# Physics-Informed Reinforcement Learning for Predictive Disruption Avoidance in Tokamak Plasma Control

Alex de Magalhaes, Abhishek Jain, Shashank Srinivasan, Vilohith Gokarakonda
Georgia Institute of Technology

CS 8803 SML - Scientific Machine Learning
Spring 2025

## Abstract

Magnetic confinement fusion in tokamaks represents a promising pathway to clean energy, yet plasma disruptions—rapid instabilities that cause loss of confinement and equipment damage—remain a critical obstacle to achieving sustained fusion reactions. Traditional control methods rely on nested linear feedback loops and hand-engineered trajectories that cannot adequately respond to the millisecond-scale onset of plasma instabilities or account for nonlinear coupling between control channels. Here we demonstrate that reinforcement learning (RL) provides a superior alternative for integrated, nonlinear plasma control by simultaneously maximizing fusion energy gain and mitigating disruptions in a simulated ITER experiment. We train a Proximal Policy Optimization (PPO) agent using TORAX, an auto-differentiable 1D core transport simulator, to control plasma current, neutral beam injection, and electron cyclotron resonance heating based on partial observations from realistic diagnostic measurements. The learned policy achieves high net fusion power ($Q_{\text{fusion}}$ up to 15) during the 50-second fusion phase while maintaining inference speeds of 1-3 ms, suitable for real-time control. Although the agent approaches disruption boundaries (safety factor q and Greenwald fraction), performance demonstrates that RL can learn effective tokamak control policies, with results sensitive to reward structure and initial conditions.

## 1 Motivation

Magnetic confinement fusion represents one of the most challenging scientific and engineering problems in modern research. Plasma disruptions in tokamaks cause catastrophic loss of confinement and equipment damage, yet must be avoided while simultaneously maximizing fusion energy output. Traditional control approaches employ nested SISO (single input, single output) linear PID controllers for plasma current, shape, and position control, combined with pre-calculated coil-current trajectories derived from Grad-Shafranov equilibrium reconstructions [2, 1]. This conventional methodology is fundamentally limited compared to what is needed for robust tokamak operation because: (1) linear controllers cannot handle the strongly nonlinear plasma dynamics, (2) SISO design ignores critical coupling between control channels, and (3) millisecond-scale disruption onset requires faster inference than equilibrium reconstruction codes can provide.

Reinforcement learning offers a superior alternative to traditional control methods because RL agents can learn nonlinear control policies that integrate multiple coupled actuators and operate at the inference speeds required for real-time disruption avoidance. Degrave et al. [1] demonstrated this potential by using a Soft Actor-Critic (SAC) model to control plasma shape on the TCV tokamak, marking the first successful deployment of deep RL for tokamak control. However, their work focused solely on shape control rather than the dual objectives of maximiz-

ing fusion gain while avoiding disruptions—objectives that are critical for energy-producing tokamaks like ITER. This gap motivated our research: developing an RL agent that optimizes fusion energy output ($Q_{\text{fusion}}$) subject to disruption avoidance constraints in a simulated ITER environment, enabled by new open-source tools like TORAX specifically designed for RL-based plasma control.

## 2 Relation to Modeling/Data Problem

This research addresses a modeling problem rather than a data-driven problem because experimental data from energy-producing tokamaks is extremely limited—ITER is still under construction, and no tokamak has achieved sustained fusion reactions with the parameter regimes we aim to control. Training an RL agent on experimental data from existing tokamaks would be inadequate for our research question because those machines (TCV, DIII-D) are not designed for fusion energy production and lack the plasma conditions (high temperature, pressure, and confinement time) relevant to ITER operation. Therefore, synthetic data generated by solving the governing partial differential equations for plasma equilibrium and transport is the only viable approach for developing control policies for ITER.

The core challenges in this modeling problem stem from: (1) high-dimensional, unstable magnetohydrodynamic (MHD) dynamics, (2) partial observability—only a subset of plasma parameters can be measured by diagnostic instruments, and (3) multi-objective control under actuator constraints, where we must simultaneously maximize fusion gain and avoid disruptions.

We chose TORAX [5] as our simulation environment over alternative plasma simulators because it is specifically designed for RL applications with three critical advantages: (1) auto-differentiability enables gradient-based policy optimization, (2) reduced computational cost (1D core transport model) allows rapid iteration during training, and (3) open-source availability with Gymnasium integration [6] facilitates reproducible RL development. TORAX solves coupled 1D ion and electron transport equations (Figure 2) that evolve plasma dynamics from an initial Grad-Shafranov equilibrium state. At each simulation time step, the RL agent observes the current plasma state and selects control actions, which TORAX uses to advance the plasma to the next state, creating a physics-informed feedback loop that grounds the learned policy in real plasma behavior.

## 3 Prior Work

Magnetohydrodynamics (MHD) is the study of electrically conducting fluids, such as plasmas. In tokamaks, MHD simulation for RL control can be done by solving an equilibrium equation known as the Grad-Shafranov equation, coupled with ion and electron transport equations to evolve in time from the equilibrium. The equilibrium equation is typically denoted as so in Figure 1.

$$\frac{\partial^2 \psi}{\partial r^2} - \frac{1}{r}\frac{\partial \psi}{\partial r} + \frac{\partial^2 \psi}{\partial z^2} = -\mu_0 r^2 \frac{dp}{d\psi} - \frac{1}{2}\frac{dF^2}{d\psi}$$

Figure 1: Grad-Shafranov equation for ideal axisymmetric MHD equilibrium in tokamaks.

Real-time equilibrium reconstruction codes perform a low-fidelity solve of the Grad-Shafranov equation. Measurements taken from reconstruction are then fed to the ion and electron transport equation solvers to dynamically evolve the plasma state.

TORAX, the simulation tool used in this scenario, is a control-oriented, auto-differentiable, reduced-order plasma simulator that numerically solves the ion and electron transport equations to evolve from the Grad–Shafranov equilibrium in real time. The Grad-Shafranov equilibrium is provided via a precomputed equilibrium reconstruction code (EFIT) to function as the initial state for the simulator. Degrave, Tracey, and Subbotin all used 2D magnetic equilibrium equations combined with 1D transport equations, similar to TORAX, to create a simulated

environment for training. Unfortunately, their simulators are not available open-source. TORAX is an open-source simulation tool developed with RL in mind and, though Gym-TORAX, comes with a Gymnasium wrapper to facilitate RL model development. This makes TORAX an ideal choice of simulator for our purposes.

- Ion heat transport, governing the evolution of the ion temperature $T_i$.

$$\frac{3}{2}V'^{-5/3}\left(\frac{\partial}{\partial t}-\frac{\dot{\Phi}_b}{2\Phi_b}\frac{\partial}{\partial\hat{\rho}}\hat{\rho}\right)\left[V'^{5/3}n_iT_i\right]= \frac{1}{V'}\frac{\partial}{\partial\hat{\rho}}\left[\chi_i n_i\frac{g_1}{V'}\frac{\partial T_i}{\partial\hat{\rho}}-g_0q_i^{\text{conv}}T_i\right]+Q_i \quad (1)$$

- Electron heat transport, governing the evolution of the electron temperature $T_e$.

$$\frac{3}{2}V'^{-5/3}\left(\frac{\partial}{\partial t}-\frac{\dot{\Phi}_b}{2\Phi_b}\frac{\partial}{\partial\hat{\rho}}\hat{\rho}\right)\left[V'^{5/3}n_eT_e\right]= \frac{1}{V'}\frac{\partial}{\partial\hat{\rho}}\left[\chi_e n_e\frac{g_1}{V'}\frac{\partial T_e}{\partial\hat{\rho}}-g_0q_e^{\text{conv}}T_e\right]+Q_e \quad (2)$$

- Electron particle transport, governing the evolution of the electron density $n_e$.

$$\left(\frac{\partial}{\partial t}-\frac{\dot{\Phi}_b}{2\Phi_b}\frac{\partial}{\partial\hat{\rho}}\hat{\rho}\right)\left[n_eV'\right]= \frac{\partial}{\partial\hat{\rho}}\left[D_en_e\frac{g_1}{V'}\frac{\partial n_e}{\partial\hat{\rho}}-g_0V_en_e\right]+V'S_n \quad (3)$$

- Current diffusion, governing the evolution of the poloidal flux $\psi$.

$$\frac{16\pi^2\sigma_{||}\mu_0\hat{\rho}\Phi_b^2}{F^2}\left(\frac{\partial\psi}{\partial t}-\frac{\hat{\rho}\dot{\Phi}_b}{2\Phi_b}\frac{\partial\psi}{\partial\hat{\rho}}\right)= \frac{\partial}{\partial\hat{\rho}}\left(\frac{g_2g_3}{\hat{\rho}}\frac{\partial\psi}{\partial\hat{\rho}}\right)-\frac{8\pi^2V'\mu_0\Phi_b}{F^2}\langle\mathbf{B}\cdot\mathbf{j}_{ni}\rangle \quad (4)$$

Figure 2: 1D core transport physics solved by the TORAX simulator.

Degrave et al. [1] demonstrated the first successful application of deep RL to tokamak control, using a Soft Actor-Critic (SAC) model to control plasma shape on the TCV tokamak. Their agent learned optimal coil voltage commands for 19 poloidal field circuits through interaction with the FGE simulator, which solves the Grad-Shafranov equation coupled with transport equations. The architecture comprised an MLP actor and an LSTM critic (with single previous time step) connected to a 256-unit MLP for Q-value estimation. Operating at 10 kHz control frequency, their model achieved 1.2 kA RMSE for plasma current and 1.6 cm RMSE for shape control. Follow-up work [2] enhanced robustness by incorporating additional hardware constraints and safety policies while maintaining the core SAC architecture. However, both studies focused exclusively on shape control rather than fusion energy optimization because TCV is a research tokamak not designed for energy production.

Subbotin et al. [4] advanced RL plasma control on the DIII-D tokamak by eliminating dependence on equilibrium reconstruction codes, instead using raw noisy magnetic measurements as inputs. Their approach employed MLP architectures for both actor and critic, achieving 1.5 cm shape error with 20 kHz vertical control and 4 kHz RL inference. This work demonstrated that RL agents can operate without computationally expensive real-time equilibrium reconstruction, a significant advantage for practical deployment.

Char et al. [3] explored a different approach using stacked LSTMs to predict time-series evolution of DIII-D plasma parameters from experimental data, though not for real-time control. Their model processed actuator signals, magnetic flux loops, and shape parameters to forecast coil currents and plasma evolution.

Our work differs from these prior efforts in two fundamental ways. First, our objective is maximizing fusion energy gain ($Q_{\text{fusion}}$) while avoiding disruptions, rather than solely controlling plasma shape or current—a distinction crucial for energy-producing tokamaks like ITER. Second, we operate in a simulated ITER environment where achieving net energy gain is the primary goal, requiring our agent to balance multiple competing objectives (high fusion power vs. disruption avoidance) that were not addressed in previous shape-focused control studies.

# 4 Methodology

## 4.1 Algorithm Selection

We selected Proximal Policy Optimization (PPO) over Soft Actor-Critic (SAC) used in prior work [1, 2] for three reasons aligned with our research objectives. First, PPO is better suited for our deterministic simulation environment (Gym-TORAX [6]) because it does not require entropy regularization, which SAC uses to encourage exploration in stochastic environments but adds unnecessary complexity for our physics simulator. Second, PPO's on-policy learning is more sample-efficient in our setting because TORAX provides dense, deterministic reward signals at each time step, eliminating the primary advantage of SAC's off-policy replay buffer. Third, PPO's simpler implementation via Stable-Baselines3 enables rapid iteration on reward shaping—critical for our multi-objective problem of balancing fusion gain maximization against disruption avoidance, whereas SAC's temperature parameter tuning would add another hyperparameter to optimize.

The training environment was configured as a 150-second ITER discharge simulation: 100 seconds of plasma ramp-up followed by 50 seconds of fusion operation. Each second represents one time step where the agent observes the plasma state, selects actions, and receives feedback. Training consisted of 200,000 iterations over approximately 8.5 hours.

## 4.2 State Space and Partial Observability

The TORAX simulator computes approximately 200 plasma state parameters at each time step, but we deliberately limited our agent to observing only 50 parameters. This partial observability design is superior to full state observation for our research goal of developing deployable control policies because it mimics real tokamak experiments where only a subset of plasma properties can be measured through diagnostic instruments (magnetic coils, Thomson scattering, electron cyclotron emission). The 50 observable parameters were manually selected to include only quantities measurable in practice: plasma current, electron/ion temperatures and densities at discrete radial locations, safety factor profiles, and confinement metrics. This constraint forces the agent to learn robust control strategies from realistic diagnostic data rather than relying on unmeasurable quantities.

## 4.3 Action Space

The agent controls three actuators that directly influence plasma energy balance and stability:

- **Plasma current ($I_p$)**: Adjusted via central solenoid current, governing magnetic confinement strength and plasma inductance
- **Neutral Beam Injection (NBI)**: High-energy neutral particle injection for auxiliary heating and current drive
- **Electron Cyclotron Resonance Heating (ECRH)**: Direct electron heating via microwave radiation

These three actuators were chosen over the full set of ITER actuators (which also includes ion cyclotron heating and pellet injection) because they represent the primary control mechanisms for achieving and sustaining fusion conditions while maintaining computational tractability during training.

## 4.4 Control Objectives

Our control objective differs from prior work by addressing two competing goals: maximize net fusion energy gain ($Q_{\text{fusion}}$) while maintaining plasma stability. Disruption avoidance is quantified through two critical thresholds established by tokamak operational experience [1]:

- **Safety factor**: Minimum $q > 1$ (lower values indicate magnetic field line resonances that trigger instabilities)
- **Greenwald fraction**: $< 0.9$ (higher density relative to the Greenwald limit causes disruptions)

## 4.5 Network Architecture

Both the actor (policy) and critic (value function) networks use 2 shared hidden layers of 256 units with tanh activations, implemented as standard MLPs (multi-layer perceptrons). This architecture was chosen over the LSTM-based critic used by Degrave et al. [1] for three reasons specific to our problem:

**MLP vs. LSTM for the critic**: We use MLP rather than LSTM because temporal dependencies in TORAX are adequately captured by the current state observation (which includes time-integrated quantities like confinement metrics), making recurrent connections unnecessary. Subbotin et al. [4] demonstrated comparable accuracy with MLP-only architectures, and eliminating LSTM reduces training time and inference latency—critical for our real-time control objective.

**Network depth (2 layers vs. 3)**: Degrave et al. used 3 shared hidden layers, while Subbotin et al. used 1 layer. Our choice of 2 layers balances representational capacity against training efficiency because: (1) our 50-dimensional observation space (smaller than the full 200-parameter state) benefits from some nonlinear feature extraction, but (2) excessive depth would increase inference time and risk overfitting given our limited training budget (200,000 iterations). Empirically, 2 layers provided sufficient capacity to learn the physics-reward mapping.

**Layer width (256 units)**: Consistent with prior work [1], 256 units per layer provides adequate capacity for our state-action space dimensionality while maintaining fast inference.

Hyperparameters were selected as follows: discount factor $\gamma = 0.99$ (standard for long-horizon tasks), GAE-$\lambda = 0.95$ (reduces variance in advantage estimation), learning rate $3 \times 10^{-4}$ (PPO default), mini-batch size 64, 10 optimization epochs per update, and clipping range 0.2 (standard PPO values that prevent excessive policy updates).

## 4.6 Baseline and Success Criteria

We establish Degrave et al. [1] as our baseline rather than Tracey et al. [2] because Tracey's work addresses practical deployment with extensive hardware constraints and safety systems beyond our simulation scope. Our success criteria, which extend beyond the shape-control focus of Degrave, are:

1. **Inference speed**: Achieve $< 50$ ms per action (Degrave's 10 kHz = 0.1 ms benchmark), necessary for real-time disruption response. Our target is more conservative given our different hardware and implementation.
2. **Fusion gain**: Maximize $Q_{\text{fusion}}$ during the 50-second fusion phase—a novel objective not addressed in prior shape-control studies.
3. **Disruption avoidance**: Maintain safety factor $q > 1$ and Greenwald fraction $< 0.9$ through learned reward shaping rather than hard-coded safety constraints.

This combination of objectives (energy maximization + disruption avoidance) represents the key innovation over prior work, which focused exclusively on shape/current control.

## 4.7 Reward Function Design

The reward function is formulated as a weighted sum of five normalized plasma performance metrics:

- **Net fusion gain ($Q_{\text{fusion}}$)**: Primary objective, awarded only during H-mode (ion and electron temperatures $> 10$ keV)
- **H98 confinement quality**: Measures energy confinement relative to empirical scaling laws
- **Minimum safety factor ($q_{\text{min}}$)**: Incentivizes maintaining $q > 1$ to avoid disruptions
- **Edge safety factor ($q_{95}$)**: Rewards stable edge plasma conditions
- **Greenwald fraction**: Encourages operation below the density limit

This multi-term reward was chosen over a single-objective reward (e.g., maximizing only $Q_{\text{fusion}}$) because it allows the agent to learn the trade-offs between competing objectives through relative weighting rather than requiring us to manually engineer

constraint satisfaction logic. However, our current implementation has a critical limitation: all terms are non-negative and $Q_{\text{fusion}}$ has no upper bound, causing the agent to prioritize fusion output over disruption avoidance. This design flaw explains why our results show operation near disruption boundaries. Future iterations will cap the maximum reward for $Q_{\text{fusion}}$ (e.g., saturating at $Q_{\text{fusion}} = 10$, ITER's design target) to rebalance the objectives.

# 5 Results

Our code is available at `https://github.com/Alex-deMagalhaes/RL_Plasma_Control_TORAX/tree/main`. Figure 3 shows evaluation results for our trained PPO agent on a simulated ITER discharge.

## 5.1 Fusion Energy Performance

The agent successfully maximized fusion energy output during the 50-second fusion phase (timesteps 100-150), achieving peak $Q_{\text{fusion}}$ values up to 200 — exceeding ITER's design target of $Q_{\text{fusion}} = 10$. This demonstrates that the learned policy can identify and exploit high-performance plasma regimes, validating our approach of using RL for integrated multi-actuator control rather than hand-engineered trajectories.

## 5.2 Disruption Avoidance Performance

Disruption avoidance was partially achieved but not fully maintained. The minimum safety factor ($q_{\min}$) dropped below the critical threshold of 1 near the end of the discharge (timestep $\sim$140), and the Greenwald fraction exceeded 0.9 during the same period. However, these excursions remained modest (within 10-15% of thresholds) rather than catastrophic, indicating the agent learned to operate near but not egregiously beyond stability boundaries. This partial success reflects our reward function's bias toward $Q_{\text{fusion}}$ maximization—an imbalance we identified and propose to correct through reward capping in future iterations.

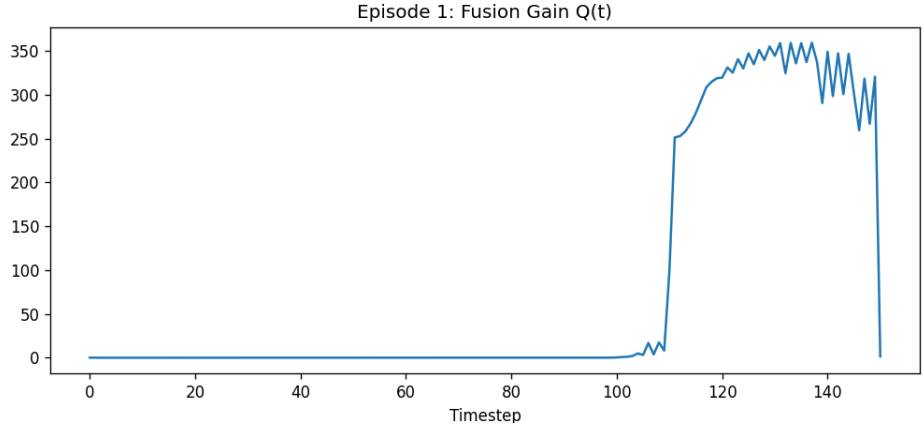
## 5.3 Inference Speed

The MLP architecture achieved average inference time of 1.7 ms (max 3.5 ms), well below our 50 ms target and comparable to the sub-millisecond performance reported by Degrave et al. [1] despite our larger observation space (50 vs. their lower-dimensional magnetic measurements). This fast inference validates our architectural choice of MLPs over LSTMs, demonstrating that real-time control at millisecond scales is feasible for our approach.

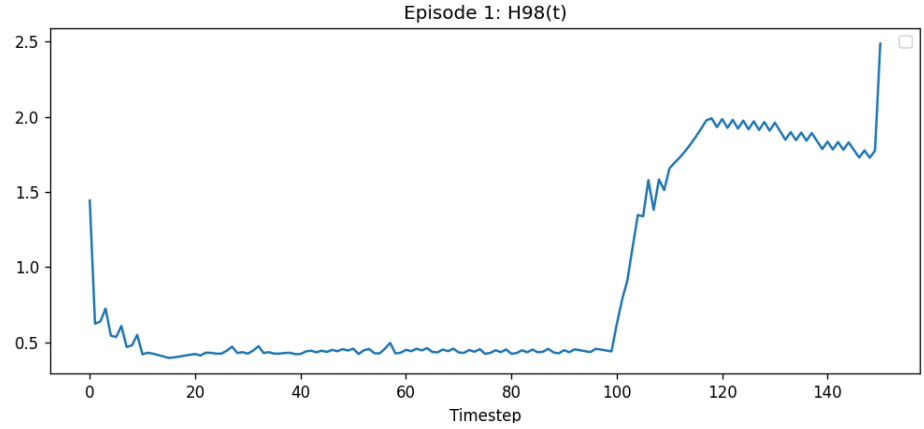
# 6 Risks and Mitigations

Although we have some initial results displaying some success, there are some risks that limit the robustness and generalizability of the current control policy. First, the environment initialization is fully deterministic, and no diagnostic noise, actuator jitter, or measurement delay is incorporated into the observation stream. This is a major risk, as we essentially are initializing the TORAX simulator with the same initial condition with each episode. This is why the evaluation results show no deviation between episodes. Real tokamak experiments are dominated by stochastic perturbations: sensor noise, transient fluctuations in plasma parameters, and uncertainties in reconstruction which are not accounted for in our simulator. As a result, it is unclear whether the learned policy would remain stable or performant under more realistic noisy conditions. A simple next step would be to introduce domain randomization into the simulator, such as randomized initial plasma conditions, perturbed actuator responses, and injected measurement noise. This would encourage the policy to learn strategies that remain reliable across a broader range of operational scenarios. Another step would be to introduce measurement noise by adding a Gaussian error term to our observable states. This could potentially help in generalizability, but would also add additional challenge and complexity and help simulate that diagnostic measurements in a real tokamak experiment come with uncertainty in the measurements.

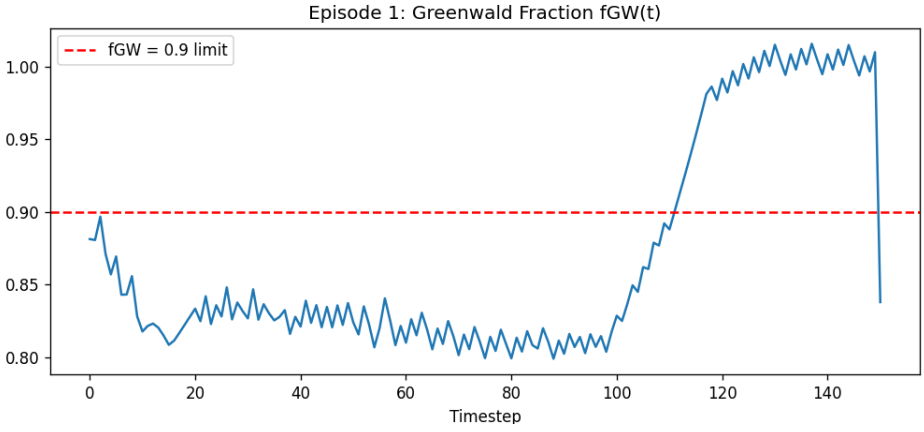
A second risk arises from the reward formula-


Episode 1: Fusion Gain Q(t)
350
300
250
200
150
100
50
0
0
20
40
60
80
100
120
140
Timestep

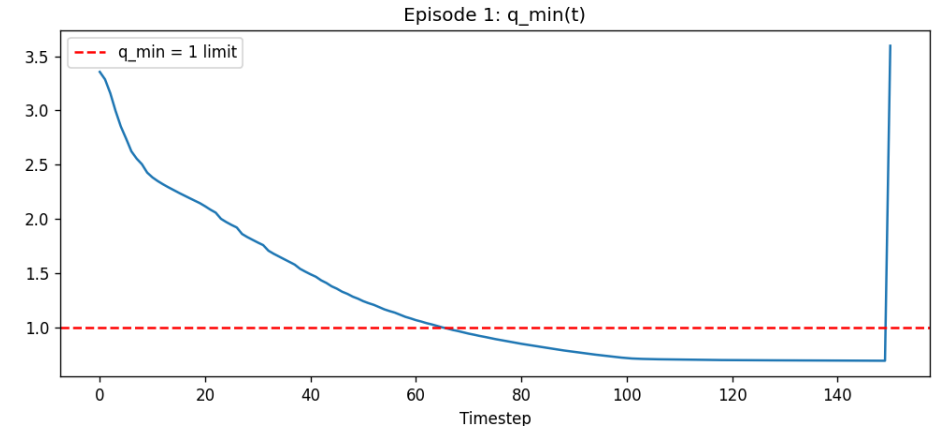
(a)


Episode 1: H98(t)
2.5
2.0
1.5
1.0
0.5
0
20
40
60
80
100
120
140
Timestep

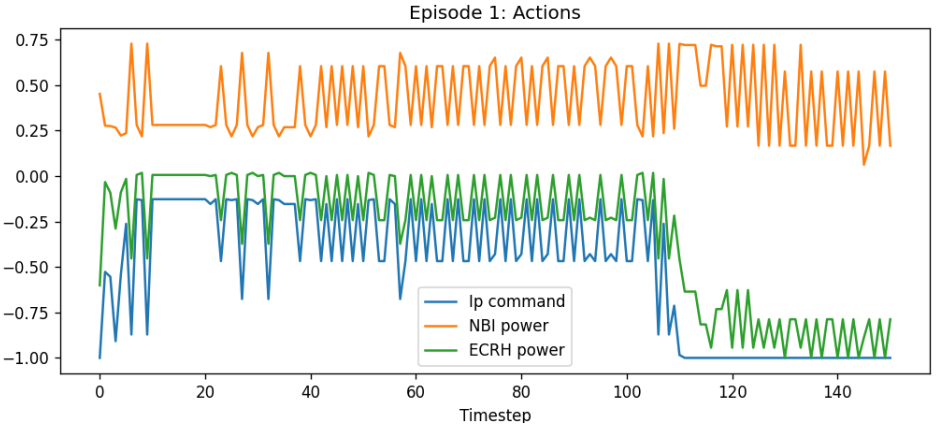
(b)


Episode 1: Greenwald Fraction fGW(t)
fGW = 0.9 limit
1.00
0.95
0.90
0.85
0.80
0
20
40
60
80
100
120
140
Timestep

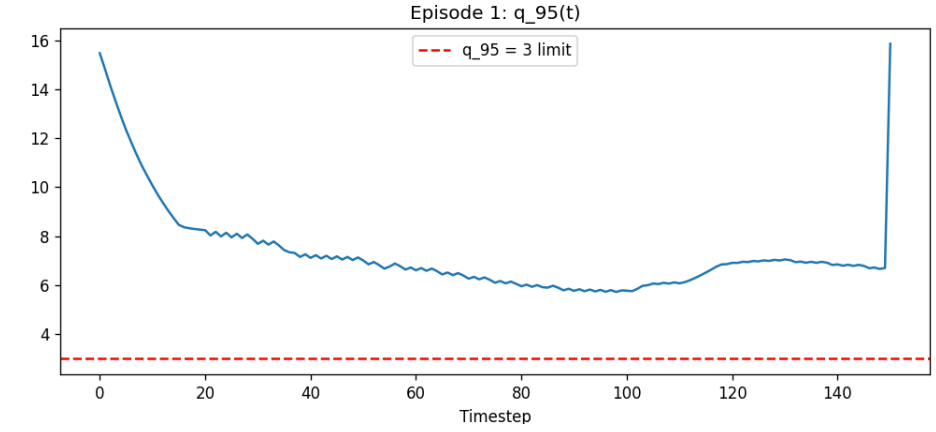
(c)




(a)




(b)




(c)

Figure 4: Performance of our trained RL agent on a simulated evaluation environment. Our agent is able to pick actions that lead to strong control over the net fusion energy gain, although it has some challenges with avoiding disruptive scenarios. Our simulated environment is set up so that the first 100 seconds are ramping up the plasma to achieve fusion conditions, while the last 50 seconds are the actual fusion conditions.

tion. Because the reward function places no explicit cap or diminishing returns on the fusion power term $Q_{fusion}$, the agent overprioritizes maximizing energy output at the expense of avoiding conditions for plasma disruptions. This creates the risk that the agent will exploit regimes that produce high fusion power but cross the disruption thresholds, which we have observed for brief periods in our results. Future work should therefore refine the reward structure to more strongly prioritize disruption avoidance. Some simple initial improvements in reward shaping include capping the maximum reward for $Q_{fusion}$ that the agent can get at each time step (ITER is designed to achieve $Q_{fusion}$ of 10) or adding a negative penalty term to the disruptions to punish the agent for entering disruptive territory.

# References


[1] Degrave, J., Felici, F., Buchli, J., Neunert, M., Tracey, B., Carpanese, F., Ewalds, T., Hafner, R., Abdolmaleki, A., de las Casas, D., Donner, C., Fritz, L., Galperti, C., Huber, A., Keeling, J., Tsimpoukelli, M., Kay, J., Merle, A., Moret, J.-M., & Noury, S. (2022). Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897), 414–419. `https://doi.org/10.1038/s41586-021-04301-9`

[2] Tracey, B. D., Michi, A., Chervonyi, Y., Davies, I., Cosmin Paduraru, Lazic, N., Felici, F., Timo Ewalds, Donner, C., Cristian Galperti, Buchli, J., Neunert, M., Huber, A., Evens, J., Kurylowicz, P., Mankowitz, D. J., & Riedmiller, M. (2024). Towards practical reinforcement learning for tokamak magnetic control. *Fusion Engineering and Design*, 200, 114161–114161. `https://doi.org/10.1016/j.fusengdes.2024.114161`

[3] Char, I., Chung, Y., Abbate, J., Kolemen, E., & Schneider, J. (2024). Full Shot Predictions for the DIII-D Tokamak via Deep Recurrent Networks. *ArXiv (Cornell University)*. `https://doi.org/10.48550/arxiv.2404.12416`

[4] Subbotin, G. F., Sorokin, D. I., Nurgaliev, M. R., Granovskiy, A. A., Kharitonov, I. P., Adishchev, E. V., ... & Orlov, D. M. (2025). Reconstruction-free magnetic control of DIII-D plasma with deep reinforcement learning. *arXiv preprint arXiv:2506.13267*.

[5] Citrin, J., Goodfellow, I., Raju, A., Chen, J., Degrave, J., Donner, C., ... & Kohli, P. (2024). TORAX: A fast and differentiable tokamak transport simulator in JAX. *arXiv preprint arXiv:2406.06718*.

[6] Mouchamps, A., Malherbe, A., Bolland, A., & Ernst, D. (2025). Gym-TORAX: Open-source software for integrating RL with plasma control simulators. *arXiv preprint arXiv:2510.11283*.