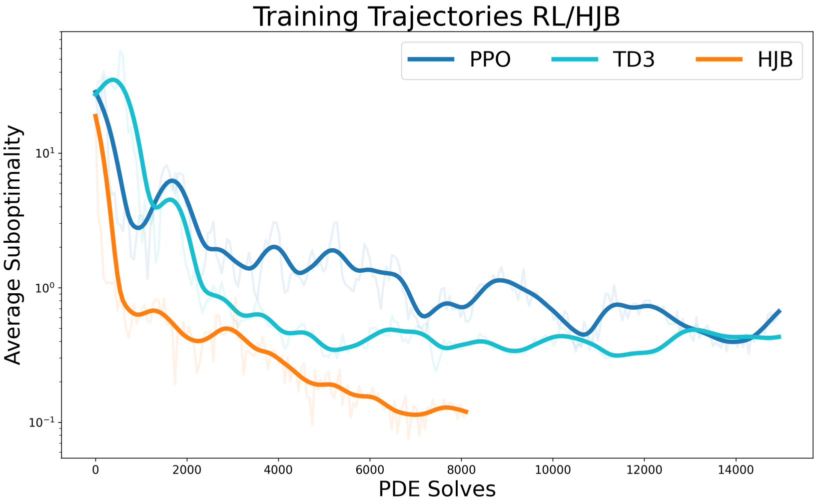
RL/HJB Comparison: Example 1



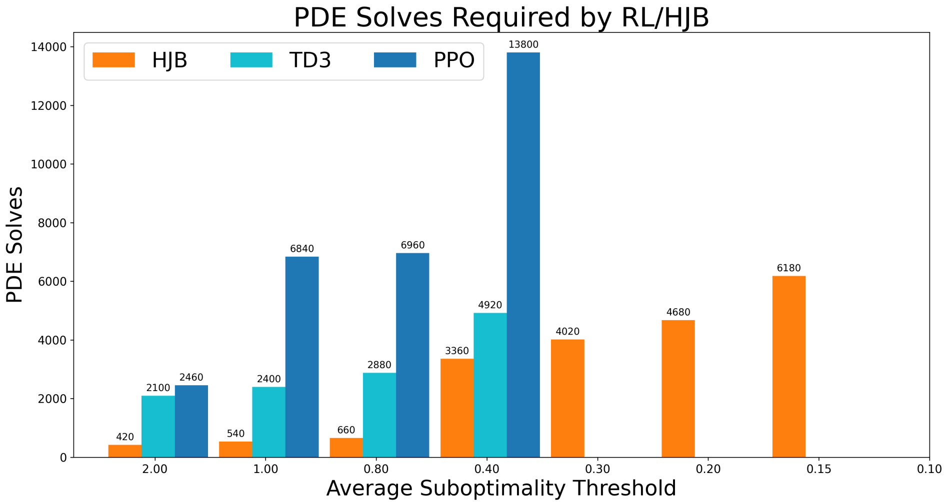
RL/HJB Comparison: Example 1



RL/HJB Comparison: Example 2



RL/HJB Comparison: Example 2



Summary and Concluding Remarks

Gradient-Based Methods	Reinforcement Learning	Approximate HJB
Single PDE configuration	Multiple configurations	Multiple configurations
Involved implementation (gradient calculations)	Simple implementation (black-box/no gradients)	Involved implementation (gradient calculations)
Optimal control solutions (relative to RL/HJB)	Suboptimal solutions (with more PDE solves)	Near optimal solutions (with less PDE solves)

→ Local methods yield most accurate solutions, but require precise knowledge of system configuration

→ HJB leverages offline calculations to form policies which can *adapt to a broad range of configurations*