Plasmax: DIFFERENTIABLE & PARALLELIZABLE ENVIRONMENTS FOR TOKAMAK CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Controlling unstable plasma in tokamaks to improve nuclear fusion performance remains an open problem, and any progress in this area brings us closer to efficient nuclear fusion power for carbon-free energy. While Reinforcement Learning (RL) has great potential to tackle such complex dynamical control problems, research on its application to fusion remains sparse. One significant barrier is the lack of fast, configurable RL environments tailored to fusion applications. In this work, we introduce *Plasmax*: 17 differentiable, parallelizable environments for tokamak kinetic control with 5 physics models for dynamics simulation. Plasmax enables benchmarking RL methods against other open- and closed-loop strategies, evaluating robustness to real-world-inspired observation corruption (e.g., noise or delays), and studying transfer between simulators with different fidelities. We show that Plasmax is significantly faster than prior work, enabling up to a $1103\times$ speedup and training an RL agent in 7 minutes on hardware accelerators. We benchmark a wide array of trajectory and policy optimization methods, including RL, Evolution Strategies (ES), and differentiating through the environment. We find that open-loop control optimized with CMA-ES performs 32% better than the best RL method we evaluate (SAC). We also investigate how trained policies generalize to different backends, how real-world-inspired observation corruptions affect training, and how exploration can be used as a bug-hunting tool to improve simulators. We hope Plasmax will foster collaboration between machine learning and fusion researchers to develop practical fusion energy.

1 INTRODUCTION

Controlling higher-energy, but more unstable plasma effectively is a key route for improving nuclear fusion performance in tokamaks (Sorbom et al., 2015; Humphreys et al., 2020; Seo et al., 2024). Researchers commonly develop tokamak control strategies through *trajectory optimization* (Mitchell et al., 2023; Van Mulders et al., 2023). Researchers then deploy these trajectories as an *open-loop* control strategy for low-level controllers to track. However, higher-pressure plasmas provide more power but can cause significant, hard-to-predict instabilities (Seo et al., 2024). Open-loop trajectories are fixed in advance, so they cannot adapt to disturbances, model mismatch, or deviations from the predicted plasma evolution (Wang et al., 2025). Instead, *closed-loop*, dynamic control strategies adapt to the current plasma state and react online to such deviations (Barton et al., 2013; Wang et al., 2025).

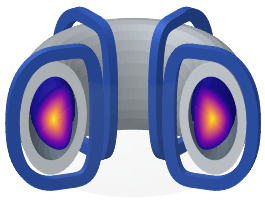


Figure 1: Simplified schematic of a tokamak. Plasma profile in yellow-purple.

Reinforcement Learning (RL) is a framework for learning closed-loop control strategies, known as *policies*, through interaction with an *environment* or from datasets of logged environment interactions. RL has shown early signs of success on real tokamaks (Degrave et al., 2022; Seo et al., 2021; 2024). Unlike robotics, which has standard benchmarks (Brockman et al., 2016; Tunyasuvunakool et al., 2020; Freeman et al., 2021), progress in learning-based control for tokamaks is slowed by a lack of unified evaluations, making results and methods hard to compare. In particular, plasma simulators either only accept fixed actuator trajectories (Citrin et al., 2024) or are developed using programming

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

languages that are less common among machine learning researchers, such as MATLAB or Fortran (Felici et al., 2018; Heiß et al., 2025; Militello Asp et al., 2022; Citrin et al., 2024). Therefore, learning or designing dynamic controllers requires cross-domain knowledge of control engineering, nuclear fusion physics, and programming within large codebases, which hinders research.

TORAX (Citrin et al., 2024) is a state-of-the-art JAX-based plasma simulation toolkit for tokamak researchers. It is designed primarily for physicists to couple physics models and study open-loop control for tokamaks. Configuring a TORAX simulation therefore requires translating a researcher's objectives into concrete parameter choices, model assumptions, and simulation settings, a task that remains challenging for current agentic systems. Additionally, TORAX does not natively support dynamic control, as the actuator trajectory is part of the configuration that is integrated forward.

In this work, we introduce *Plasmax*, an open-source Python environment interface for open- and closed-loop tokamak control. Plasmax enables dynamic and differentiable control of tokamak shots with a Gym-style interface (Brockman et al., 2016). We define different tokamak control tasks, each with an associated reward function, as distinct *environments*. Moreover, inspired by the sim-to-real problem in robotics (Peng et al., 2018), we design 5 *backends* that use different simulator models to test zero-shot generalization (Kirk et al., 2023), with the aim of deploying policies on a real tokamak. Plasmax also includes environment wrappers intended to help training or improve realism (Dulac-Arnold et al., 2019), including missing, partial, and noisy observations. Finally, we include operational constraints informed by real-world tokamak devices, such as termination conditions and actuator limits.

We show that Plasmax is much faster than comparable prior work, with $24\times$ to $1103\times$ speedup depending on hardware and parallelization. We benchmark all 17 Plasmax environments with six methods: 2 open-loop and 4 closed-loop methods, including 2 RL methods. Our experiments show that open-loop optimization with CMA-ES performs best for both in-distribution and zero-shot generalization; we highlight optimization issues that prevent closed-loop methods from performing as well as open-loop, and we encourage the community to develop novel bespoke algorithms for tokamak control. Finally, we show how policy optimization can uncover bugs and issues in the underlying simulator, and we demonstrate this for TORAX. Our contributions are fivefold:

1. We release Plasmax[1], an open-source, Gym-style environment interface for tokamak control with 17 environments across 4 distinct tokamaks with 5 pre-configured simulator backends that aim to reproduce real tokamak firing conditions.
2. We show that Plasmax is parallelizable and can be compiled end-to-end, enabling $24\times$ to $1103\times$ speedups over prior work and training an RL policy in 7 minutes on an H100 GPU.
3. We run comprehensive evaluations with six control and RL algorithms in the 17 current environments, showing that an open-loop trajectory optimized with CMA-ES performs best in 11/17 environments.
4. Next, we show that Plasmax can serve as a benchmark for *zero-shot* generalization, when treating the different simulator backends as different *contexts*, facilitating potential transfer to a real tokamak.
5. Finally, we highlight how exploration during learning can be used to find bugs and therefore improve underlying simulators.

2 Background

2.1 A Quick Primer on Fusion in Tokamaks

Nuclear fusion is a process by which light atomic nuclei combine to create energy, but achieving enough fusion to make useful amounts of energy requires high temperatures (150M Kelvin), at which the fuel forms a *plasma*: an ionized gas of electrons and ions (Campbell, 2007; Wesson, 2011). Tokamaks (Figure 1) use magnetic fields to confine this plasma in a torus-shaped chamber. Within the plasma, temperature, density, and electrical current vary; Figure 2 shows a temperature *profile* (2D slice) of the ITER tokamak. Different actuators can modify these profiles, but heat and particles naturally redistribute throughout the profile in a process known as *transport*. Turbulent transport is particularly difficult to predict, motivating simulators with different physical approximations.

[1] https://anonymous.4open.science/r/plasmax-0F41/README.md

The performance of a tokamak plasma depends jointly on temperature, density, and how effectively the plasma retains its energy (Wurzel & Hsu, 2022). A tokamak shot progresses through *ramp-up, flat-top, and ramp-down*. These phases require, respectively, establishing suitable operating conditions, sustaining performance, and safely shutting down.

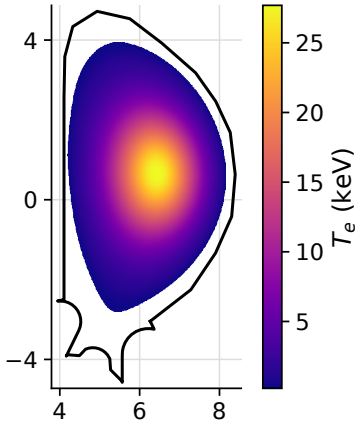


Figure 2: Sample temperature profile of the ITER tokamak.

2.2 Plasma Simulators

Plasma *transport* simulators such as RAPTOR (Felici et al., 2011; 2018), ASTRA (Tardini et al., 2026) or JINTRAC (Militello Asp et al., 2022) evolve temperature and density profiles using turbulent transport models such as QuaLiKiz (Bourdelle et al., 2007) and a fixed boundary for the plasma geometry. This contrasts with *free-boundary Grad-Shafranov*-based solvers, such as FGE (Heiß et al., 2025) or FreeGSNKE (Amorisco et al., 2024) that explicitly solve for the plasma geometry. In this work, we use FreeGSNKE to generate pre-computed geometries, and then interpolate between them to evolve the state with different transport simulators.

2.3 Control in Tokamaks

The control task can be broadly separated into two sub-problems, often viewed independently (Lister et al., 2006). In *magnetic control*, magnetic coils control the *geometry* (or informally, the shape) of the plasma by inducing a magnetic field. Plasma geometry affects heat retention, stability, and the distribution of heat on the plasma-facing wall, making shape control important for both plasma performance and protection of the device (Degrave et al., 2022; Tracey et al., 2024). Complementarily, *kinetic control* refers to controlling the temperature, density, and other state variables of the plasma for a given geometry. These variables evolve relatively slowly, on the order of 0.1-1s (Abbate et al., 2023), and control is generally operated on a similar scale (Humphreys et al., 2015). In this work, we focus on kinetic control.

2.4 TORAX

TORAX (Citrin et al., 2024) is a plasma transport simulation suite written in JAX (Bradbury et al., 2018), with native support for hardware accelerators, JIT compilation, parallelization, and automatic differentiation. It exposes a broad selection of physics/surrogate models and solvers, leaving their configuration to the user's domain expertise. As a simulator, TORAX takes a specified configuration, including a fixed actuator trajectory, and integrates it forward. As such, actuator values are predetermined in an open-loop fashion that cannot be altered during the episode.

Plasmax provides: (i) an adapter that suspends the solver at controller decision points and resumes it with updated actuator values; (ii) preserved gradient propagation through this adapter, (iii) default differentiable simulator configurations that mimic real-world operation, which we name *backends*, (iv) a suite of pre-defined environments, including observation and action spaces, reward functions, initial-state distributions, geometries, and termination criteria; and (v) a Gym-style interface, with environment wrappers so that standard RL algorithms apply without modification. The relationship between TORAX and Plasmax is like that between the physics engine `MuJoCo` (Todorov et al., 2012) and the control suite `Brax` (Freeman et al., 2021): in Brax, environments can use different physics-simulation backends, including MuJoCo. The current version of Plasmax mostly relies on TORAX to simulate the environments' physics. Nevertheless, we design the interface to be modular and compatible with any JAX-based simulator; to demonstrate this, we include a pre-trained world model environment trained on the KSTAR tokamak (Seo et al., 2021; Gao et al., 2025). We plan to include more underlying simulators in future versions to enable controllers that are robust to different simulator implementations and assumptions.

3 Related Work

3.1 Reinforcement Learning for Tokamak Control

Degrave et al. (2022) train a policy using Maximum-a-Posteriori Policy Optimization (Abdolmaleki et al., 2018a;b, MPO) on the FGE simulator (Carpanese, 2021; Heiß et al., 2025) configured for the Tokamak à Configuration Variable (TCV). Nevertheless, the simulator's computational cost makes training expensive: follow-up work by Tracey et al. (2024) reports about two seconds per simulation step on a single CPU core during training, with policy training taking days to weeks. In contrast, Plasmax prioritizes speed and parallelization on hardware accelerators, achieving speeds of up to 193k steps per second on an H100 GPU. This enables RL agent training in minutes on hardware accelerators, making research more widely accessible. Furthermore, these works address *geometric* control, whereas Plasmax addresses *kinetic* control.

Seo et al. (2021; 2022) first apply RL to tokamak kinetic control on the KSTAR tokamak. They train a world model on five years of data. They then use this world model to train an RL policy with TD3 (Fujimoto et al., 2018). They use the policy to generate actuator trajectories that are deployed on the real KSTAR tokamak. As expected, when the policy drifts outside the training-data regime, it can no longer control instabilities effectively. Gao et al. (2025) use this world model as one of their real-world benchmark environments, which we include in the environment suite. Seo et al. (2023) develop a learned model that predicts tearing instabilities for the DIII-D tokamak. They then use this model to train an RL policy that learns to avoid tearing instabilities (Seo et al., 2024) and successfully deploy this on the DIII-D tokamak. Unlike these works, Plasmax is not restricted to machine-specific datasets, though each backend is only physically plausible or numerically stable within certain state boundaries.

3.2 Tokamak RL Environment Interfaces

Wang et al. (2025) develop *PopDownGym*, an environment interface for *ramping down* tokamak shots. PopDownGym is a hybrid physics/machine-learning model trained on simulations. In contrast, Plasmax introduces environments for up to three phases of tokamak operation (ramp-up, flat-top, ramp-down) across four different tokamaks and up to three operational scenarios. Notably, the authors randomize the simulation physics parameters at agent training time for *Domain Randomization* (Tobin et al., 2017; Peng et al., 2018), an approach we draw inspiration from in this work (see Section 4).

Mouchamps et al. (2026) design Gym-TORAX, a fusion control environment using TORAX (Citrin et al., 2024) as a physics backend. The authors implement a single environment and physics backend: a hybrid scenario on the ITER tokamak with a QLKNN-based simulator. Plasmax integrates four tokamaks, with up to three scenarios, three tasks, and five backends, for 63 backend-environment combinations, with many planned additions. Additionally, Plasmax is differentiable and obtains $24\times$ to $1103\times$ speedups over Gym-TORAX in steps per second, as shown in Section 5.1. Finally, Plasmax focuses on real-world RL algorithm development, following recommendations from Dulac-Arnold et al. (2019) for environment design, such as delayed, partial, and noisy observations, implemented through a comprehensive *wrapper* ecosystem.

4 Environment Design

Backends Through a flexible API design, Plasmax supports different dynamics for stepping through the environments, which we call *backends*. We use TORAX to design different configurations to mimic real-world tokamak operation. The TORAX-based backends mostly modify the *transport model* and require forward integration with a solver. The five backends we include are:

- *CGM*: Critical Gradient Model, which relies on a simplified theory-based analytic transport model (Guo & Romanelli, 1993).
- *Bohm-GyroBohm*: relies on a semi-empirical analytic model, which combines Bohm and gyro-Bohm transport scalings with empirical coefficients (Tholerus et al., 2024). A variant is available for spherical tokamaks.
- *QLKNN*, which uses a 73k-parameter neural network transport model trained with outputs from the QuaLiKiz transport model (Citrin et al., 2024; 2017; Plassche et al., 2020).

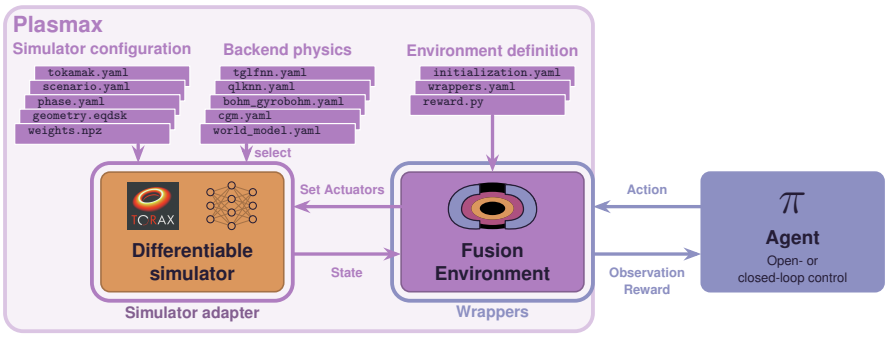


Figure 3: Plasmax supports the development of both open- and closed-loop controllers for tokamak kinetic control. JAX-native interface enables compilation, automatic differentiation, parallelization, and GPU support.

- *TGLFNN* uses a 15M-parameter neural network transport model trained on the TGLF transport model (Zanisi et al., 2025; Staebler et al., 2007). The TGLFNN backend comes in three variants: one using a linear solver, another using a more expensive Newton-Raphson solver (TGLFNN-NR), and a third variant for spherical tokamaks.
- *World Model*: a world model as an environment surrogate; it does not use a solver like the other backends. Currently, it only supports the 5.5M-parameter LSTM world model from Seo et al. (2021); Gao et al. (2025).

We highlight the trade-offs of these backends in Section 5.1. We also implement a physics randomization wrapper that randomizes a subset of the simulator's physics parameters at each step, enabling more robust learning (Tobin et al., 2017).

Observation Space The underlying simulator states (TORAX or world model) contain a mixture of scalar and *profile* variables, including temperature, density, electric current, and derived values such as the *safety factor*. Certain values are unobtainable during real-time control operation; we therefore remove these variables from the observation. Additionally, we downsample the observed profiles to reflect the limited data acquisition in a real tokamak. We provide default settings based on work from the ITER tokamak team (Mazon et al., 2025). These changes are configurable through the wrapper API, which we detail in Section A.4. We show how ablating these wrappers affects agent training in Section 5.4.

Action Space We use up to four actuators for the TORAX-based environments:

- A fuelling channel, which increases the plasma density.
- A heating channel, which increases the temperature of the plasma.
- A localized energy deposition, which increases the plasma current at configurable locations.
- The location of the energy deposition channel defined above.

We use the values from Van Mulders et al. (2021) to set bounds on the actuators. We also impose a maximum change the agent can apply to the actuator, reflecting real-world actuators that cannot change values instantaneously.

Reward functions We provide default hand-crafted reward functions for each environment based on prior works and domain knowledge. Additionally, we also penalize crossing common operational constraints such as the *minimum safety factor* $q_{\min} > 0.8$ (Park & Monticello, 1990; Paruchuri et al., 2024) or the *Greenwald Density Limit Fraction* $f_{GW} < 1$ (Greenwald et al., 1988; Greenwald, 2002). Crucially, we also penalize the agent for encountering numerically unstable states to encourage it to stay within the simulator's domain of validity. To enable the penalty to be differentiable, we first enforce all rewards to be positive through a *square-plus* transformation and then use a *logarithm* transformation when the agents cross a constraint or encounter a numerically unstable state. While these default reward functions are sufficient, we do not claim they shape optimal behavior: reward design remains an open question. We detail all default reward functions in Section A.3.

5 EXPERIMENTS

5.1 IMPLEMENTATION PERFORMANCE

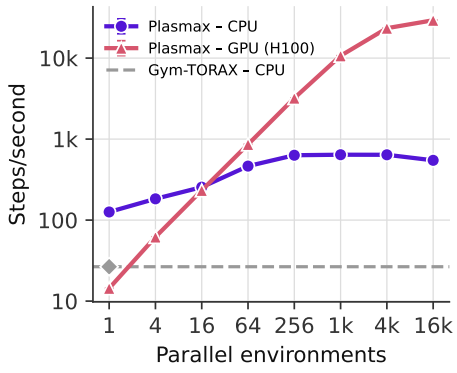


Figure 4: Step throughput comparison of Plasmax and Gym-TORAX on an equivalent backend (QLKNN). Compilation and parallelization enable large speedups on GPUs.

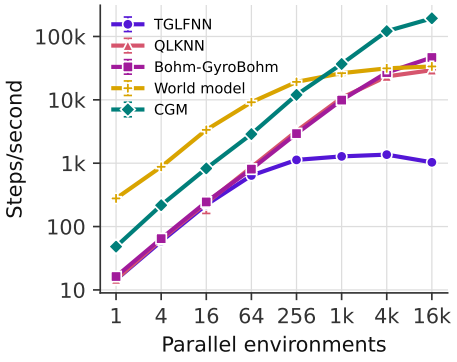


Figure 5: Speed in steps per second of different backends on an H100 GPU. The CGM backend peaks at 193k sps.

Figure 4 shows that, thanks to the JAX-native implementation, step throughput increases with parallelization (up to 16k parallel environments). This contrasts sharply with Gym-TORAX (Mouchamps et al., 2026), where no parallelization (or differentiation) is possible. Additionally, our implementation can be compiled end-to-end, achieving $4.7\times$ more steps/second compared to Gym-TORAX for a single environment; this extends to $24\times$ for 256 parallel environments on CPU and $1103\times$ for 16k parallel environments on GPU. In Section C.1, we measure speed when backpropagating gradients through the environment, achieving up to 3025 steps/second. In Figure 5, we see how backends scale linearly with parallelization up to a different number of parallel environments. TGLFNN is the most expensive backend, and parallelization effectively plateaus beyond 256 parallel environments, as the GPU memory saturates, peaking at 1365 steps/second. In contrast, CGM peaks at 193k steps/second. The world model backend does not rely on a numerical solver, which helps explain its speed at low parallelization; at high parallelization, its speed plateaus, similarly to TGLFNN. QLKNN uses a much smaller neural network surrogate, which explains why the speed plateau is less pronounced. Detailed numbers and RL policy training times are in Section C.1.

Figure 6 shows a trade-off between speed and agreement across simulator backends. We use the TGLFNN backend with a Newton–Raphson (NR) solver as the reference backend (TGLFNN-NR). Replacing the NR solver with a linear one (TGLFNN-linear) yields a substantial speedup with little change in the resulting profiles; we call this TGLFNN from here on out. At higher throughput, Bohm–GyroBohm provides the best observed compromise between speed and agreement relative to TGLFNN-NR and, on this metric, outperforms both QLKNN and CGM, making this an attractive backend for training. We therefore run the main experiments in Section 5.2 with the Bohm-GyroBohm backend. The disagreement among similarly fast backends highlights how differences in modeling assumptions lead to different state distributions. We show how the state distributions of different backends and environments vary in Section C.2.

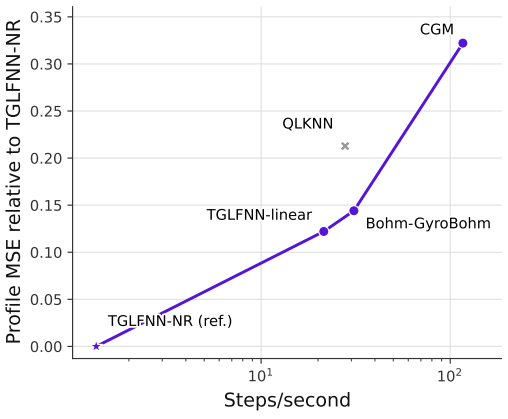


Figure 6: Pareto frontier of agreement with a reference backend versus speed for different backends. See Section C.1 for details.

5.2 CONTROL PERFORMANCE

All backends in Plasmax are specifically designed to be *automatically differentiable* (Baydin et al., 2018; Newbury et al., 2024). Because the reward is a function of state and action, we can backpropagate through the environment to learn a policy or actuator values directly. Additionally, we can obtain approximate gradients through Evolution Strategies (ES), which estimate gradients by aggregating results from a population of perturbed parameters. These approximate gradients have been argued to be more robust than backpropagating through simulators directly (Metz et al., 2022). In contrast, RL does not require a differentiable environment to obtain gradients. Despite this, RL is known to have high variance in performance (Agarwal et al., 2021) and high sensitivity to hyperparameters (Eimer et al., 2023).

We compare control performance between closed-loop and open-loop controllers. To learn open-loop trajectories, we treat actuator values as learnable parameters. We compare this to closed-loop control methods such as learning an adaptive controller with and without RL. We use the following methods:

- PPO: An on-policy RL method (Schulman et al., 2017; Liesen et al., 2024).
- SAC: An off-policy RL method (Haarnoja et al., 2018; Liesen et al., 2024).
- Closed-loop (Backprop): Optimizing the parameters of a feedback policy by back-propagating through the simulator.
- Open-Loop (Backprop): Optimizing actuator values by backpropagating through the simulator. We use 10 control knots, with actuator values interpolated between them.
- Closed-loop (CMA-ES): Optimizing the parameters of a feedback policy using CMA-ES (Hansen & Ostermeier, 2001; Lange, 2022).
- Open-Loop (CMA-ES): Optimizing a predetermined actuator trajectory (10 knots) using CMA-ES.

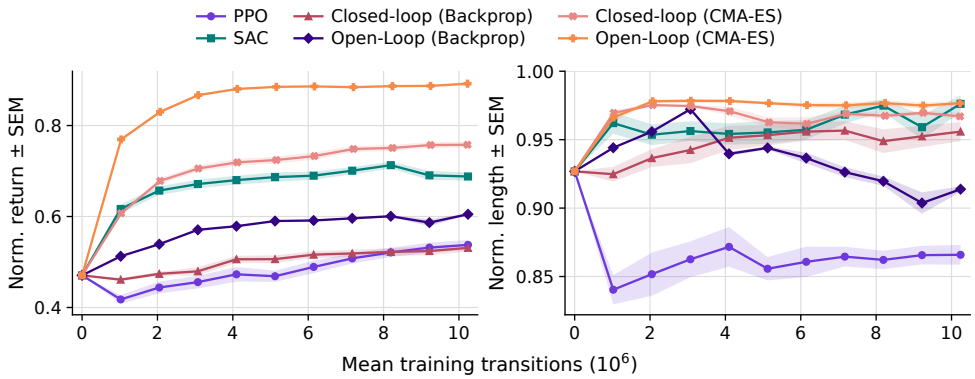


Figure 7: Open-loop trajectories trained with CMA-ES outperform the other methods on average. Normalized mean and episode length across all 17 environments with SEM. Returns are normalized between 0 and 1 using the worst and best return across all algorithms during training. We normalize episode length by the maximum episode length. We trained each algorithm with 10 seeds per environment. The individual environment results are in Section C.2.

We show normalized results across all 17 environments in Figure 7; detailed results are in Section C.2. Open-Loop (CMA-ES) stands out as the best method across environments, achieving the highest returns in 11/17 environments and the highest episode lengths in 13/17 environments; it outperforms the next best method, Closed-loop (CMA-ES), by 17%, and the third, SAC, by 32%. We attribute this difference in performance simply to the much easier optimization problem (40 versus 6000 learnable parameters); it can be seen in certain environments such as ITER hybrid ramp-down that Open-Loop (CMA-ES)'s return increases much faster than Closed-loop (CMA-ES)'s. PPO achieves the shortest episodes, but this mean is driven down by learning instabilities in two ITER environments. For computational cost reasons, we used untuned configurations for the main comparison, but tuning results in Section C.3 suggest that tuning can significantly improve PPO and SAC performance, consistent with prior RL work (Engstrom et al., 2020). We note that Backprop methods produce high-variance gradients, with variances up to 10^{16} in certain runs, which could explain the performance gap

relative to CMA-ES methods. Pascanu et al. (2013) previously analyzed this phenomenon and showed that repeated backpropagation through the same function can lead to chaotic gradient divergence if its Jacobian has eigenvalues with magnitudes greater than one. Metz et al. (2022) showed that ES gradients do not suffer from this flaw. We find evidence of Jacobian eigenvalue magnitudes above 1 under certain conditions, as shown in Section C.4; this does not confirm that high gradient variance is due to this, but is a possible explanation.

5.3 Benchmarking Zero-Shot Generalization

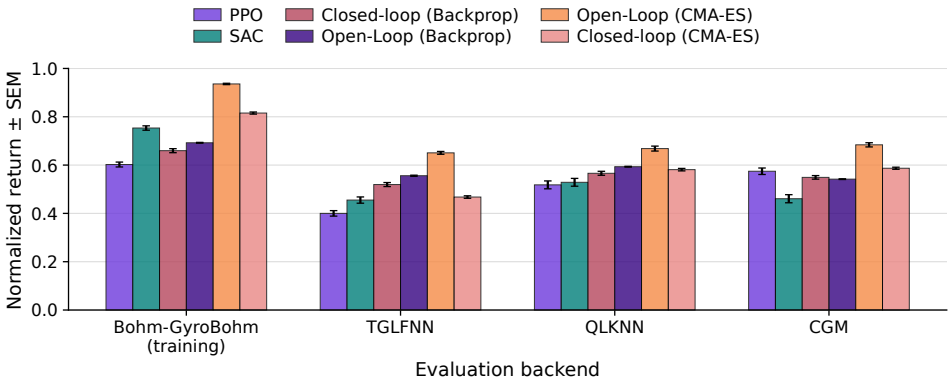


Figure 8: Open-loop (CMA-ES) performs best when evaluating on unseen backends. Bohm-GyroBohm (BgB) is the training backend for all policies. Normalized returns between 0 and 1 using the worst and best policies seen during evaluation per environment across backends and algorithms, averaged across environments.

Following Hallak et al. (2015), we treat the different environments as *Contextual Markov Decision Processes* and the different backends as *contexts*. We use this to define Zero-Shot Generalization (Kirk et al., 2023, ZSG) in Plasmax as training on one or more backends and evaluating on other, unseen backends. We evaluate the policies trained in Section 5.2 on all other backends for all environments (except KSTAR, which has only one backend); the full per-environment results are in Section D.

We see in Figure 8 that Open-Loop (CMA-ES) performs the best on all backends by a large margin. For TGLFNN, the most expensive backend, Open-Loop (Backprop) is the second-best method, suggesting that our open-loop methods are more robust than our closed-loop implementations in this setting.

5.4 Further Analysis

In Figure 9, we see that cumulatively adding realism wrappers first improves performance relative to the base environment (with physics randomization and observation noise added), but downsampling the observation reduces performance significantly, and delaying observations maintains performance; the individual environment results for this plot are in Section C.2.

In Figure 10, we train PPO on different backends and then evaluate the policies on the other backends. In this experiment, QLKNN policies perform within error of the best performances across all other backends except Bohm-GyroBohm. Additionally, training on the CGM backend and evaluating on Bohm-GyroBohm matches, within error, training directly on Bohm-GyroBohm. Both results suggest the backends are sufficiently similar that certain policies could generalize to *all* backends.

5.5 Policy Optimization as a Bug-Finding Tool

Policy optimization exposes simulators to a wide variety of actuator sequences and plasma states, creating opportunities to identify numerical or physically implausible state issues that would otherwise be unlikely to occur. This gives exploration a useful diagnostic role alongside its role in learning effective control strategies. Using RL exploration to find issues and subsequently improve simulators has shown promise, e.g., Güzel et al. (2026).

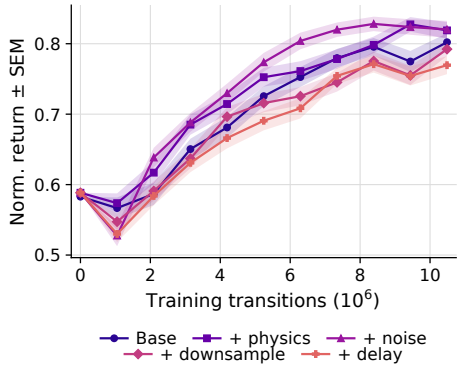


Figure 9: Cumulatively adding wrappers inspired by Dulac-Arnold et al. (2019) modifies PPO performance. Normalized mean and SEM across 16 environments, with 15 training seeds. Returns are normalized per environment between 0 and 1 using the worst and best return across all wrapper configurations during training.

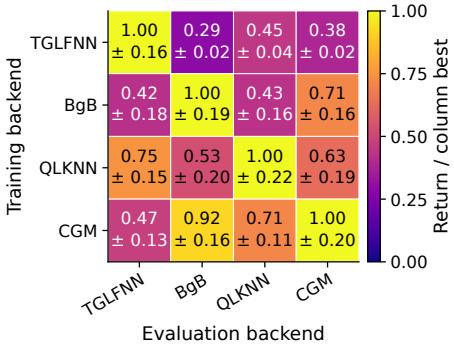


Figure 10: PPO learns policies that generalize to other unseen backends on `ITER/hybrid/rampup`. Mean and SEM over 5 training seeds per backend and 128 evaluation seeds. Return is normalized per column; raw returns in Section C.2.

During the development and benchmarking of Plasmax, we identified issues affecting numerical stability, profile interpolation, and gradient computation across different backend configurations. These manifested as solver failures, unexpected terminations, or non-finite gradients during policy optimization. Investigating these cases helped us find important issues in the underlying TORAX code. We provide technical details of these issues in Section C.5.

6 Conclusion

In this work, we introduced *Plasmax*, an open-source set of parallelizable and differentiable environments for tokamak control. We showed that Plasmax is faster than prior work through parallelization and end-to-end compilation with JAX, while remaining differentiable. Using Plasmax, we benchmarked various learned controllers and showed that open-loop controllers trained with CMA-ES outperform closed-loop controllers in both in-distribution performance and zero-shot generalization. Finally, we highlighted how exploration and benchmarking helped us find issues in the underlying simulator and therefore improve it.

Our evaluation of control performance in different tasks is not exhaustive. Plasmax includes 63 possible task-backend combinations, plus hundreds of configurable parameters, which we could not fully explore due to limited computational resources. While we researched existing tokamak references to set realistic environment parameters, we did not change many default parameters for which we could not find references. Although Plasmax aims to provide realistic observations, the wrappers do not model the physical processes that produce imperfect observations in practice and instead use simple approximations (e.g., Gaussian noise, Bernoulli dropout). Plasmax also inherits several limitations from its simulator backend, TORAX, including solver and backpropagation instabilities, as well as realism limitations, since TORAX does not model all relevant physical phenomena, especially magnetohydrodynamic instabilities.

We believe Plasmax provides an important and necessary interface for tackling plasma control with RL methods. It provides ready-to-use control tasks across a range of tokamaks, operating scenarios, and simulation backends. We hope it encourages RL researchers to engage with plasma control without requiring extensive domain-specific infrastructure, while also making it easier for plasma researchers to evaluate and develop modern learning-based control methods.

AI USE STATEMENT

In this work, we used generative AI tools to propose or refine hypotheses, design or provide feedback on research methodology or experiments, interpret results, and implement methods. We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, or assist in the writing of proofs. Additionally, we used generative AI tools to create or modify scientific figures or images, suggest experimental parameters, create or edit software code, create artifacts, summarize or analyze existing literature, discover research topics or identify gaps, brainstorm, sourcing/searching for information. We have reviewed all AI-assisted work, including the literature, the algorithm implementations, and the backends' parameters. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Joseph Abbate, Rory Conlin, Ricardo Shousha, Keith Erickson, and Egemen Kolemen. A general infrastructure for data-driven control design and implementation in tokamaks. *Journal of Plasma Physics*, 89:895890102, February 2023. ISSN 0022-3778. doi: 10.1017/S0022377822001040. URL https://ui.adsabs.harvard.edu/abs/2023JPlPh..89a8902A. ADS Bibcode: 2023JPlPh..89a8902A.

Abbas Abdolmaleki, Jost Tobias Springenberg, Jonas Degrave, Steven Bohez, Yuval Tassa, Dan Belov, Nicolas Heess, and Martin Riedmiller. Relative Entropy Regularized Policy Iteration, December 2018a. URL http://arxiv.org/abs/1812.02256. arXiv:1812.02256 [cs.LG].

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a Posteriori Policy Optimisation, June 2018b. URL http://arxiv.org/abs/1806.06920. arXiv:1806.06920 [cs].

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29304–29320, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f514cec81cb148559cf475e7426eed5e-Abstract.html.

N. C. Amorisco, A. Agnello, G. Holt, M. Mars, J. Buchanan, and S. Pamela. FreeGSNKE: A Python-based dynamic free-boundary toroidal plasma equilibrium solver. *Physics of Plasmas*, 31 (4):042517, April 2024. ISSN 1070-664X. doi: 10.1063/5.0188467. URL https://doi.org/10.1063/5.0188467.

Justin E. Barton, Wenyu Shi, Karim Besseghir, Jo Lister, Arnold Kritz, Eugenio Schuster, Tim C. Luce, Michael L. Walker, David A. Humphreys, and John R. Ferron. Physics-based control-oriented modeling of the safety factor profile dynamics in high performance tokamak plasmas. In *52nd IEEE Conference on Decision and Control*, pp. 4182–4187, Firenze, December 2013. IEEE. ISBN 978-1-4673-5717-3 978-1-4673-5714-2 978-1-4799-1381-7. doi: 10.1109/CDC.2013.6760531. URL http://ieeexplore.ieee.org/document/6760531/.

Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18(153):1–43, 2018. URL https://www.jmlr.org/papers/v18/17-468.html.

C. Bourdelle, X. Garbet, F. Imbeaux, A. Casati, N. Dubuit, R. Guirlet, and T. Parisot. A new gyrokinetic quasilinear transport model applied to particle transport in tokamak plasmas. *Physics of Plasmas*, 14:112501, November 2007. ISSN 1070-664X. doi: 10.1063/1.2800869. URL https://ui.adsabs.harvard.edu/abs/2007PhPl...14k2501B. ADS Bibcode: 2007PhPl...14k2501B.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Yash Katariya, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs. Software, 2018. URL https://github.com/jax-ml/jax.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

Theodore Brown, Francis Casson, Agnieszka Hudoba, Stephen Marsden, Emmi Tholerus, Thomas Wilson, and The STEP team. Openstep: public data release of the step prototype powerplant scenario spp-001, 2025. URL https://doi.org/10.14468/07jt-s540. Dataset.

D. J. Campbell. ITER physics. Lecture slides, First ITER Summer School, Aix-en-Provence, France, 16–20 July 2007, 2007. URL https://www.iter.org/sites/default/files/education/ITER_Physics_Campbell.pdf.

Francesco Carpanese. *Development of free-boundary equilibrium and transport solvers for simulation and real-time interpretation of tokamak experiments*. Phd thesis, École polytechnique fédérale de Lausanne, Lausanne, Switzerland, 2021. URL https://infoscience.epfl.ch/entities/publication/d5d51ec0-9b9f-4da1-8f82-b3b718408dbf.

J Citrin, C Bourdelle, F J Casson, C Angioni, N Bonanomi, Y Camenen, X Garbet, L Garzotti, T Görler, O Gürcan, F Koechl, F Imbeaux, O Linder, K van de Plassche, P Strand, G Szepesi, and J. E. T. Contributors. Tractable flux-driven temperature, density, and rotation profile evolution with the quasilinear gyrokinetic transport model QuaLiKiz. *Plasma Physics and Controlled Fusion*, 59(12):124005, November 2017. ISSN 0741-3335. doi: 10.1088/1361-6587/aa8aeb. URL https://doi.org/10.1088/1361-6587/aa8aeb.

Jonathan Citrin, Ian Goodfellow, Akhil Raju, Jeremy Chen, Jonas Degrave, Craig Donner, Federico Felici, Philippe Hamel, Andrea Huber, Dmitry Nikulin, David Pfau, Brendan Tracey, Martin Riedmiller, and Pushmeet Kohli. TORAX: A Fast and Differentiable Tokamak Transport Simulator in JAX, December 2024. URL http://arxiv.org/abs/2406.06718. arXiv:2406.06718 [physics].

Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de las Casas, Craig Donner, Leslie Fritz, Cristian Galperti, Andrea Huber, James Keeling, Maria Tsimpoukelli, Jackie Kay, Antoine Merle, Jean-Marc Moret, Seb Noury, Federico Pesamosca, David Pfau, Olivier Sauter, Cristian Sommariva, Stefano Coda, Basil Duval, Ambrogio Fasoli, Pushmeet Kohli, Koray Kavukcuoglu, Demis Hassabis, and Martin Riedmiller. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, February 2022. ISSN 1476-4687. doi: 10.1038/s41586-021-04301-9. URL https://www.nature.com/articles/s41586-021-04301-9.

Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of Real-World Reinforcement Learning, April 2019. URL http://arxiv.org/abs/1904.12901. arXiv:1904.12901 [cs.LG].

Theresa Eimer, Marius Lindauer, and Roberta Raileanu. Hyperparameters in reinforcement learning and how to tune them. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 9104–9149. PMLR, 2023. URL https://proceedings.mlr.press/v202/eimer23a.html.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation Matters in Deep Policy Gradients: A Case Study on PPO and TRPO, May 2020. URL http://arxiv.org/abs/2005.12729. arXiv:2005.12729 [cs.LG].

F. Felici, O. Sauter, S. Coda, B.P. Duval, T.P. Goodman, J-M. Moret, and J.I. Paley. Real-time physics-model-based simulation of the current density profile in tokamak plasmas. *Nuclear Fusion*, 51(8): 083052, August 2011. ISSN 0029-5515, 1741-4326. doi: 10.1088/0029-5515/51/8/083052. URL https://iopscience.iop.org/article/10.1088/0029-5515/51/8/083052.

Federico Felici, J. Citrin, A.A. Teplukhina, J. Redondo, C. Bourdelle, F. Imbeaux, O. Sauter, H. Meyer, T. Eich, M. Beurskens, S. Coda, Antti Hakola, P. Martin, J. Adamek, M. Agostini, D. Aguiam, J. Ahn, Leena Aho-Mantila, R. Akers, R. Albanese, R. Aledda, E. Alessi, S. Allan, D. Alves, R. Ambrosino, L. Amicucci, H. Anand, G. Anastassiou, Y. Andrèbe, C. Angioni, G. Apruzzese,

M. Ariola, H. Arnichand, W. Arter, A. Baciero, M. Barnes, L. Barrera, R. Behn, A. Bencze, J. Bernardo, Juuso Karhunen, Aki Lahtinen, Y.Q. Liu, Antti Salmi, Tuomas Tala, Markus Airila, Aaro Järvinen, H.T. Kim, H.S. Kim, Seppo Koivuranta, Jari Likonen, Y.Q. Liu, T. Makkonen, H. Nordman, M.I.K. Santala, Paula Sirén, JET Contributors, and EUROfusion MST1 Team. Real-time-capable prediction of temperature and density profiles in a tokamak using RAPTOR and a first-principle-based transport model. *Nuclear Fusion*, 58(9), July 2018. ISSN 0029-5515. doi: 10.1088/1741-4326/aac8f0. URL https://www.scopus.com/pages/publications/85051239087.

C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax: A differentiable physics engine for large scale rigid body simulation. Software, 2021. URL https://github.com/google/brax.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing Function Approximation Error in Actor-Critic Methods, October 2018. URL http://arxiv.org/abs/1802.09477. arXiv:1802.09477 [cs.AI].

Songyi Gao, Zuolin Tu, Rong-Jun Qin, Yi-Hao Sun, Xiong-Hui Chen, and Yang Yu. NeoRL-2: Near Real-World Benchmarks for Offline Reinforcement Learning with Extended Realistic Scenarios, March 2025. URL http://arxiv.org/abs/2503.19267. arXiv:2503.19267 [cs.LG].

M. Greenwald, J.L. Terry, S.M. Wolfe, S. Ejima, M.G. Bell, S.M. Kaye, and G.H. Neilson. A new look at density limits in tokamaks. *Nuclear Fusion*, 28(12):2199, December 1988. ISSN 0029-5515. doi: 10.1088/0029-5515/28/12/009. URL https://doi.org/10.1088/0029-5515/28/12/009.

Martin Greenwald. Density limits in toroidal plasmas. *Plasma Physics and Controlled Fusion*, 44 (8):R27–R53, August 2002. ISSN 07413335. doi: 10.1088/0741-3335/44/8/201. URL https://iopscience.iop.org/article/10.1088/0741-3335/44/8/201.

S. C. Guo and F. Romanelli. The linear threshold of the ion-temperature-gradient-driven mode. *Physics of Fluids B: Plasma Physics*, 5(2):520–533, February 1993. ISSN 0899-8221. doi: 10.1063/1.860537. URL https://doi.org/10.1063/1.860537.

Ahmet H. Güzel, Jenny Seidenschwarz, Benjamin Graham, Jonathan Sadeghi, Jeffrey Hawke, and Ilija Bogunovic. PROWL: Prioritized regret-driven optimization for world model learning. arXiv preprint arXiv:2605.18803, 2026. URL https://arxiv.org/abs/2605.18803.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor, August 2018. URL http://arxiv.org/abs/1801.01290. arXiv:1801.01290 [cs.LG].

Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual Markov Decision Processes, February 2015. URL http://arxiv.org/abs/1502.02259. arXiv:1502.02259 [stat.ML].

Nikolaus Hansen and Andreas Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, June 2001. ISSN 1063-6560. doi: 10.1162/106365601750190398. URL https://doi.org/10.1162/106365601750190398.

Cosmas Heiß, Antoine Merle, Francesco Carpanese, Federico Felici, Craig Donner, Stefano Marchioni, Alessandro Mari, and Olivier Sauter. FGE: A Fast Free-Boundary Grad-Shafranov Evolutive Solver, December 2025. URL http://arxiv.org/abs/2512.06847. arXiv:2512.06847 [physics].

D. Humphreys, G. Ambrosino, P. de Vries, F. Felici, S. H. Kim, G. Jackson, A. Kallenbach, E. Kolemen, J. Lister, D. Moreau, A. Pironti, G. Raupp, O. Sauter, E. Schuster, J. Snipes, W. Treutterer, M. Walker, A. Welander, A. Winter, and L. Zabeo. Novel aspects of plasma control in ITER. *Physics of Plasmas*, 22(2):021806, February 2015. ISSN 1070-664X. doi: 10.1063/1.4907901. URL https://doi.org/10.1063/1.4907901.

David Humphreys, A. Kupresanin, M. D. Boyer, J. Canik, C. S. Chang, E. C. Cyr, R. Granetz, J. Hittinger, E. Kolemen, E. Lawrence, V. Pascucci, A. Patra, and D. Schissel. Advancing fusion with machine learning research needs workshop report. *Journal of Fusion Energy*, 39(4):

123–155, 2020. doi: 10.1007/s10894-020-00258-1. URL https://doi.org/10.1007/s10894-020-00258-1.

Matthew Thomas Jackson, Uljad Berdica, Jarek Liesen, Shimon Whiteson, and Jakob Nicolaus Foerster. A Clean Slate for Offline Reinforcement Learning, April 2025. URL http://arxiv.org/abs/2504.11453. arXiv:2504.11453 [cs.LG].

Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A Survey of Zero-shot Generalisation in Deep Reinforcement Learning. *Journal of Artificial Intelligence Research*, 76: 201–264, January 2023. ISSN 1076-9757. doi: 10.1613/jair.1.14174. URL http://arxiv.org/abs/2111.09794. arXiv:2111.09794 [cs.LG].

Robert Tjarko Lange. evosax: JAX-based Evolution Strategies, December 2022. URL http://arxiv.org/abs/2212.04180. arXiv:2212.04180 [cs.NE].

Jarek Liesen, Chris Lu, and Robert Lange. rejax, 2024. URL https://github.com/keraJLi/rejax.

J.B. Lister, A. Portone, and Y. Gribov. Plasma control in ITER. *IEEE Control Systems Magazine*, 26(2):79–91, April 2006. ISSN 1941-000X. doi: 10.1109/MCS.2006.1615274. URL https://ieeexplore.ieee.org/document/1615274.

Y. R. Martin, T. Takizuka, and ITPA CDBM H-mode Threshold Database Working Group. Power requirement for accessing the H-mode in ITER. *Journal of Physics: Conference Series*, 123(1): 012033, 2008. doi: 10.1088/1742-6596/123/1/012033. URL https://doi.org/10.1088/1742-6596/123/1/012033.

D. Mazon, G. Vayakis, M. Walsh, G. Yun, S.-H. Hong, B. Peterson, M.H. Aumeunier, A. Bultel, C. Klepper, D. Rasmussen, H. Choi, C. Grisolia, K. Kim, S. Oh, C. Sun, M. Scholz, B. Esposito, D. Marocco, F. Belli, L. Bertalot, B. Coriton, V. Ginv, D. Gin, J. Dankowski, A. Hjalmarsson, V. Krasilnikov, G. Ericsson, M. Tardochi, D. Rigamonti, M. Nocente, M. Garcia-Munoz, M. Ishikawa, M. Cheon, J. Jo, S. Zoletnik, O. Asztalos, M. Bandyopadhay, P. Bharathi, M. De Bock, O. Ford, M.G. Von Hellermann, D.W. Johnson, J. Ko, S. Menmuir, Ph. Mertens, A.H. Nielsen, G.I. Pokol, S.V. Serov, M.J. Singh, S.N. Tugarinov, G.L. Vyas, M.G. O'Mullane, Ling Zhang, R. Barnsley, R. Tieulent, D. Colette, V.S. Neverov, R. Scannell, H. Liu, E. Mukhin, E. Yatsuka, A. Gorbunov, L. Giudicotti, G. Kurskiev, J. Chen, M.A. Van Zeeland, D. Finkenthal, R. Imazawa, D. Brower, A. Sirinelli, T. Akiyama, T. Carlstrom, M. Lesher, C. Watts, M. Bassan, M. Austin, S.B. Korsholm, Y. Liu, S. Danani, C. Muscatello, W.L. Rowan, V. Vershkov, G. Wang, J.L. Xie, M. Zerbini, Y.L. Zhu, I. Ďuran, A. Gusarov, K. Vukolov, A. Litnovsky, L. Moser, N. Babinov, A. Dmitriev, B. Kim, L. Marot, A. Razdobarin, A. Rogov, D. Samsonov, C. Seon, K. Soni, R. Yan, M.R. De Baar, L. Zabeo, M. Schneider, Th. Blanken, Th. Bosman, T. Ravensbergen, B. Van De Boorn, C. Orrico, R. Fischer, A. Bock, S.S. Denk, A. Medvedeva, M. Salewski, D. Stieglitz, the ASDEX Upgrade Team, and the WEST Team. Diagnostics: Chapter 8 of the special issue: on the path to tokamak burning plasma operation. *Nuclear Fusion*, 65(11): 113001, November 2025. ISSN 0029-5515, 1741-4326. doi: 10.1088/1741-4326/adfc7c. URL https://iopscience.iop.org/article/10.1088/1741-4326/adfc7c.

Luke Metz, C. Daniel Freeman, Samuel S. Schoenholz, and Tal Kachman. Gradients are Not All You Need, January 2022. URL http://arxiv.org/abs/2111.05803. arXiv:2111.05803 [cs.LG].

E. Militello Asp, G. Corrigan, P. da Silva Aresta Belo, L. Garzotti, D.M. Harting, F. Köchl, V. Parail, M. Cavinato, A. Loarte, M. Romanelli, and R. Sartori. JINTRAC integrated simulations of ITER scenarios including fuelling and divertor power flux control for H, He and DT plasmas. *Nuclear Fusion*, 62(12):126033, October 2022. ISSN 0029-5515. doi: 10.1088/1741-4326/ac90d4. URL https://doi.org/10.1088/1741-4326/ac90d4.

J. Mitchell, A. Parrott, F.J. Casson, F.E. Eriksson, F. Koechl, M. Lennholm, O. Bardsley, H.F. Meyer, and S. Aleiferis. Scenario trajectory optimization and control on STEP. *Fusion Engineering and Design*, 192:113777, July 2023. ISSN 09203796. doi: 10.1016/j.fusengdes.2023.113777. URL https://linkinghub.elsevier.com/retrieve/pii/S0920379623003599.

Antoine Mouchamps, Arthur Malherbe, Adrien Bolland, and Damien Ernst. Gym-TORAX: Open-source software for integrating RL with plasma control simulators. *Software Impacts*, 27:100829, April 2026. ISSN 26659638. doi: 10.1016/j.simpa.2026.100829. arXiv: 2510.11283 [cs].

Marco Muraca, Pablo Rodriguez-Fernandez, Nathaniel T. Howard, Joe Hall, Emiliano Fable, and Giovanni Tardini. Integrated Modeling of SPARC H-mode Scenarios: Exploration of the Impact of Modeling Assumptions on Predicted Performance, January 2025. URL http://arxiv.org/abs/2502.00187. arXiv:2502.00187 [physics.plasm-ph].

Tom F. Neiser, Orso Meneghini, Tim Slendebroek, Sterling P. Smith, Joseph T. McClenaghan, Adriana G. Ghiozzi, Bhavin S. Patel, Andrew Oakleigh O. Nelson, Galina Avdeeva, Colin M. Roach, Francis J. Casson, Harry G. Dudding, Gary M. Staebler, Joseph Hall, Brendan C. Lyons, Emily A. Belli, Jeff Candy, Rose Yu, Brian Sammuli, Raffi M. Nazikian, and Tom H. Osborne. Large database validation of TGLF on DIII-D and MAST-U plasmas. Poster, 66th Annual Meeting of the APS Division of Plasma Physics, October 2024. URL https://meetings-archive.aps.org/dpp/2024/pp12/97/.

Rhys Newbury, Jack Collins, Kerry He, Jiahe Pan, Ingmar Posner, David Howard, and Akansel Cosgun. A Review of Differentiable Simulators, July 2024. URL http://arxiv.org/abs/2407.05560. arXiv:2407.05560 [cs.RO].

W. Park and D. A. Monticello. Sawtooth oscillations in tokamaks. *Nuclear Fusion*, 30(11):2413–2418, 1990. doi: 10.1088/0029-5515/30/11/019.

Sai Tej Paruchuri, Andres Pajares, Tariq Rafiq, and Eugenio Schuster. Nonlinear control of the minimum safety factor in tokamaks by optimal allocation of spatially moving electron cyclotron current drive. *Fusion Engineering and Design*, 207:114612, October 2024. ISSN 0920-3796. doi: 10.1016/j.fusengdes.2024.114612. URL https://www.sciencedirect.com/science/article/pii/S0920379624004630.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28(3) of *Proceedings of Machine Learning Research*, pp. 1310–1318. PMLR, 2013. URL https://proceedings.mlr.press/v28/pascanu13.html.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3803–3810, May 2018. doi: 10.1109/ICRA.2018.8460528. URL http://arxiv.org/abs/1710.06537. arXiv:1710.06537 [cs.RO].

G. V. Pereverzev and G. Corrigan. Stable numeric scheme for diffusion equation with a stiff transport. *Computer Physics Communications*, 179(8):579–585, October 2008. ISSN 0010-4655. doi: 10.1016/j.cpc.2008.05.006. URL https://www.sciencedirect.com/science/article/pii/S001046550800218X.

Karel Lucas van de Plassche, Jonathan Citrin, Clarisse Bourdelle, Yann Camenen, Francis J. Casson, Victor I. Dagnelie, Federico Felici, Aaron Ho, Simon Van Mulders, and J. E. T. Contributors. Fast modeling of turbulent transport in fusion plasmas using neural networks. *Physics of Plasmas*, 27(2):022310, February 2020. ISSN 1070-664X, 1089-7674. doi: 10.1063/1.5134126. URL http://arxiv.org/abs/1911.05617. arXiv:1911.05617 [physics.plasm-ph].

O. Sauter, C. Angioni, and Y. R. Lin-Liu. Neoclassical conductivity and bootstrap current formulas for general axisymmetric equilibria and arbitrary collisionality regime. *Physics of Plasmas*, 6(7):2834–2839, July 1999. ISSN 1070-664X, 1089-7674. doi: 10.1063/1.873240. URL https://pubs.aip.org/pop/article/6/7/2834/464899/Neoclassical-conductivity-and-bootstrap-current.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, August 2017. URL http://arxiv.org/abs/1707.06347. arXiv:1707.06347 [cs.LG].

J. Seo, Y.-S. Na, B. Kim, C.Y. Lee, M.S. Park, S.J. Park, and Y.H. Lee. Development of an operation trajectory design algorithm for control of multiple 0D parameters using deep reinforcement learning in KSTAR. *Nuclear Fusion*, 62(8):086049, July 2022. ISSN 0029-5515. doi: 10.1088/1741-4326/ac79be. URL https://doi.org/10.1088/1741-4326/ac79be.

Jaemin Seo, Y.-S. Na, B. Kim, C.Y. Lee, M.S. Park, S.J. Park, and Y.H. Lee. Feedforward beta control in the KSTAR tokamak by deep reinforcement learning. *Nuclear Fusion*, 61(10):106010, September 2021. ISSN 0029-5515. doi: 10.1088/1741-4326/ac121b. URL https://doi.org/10.1088/1741-4326/ac121b.

Jaemin Seo, Rory Conlin, Andrew Rothstein, SangKyeun Kim, Joseph Abbate, Azarakhsh Jalalvand, and Egemen Kolemen. Multimodal Prediction of Tearing Instabilities in a Tokamak. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, June 2023. doi: 10.1109/IJCNN54540.2023.10191359.

Jaemin Seo, SangKyeun Kim, Azarakhsh Jalalvand, Rory Conlin, Andrew Rothstein, Joseph Abbate, Keith Erickson, Josiah Wai, Ricardo Shousha, and Egemen Kolemen. Avoiding fusion plasma tearing instability with deep reinforcement learning. *Nature*, 626(8000):746–751, February 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07024-9. URL https://www.nature.com/articles/s41586-024-07024-9.

B. N. Sorbom, J. Ball, T. R. Palmer, F. J. Mangiarotti, J. M. Sierchio, P. Bonoli, C. Kasten, D. A. Sutherland, H. S. Barnard, C. B. Haakonsen, J. Goh, C. Sung, and D. G. Whyte. ARC: A compact, high-field, fusion nuclear science facility and demonstration power plant with demountable magnets. *Fusion Engineering and Design*, 100:378–405, November 2015. ISSN 09203796. doi: 10.1016/j.fusengdes.2015.07.008. URL http://arxiv.org/abs/1409.3540. arXiv:1409.3540 [physics.plasm-ph].

G. M. Staebler, J. E. Kinsey, and R. E. Waltz. A theory-based transport model with comprehensive physics. *Physics of Plasmas*, 14(5):055909, May 2007. ISSN 1070-664X. doi: 10.1063/1.2436852. URL https://doi.org/10.1063/1.2436852.

G Tardini, E Fable, C Angioni, M Bergmann, D Fajardo, and T Luda. ASTRA-8: a modern framework for transport analysis and modelling in fusion devices. *Plasma Physics and Controlled Fusion*, 68(6):065024, June 2026. ISSN 0741-3335. doi: 10.1088/1361-6587/ae7640. URL https://doi.org/10.1088/1361-6587/ae7640.

E. Tholerus, F.J. Casson, S.P. Marsden, T. Wilson, D. Brunetti, P. Fox, S.J. Freethy, T.C. Hender, S.S. Henderson, A. Hudoba, K.K. Kirov, F. Koechl, H. Meyer, S.I. Muldrew, C. Olde, B.S. Patel, C.M. Roach, S. Saarelma, G. Xia, and the STEP team. Flat-top plasma operational space of the STEP power plant. *Nuclear Fusion*, 64(10):106030, August 2024. ISSN 0029-5515. doi: 10.1088/1741-4326/ad6ea2. URL https://doi.org/10.1088/1741-4326/ad6ea2.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World, March 2017. URL http://arxiv.org/abs/1703.06907. arXiv:1703.06907 [cs.RO].

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Brendan D. Tracey, Andrea Michi, Yuri Chervonyi, Ian Davies, Cosmin Paduraru, Nevena Lazic, Federico Felici, Timo Ewalds, Craig Donner, Cristian Galperti, Jonas Buchli, Michael Neunert, Andrea Huber, Jonathan Evens, Paula Kurylowicz, Daniel J. Mankowitz, and Martin Riedmiller. Towards practical reinforcement learning for tokamak magnetic control. *Fusion Engineering and Design*, 200:114161, March 2024. ISSN 0920-3796. doi: 10.1016/j.fusengdes.2024.114161. URL https://www.sciencedirect.com/science/article/pii/S0920379624000140.

Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020. ISSN 2665-9638. doi: https://doi.org/10.1016/j.simpa.2020.100022. URL https://www.sciencedirect.com/science/article/pii/S2665963820300099.

S. Van Mulders, F. Felici, O. Sauter, J. Citrin, A. Ho, M. Marin, and K.L. van de Plassche. Rapid optimization of stationary tokamak plasmas in RAPTOR: demonstration for the ITER hybrid scenario with neural network surrogate transport model QLKNN. *Nuclear Fusion*, 61(8):086019, July 2021. ISSN 0029-5515. doi: 10.1088/1741-4326/ac0d12. URL https://doi.org/10.1088/1741-4326/ac0d12.

S Van Mulders, O Sauter, C Contré, F Felici, R Fischer, T Pütterich, B Sieglin, A A Teplukhina, and the ASDEX Upgrade Team. Scenario optimization for the tokamak ramp-down phase in RAPTOR: Part A. Analysis and model validation on ASDEX Upgrade. *Plasma Physics and Controlled Fusion*, 66(2):025006, December 2023. ISSN 0741-3335. doi: 10.1088/1361-6587/ad157d. URL https://doi.org/10.1088/1361-6587/ad157d.

Allen M. Wang, Cristina Rea, Oswin So, Charles Dawson, Darren T. Garnier, and Chuchu Fan. Active ramp-down control and trajectory design for tokamaks with neural differential equations and reinforcement learning. *Communications Physics*, 8(1):231, June 2025. ISSN 2399-3650. doi: 10.1038/s42005-025-02146-6. URL https://www.nature.com/articles/s42005-025-02146-6.

John Wesson. *Tokamaks*, volume 149 of *International Series of Monographs on Physics*. Oxford University Press, Oxford, fourth edition, 2011. ISBN 9780199592234.

Samuel E. Wurzel and Scott C. Hsu. Progress toward fusion energy breakeven and gain as measured against the Lawson criterion. *Physics of Plasmas*, 29(6):062103, June 2022. ISSN 1070-664X, 1089-7674. doi: 10.1063/5.0083990. URL http://arxiv.org/abs/2105.10954. arXiv:2105.10954 [physics.plasm-ph].

L. Zanisi, F. J. Casson, B. S. Patel, H. G. Dudding, Z. Stancar, C. M. Roach, A. P. Snodin, C. Siddle, J. Buchanan, S. Pamela, T. Brown, A. Järvinen, A. Kit, M. Marin, A. Ho, G. Snoep, E. Vergnaud, C. Bourdelle, O. Meneghini, T. Neiser, and JET contributors. STEP and multimachine TGLF neural surrogates. In *30th IAEA Fusion Energy Conference*, Chengdu, China, 2025. URL https://conferences.iaea.org/event/392/papers/36059/files/13368-IAEA_FEC_2025_manuscript.pdf. Conference preprint.

A Environment Details

A.1 Overview

Table 1: Default environment configurations. Horizons assume no early termination. State dimensions refer to policy observations, shown as base / realistic. Actuator defaults are physical reset values. Actuator order: ITER/SPARC use $(P_{\mathrm{nbi}}, P_{\mathrm{eccd}}, \rho_{\mathrm{eccd}}, \Gamma_{\mathrm{gas}})$; STEP uses $(P_{\mathrm{eccd}}, \rho_{\mathrm{eccd}}, \Gamma_{\mathrm{pellet}})$. Powers are in MW and particle rates in 10^{21} particles/s. On SPARC, P_{nbi} denotes ICRF heating. KSTAR uses $(I_p, \kappa, \delta_u, \delta_l, R_{\mathrm{in}}, R_{\mathrm{out}})$, with I_p in MA and radii in meters.

Environment	Duration (s)	Steps	Frequency (Hz)	Reward	Actuator defaults	State dim.	Action dim.
`iter/baseline/rampup`	60	600	10	`lh_transition`	$(20, 0, 0.3, 6)$	134 / 24	4
`iter/baseline/flattop`	200	2000	10	`P_diff`	$(51, 0, 0.1274, 0.1)$	134 / 24	4
`iter/baseline/rampdown`	40	400	10	`rampdown`	$(32.2, 16.88, 0.1809, 0)$	134 / 24	4
`iter/hybrid/rampup`	100	1000	10	`lh_transition`	$(20, 0, 0.1274, 0)$	134 / 24	4
`iter/hybrid/flattop`	440	4400	10	`P_diff`	$(51, 0, 0.1274, 6)$	134 / 24	4
`iter/hybrid/rampdown`	60	600	10	`rampdown`	$(23.31, 27.69, 0.1274, 6)$	134 / 24	4
`iter/advanced/rampup`	100	1000	10	`lh_transition`	$(20, 0, 0.5, 0)$	134 / 24	4
`iter/advanced/flattop`	840	8400	10	`P_diff`	$(51, 0, 0.1274, 0.1)$	134 / 24	4
`iter/advanced/rampdown`	60	600	10	`rampdown`	$(34, 34, 0.5, 6)$	134 / 24	4
`sparc/prd/rampup`	10	100	10	`lh_transition`	$(11, 0, 0.3, 4)$	134 / 24	4
`sparc/prd/flattop`	15	150	10	`P_diff`	$(11, 0, 0.3, 4)$	134 / 24	4
`sparc/prd/rampdown`	5	50	10	`rampdown`	$(11, 0, 0.3, 4)$	134 / 24	4
`sparc/reduced_field/rampup`	10	100	10	`lh_transition`	$(25, 0, 0.3, 0)$	134 / 24	4
`sparc/reduced_field/flattop`	15	150	10	`P_diff`	$(25, 0, 0.3, 0)$	134 / 24	4
`sparc/reduced_field/rampdown`	5	50	10	`rampdown`	$(25, 0, 0.3, 0)$	134 / 24	4
`step/spp_001_ec_hd/flattop`	400	4000	10	`P_diff`	$(150, 0.3384, 3)$	134 / 24	3
`kstar/nominal/tracking`	190	100	0.526	Native tracking	$(0.5, 1.7, 0.3, 0.75, 1.32, 2.22)$	15 / 15	6

Table 2: Available tokamak–backend combinations. The number in brackets is the number of environments, determined by the available scenario–phase combinations. There are 63 backend–environment combinations.

Tokamak	**CGM**	**QLKNN**	**Bohm–GyroBohm**	**TGLFNN**	**World Model**
ITER (9)	✓	✓	✓	✓	×
SPARC (6)	✓	✓	✓	✓	×
STEP (1)	×	×	✓	✓	×
KSTAR (1)	×	×	×	×	✓

A.2 Phases

A tokamak discharge has three phases.

1. The *ramp-up*, where the plasma current is progressively increased. The goal of this phase is to heat up the plasma until it transitions from *L-mode* to *H-mode*, in which energy confinement time increases significantly.
2. The *flat-top*, where plasma current is held constant. This phase generates nuclear fusion power.
3. The *ramp-down*, the current is progressively decreased. The goal is to decrease the plasma current without disrupting the plasma.

Plasmax implements all these phases for the ITER and SPARC tokamaks. These tasks have different objectives and therefore different reward functions.

A.3 Reward Design

Each Plasmax task defines a scalar reward z_t at each time step. The default reward functions are `lh_transition` for ramp-up, `P_diff` for flat-top, and `rampdown` for ramp-down. We modify these scalar rewards to ensure (i) automatic differentiation and (ii) remove per-task tuning. Strictly positive rewards on standard transitions also incentivize long episodes, which is desirable for tokamak shots.

Reward Transformation All built-in Plasmax rewards use the square-plus utility

$$U(z) = \frac{z + \sqrt{z^2 + 4}}{2}, \tag{1}$$

$$r_t = \begin{cases} U(z_t), & c_t = -1, \\ \log U(z_t) = \operatorname{asinh}(z_t/2), & c_t \in \{1, 2, 3\}, \\ 0, & c_t = 4, \end{cases} \tag{2}$$

where $c_t = -1$ denotes an ordinary transition, $c_t = 1, 2$ denote safety-factor and Greenwald-limit disruptions, respectively, $c_t = 3$ denotes solver failure (provided by TORAX), and $c_t = 4$ denotes a state with non-finite values. We choose this slightly contrived formulation for multiple reasons. Early on, we found that scalar penalties required per-environment tuning because each environment has a different episode length. We therefore wanted a function that satisfies $f(x) \leq x$, is differentiable over its domain, and has no parameters; the natural logarithm has these properties, but it requires a positive reward. We therefore chose a square-plus transformation, which has a larger range of values in the negative domain than other such functions, e.g., softplus. We evaluate the logarithm of the squareplus with the arcsinh function, which is equivalent but more numerically stable. Non-finite inputs are masked before reward arithmetic, yielding zero reward and zero reward-branch gradient.

Soft constraints The ramp-up and ramp-down scores include log-sigmoid barriers inspired by Wang et al. (2025). For a quantity x with upper limit y, define

$$B(x; y) = \log \sigma[-40\,(\max(x/y, 0) - 0.9)]\,, \tag{3}$$

where σ is the logistic sigmoid. The penalty is near zero comfortably below $0.9y$ and becomes increasingly negative as the limit is approached or exceeded.

Ramp-up The goal is a stable H-mode plasma suitable for power production. Rewarding the L–H transition alone can encourage an early crossing while the current is still ramping, followed by low $q_{\min}$ or high f_{GW}. We therefore weight the transition terms by time and include soft barriers throughout the ramp:

$$z_t^{\mathrm{LH}} = \max\left(\frac{P_{\mathrm{SOL}}}{P_{\mathrm{LH}}}, 0\right) \tau^2 + \mathbb{I}_{\mathrm{H}}\tau - 2\,\mathbb{I}_{\mathrm{H}\to\mathrm{L}} + B(f_{\mathrm{GW}}; 1) + B(q_{\min}^{-1}; 1/1.6), \tag{4}$$

where $\tau = t/t_{\mathrm{final}}$, t is the simulation time, and t_{final} is the scenario's configured final time. Here P_{SOL} is the power crossing the separatrix into the scrape-off layer and P_{LH} is the empirical threshold for H-mode access (Martin et al., 2008). The quadratic time weight suppresses this term early in the ramp. The H-mode indicator $\mathbb{I}_{\mathrm{H}}$ rewards sustained H-mode, while $\mathbb{I}_{\mathrm{H}\to\mathrm{L}}$ penalizes a transition back towards L-mode. The barriers discourage a high Greenwald fraction $f_{\mathrm{GW}} = \bar{n}_e/n_{\mathrm{GW}}$ and a low minimum safety factor.

Flat-top The objective is fusion output minus the auxiliary heating power spent to sustain it, in gigawatts:

$$z_t^{P_{\mathrm{diff}}} = \frac{P^{\mathrm{fusion}} - P^{\mathrm{aux}}}{10^9\ \mathrm{W}}. \tag{5}$$

We do not use the more standard $Q = P^{\mathrm{fusion}}/P^{\mathrm{aux}}$ as agents control the denominator, which led to reward-hacking in experiments. Using the power difference avoids this particular incentive.

Ramp-down The objective is to remain stable while the plasma current follows a prescribed ramp-down schedule. We use the stability barriers inspired from *PopDownGym* Wang et al. (2025):

$$z_t^{\mathrm{rampdown}} = B(f_{\mathrm{GW}}; 1) + B(\ell_{i3}; 1.5) + B(q_{\min}^{-1}; 1) + B(\beta_N; 3), \tag{6}$$

where ℓ_{i3} is the normalized internal inductance and β_N is the normalized toroidal beta. The safety-factor limit relaxes from 1.6 to 1 for ramp-down.

The KSTAR world model retains its native reward without the Plasmax transformation.

A.4 WRAPPERS

Following recommendations for real-world RL (Dulac-Arnold et al., 2019), optional wrappers model imperfect measurements and uncertain plasma physics. They can be combined as needed, alongside training convenience wrappers for actions and episode length:

- **Physics randomization:** Samples physics parameters uniformly before each step, using configured ranges or factors relative to default values. We use uncertainties from Sauter et al. (1999); Neiser et al. (2024); Plassche et al. (2020).
- **Observation noise:** Adds independent Gaussian noise scaled to each reading's magnitude. Default noise levels are informed by ITER measurement specifications (Mazon et al., 2025).
- **Observation filtering:** Keeps selected measurements and samples profiles at fewer, roughly evenly spaced points, reflecting limits on what sensors can measure.
- **Observation delay:** Independently reuses each measurement's last reported value with a configurable probability, mimicking missing or delayed readings.
- **Observation history:** Includes the last k observations and actions, helping the controller infer changes that a single observation may miss.
- **Action rescaling:** Maps actions from $[-1, 1]$ onto each actuator's physical range using a linear rescaling.
- **Action quantization:** Provides a configurable number of evenly spaced settings for each actuator, allowing controllers to choose discrete actions.
- **Time awareness:** Adds time since the episode began to observations; the KSTAR world model uses elapsed step count.

A.5 GEOMETRIES AND INITIALIZATIONS

Each plasma current value induces a different plasma geometry. Therefore, for each scenario and phase, we regenerate geometries using FreeGSNKE (Amorisco et al., 2024). During the ramp-up and ramp-down phases, the plasma current increases and decreases progressively. We therefore generate three geometries at different current values and interpolate between them as the current increases or decreases. We show the geometries in Figure 11.

Prior work in tokamak modeling has determined initializations for different phases; we use available state and actuator values and dummy values for missing information (Citrin et al., 2024; Muraca et al., 2025; Brown et al., 2025). We then run the simulator forward and let it relax to stable states, which we use as initializations.

Table 3: Default reset schemes across environments. Percentages denote relative standard deviations of independent Gaussian perturbations; these provide diversity without causing solver failures. We do not report the full 134 variable initial values here; we refer to these as *cold* and *hot*.

Environment	Initialization	Reset randomization
ITER baseline (all phases)	Cold for ramp-up; hot for flattop and ramp-down.	T_e, T_i: 5%; n_e: 3%.
ITER hybrid (all phases)	Cold for ramp-up; hot for flat-top and ramp-down.	T_e, T_i: 5%; n_e: 3%.
ITER advanced (all phases)	Cold for ramp-up; hot for flattop and ramp-down.	T_e, T_i: 5%; n_e: 3%.
SPARC PRD (all phases)	Cold for ramp-up; hot for flattop and ramp-down.	T_e, T_i: 5%; n_e: 3%.
SPARC reduced-field (all phases)	Cold for ramp-up; hot for flattop and ramp-down.	T_e, T_i: 5%; n_e: 3%.
STEP flattop	Hot OpenSTEP snapshot.	T_e, T_i: 5%; n_e: 3%.
KSTAR world model	Fixed engineering inputs and LSTM history.	$\beta_p^\star \sim \mathcal{U}(1.1, 2.1)$, $q_{95}^\star \sim \mathcal{U}(3.8, 6.2)$, $l_i^\star \sim \mathcal{U}(0.84, 1.06)$.

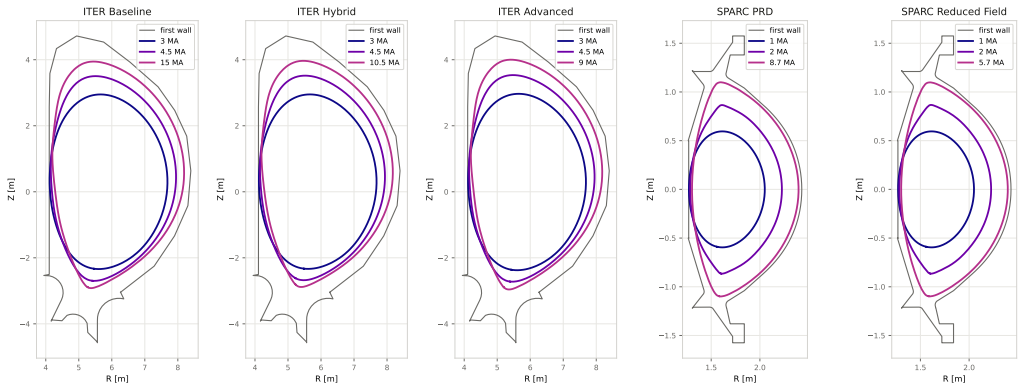


Figure 11: Plasma geometries generated with FreeGSNKE (Amorisco et al., 2024) for different tokamaks & scenarios at different plasma currents.

A.6 SCENARIOS

Each tokamak is associated with a set of *scenarios*. These detail the experimental conditions for the plasma discharge. For example, the *advanced* scenario for ITER uses less plasma current than the *baseline* scenario, making the plasma harder to control. ITER has three main scenarios

- *Baseline*: operates at a current of 15 MA. High Q for shorter operation.
- *Hybrid*: operates at a current of 10.5 MA. Halfway between the other two scenarios.
- *Advanced*: operates at a current of 9 MA. Hardest to control but longest potential operation.

In Plasmax, we use two SPARC scenarios:

- *Reduced Field*: a test scenario at only 5.7 MA and 8T. Projected Q at 1.6.
- *Primary Reference Discharge* (PRD): the flagship SPARC scenario at 8.7 MA and 12.2T, with projected Q of 11.

We only include a single STEP scenario (Brown et al., 2025, SPP-001) and phase (flat-top). Future Plasmax releases aim to build on this.

B EXPERIMENTAL DETAILS

Table 4: PPO hyperparameters. Settings are shared across the plotted PPO runs.

Hyperparameter	Value
Training budget	10,240,000 environment steps
Parallel environments	1024
Rollout length	100 steps per environment
Epochs per iteration	4
Minibatches per epoch	4
Minibatch size	25,600
Policy network	MLP $(256, 256)$
Critic network	MLP $(256, 256)$
Activation	Swish
Policy distribution	Diagonal Gaussian; actions clipped to bounds
Initial log standard deviation	0 (initial standard deviation 1)
Learning rate	3×10^{-4}, constant
Adam $(\beta_1, \beta_2, \epsilon)$	$(0.9,\ 0.999,\ 10^{-8})$
Discount factor γ	0.99
GAE parameter λ	0.95
PPO policy/value clipping ϵ	0.2
Value-loss coefficient	0.5
Entropy coefficient	0.01
Gradient clipping	Elementwise to $[-1, 1]$
Reward normalization	Enabled
Observation normalization	Enabled
Evaluation interval	1,024,000 environment steps
Evaluation episodes	128 per seed and checkpoint
Evaluation seed	42
Evaluation action selection	Deterministic

Table 5: SAC hyperparameters. Settings are shared across the plotted SAC runs; budgets and buffer sizes are per training seed.

Hyperparameter	Value
Training budget	10,240,000 environment steps
Parallel environments	64
Minibatch size	256
Replay-buffer capacity	1,000,000 transitions
Replay-buffer warm-up	10,000 transitions
Policy parameterization	MLP $(256, 256)$
Critic networks	Two MLPs $(256, 256)$
Activation	Swish
Policy distribution	Tanh-squashed diagonal Gaussian
Policy log standard deviation	Clipped to $[-10, 2]$
Learning rate	3×10^{-4}, constant
Adam $(\beta_1, \beta_2, \epsilon)$	$(0.9,\ 0.999,\ 10^{-8})$
Discount factor γ	0.99
Polyak factor	0.995
Gradient clipping	Elementwise to $[-1, 1]$
Entropy temperature α	Learned; initialized to 1
Reward normalization	Enabled
Observation normalization	Enabled
Evaluation interval	1,024,000 environment steps
Evaluation episodes	128 per seed and checkpoint
Evaluation seed	42
Evaluation action selection	Deterministic

Table 6: Direct-gradient baseline hyperparameters. Both methods optimize undiscounted reward through 25-step simulator chunks, stopping gradients through the carried simulator state between chunks.

Hyperparameter	Backprop (closed-loop)	Backprop (open-loop)
Policy parameterization	$(256, 256)$ MLP	10 knots per actuator
Activation	Swish	Not applicable
Training rollouts per update	1024	1024
Gradient horizon	25 steps	25 steps
Training budget	10,240,000 environment steps	10,240,000 environment steps
Learning rate	10^{-4}	5×10^{-2}
Adam $(\beta_1, \beta_2, \epsilon)$	$(0.9,\ 0.999,\ 10^{-8})$	$(0.9,\ 0.999,\ 10^{-8})$
Gradient clipping	Global norm 1.0	Global norm 1.0
Minimum knot update scale	Not applicable	10^{-3}
Observation normalization	Disabled	Disabled
Reward normalization	Disabled	Disabled
Evaluation interval	1,024,000 steps	1,024,000 steps
Evaluation episodes	128	128
Evaluation seed	42	42

Table 7: CMA-ES hyperparameters. Both controllers maximize undiscounted full-episode return using full-covariance CMA-ES.

Hyperparameter	CMA-ES (closed-loop)	CMA-ES (open-loop)
Policy parameterization	$(64, 64)$ MLP	10 knots per actuator
Activation	Swish	Not applicable
Population size λ	64	64
Positive-weight parents μ	32	32
Initial search scale σ_0	0.01	0.01
Initial covariance C_0	Identity	Identity
Mean learning rate c_m	1.0	1.0
Fitness episodes per candidate	1	1
Fitness horizon	Full environment episode	Full environment episode
Observation normalization	Disabled	Disabled
Reward normalization	Disabled	Disabled
Evaluation episodes	128	128
Evaluation seed	42	42

C Additional Results

C.1 Environment throughput

Table 8: Environment throughput in steps/second, computed from median durations over five repeats. Compilation and warmup are excluded. We tried our best to match Gym-TORAX's backend implementation to ours, including the solver, steps, and configuration.

			Parallel environments							
Backend	Implementation	Device	1	4	16	64	256	1,024	4,096	16,384
QLKNN	Gym-TORAX	CPU	26.57	–	–	–	–	–	–	–
QLKNN	Plasmax	CPU	125.99	183.02	255.01	464.03	631.97	641.35	640.09	546.98
QLKNN	Plasmax	H100 GPU	14.39	61.87	232.82	860.10	3,206.89	10,680.65	23,398.04	29,319.02
TGLFNN	Plasmax	H100 GPU	14.50	58.29	220.15	639.48	1,127.52	1,280.99	1,365.35	1,033.02
Bohm-GyroBohm	Plasmax	H100 GPU	16.27	64.50	244.65	806.24	2,931.65	9,910.99	26,915.59	46,594.67
CGM	Plasmax	H100 GPU	48.27	216.99	827.31	2,877.99	11,982.47	36,783.90	121,180.29	193,188.95
World model	Plasmax	H100 GPU	278.09	876.85	3,357.48	9,163.28	19,196.37	26,370.47	31,406.10	33,644.80

Table 9: Backpropagation throughput on one H100 for ITER/advanced/flattop with Bohm–GyroBohm. Values are simulator steps per second for forward-and-backward passes through the environment; compilation is excluded.

Rollout steps	64 envs	256 envs	1024 envs
8	157.9	681.9	2,774.6
16	170.7	721.3	2,542.5
32	188.9	732.6	2,984.3
64	185.9	674.2	2,600.9
128	173.7	682.8	**3,025.1**

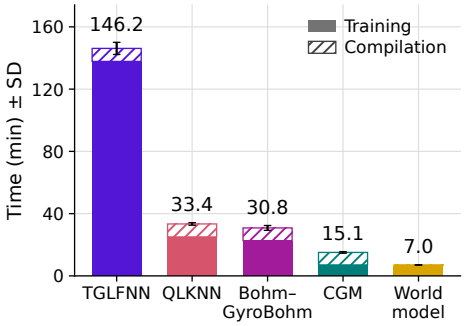


Figure 12: PPO wall-clock time for 10M environment steps using one H100 GPU per run.

Table 10: Configuration for the backend agreement and throughput benchmark.

Setting	Configuration
Environment	ITER/hybrid/flattop
Physics Randomization	Off
Fixed actuators	NBI: 51 MW; ECCD: 0 MW; gas puff: 6×10^{21} s^{-1}
Rollout	500 control steps of 0.1 s (50 s total)
Agreement metric	MSE averaged over radius and time, with equal weighting of normalized T_e, T_i, n_e, and q profiles
Throughput	Median of three warm compiled rollouts; compilation excluded

C.2 Individual Environment Results

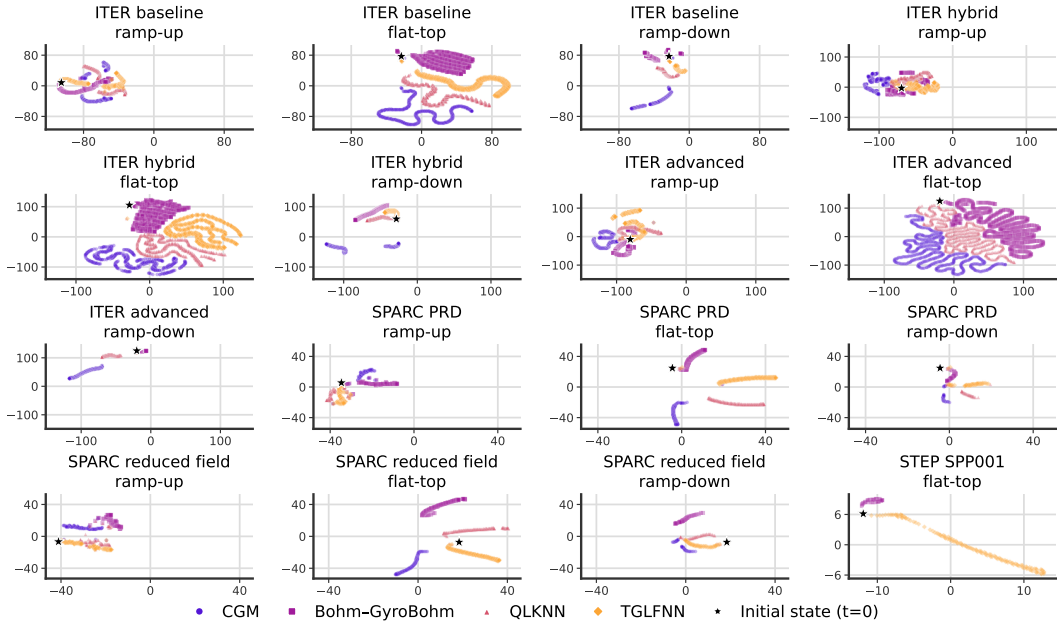


Figure 13: T-SNE embeddings of rollouts for different backends and environments. The embeddings are computed per scenario.

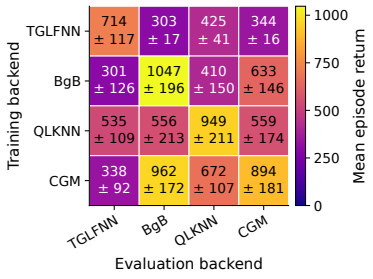


Figure 14: Unnormalized returns of Figure 10.

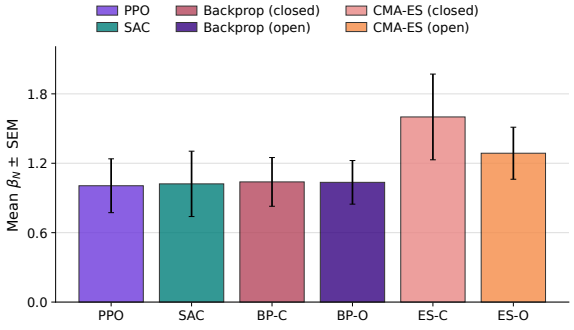


Figure 15: Closed-loop CMA-ES achieves higher β_N in flat-top environments, suggesting that closed-loop is able to control higher-pressure, more unstable plasmas. Mean $\beta_N \pm$ SEM across flat-top environments with Bohm-GyroBohm (5 seeds).

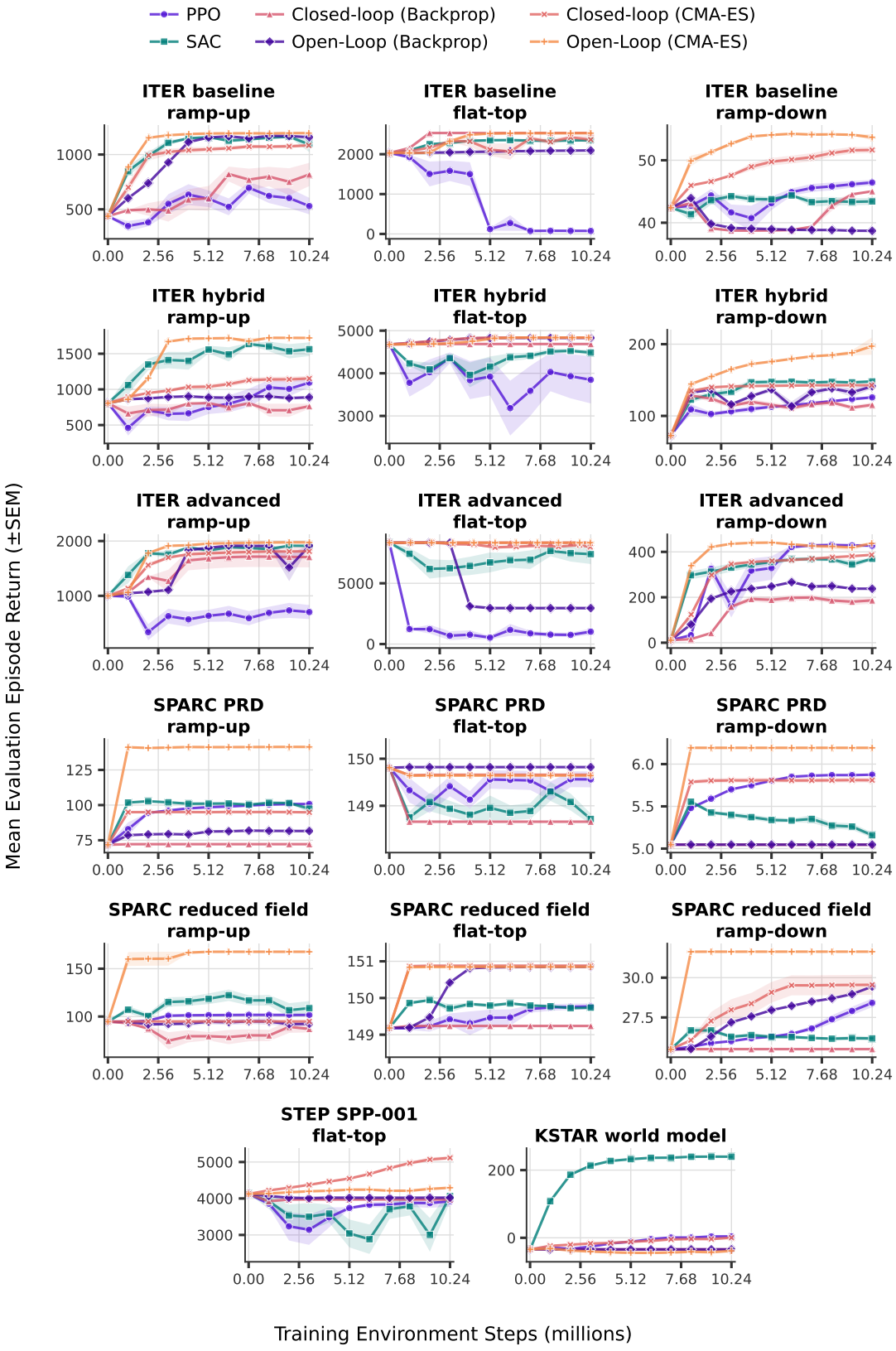


Figure 16: Environment Returns for different algorithms. Mean and SEM across 10 seeds.

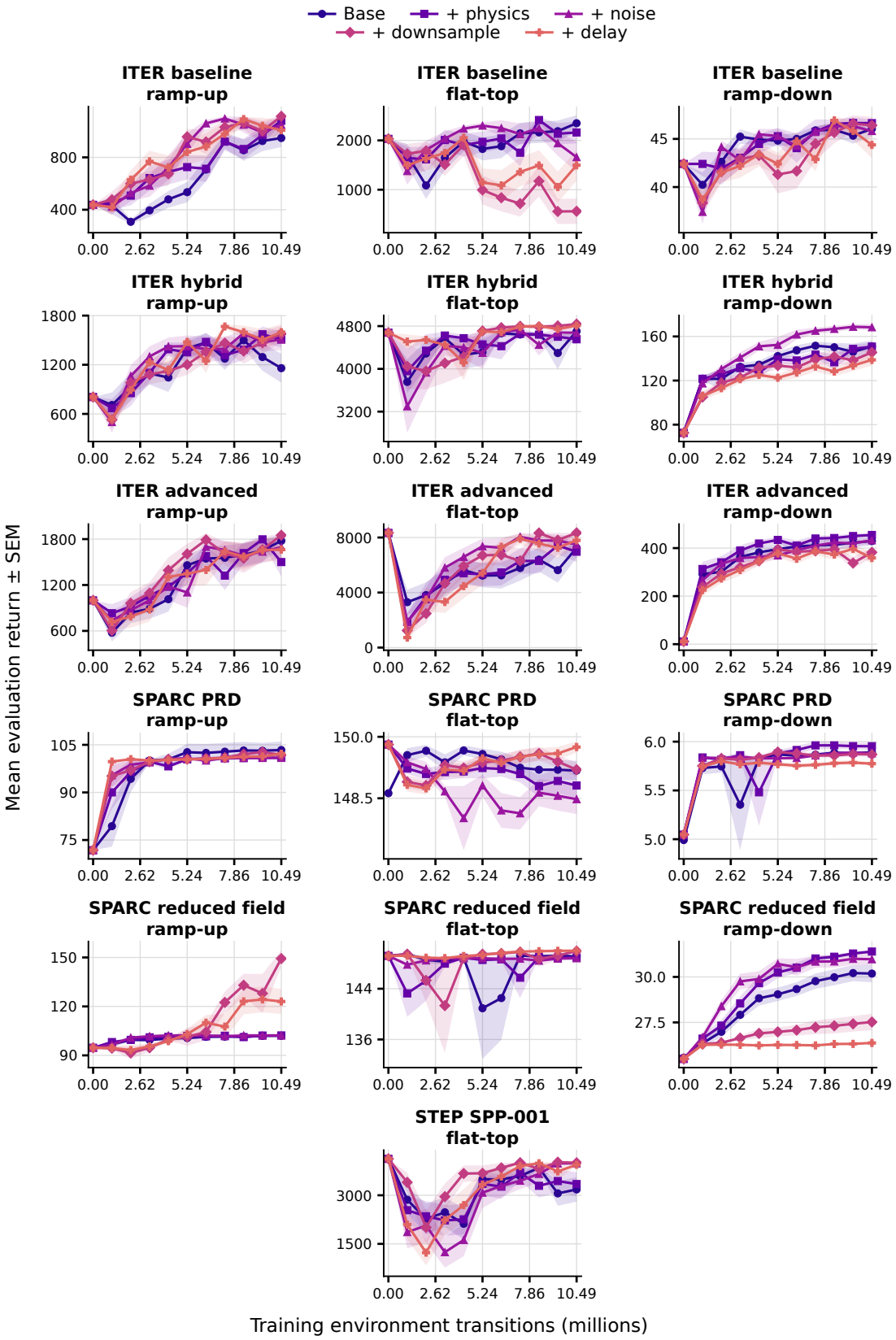


Figure 17: Wrapper ablations for PPO on all environments. Mean and SEM across 15 seeds.

C.3 Hyperparameter Tuning

We investigate how hyperparameter tuning can improve RL performance. We use the ITER/hybrid/flattop task, where both SAC and PPO perform poorly. We define the hyperparameter search space inspired by reference values from Liesen et al. (2024); detailed in Table 11. We then use random search with 5 training seeds in parallel per run. We budget 100 runs per algorithm and find that PPO needs only 14 runs to reach optimal performance. SAC needs closer to 40 tuning runs to reach a lower return. These results support two hypotheses: (i) PPO is a strong baseline, once tuned. (ii) PPO is more performant than SAC. More experimentation is necessary to confirm these hypotheses. We represent this search in Figure 18, inspired by (Jackson et al., 2025).

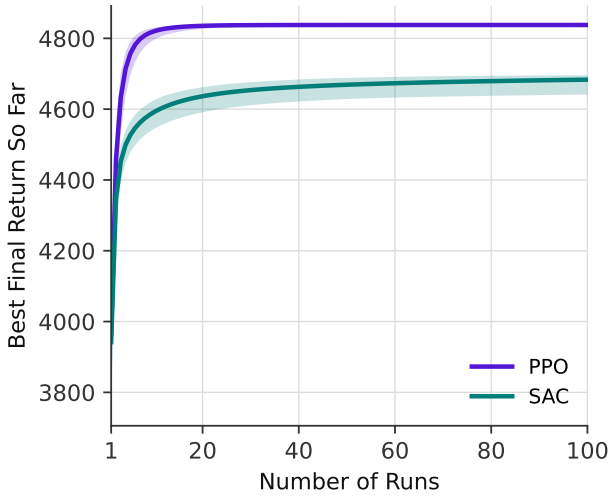


Figure 18: Tuning PPO and SAC can substantially improve performance. Bootstrapped random search with 95% confidence intervals.

Table 11: Tuned hyperparameter search spaces for PPO and SAC.

Hyperparameter	Search space
PPO	
Learning rate	$[10^{-5}, 10^{-3}]$ (log-uniform)
Discount factor γ	$\{0.95, 0.97, 0.99, 0.995, 0.999\}$
GAE parameter λ	$\{0.90, 0.95, 0.97, 0.99\}$
PPO policy/value clipping ϵ	$\{0.1, 0.2, 0.3\}$
Entropy coefficient	$\{0, 10^{-4}, 10^{-3}, 10^{-2}, 0.03\}$
Value-loss coefficient	$\{0.25, 0.5, 1.0\}$
Gradient clipping	$\{0.5, 1.0, 2.0\}$
Epochs per iteration	$\{2, 4, 8\}$
Rollout length	$\{8, 32, 64, 128\}$ steps per environment
Minibatches per epoch	$\{4, 8, 16\}$
Observation normalization	Enabled, disabled
Reward normalization	Enabled, disabled
SAC	
Learning rate	$[10^{-5}, 10^{-3}]$ (log-uniform)
Discount factor γ	$\{0.95, 0.97, 0.99, 0.995, 0.999\}$
Gradient updates per vector step	$\{16, 32, 64, 128\}$
Minibatch size	$\{128, 256, 512\}$
Replay-buffer capacity	$\{2^{17}, 2^{18}, 2^{20}\}$ transitions
Target-network retention	$\{0.99, 0.995, 0.999\}$
Gradient clipping	$\{1.0, 5.0, 10.0\}$
Observation normalization	Enabled, disabled
Reward normalization	Enabled, disabled

Table 12: Best SAC configuration found by random search, 5 seeds per setting. Return is reported as the mean ± standard deviation across five seeds. † indicates hyperparameters that were fixed during tuning.

Hyperparameter	Value
Learning rate	1.24376×10^{-4}
Discount factor γ	0.95
Gradient updates per vector step	64
Minibatch size	128
Replay-buffer capacity	1,048,576 transitions
Replay-buffer warm-up threshold†	10,000 transitions
Target-network retention	0.99
Target update interval†	1 vector step
Gradient clipping	Elementwise to $[-10, 10]$
Observation normalization	Disabled
Reward normalization	Disabled
Parallel environments†	64
Actor/critic hidden sizes†	$(256, 256)$
Activation†	Swish
Training budget†	10,485,760 environment steps
Final evaluation return	4697.44 ± 51.53

Table 13: Best PPO configuration found by random search, 5 seeds per setting. Return is reported as the mean ± standard deviation across five seeds. † indicates hyperparameters that were fixed during tuning.

Hyperparameter	Value
Learning rate	1.97405×10^{-5}
Discount factor γ	0.97
GAE parameter λ	0.99
PPO policy/value clipping ϵ	0.3
Entropy coefficient	0.001
Value-loss coefficient	0.5
Gradient clipping	Elementwise to $[-1, 1]$
Epochs per iteration	4
Rollout length	64 steps per environment
Minibatches per epoch	16
Observation normalization	Disabled
Reward normalization	Enabled
Actor/critic hidden sizes†	(256, 256)
Activation†	Swish
Parallel environments†	1,024
Training budget†	10,485,760 environment steps
Final evaluation return	$4,837.71 \pm 0.00$

C.4 JACOBIAN EIGENVALUES

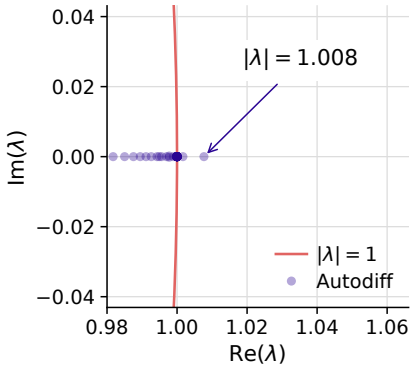


Figure 19: We find Jacobian eigenvalues that have a higher magnitude than 1 on ITER/hybrid/flattop when energy input is high and particle fueling is low. Nevertheless, we could not draw a direct causal link between Jacobian eigenvalues and high gradient variance.

C.5 BUGS IN TORAX

Policy optimization explores and exploits environments and therefore can be used as a tool to (i) comprehensively cover the state space of a simulator (ii) exploit flaws in the simulator to obtain a higher reward; reward-hacking. During the development and subsequent benchmarking of Plasmax, we uncovered multiple bugs and monkey-patched them before running experiments.

1. Scaling of Pereverzev-Corrigan transport coefficients in the pedestal. These coefficients enable numerical stability (Pereverzev & Corrigan, 2008) and therefore should not be scaled with other pedestal transport coefficients. We found this bug while investigating many numerically unstable runs.
2. Safety factor profile polynomial interpolation overfits the data, allowing, e.g., negative values, which is physically impossible. TORAX derives the minimum safety factor ($q_{\min}$) from this interpolated profile, and it is one of the most important metrics for plasma stability (Paruchuri et al., 2024). We found this by investigating runs that hit the $q_{\min}$ threshold more often than expected.
3. Unsafe operations for automatic differentiation in the simulator. We found these by analyzing non-finite gradients when backpropagating through the simulator.

We submitted a fix for the Pereverzev-Corrigan coefficient bug, and the maintainers agreed. The safety factor interpolation has since been fixed independently in TORAX 1.4.3 (we use 1.4.2 in this paper). We have also submitted fixes for the automatic differentiation issues. We do not link the PRs to maintain double-blind anonymity.

D FULL ENVIRONMENT TRANSFER RESULTS

Table 14: Policy returns for all algorithms trained on the Bohm-GyroBohm backend. We also evaluate the trained policies on the other three backends. Values are the mean ± SEM across training seeds with 128 evaluation episodes per seed. 10 trained policies × 128 evaluation seeds = 1280 episodes evaluated per table entry. We chose to omit the 100-knot results from the main text for clarity, as the results are not meaningfully different from the 10-knot ones for the majority of environments. We note that the SEMs of 0 result from decimal truncation.

Environment	Backend	PPO	SAC	Direct backprop	Backprop 10 knots	Backprop 100 knots	CMA-ES 10 knots	CMA-ES policy
ITER baseline rampup	BgB	529.7 ± 75.3	1092 ± 70	818.1 ± 106.3	1158 ± 5	1160 ± 1	$\mathbf{1195 \pm 1}$	1083 ± 12
	CGM	647.8 ± 83.3	607.1 ± 40.1	712.4 ± 74.0	$\mathbf{950.9 \pm 2.7}$	941.9 ± 1.1	928.9 ± 2.3	656.2 ± 15.7
	QLKNN	389.8 ± 50.1	809.1 ± 63.3	840.3 ± 90.6	$\mathbf{1229 \pm 4}$	1217 ± 1	1086 ± 53	602.5 ± 23.2
	TGLFNN	401.5 ± 75.3	880.6 ± 79.8	870.2 ± 89.2	$\mathbf{1166 \pm 3}$	1161 ± 1	1131 ± 1	690.6 ± 44.0
ITER baseline flattop	BgB	76.21 ± 1.00	2349 ± 10	$\mathbf{2533 \pm 0}$	2095 ± 0	2035 ± 0	2528 ± 0	2366 ± 105
	CGM	1189 ± 287	1729 ± 165	1887 ± 0	1893 ± 0	$\mathbf{1893 \pm 0}$	1856 ± 0	1889 ± 0
	QLKNN	664.9 ± 275.6	1525 ± 235	1913 ± 0	1906 ± 0	1907 ± 0	$\mathbf{1948 \pm 0}$	1911 ± 0
	TGLFNN	643.8 ± 270.2	1518 ± 253	2087 ± 4	1950 ± 0	1946 ± 0	$\mathbf{2318 \pm 2}$	2035 ± 12
ITER baseline rampdown	BgB	46.44 ± 0.43	43.41 ± 0.56	45.03 ± 0.07	38.70 ± 0.02	43.12 ± 0.00	$\mathbf{53.68 \pm 0.47}$	51.65 ± 0.50
	CGM	40.51 ± 2.19	35.12 ± 2.40	$\mathbf{48.05 \pm 0.01}$	48.03 ± 0.00	47.11 ± 0.00	41.29 ± 1.70	46.01 ± 0.47
	QLKNN	41.36 ± 0.86	24.58 ± 2.73	41.72 ± 0.05	41.26 ± 0.00	39.95 ± 0.00	33.33 ± 0.21	$\mathbf{43.22 \pm 2.12}$
	TGLFNN	29.41 ± 2.64	18.29 ± 2.46	34.01 ± 0.24	$\mathbf{34.65 \pm 0.00}$	32.80 ± 0.00	24.71 ± 0.05	21.88 ± 1.25
ITER hybrid rampup	BgB	1094 ± 65	1563 ± 96	764.3 ± 35.3	889.3 ± 16.9	798.9 ± 11.8	$\mathbf{1721 \pm 3}$	1153 ± 36
	CGM	929.9 ± 78.3	744.2 ± 112.3	677.4 ± 67.0	827.7 ± 13.6	779.7 ± 9.4	$\mathbf{1356 \pm 140}$	924.6 ± 11.6
	QLKNN	554.1 ± 127.6	928.6 ± 121.8	653.9 ± 50.8	744.9 ± 23.4	696.9 ± 11.3	$\mathbf{1135 \pm 170}$	954.4 ± 36.3
	TGLFNN	238.9 ± 28.4	328.6 ± 54.5	454.6 ± 52.5	$\mathbf{497.9 \pm 49.4}$	439.5 ± 23.1	302.9 ± 81.5	216.9 ± 27.4
ITER hybrid flattop	BgB	3847 ± 554	4487 ± 53	4687 ± 0	4831 ± 0	4772 ± 0	$\mathbf{4838 \pm 0}$	$\mathbf{4838 \pm 0}$
	CGM	4272 ± 59	3448 ± 331	4297 ± 0	4297 ± 0	4297 ± 0	$\mathbf{4298 \pm 0}$	4297 ± 0
	QLKNN	3924 ± 429	3300 ± 445	$\mathbf{4343 \pm 0}$	4342 ± 0	4342 ± 0	4342 ± 0	4343 ± 0
	TGLFNN	4054 ± 441	4098 ± 135	4464 ± 0	4484 ± 0	4477 ± 0	$\mathbf{4485 \pm 0}$	4478 ± 0
ITER hybrid rampdown	BgB	125.8 ± 2.2	148.0 ± 2.2	115.4 ± 3.4	141.3 ± 3.2	146.5 ± 1.1	$\mathbf{197.2 \pm 12.0}$	142.8 ± 0.2
	CGM	68.99 ± 2.45	71.23 ± 14.87	73.67 ± 1.60	86.92 ± 1.65	89.77 ± 1.05	$\mathbf{126.7 \pm 1.9}$	57.83 ± 0.24
	QLKNN	61.07 ± 4.05	65.43 ± 11.02	83.62 ± 1.66	102.3 ± 3.4	98.30 ± 1.97	$\mathbf{127.0 \pm 2.1}$	47.42 ± 0.29
	TGLFNN	43.31 ± 3.96	57.14 ± 9.26	81.42 ± 1.91	96.80 ± 3.45	99.64 ± 2.71	$\mathbf{122.7 \pm 1.9}$	46.44 ± 1.10
ITER advanced rampup	BgB	704.5 ± 127.9	1911 ± 6	1708 ± 188	1916 ± 9	1592 ± 172	$\mathbf{1975 \pm 4}$	1812 ± 9
	CGM	900.3 ± 119.9	877.6 ± 143.7	1554 ± 170	1646 ± 7	1479 ± 7	$\mathbf{1686 \pm 13}$	1209 ± 28
	QLKNN	875.1 ± 95.1	1085 ± 186	1788 ± 196	$\mathbf{2059 \pm 8}$	1963 ± 9	1516 ± 222	1444 ± 31
	TGLFNN	802.6 ± 12.0	1132 ± 216	1532 ± 168	$\mathbf{1940 \pm 33}$	1816 ± 15	1695 ± 147	1295 ± 69
ITER advanced flattop	BgB	1020 ± 255	7399 ± 814	$\mathbf{8339 \pm 0}$	2955 ± 121	8336 ± 0	8338 ± 0	7989 ± 108
	CGM	7413 ± 777	5551 ± 945	8204 ± 0	2907 ± 119	$\mathbf{8207 \pm 0}$	8204 ± 0	8171 ± 1
	QLKNN	4990 ± 1162	7066 ± 781	8028 ± 7	2914 ± 118	8023 ± 0	$\mathbf{8149 \pm 0}$	8016 ± 10
	TGLFNN	87.67 ± 33.77	863.1 ± 523.3	3935 ± 578	1803 ± 77	4248 ± 0	$\mathbf{5146 \pm 0}$	240.2 ± 5.0
ITER advanced rampdown	BgB	427.4 ± 4.5	369.7 ± 5.4	186.3 ± 14.5	237.8 ± 1.6	274.1 ± 3.4	$\mathbf{437.5 \pm 4.1}$	386.8 ± 0.7
	CGM	$\mathbf{506.3 \pm 7.2}$	168.4 ± 49.2	227.3 ± 7.2	271.3 ± 2.6	323.4 ± 6.5	386.1 ± 42.3	500.0 ± 0.4
	QLKNN	41.86 ± 3.48	$\mathbf{88.52 \pm 19.30}$	13.28 ± 0.76	43.89 ± 3.21	16.60 ± 0.47	84.60 ± 4.14	81.52 ± 5.86
	TGLFNN	$\mathbf{-1.725 \pm 0.001}$	-1.736 ± 0.002	-1.744 ± 0.000	-1.739 ± 0.001	-1.732 ± 0.000	-1.730 ± 0.003	-1.735 ± 0.001
SPARC prd rampup	BgB	100.8 ± 0.0	97.43 ± 2.74	72.19 ± 0.01	81.54 ± 0.05	80.37 ± 0.05	$\mathbf{141.4 \pm 0.1}$	94.84 ± 0.22
	CGM	96.83 ± 2.06	85.53 ± 5.70	54.49 ± 0.00	62.11 ± 0.05	53.52 ± 0.04	$\mathbf{101.5 \pm 0.1}$	61.37 ± 0.13
	QLKNN	91.62 ± 5.89	91.02 ± 3.55	57.70 ± 0.00	64.33 ± 0.04	59.38 ± 0.06	$\mathbf{99.03 \pm 0.32}$	66.74 ± 0.69
	TGLFNN	62.34 ± 6.99	90.86 ± 3.79	63.91 ± 0.00	69.74 ± 0.04	67.67 ± 0.02	$\mathbf{103.6 \pm 0.2}$	77.46 ± 0.66
SPARC prd flattop	BgB	149.6 ± 0.2	148.7 ± 0.1	148.7 ± 0.0	$\mathbf{149.8 \pm 0.0}$	149.8 ± 0.0	149.6 ± 0.0	149.7 ± 0.0
	CGM	147.5 ± 0.3	147.0 ± 0.4	$\mathbf{149.2 \pm 0.0}$	148.1 ± 0.0	148.1 ± 0.0	147.9 ± 0.0	148.5 ± 0.0
	QLKNN	$\mathbf{148.7 \pm 0.0}$	147.8 ± 0.2	148.1 ± 0.0	148.2 ± 0.0	148.1 ± 0.0	147.0 ± 0.0	147.1 ± 0.0
	TGLFNN	$\mathbf{149.8 \pm 0.1}$	149.1 ± 0.1	149.4 ± 0.0	149.5 ± 0.0	149.5 ± 0.0	148.7 ± 0.0	148.7 ± 0.0
SPARC prd rampdown	BgB	5.874 ± 0.002	5.158 ± 0.031	5.046 ± 0.000	5.047 ± 0.000	5.047 ± 0.000	$\mathbf{6.193 \pm 0.012}$	5.810 ± 0.029
	CGM	4.954 ± 0.039	4.632 ± 0.113	4.717 ± 0.000	4.715 ± 0.000	4.715 ± 0.000	$\mathbf{5.524 \pm 0.017}$	5.045 ± 0.097
	QLKNN	5.841 ± 0.018	5.430 ± 0.091	5.297 ± 0.000	5.292 ± 0.000	5.292 ± 0.000	$\mathbf{5.921 \pm 0.014}$	5.818 ± 0.077
	TGLFNN	5.833 ± 0.011	5.442 ± 0.052	5.292 ± 0.000	5.292 ± 0.000	5.292 ± 0.000	$\mathbf{6.050 \pm 0.012}$	5.791 ± 0.051
SPARC reduced field rampup	BgB	101.6 ± 0.1	108.7 ± 7.4	87.02 ± 3.61	92.43 ± 2.43	88.85 ± 1.35	$\mathbf{167.7 \pm 0.4}$	94.91 ± 0.06
	CGM	$\mathbf{98.03 \pm 2.05}$	63.17 ± 8.54	23.15 ± 2.63	38.78 ± 1.87	33.34 ± 3.23	85.93 ± 7.44	32.58 ± 0.01
	QLKNN	$\mathbf{99.10 \pm 0.60}$	71.58 ± 6.23	45.51 ± 0.70	51.16 ± 0.72	50.55 ± 1.65	88.90 ± 3.13	44.97 ± 0.06
	TGLFNN	63.17 ± 1.24	67.56 ± 6.59	58.54 ± 0.96	59.19 ± 0.44	58.60 ± 1.19	$\mathbf{118.4 \pm 5.8}$	64.05 ± 0.23
SPARC reduced field flattop	BgB	149.8 ± 0.1	149.7 ± 0.0	149.2 ± 0.0	150.8 ± 0.0	150.8 ± 0.0	150.8 ± 0.0	$\mathbf{150.9 \pm 0.0}$
	CGM	$\mathbf{149.4 \pm 0.2}$	149.0 ± 0.2	148.4 ± 0.0	148.3 ± 0.0	148.5 ± 0.2	148.9 ± 0.0	149.2 ± 0.0
	QLKNN	148.6 ± 0.8	148.7 ± 0.6	148.7 ± 0.0	149.6 ± 0.0	$\mathbf{149.6 \pm 0.0}$	149.5 ± 0.0	149.6 ± 0.0
	TGLFNN	149.0 ± 0.3	149.2 ± 0.1	148.9 ± 0.0	149.4 ± 0.0	$\mathbf{149.4 \pm 0.0}$	149.3 ± 0.0	149.3 ± 0.0
SPARC reduced field rampdown	BgB	28.42 ± 0.27	26.18 ± 0.13	25.51 ± 0.00	29.44 ± 0.01	27.78 ± 0.00	$\mathbf{31.63 \pm 0.02}$	29.55 ± 0.63
	CGM	20.23 ± 0.36	19.96 ± 0.67	20.82 ± 0.00	$\mathbf{22.78 \pm 0.02}$	21.93 ± 0.03	22.72 ± 0.27	21.12 ± 0.61
	QLKNN	27.68 ± 0.25	25.63 ± 0.38	25.47 ± 0.00	28.38 ± 0.01	27.56 ± 0.03	$\mathbf{29.94 \pm 0.05}$	28.19 ± 0.53
	TGLFNN	27.47 ± 0.28	26.39 ± 0.14	25.98 ± 0.00	28.86 ± 0.01	27.58 ± 0.01	$\mathbf{30.06 \pm 0.09}$	29.22 ± 0.76
STEP spp 001 ec hd flattop	BgB	3926 ± 149	4067 ± 173	3973 ± 0	4022 ± 0	4028 ± 0	4295 ± 29	$\mathbf{5113 \pm 53}$
	TGLFNN	80.21 ± 11.47	840.3 ± 336.3	923.0 ± 2.9	1117 ± 1	920.1 ± 1.9	$\mathbf{3572 \pm 5}$	520.6 ± 10.3
KSTAR	World model	4.121 ± 2.293	$\mathbf{239.7 \pm 1.7}$	-34.19 ± 0.00	-34.19 ± 0.00	-34.19 ± 0.00	-38.80 ± 2.10	0.2395 ± 3.2928