Offline Reinforcement Learning for Plasma Control in Nuclear Fusion: Codebase and Benchmark

Yang Fu[1,†] **Haomin Bao**[2,†] **Rohit Sonker**[3] **Xiaoyan Hu**[3]
Aravind Venugopal[3] **Jeff Schneider**[3] **Jiayu Chen**[4,‡]

[1]Central South University [2]Chongqing University
[3]Carnegie Mellon University [4]The University of Hong Kong

[†]Equal contribution. [‡]Corresponding author.

Abstract

Offline reinforcement learning (RL) offers a promising route for developing plasma controllers from historical tokamak data, since online trial-and-error on real devices is costly and risky. However, progress in this direction remains difficult to measure due to the lack of a standardized offline RL benchmark for realistic multi-actuator, long-horizon plasma control problems in nuclear fusion. We introduce RL4F, an Offline Reinforcement Learning Benchmark for Plasma Control in Nuclear Fusion, providing closed-loop evaluation environments and baseline comparisons across four full-profile tracking tasks: rotation, density, temperature, and pressure. The dynamics function underlying the evaluation environment is built from historical discharge data from DIII-D, a real-world Tokamak. We evaluate a broad set of imitation learning and offline RL baselines under a unified protocol. We find that offline model-based RL methods obtain the best average performance on most objectives, although no single method dominates all tasks, highlighting the importance of dynamics modeling in complex, long-horizon plasma control tasks. To foster further research, we open-source the codebase, datasets, and evaluation framework, providing a benchmark not only for the fusion community but also for algorithm development in offline RL.

Code: https://github.com/LucasCJYSDL/Offline-RL-Kit-for-Nuclear-Fusion
Correspondence: Jiayu Chen (jiayuc@hku.hk)

1 Introduction

Nuclear fusion offers a potential route toward abundant, low-carbon energy by harnessing the reactions that power stars (Gi et al., 2020). The tokamak is one of the most promising confinement devices for achieving controllable nuclear fusion, but its operation requires real-time control of hot, unstable, and strongly coupled plasmas. Recent work has shown that reinforcement learning (RL) can be used to train such controllers. In particular, Degrave et al. (2022) demonstrated deep RL-based magnetic control on the TCV tokamak, and subsequent studies have extended RL-based plasma control toward tearing-mode avoidance, profile tracking, and ramp-down control (Tracey et al., 2024; Seo et al., 2024; Char et al., 2023; Wang et al., 2025). These advances suggest that RL can complement conventional plasma-control design by directly optimizing feedback policies in high-dimensional tokamak control problems.

Developing RL controllers directly on real tokamaks is difficult to scale: tokamak operation is expensive, time-limited, and safety-critical. A natural way is to train candidate policies and evaluate their closed-loop behavior in simulation before possible deployment. Physics-based simulators,

such as RAPTOR (Felici et al., 2011), Forward Grad-Shafranov Static (FGE) simulator (Carpanese, 2021), and the more recent TORAX (Citrin et al., 2024), provide reliable tools for forward modeling, trajectory optimization, and controller development. Previous work has demonstrated promising control performance for RL policies trained on these simulators (Degrave et al., 2022; Tracey et al., 2024). However, physics-based simulators are significantly more computationally expensive than traditional RL simulators, especially when RL agents explore the actuator space stochastically during training, which often results in slower convergence for the iterative solver. Moreover, adapting these simulators to a specific tokamak can require substantial modeling choices, parameter identification, and calibration. An alternative approach is to learn control-oriented dynamics models from historical experimental data, yielding simulation environments tied more directly to a particular device (Char et al., 2023; Sonker et al., 2026). Despite recent progress, RL-based profile control remains challenging: plasma profiles are high-dimensional spatial quantities, their dynamics are nonlinear and coupled across multiple actuators, and practical RL controllers still face difficulties in reward specification, steady-state tracking bias, and sample efficiency (Tracey et al., 2024). These challenges motivate a standardized benchmark for data-driven training and offline evaluation of RL algorithms for plasma control in tokamaks.

Our primary contribution is RL4F, a unified benchmark for offline RL in tokamak profile control. A profile refers to the radial spatial variation of a plasma quantity from the core, i.e., the innermost region of the plasma, to the edge, i.e., the outermost region. It is characterized by four key features. 1) **Realistic pre-deployment workflow.** We first train a reference dynamics model from historical DIII-D experimental discharges, and then use this model to generate trajectories for offline policy learning. Candidate algorithms learn only from the model-generated datasets and are evaluated in closed loop on the reference dynamics model, mirroring the practical setting in which controllers must be developed from past experimental data before any real-machine test. 2) **Multi-task profile tracking.** The benchmark covers four full-profile tracking tasks, including rotation, density, temperature, and pressure, which expose different control difficulties and plasma response channels. 3) **Scenario-relevant actuator space.** Policies act through a shared action space consisting of neutral-beam power, neutral-beam torque, gas puffing, and electron-cyclotron heating, covering the main heating, momentum-input, and fueling channels relevant to profile control for DIII-D, a tokamak device located in San Diego. 4) **Large-scale data.** The benchmark contains 5,882 shots and 945,828 transitions after filtering, with fixed training, validation, and test splits. To our knowledge, this is the first benchmark specifically designed for offline RL in fusion plasma control.

We evaluate representative baselines spanning imitation learning, model-free offline RL, and model-based offline RL under the same closed-loop protocol and profile-level tracking metrics. Our evaluation shows that model-based methods generally outperform model-free baselines, while no single algorithm dominates across all profile objectives, highlighting tokamak profile control as a challenging offline RL benchmark.

2 Related Work

Fusion Plasma Control. Nuclear fusion is a leading candidate for sustainable power generation. A central challenge is the control of plasma profiles to achieve stable, high-performance operation. Conventional plasma control typically relies on precomputed feedforward coil current trajectories (Walker and Humphreys, 2006) together with feedback loops for individual target quantities. Such profile control systems have been implemented on several tokamaks, including JET for safety q-profile control (Moreau et al., 2003), TCV for q-profile and electron-temperature control (Barton et al., 2015), and EAST for q-profile control (Wang et al., 2021). These approaches have demonstrated promising performance in a wide range of discharges, but design can be challenging and time-consuming, especially in plasma scenarios where the control quantities are high-dimensional or strongly coupled. More recently, reinforcement learning (RL) has emerged as a new framework for designing feedback controllers in fusion systems. Degrave et al. (2022) train an RL-based controller under the Maximum-a-Posteriori (MPO) Abdolmaleki et al. (2018) framework to track the location, current, and shape across a diverse set of plasma configurations. This line of work is further extended by Tracey et al. (2024). Char et al. (2023) develop an offline RL framework for tracking β_N and plasma rotation quantities and train the controller using Proximal Policy Optimization (PPO), which is later utilized to design a Bayesian optimization (BO)-style controller (Sonker et al., 2025) for mitigating tearing instabilities. Seo et al. (2024) apply the Deep Deterministic Policy Gradient (DDPG) Lillicrap et al.

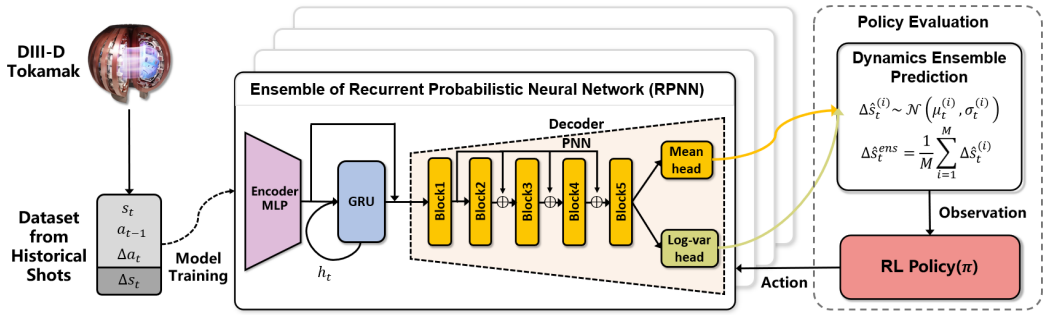


Figure 1: The reference RPNN dynamics ensemble is trained from historical DIII-D operational data and used as the closed-loop environment for evaluating trained policies.

(2016) approach to maintain high-pressure plasma at the H-mode performance while avoiding tearing instabilities. Wang et al. (2025) trains RL policies to avoid disruptions during the ramp-down phase.

Benchmarks for Offline RL. The well-known D4RL dataset (Fu et al., 2021a) benchmarks offline RL in challenging robotic control scenarios with biased data distributions. The Dope benchmark (Fu et al., 2021b), which builds upon D4RL and RL Unplugged (Gulcehre et al., 2020), focuses on off-policy evaluation. Qin et al. (2022) propose NeoRL to mitigate the gap between earlier offline RL benchmarks and real-world scenarios. Liu et al. (2023) present a benchmarking suite that facilitates the development and evaluation of offline safe RL algorithms in both the training and deployment phases. Park et al. (2025) propose OGBench for offline goal-conditioned RL. Compared with these existing benchmarks, fusion plasma control tasks present unique challenges, including highly nonlinear and stochastic dynamics, partial observability, and safety-critical operating constraints. To the best of our knowledge, RL4F is the first offline RL benchmark tailored to fusion plasma control, a critical real-world operational scenario.

3 Simulator

Problem Setup. We formulate tokamak profile control as a finite-horizon Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \rho_0, \gamma, H)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P(s'|s,a)$ is the transition distribution, $r(s,a)$ is the reward function, ρ_0 is the initial-state distribution, γ is the discount factor, and H is the episode horizon. A policy $\pi(a|s)$ aims to maximize the expected discounted return

$$J(\pi) = \mathbb{E}_{\pi,P,\rho_0}\left[\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t)\right]. \tag{1}$$

In offline reinforcement learning, the agent cannot interact with the real environment during training (Levine et al., 2020) and is instead given a fixed dataset of trajectories $\mathcal{D} = \{\tau_i\}_{i=1}^N$, where each trajectory consists of transitions (s_t, a_t, r_t, s_{t+1}) collected by an unknown behavior policy. This setting is well matched to tokamak control, where online trial-and-error is expensive and risky.

We use a recurrent probabilistic neural network (RPNN) to learn plasma dynamics from trajectory data, following prior data-driven tokamak dynamics modeling and offline model-based control work (Char et al., 2023; Sonker et al., 2026). Given the current plasma state s_t, the previous actuator setting a_{t-1}, and the actuator increment $\Delta a_t = a_t - a_{t-1}$, the model predicts a distribution over the next state change, $\Delta s_t = s_{t+1} - s_t$. Specifically, the RPNN outputs the parameters of a Gaussian distribution, $\mu_t, \log\sigma_t^2 = f_\theta(s_t, a_{t-1}, \Delta a_t)$, and the next state is advanced autoregressively by adding the predicted state change to the current state. This recurrent probabilistic formulation allows the model to capture history-dependent plasma evolution while providing uncertainty estimates for long-horizon rollouts. A schematic of the dynamics-model training workflow is shown in Figure 1.

Dynamics Modeling. Given the limited accessibility of real tokamak devices, we train a reference dynamics model as a digital twin. The reference dynamics model is trained from historical DIII-D experimental discharges and is used to generate offline training data and to provide the closed-loop evaluation environment. As shown in Figure 2, we adopt a two-stage RPNN training procedure (Sonker et al., 2026).

The first stage optimizes the predictive mean using a mean-squared error objective. The second stage initializes from the first-stage checkpoint, freezes the predictive backbone, and trains the log-variance head using a negative log-likelihood objective, thereby calibrating uncertainty without changing the learned mean dynamics. Following prior model-based control practice (Chen et al., 2025), we train a bootstrapped ensemble of RPNNs; the predicted log variance captures aleatoric uncertainty, while disagreement across ensemble members provides an estimate of epistemic uncertainty. More training details are given in Appendix B.

We train a 25-member RPNN ensemble on roughly 18,000 historical DIII-D experimental discharges, spanning nearly a decade of data collection. Each shot contains approximately four seconds of data sampled at 20 ms. For profile quantities, including electron temperature, ion temperature, density, pressure, rotation, and the safety-factor q profile, we use ZipFIT reconstructions (Logan et al., 2018), which provide smooth, physics-constrained profile estimates. Following prior tokamak dynamics modeling work (Char et al., 2023), we reduce the dimension of profile quantities with PCA before dynamics-model training. The held-out predictive fidelity of the trained ensemble is reported in Appendix B.1. To simulate a tokamak experiment, i.e., a "shot", we use the ensemble of dynamics models $\{f_{\theta_i}\}_{i=1}^{25}$. Each ensemble member represents a plausible version of the device dynamics, and the ensemble captures uncertainty in the true transition model. Notably, RL policies trained on the same ensemble of dynamics models have been verified to be effective for profile control on DIII-D (Sonker et al., 2026).

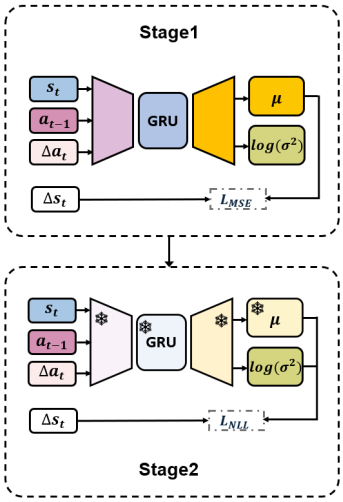


Figure 2: Two-stage training procedure for the dynamics model.

Synthetic Dataset for Offline RL. The offline dataset is generated by initializing rollouts from real flat-top shot states after a warm-up period, replaying the real actuator sequence, and autoregressively rolling out the reference dynamics model. At timestep t, the measured next state is replaced by the ensemble prediction

$$\tilde{\Delta s}_t^{(m)} \sim \mathcal{N}\big(\mu_t^{(m)}, \mathrm{diag}((\sigma_t^{(m)})^2)\big), \qquad \hat{s}_{t+1} = \hat{s}_t + \frac{1}{M}\sum_{m=1}^{M} \tilde{\Delta s}_t^{(m)}, \quad M = 25, \tag{2}$$

where $\mu_t^{(m)}$ and $\sigma_t^{(m)}$ denote the mean and standard deviation predicted by the m-th RPNN ensemble member at time step t, and $\tilde{\Delta s}_t^{(m)}$ denotes a sampled state increment drawn from the corresponding Gaussian predictive distribution. The generated state $\hat{s}_{t+1}$ is then fed back as the input for the next timestep until the end of the shot segment. This procedure preserves the actuator schedules and shot segmentation of the real experiments, while replacing the measured plasma evolution with model-generated dynamics.

We restrict the synthesized data to the flat-top phase of each discharge and remove outlier values, resulting in 5,882 shots and 945,828 timesteps. From these filtered shots, we randomly sample 300 shots for validation and 300 shots for testing, and use the remaining 5,282 shots for training. This yields 849,977 training timesteps, 48,010 validation timesteps, and 47,841 test timesteps, with a maximum shot length of 180 timesteps after filtering. Offline RL and imitation learning algorithms are trained only on the generated training dataset and evaluated on the reference dynamics model.

4 Benchmarking Tasks

We benchmark algorithms on full-profile tracking tasks in tokamak plasma control. Unlike scalar tracking, profile tracking requires regulating high-dimensional spatial quantities across the normalized radial coordinate. This is challenging for offline RL because plasma profiles evolve over long horizons under nonlinear, stochastic dynamics, while policies must infer corrective actions from a fixed dataset.

In this work we address profile tracking across four plasma quantities - rotation, density, electron temperature, and pressure. **Rotation profile control** is beneficial as differential rotation control

impacts suppression of tearing instabilities (Richner et al., 2024). Moreover, edge rotation affects penetration of neutral gas, leading to asymmetric fueling (Emdee et al., 2024; Wilkie et al., 2024), a process crucial for modern high-performance fusion scenarios because it helps manage plasma exhaust and maintain the high density required for fusion without damaging the reactor walls. **Density profile control** regulates the core fuel inventory and fusion reaction rate while avoiding excessive edge density, radiation cooling, confinement degradation, and density-limit disruptions. Experiments and density-limit studies show that peaked or optimized density profiles can extend the operational range and support high-density, high-confinement tokamak plasmas (Greenwald, 2002). **Electron temperature profile control** is important because the profile gradient directly affects heat transport and regulation of this profile helps maintain core thermal energy and confinement while keeping the plasma within favorable regimes (Chapman-Oplopoiou et al., 2025). Finally, **Pressure profile control** is important because the achievable normalized pressure and MHD stability limit depend strongly on the shape of the pressure and current-density profiles. A poorly placed pressure gradient can drive instabilities, while an optimized pressure profile helps sustain high stored energy without crossing stability limits (Strait, 1994). The profile tracking tasks can be modeled as MDPs. We define their observation space, action space, and reward function as follows, the effectiveness of which has been testified in real-world fusion control experiments (Sonker et al., 2026).

Observation Space. We include the present profile, the current target profile, and a future target profile shifted 10 timesteps ahead. Following the PID-inspired observation design used in practical tokamak RL control (Tracey et al., 2024), we also include proportional error terms (P-terms) for both the current and future targets. These terms are defined as the difference between the target profile and the present profile, and provide the policy with explicit feedback on the current tracking error.

Target Sampling. Randomly set targets may be infeasible within a given regime. To ensure feasibility, we construct targets from reference trajectories by sampling two time points and forming a step function with three segments (first–second–first). This strategy ensures that all generated targets are consistent with the system dynamics while introducing sufficient variation to enrich the training distribution.

Action Space. The policy controls NBI power, NBI torque, gas-puffing voltage, and total ECH power. NBI torque directly influences the plasma rotation profile at the beam deposition location, which is typically situated near the plasma core. Both torque and power originate from the same neutral beam source, playing a central role in sustaining the plasma. We follow a counter-beam configuration of neutral beams, which allows independent control of injected power and torque.

Reward Function. The reward function is the mean squared tracking error on the whole profile (33 dimensions per timestep), which is reconstructed from corresponding PCA components.

$$r_t = - \left\| p_{\text{target}}(t) - p(t) \right\|_2^2 \tag{3}$$

where $p_{\text{target}}(t)$ is the target profile at time t, and $p(t)$ is the actual profile.

5 Benchmarking Results

We benchmark diverse baselines on four profile-tracking tasks. All policies are trained on a static dataset synthesized by the reference dynamics model and evaluated in closed loop on the same reference model, as illustrated in Figure 1. Following Sonker et al. (2026), evaluation rollouts use the mean prediction of the reference ensemble as the next-state estimate, and performance is measured by profile-tracking error on held-out test shots.

Baselines The evaluated methods include model-free offline RL baselines –TD3+Behavior Cloning (TD3BC) (Fujimoto and Gu, 2021), Conservative Q-Learning (CQL) (Kumar et al., 2020), Implicit Q-Learning (IQL) (Kostrikov et al., 2022), Ensemble-Diversified Actor-Critic (EDAC) (An et al., 2021), and Mildly Conservative Q-Learning (MCQ) (Lyu et al., 2022); model-based offline RL baselines – PPO (Schulman et al., 2017), Conservative Offline Model-Based Policy Optimization (COMBO) (Yu et al., 2021), Model-based Offline Policy Optimization (MOPO) (Yu et al., 2020), Model-Bellman Inconsistency (MOBILE) (Sun et al., 2023), Robust Adversarial Model-Based Offline Reinforcement Learning (RAMBO) (Rigter et al., 2022), and Bayes Adaptive Monte Carlo Tree Search (BAMCTS) (Chen et al., 2026); and Goal-Conditioned Imitation Learning (GCIL) (Ding et al., 2019). Full experiment details in Appendix C

Model-based offline RL baselines train their own dynamics models from the synthetic trajectories generated by the reference model. We use the same RPNN ensemble architecture and two-stage training pipeline described in Section 3, again training a 25-member ensemble. The held-out predictive fidelity of the learned dynamics models (for offline RL) is summarized in Appendix B.1. During model-based RL training, for each rollout, we sample one ensemble member and then generate the trajectory autoregressively by sampling from that member's predictive distribution, parameterized by its mean and log-variance. A new ensemble member is selected whenever a new rollout trajectory is sampled. Following Chua et al. (2018), this procedure trains policies to perform across the range of plausible learned dynamics.

Evaluation Metrics We evaluate the baselines on each profile tracking task using closed-loop simulation over 300 held-out test shots, with 10 random seeds per shot. At each timestep, the environment provides the target profile and the agent-achieved profile.

We compute the tracking error at each normalized radial location ψ_N, where N ranges uniformly from 0 (core) to 1 (edge). There are 33 radial locations in total. The per-location tracking metric is RMSE, computed from the squared deviation between the target and achieved profiles over the rollout horizon. We report the mean RMSE and its standard error (SE) across all rollout instances at six selected radial locations. Lower values indicate better tracking performance.

$$\overline{\mathrm{RMSE}}(\psi_N) = \frac{1}{NS} \sum_{i=1}^{N} \sum_{j=1}^{S} \mathrm{RMSE}^{(i,j)}(\psi_N), \tag{4}$$

where N is the number of test shots and S is the number of random seeds per shot. $\mathrm{RMSE}^{(i,j)}(\psi_N)$ denotes the per-rollout RMSE at radial location ψ_N for shot i and seed j.

Main Results Tables 1–4 summarize the tracking performance of all baselines across the four tasks. Across all tasks, offline RL methods achieve substantially lower tracking errors than the goal-conditioned imitation learning baseline, indicating that purely imitation-based learning is insufficient for these long-horizon profile-tracking problems.

For rotation tracking, RAMBO achieves the lowest average RMSE (8.03), narrowly outperforming COMBO and MOPO. Model-free baselines perform substantially worse. Rotation profiles have pronounced radial variation from the plasma core to the edge and respond primarily through torque actuation, requiring sustained correction over long horizons after each target switch. In this setting, conservative or behavior-regularized model-free methods are more restricted in their ability to deviate from the behavior distribution within the static training dataset when active correction is needed.

Table 1: Rotation profile tracking error (RMSE ↓). Values are reported as mean ± standard error over test shots.

Algorithm	$\psi_{0.09}$		$\psi_{0.18}$		$\psi_{0.39}$		$\psi_{0.58}$		$\psi_{0.79}$		$\psi_{0.88}$		Average	
	Mean	SE	Mean	SE	Mean	SE	Mean	SE	Mean	SE	Mean	SE	Mean	SE
GCIL	32.45	0.376	28.46	0.336	20.73	0.240	15.72	0.180	11.76	0.135	10.28	0.116	17.76	0.047
TD3BC	25.10	0.318	20.99	0.274	14.81	0.189	11.35	0.141	8.69	0.104	7.79	0.092	13.13	0.038
CQL	24.88	0.364	21.09	0.319	15.48	0.241	12.13	0.192	9.28	0.143	8.23	0.120	13.58	0.044
IQL	29.64	0.439	25.72	0.391	18.85	0.275	14.47	0.203	10.87	0.149	9.48	0.129	16.23	0.051
EDAC	26.30	0.427	20.78	0.359	14.43	0.261	11.41	0.210	9.10	0.157	8.31	0.129	13.30	0.048
MCQ	28.45	0.344	24.08	0.310	17.19	0.224	13.22	0.167	10.10	0.120	8.90	0.102	15.10	0.043
PPO	13.96	0.159	10.41	0.138	9.81	0.120	9.85	0.124	8.51	0.112	7.79	0.103	9.36	0.023
COMBO	17.67	0.309	12.85	0.252	8.02	0.160	6.29	0.118	5.29	0.085	5.43	0.074	8.09	0.032
MOPO	18.21	0.241	13.47	0.213	9.19	0.163	7.32	0.130	5.62	0.093	5.20	0.076	8.66	0.030
MOBILE	48.07	0.729	41.59	0.620	30.98	0.438	24.50	0.338	18.93	0.258	16.37	0.221	26.86	0.083
RAMBO	**17.22**	**0.280**	**12.17**	**0.216**	**7.56**	**0.139**	**6.33**	**0.106**	**5.73**	**0.082**	**5.93**	**0.078**	**8.03**	**0.029**
BAMCTS	43.47	0.488	38.65	0.433	28.92	0.318	22.24	0.244	16.28	0.181	13.69	0.157	24.38	0.063

For density tracking, COMBO achieves the lowest average RMSE of 0.691, with MOPO (0.727) and PPO (0.733) close behind. The relatively small gap between model-based and model-free methods suggests that the offline dataset provides sufficient coverage for learning effective density-control behavior. By contrast, RAMBO performs noticeably worse, with an average RMSE of 1.583,

indicating that robustness-oriented model-based training does not consistently improve performance across all profile control tasks.

Table 2: Density profile tracking error (RMSE ↓). Values are reported as mean ± standard error over test shots.

Algorithm	$\psi_{0.09}$		$\psi_{0.18}$		$\psi_{0.39}$		$\psi_{0.58}$		$\psi_{0.79}$		$\psi_{0.88}$		Average	
	Mean	SE	Mean	SE	Mean	SE	Mean	SE	Mean	SE	Mean	SE	Mean	SE
GCIL	1.120	0.0135	1.031	0.0128	0.920	0.0121	0.864	0.0117	0.792	0.0115	0.760	0.0113	0.860	0.0021
TD3BC	1.093	0.0132	1.009	0.0126	0.900	0.0117	0.840	0.0113	0.761	0.0109	0.727	0.0107	0.836	0.0021
CQL	1.127	0.0140	1.033	0.0132	0.922	0.0123	0.868	0.0120	0.807	0.0120	0.775	0.012	0.865	0.0022
IQL	1.088	0.0130	0.998	0.0124	0.885	0.0118	0.830	0.0117	0.768	0.0117	0.738	0.0116	0.831	0.0021
EDAC	1.281	0.0157	1.197	0.0151	1.099	0.0144	1.051	0.0142	0.998	0.0145	0.969	0.0145	1.037	0.0026
MCQ	1.059	0.0119	0.983	0.0115	0.891	0.0109	0.840	0.0106	0.768	0.0104	0.735	0.0102	0.828	0.0019
PPO	1.048	0.0131	0.914	0.0125	0.770	0.0115	0.717	0.0108	0.661	0.0100	0.630	0.0096	0.733	0.0020
COMBO	**0.966**	**0.0115**	**0.870**	**0.0107**	**0.745**	**0.0098**	**0.676**	**0.0094**	**0.602**	**0.0091**	**0.578**	**0.0088**	**0.691**	**0.0017**
MOPO	0.988	0.0108	0.898	0.0105	0.784	0.0103	0.723	0.0102	0.648	0.0096	0.621	0.0093	0.727	0.0018
MOBILE	1.325	0.0144	1.273	0.0139	1.225	0.0135	1.206	0.0137	1.195	0.0143	1.197	0.0145	1.184	0.0025
RAMBO	1.576	0.0147	1.596	0.0138	1.671	0.0132	1.699	0.0127	1.638	0.0115	1.600	0.0109	1.583	0.0024
BAMCTS	1.220	0.0143	1.165	0.0142	1.101	0.0139	1.058	0.0135	0.975	0.0128	0.938	0.0125	1.023	0.0024

For temperature tracking, MOPO achieves the best average RMSE (0.193), outperforming PPO (0.240), MCQ (0.244), and COMBO (0.264). Temperature dynamics involve delayed responses and accumulated transport effects, making long-horizon accuracy particularly important. Methods that penalize uncertain model predictions (MOPO) are well suited to this regime, as they avoid compounding errors from unreliable model regions.

Table 3: Temperature profile tracking error (RMSE ↓). Values are reported as mean ± standard error over test shots.

Algorithm	$\psi_{0.09}$		$\psi_{0.18}$		$\psi_{0.39}$		$\psi_{0.58}$		$\psi_{0.79}$		$\psi_{0.88}$		Average	
	Mean	SE	Mean	SE	Mean	SE	Mean	SE	Mean	SE	Mean	SE	Mean	SE
GCIL	0.492	0.0055	0.461	0.0052	0.393	0.0045	0.325	0.0037	0.238	0.0028	0.206	0.0025	0.323	0.0008
TD3BC	0.468	0.0051	0.429	0.0047	0.360	0.0039	0.298	0.0033	0.223	0.0026	0.195	0.0024	0.301	0.0007
CQL	0.563	0.0054	0.526	0.0049	0.443	0.0041	0.360	0.0034	0.259	0.0026	0.223	0.0023	0.362	0.0008
IQL	0.465	0.0052	0.429	0.0049	0.362	0.0041	0.301	0.0034	0.225	0.0027	0.196	0.0025	0.302	0.0007
EDAC	0.735	0.0073	0.709	0.0070	0.624	0.0061	0.518	0.0053	0.376	0.0043	0.330	0.0041	0.507	0.0011
MCQ	0.382	0.0046	0.349	0.0043	0.293	0.0037	0.241	0.0030	0.178	0.0023	0.157	0.0021	0.244	0.0006
PPO	0.444	0.0040	0.362	0.0034	0.245	0.0029	0.204	0.0026	0.187	0.0025	0.179	0.0026	0.240	0.0006
COMBO	0.408	0.0038	0.371	0.0035	0.310	0.0030	0.261	0.0026	0.203	0.0022	0.180	0.0021	0.264	0.0006
MOPO	**0.276**	**0.0032**	**0.246**	**0.0031**	**0.212**	**0.0030**	**0.190**	**0.0030**	**0.163**	**0.0030**	**0.156**	**0.0030**	**0.193**	**0.0005**
MOBILE	0.623	0.0066	0.580	0.0062	0.488	0.0054	0.403	0.0044	0.304	0.0033	0.273	0.0030	0.408	0.0009
RAMBO	1.044	0.0105	1.004	0.0100	0.848	0.0087	0.652	0.0071	0.406	0.0048	0.332	0.0041	0.655	0.0016
BAMCTS	0.814	0.0115	0.770	0.0107	0.658	0.0089	0.537	0.0072	0.383	0.0052	0.330	0.0046	0.534	0.0015

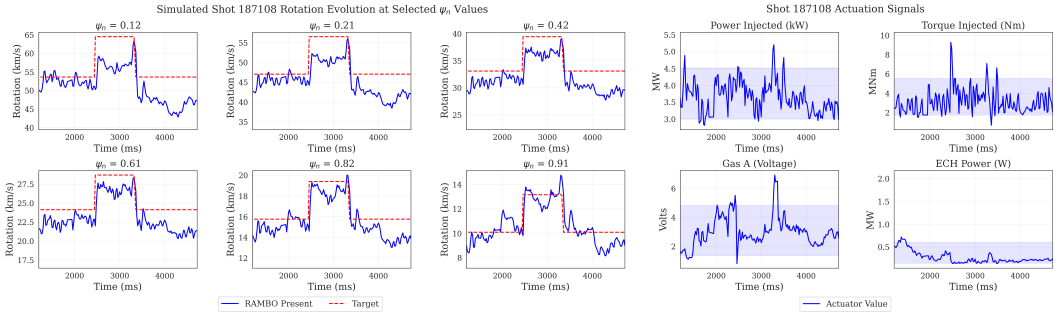
Pressure tracking is likely the most strongly coupled task. The pressure profile depends directly on both density and temperature, and it is also influenced by the current profile and actuator dynamics. In this task, MOPO achieves the lowest average RMSE of 5198.6, followed by RAMBO at 5358.2. Among model-free methods, EDAC performs best with an average RMSE of 6016.8. These results suggest that uncertainty-aware methods may be better suited to pressure tracking.

Table 4: Pressure profile tracking error (RMSE ↓). Values are reported as mean ± standard error over test shots.

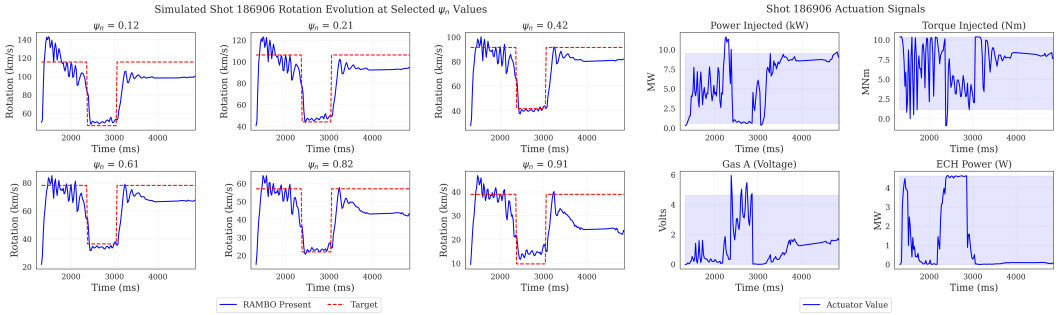
Algorithm	$\psi_{0.09}$		$\psi_{0.18}$		$\psi_{0.39}$		$\psi_{0.58}$		$\psi_{0.79}$		$\psi_{0.88}$		Average	
	Mean	SE	Mean	SE	Mean	SE	Mean	SE	Mean	SE	Mean	SE	Mean	SE
GCIL	17346.5	177.2	15052.4	160.0	10897.1	129.6	8174.0	103.5	4943.7	64.2	3300.2	42.8	8775.3	25.4
TD3BC	15496.3	154.0	13049.8	133.2	9079.8	108.5	6998.2	92.6	4558.2	61.6	3138.1	42.1	7628.9	21.8
CQL	13685.7	144.8	11030.1	123.6	6921.7	92.8	5275.7	74.2	3614.5	48.4	2544.5	33.1	6162.2	19.1
IQL	15518.4	171.7	13089.6	153.7	9041.5	126.1	6788.6	104.7	4312.7	66.9	2951.4	45.0	7525.8	23.8
EDAC	12405.6	139.3	9868.5	114.9	6624.9	81.4	5648.2	68.9	4114.2	48.9	2922.3	34.4	6016.8	17.2
MCQ	14885.6	149.0	12482.0	129.0	8536.6	101.2	6457.9	83.5	4155.5	54.1	2852.5	36.8	7174.3	20.8
PPO	14511.3	143.5	11680.0	117.8	7149.0	80.1	5178.8	63.5	3511.9	41.1	2482.2	28.2	6333.2	18.8
COMBO	14014.7	135.6	11379.7	110.5	7002.4	73.7	4959.2	56.6	3215.5	36.8	2242.7	25.5	6095.9	18.0
MOPO	**12706.3**	**131.5**	**9894.5**	**105.4**	**5534.0**	**66.7**	**3997.4**	**52.9**	**2892.6**	**37.4**	**2094.5**	**26.5**	**5198.6**	**16.5**
MOBILE	15342.5	150.7	12692.0	125.8	8178.6	90.7	5738.4	75.3	3464.5	51.9	2353.1	36.1	6854.9	20.7
RAMBO	12437.8	133.2	9786.2	109.9	5790.4	77.2	4391.8	63.4	3171.1	43.2	2276.1	30.0	5358.2	17.0
BAMCTS	18697.5	180.6	16093.4	152.6	11559.1	119.0	8715.9	103.0	5420.3	69.6	3673.5	47.7	9411.5	25.6

Methods without such stabilization mechanisms tend to obtain higher errors.

Across all the tasks, MOPO is the most robust method; COMBO is strongest on density; RAMBO excels on rotation and pressure but fails on density and temperature; PPO is a strong baseline on rotation, density, and temperature, but falls short on the more coupled pressure profile tracking task. Thus, no single algorithm is universally beneficial across plasma profile control objectives, calling for stronger offline RL algorithms.



(a) Higher rotation target



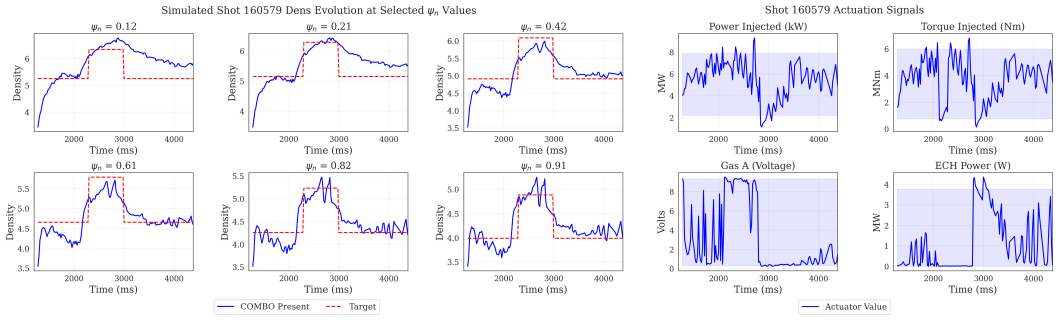
(b) Lower rotation target

Figure 3: Simulated results of RAMBO applied to Rotation profile tracking for shot 187108 and 186906 using the dynamics-model environment. Two target patterns are tested: (a) increasing the profile and returning, and (b) decreasing the profile and returning. Left plots show the rotation profile at different normalized flux values (ψ_{n}). Right plots show the RL-controlled actuator signals. Both cases demonstrate strong tracking performance in the absence of the sim-to-real gap.

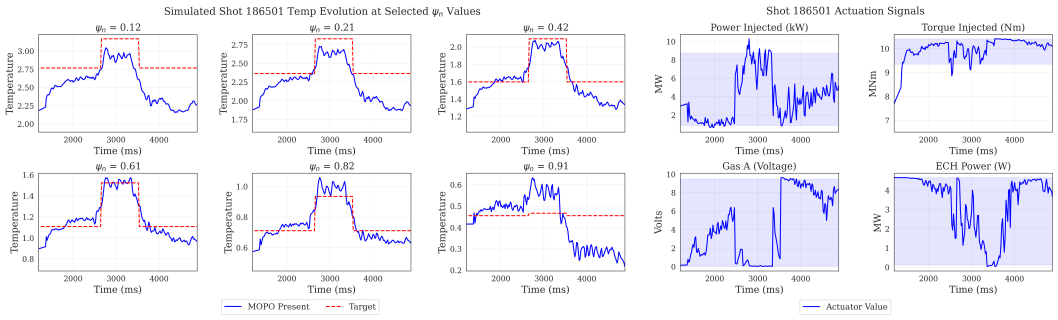
A second consistent trend is that tracking near the plasma core remains more difficult than tracking near the edge. Errors at smaller normalized flux values are systematically higher regardless of the algorithm or task, because both the profile magnitude and temporal variation are substantially larger near the core. This tendency is also qualitatively reflected in Appendix A.

The small standard errors across all tables indicate that performance is consistent across test shots and that the observed gaps reflect systematic differences between methods rather than noise or a few favorable trajectories.

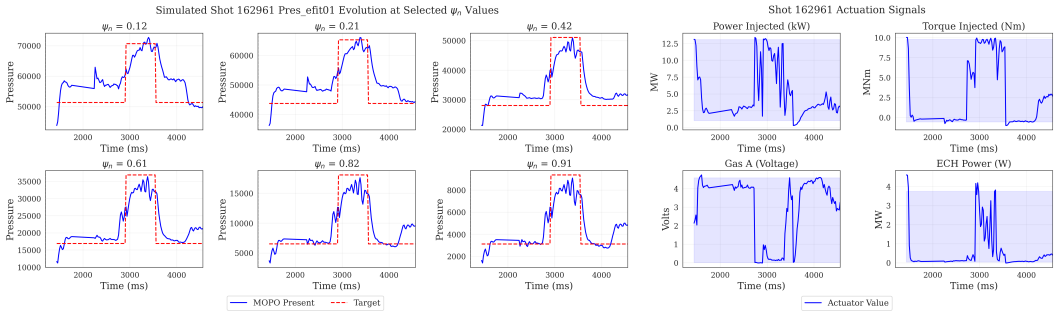
The main text shows higher- and lower-target rotation rollouts (Fig. 3), as well as higher-target rollouts for density, temperature, and pressure (Fig. 4). The corresponding lower-target cases for density, temperature, and pressure are provided in Appendix A. Across all tasks, the policies exhibit a consistent closed-loop pattern: each target switch triggers an immediate coordinated response across all four actuators, after which the profile converges smoothly to the new setpoint. The dominant actuation pathway differs by task: rotation is driven primarily by torque, density by gas puffing, temperature by co-dominant ECH and beam power with active gas puffing suppression, and pressure by balanced coordination of all four actuators. Overall, although there is a consistent offset in the tracking values, the controller responded appropriately to each step change, coordinating the actuators in a physically meaningful way.



(a) Higher density target



(b) Higher temperature target



(c) Higher pressure target

Figure 4: Simulated results for the higher-target cases of the remaining three tasks in the dynamics-model environment. From top to bottom, the panels show COMBO on density profile tracking (shot 160579), MOPO on temperature profile tracking (shot 186501), and MOPO on pressure profile tracking (shot 162961).

6 Conclusion and Discussion

We introduced RL4F, an open-source benchmark for offline RL in nuclear-fusion plasma profile control, releasing reference dynamics models trained on real-world fusion operational data, synthetic datasets for offline RL training, and the full codebase to support reproducibility and community adoption. The benchmark provides standardized task definitions and a unified evaluation protocol across four profile-tracking objectives: rotation, density, temperature, and pressure. Physics-based simulators such as RAPTOR (Felici et al., 2011) and TORAX (Citrin et al., 2024) can struggle to capture complex fusion dynamics, and calibrating them to a specific tokamak can be nontrivial. In contrast, our benchmark is built directly from experimental discharge data and designed specifically for offline RL evaluation. To the best of our knowledge, it is the first offline RL benchmark for nuclear-fusion plasma control, providing a common testbed for comparing algorithms under fixed datasets and standardized evaluation conditions.

A current limitation is that RL4F is constructed from DIII-D data alone. As a result, it remains unclear how well the learned dynamics model, or the relative performance of different algorithms, transfers to other tokamak devices. Addressing this limitation will require broader collaboration

across the fusion community and the inclusion of data from additional fusion devices. Despite these limitations, we believe this benchmark is a useful step toward accelerating progress in both nuclear fusion and RL algorithm development.

References

Abdolmaleki, A., Springenberg, J. T., Tassa, Y., Munos, R., Heess, N., and Riedmiller, M. (2018). Maximum a posteriori policy optimisation. In *International Conference on Learning Representations*.

An, G., Moon, S., Kim, J.-H., and Song, H. O. (2021). Uncertainty-based offline reinforcement learning with diversified Q-ensemble. In *Advances in Neural Information Processing Systems*, volume 34, pages 7436–7447. Curran Associates, Inc.

Barton, J. E., Wehner, W. P., Schuster, E., Felici, F., and Sauter, O. (2015). Simultaneous closed-loop control of the current profile and the electron temperature profile in the tcv tokamak. In *2015 American Control Conference (ACC)*, pages 3316–3321.

Carpanese, F. (2021). *Development of free-boundary equilibrium and transport solvers for simulation and real-time interpretation of tokamak experiments*. PhD thesis, EPFL, Lausanne.

Chapman-Oplopoiou, B., Walker, J., Hatch, D. R., Görler, T., and contributors, J. (2025). Composition of electron temperature gradient driven plasma turbulence in jet-ilw tokamak plasmas. *Phys. Rev. Res.*, 7:L012004.

Char, I., Abbate, J., Bardoczi, L., Boyer, M., Chung, Y., Conlin, R., Erickson, K., Mehta, V., Richner, N., Kolemen, E., and Schneider, J. (2023). Offline model-based reinforcement learning for tokamak control. In Matni, N., Morari, M., and Pappas, G. J., editors, *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*, volume 211 of *Proceedings of Machine Learning Research*, pages 1357–1372. PMLR.

Chen, J., Xu, L., Chen, W., and Schneider, J. (2026). Bayes adaptive monte carlo tree search for offline model-based reinforcement learning. In *International Conference on Learning Representations*. Poster.

Chen, J., Xu, L., Venugopal, A., and Schneider, J. (2025). Policy-driven world model adaptation for robust offline model-based reinforcement learning.

Chua, K., Calandra, R., McAllister, R., and Levine, S. (2018). Deep reinforcement learning in a handful of trials using probabilistic dynamics models.

Citrin, J., Goodfellow, I., Raju, A., Chen, J., Degrave, J., Donner, C., Felici, F., Hamel, P., Huber, A., Nikulin, D., Pfau, D., Tracey, B., Riedmiller, M., and Kohli, P. (2024). TORAX: A fast and differentiable tokamak transport simulator in JAX.

Degrave, J., Felici, F., Buchli, J., Neunert, M., Tracey, B., Carpanese, F., Ewalds, T., Hafner, R., Abdolmaleki, A., de las Casas, D., Donner, C., Fritz, L., Galperti, C., Huber, A., Keeling, J., Tsimpoukelli, M., Kay, J., Merle, A., Moret, J.-M., Noury, S., Pesamosca, F., Pfau, D., Sauter, O., Sommariva, C., Coda, S., Duval, B., Fasoli, A., Kohli, P., Kavukcuoglu, K., Hassabis, D., and Riedmiller, M. (2022). Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419.

Ding, Y., Florensa, C., Abbeel, P., and Phielipp, M. (2019). Goal-conditioned imitation learning. In *Advances in Neural Information Processing Systems*, volume 32, pages 15298–15309. Curran Associates, Inc.

Emdee, E., Horvath, L., Bortolon, A., and Wilkie, G. (2024). The influence of rotation and sol drifts on poloidal asymmetries of pedestal fueling. In *APS Division of Plasma Physics Meeting Abstracts*, volume 2024, pages GO06–014.

Felici, F., Sauter, O., Coda, S., Duval, B. P., Goodman, T. P., Moret, J.-M., and Paley, J. I. (2011). Real-time physics-model-based simulation of the current density profile in tokamak plasmas. *Nuclear Fusion*, 51(8):083052.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. (2021a). D4rl: Datasets for deep data-driven reinforcement learning.

Fu, J., Norouzi, M., Nachum, O., Tucker, G., Wang, Z., Novikov, A., Yang, M., Zhang, M. R., Chen, Y., Kumar, A., Paduraru, C., Levine, S., and Paine, T. L. (2021b). Benchmarks for deep off-policy evaluation.

Fujimoto, S. and Gu, S. S. (2021). A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 20132–20145. Curran Associates, Inc.

Gi, K., Sano, F., Akimoto, K., Hiwatari, R., and Tobita, K. (2020). Potential contribution of fusion power generation to low-carbon development under the paris agreement and associated uncertainties. *Energy Strategy Reviews*, 27:100432.

Greenwald, M. (2002). Density limits in toroidal plasmas. *Plasma Physics and Controlled Fusion*, 44(8):R27–R53.

Gulcehre, C., Wang, Z., Novikov, A., Paine, T., Gómez, S., Zolna, K., Agarwal, R., Merel, J. S., Mankowitz, D. J., Paduraru, C., Dulac-Arnold, G., Li, J., Norouzi, M., Hoffman, M., Heess, N., and de Freitas, N. (2020). Rl unplugged: A suite of benchmarks for offline reinforcement learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7248–7259. Curran Associates, Inc.

Kostrikov, I., Nair, A., and Levine, S. (2022). Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. (2020). Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 1179–1191. Curran Associates, Inc.

Levine, S., Kumar, A., Tucker, G., and Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2016). Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*.

Liu, Z., Guo, Z., Lin, H., Yao, Y., Zhu, J., Cen, Z., Hu, H., Yu, W., Zhang, T., Tan, J., and Zhao, D. (2023). Datasets and benchmarks for offline safe reinforcement learning.

Logan, N. C., Grierson, B. A., Haskey, S. R., Smith, S. P., Meneghini, O., and Eldon, D. (2018). OMFIT tokamak profile data fitting and physics analysis. *Fusion Science and Technology*, 74(1-2):125–134.

Lyu, J., Ma, X., Li, X., and Lu, Z. (2022). Mildly conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 1711–1724. Curran Associates, Inc.

Moreau, D., Crisanti, F., Litaudon, X., Mazon, D., Vries, P. D., Felton, R., Joffrin, E., Laborde, L., Lennholm, M., Murari, A., Pericoli-Ridolfini, V., Riva, M., Tala, T., Tresset, G., Zabeo, L., Zastrow, K., and contributors to the EFDA-JET Workprogramme (2003). Real-time control of the q-profile in jet for steady state advanced tokamak operation. *Nuclear Fusion*, 43(9):870.

Park, S., Frans, K., Eysenbach, B., and Levine, S. (2025). OGBench: Benchmarking offline goal-conditioned RL. In *The Thirteenth International Conference on Learning Representations*.

Qin, R.-J., Zhang, X., Gao, S., Chen, X.-H., Li, Z., Zhang, W., and Yu, Y. (2022). Neorl: A near real-world benchmark for offline reinforcement learning. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24753–24765. Curran Associates, Inc.

Richner, N., Bardóczi, L., Callen, J., La Haye, R., Logan, N., and Strait, E. (2024). Use of differential plasma rotation to prevent disruptive tearing mode onset from 3-wave coupling. *Nuclear Fusion*, 64(10):106036.

Rigter, M., Lacerda, B., and Hawes, N. (2022). RAMBO-RL: Robust adversarial model-based offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 16082–16097. Curran Associates, Inc.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms.

Seo, J., Kim, S., Jalalvand, A., Conlin, R., Rothstein, A., Abbate, J., Erickson, K., Wai, J., Shousha, R., and Kolemen, E. (2024). Avoiding fusion plasma tearing instability with deep reinforcement learning. *Nature*, 626(8000):746–751.

Sonker, R., Capone, A., Rothstein, A., Kaga, H. J. F., Kolemen, E., and Schneider, J. (2025). Multi-timescale dynamics model bayesian optimization for plasma stabilization in tokamaks. In *Forty-second International Conference on Machine Learning*.

Sonker, R., Kaga, H. J. F., Chen, J., Rothstein, A., Char, I., Shousha, R., Kolemen, E., and Schneider, J. (2026). Offline reinforcement learning for rotation profile control in tokamaks.

Strait, E. (1994). Stability of high beta tokamak plasmas. *Physics of Plasmas*, 1(5):1415–1431.

Sun, Y., Zhang, J., Jia, C., Lin, H., Ye, J., and Yu, Y. (2023). Model-Bellman inconsistency for model-based offline reinforcement learning. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 33177–33194. PMLR.

Tracey, B. D., Michi, A., Chervonyi, Y., Davies, I., Paduraru, C., Lazic, N., Felici, F., Ewalds, T., Donner, C., Galperti, C., Buchli, J., Neunert, M., Huber, A., Evens, J., Kurylowicz, P., Mankowitz, D. J., Riedmiller, M., and The TCV Team (2024). Towards practical reinforcement learning for tokamak magnetic control. *Fusion Engineering and Design*, 200:114161.

Walker, M. L. and Humphreys, D. A. (2006). Valid coordinate systems for linearized plasma shape response models in tokamaks. *Fusion Science and Technology*, 50(4):473–489.

Wang, A. M., Rea, C., So, O., Dawson, C., Garnier, D. T., and Fan, C. (2025). Active ramp-down control and trajectory design for tokamaks with neural differential equations and reinforcement learning. *Communications Physics*, 8(1):231.

Wang, Z., Wang, H., Schuster, E., Luo, Z., Huang, Y., Yuan, Q., Xiao, B., and Humphreys, D. (2021). Optimal shaping of the safety factor profile in the east tokamak. In *2021 IEEE Conference on Control Technology and Applications (CCTA)*, pages 63–68.

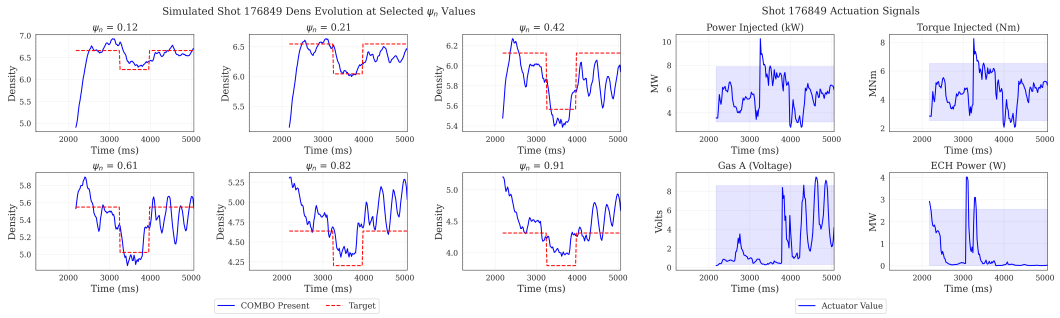
Wilkie, G., Laggner, F., Hager, R., Rosenthal, A., Ku, S.-H., Churchill, R. M., Horvath, L., Chang, C. S., and Bortolon, A. (2024). Reconstruction and interpretation of ionization asymmetry in magnetic confinement via synthetic diagnostics. *Nuclear Fusion*, 64(8):086028.

Yu, T., Kumar, A., Rafailov, R., Rajeswaran, A., Levine, S., and Finn, C. (2021). COMBO: Conservative offline model-based policy optimization. In *Advances in Neural Information Processing Systems*, volume 34, pages 28954–28967. Curran Associates, Inc.

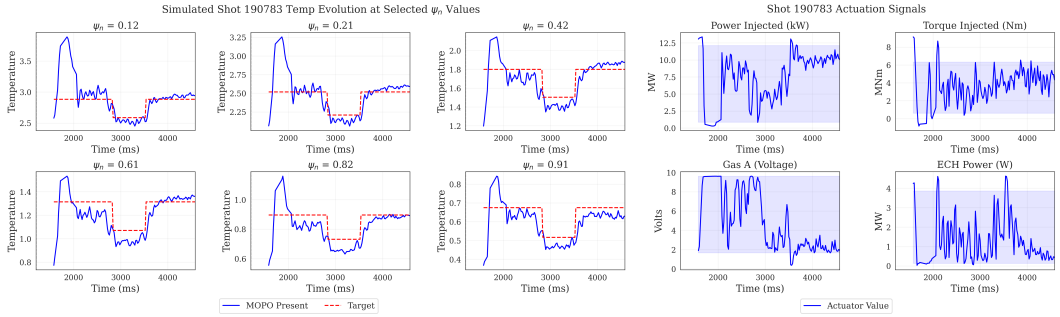
Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J. Y., Levine, S., Finn, C., and Ma, T. (2020). MOPO: Model-based offline policy optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 14129–14142. Curran Associates, Inc.

A Additional Results

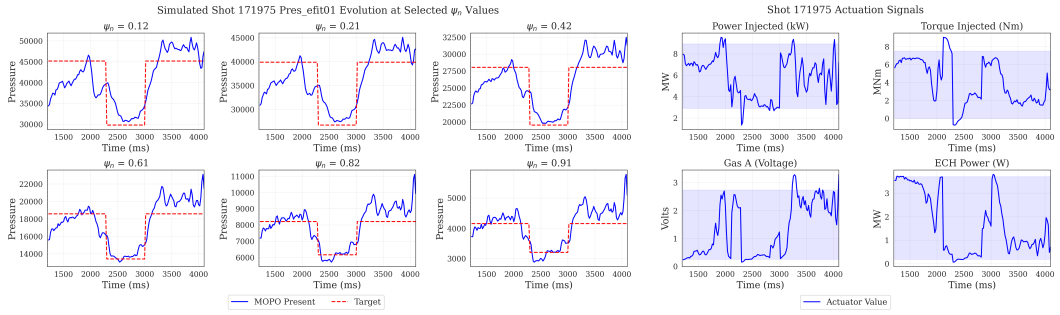
We provide additional visualizations of closed-loop profile tracking trajectories for each of the four benchmark tasks. Figures 6- 9 show the temporal evolution of the full radial profiles at selected time instances for two representative test shots per task, corresponding to a higher and a lower target profile, respectively. In each figure, solid lines denote the model-predicted profile and dashed lines indicate the target profile.



(a) Lower density target

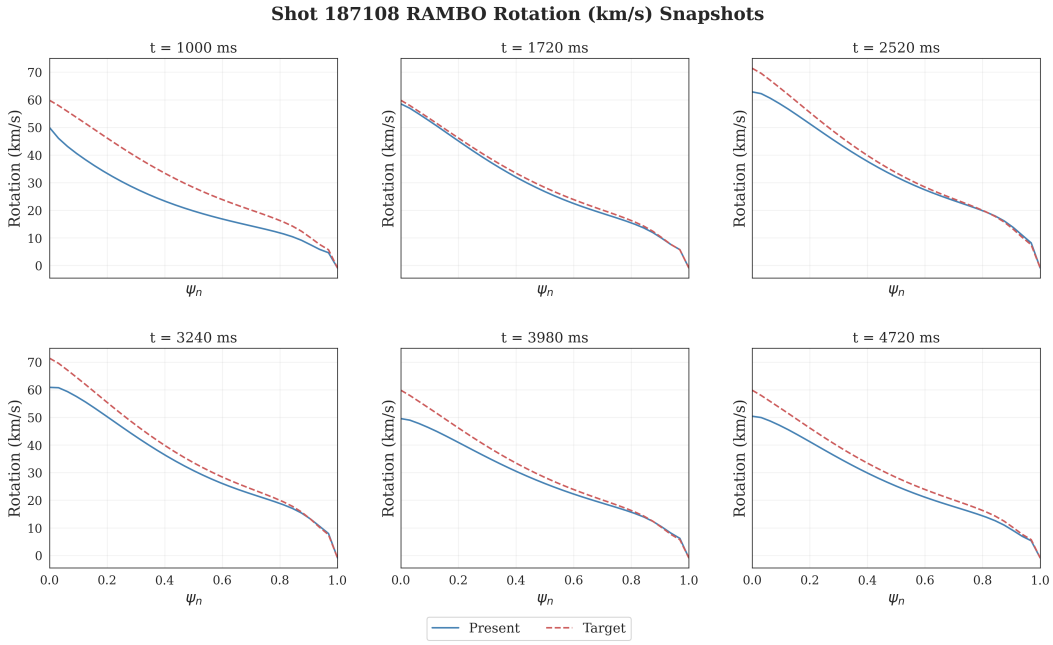


(b) Lower temperature target

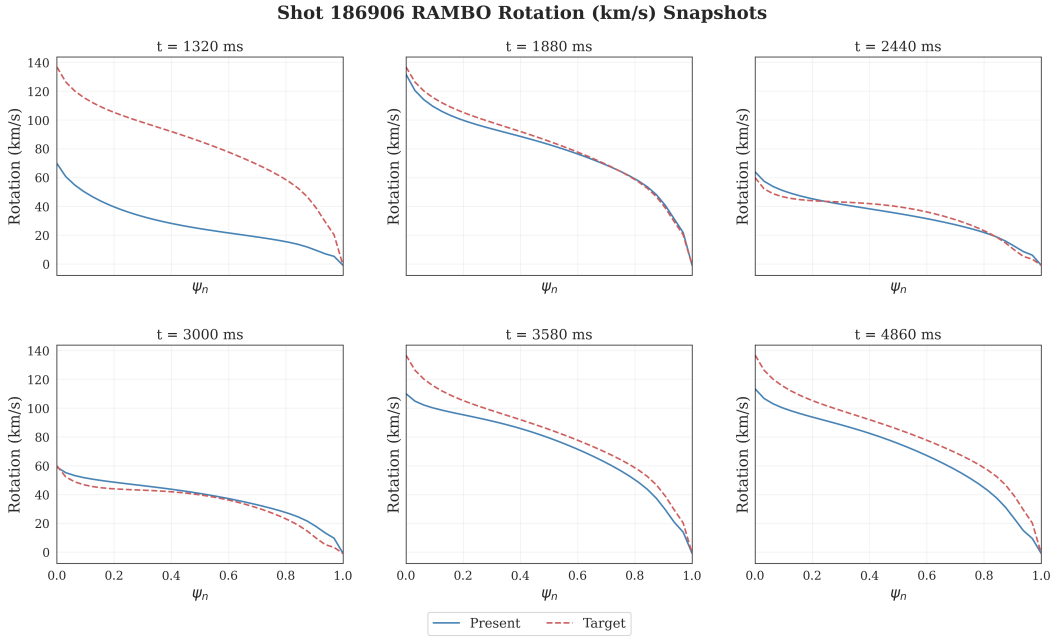


(c) Lower pressure target

Figure 5: Simulated results for the lower-target cases of the remaining three tasks in the dynamics-model environment. From top to bottom, the panels show COMBO on density profile tracking (shot 176849), MOPO on temperature profile tracking (shot 190783), and MOPO on pressure profile tracking (shot 171975).

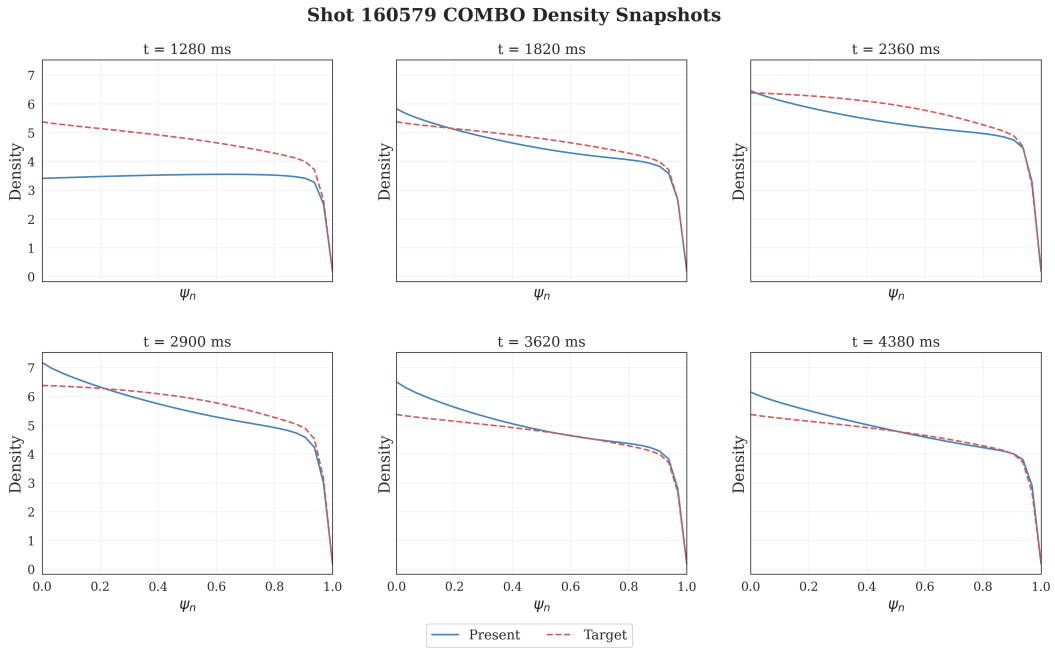


(a) Higher rotation target

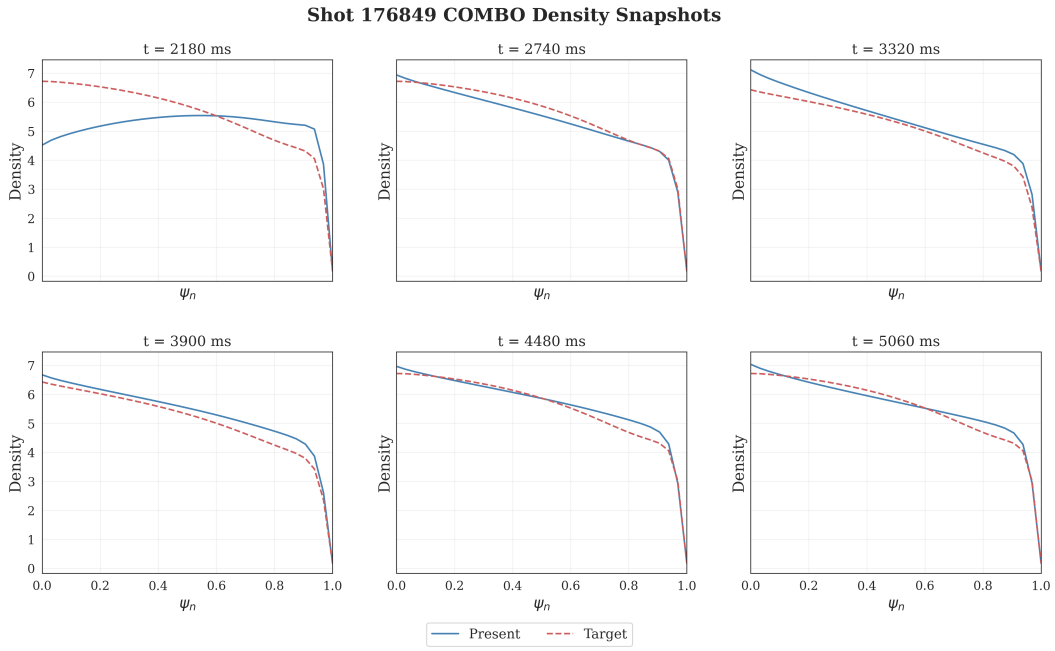


(b) Lower rotation target

Figure 6: Temporal evolution of full rotation profiles at selected time instances for two representative shots using RAMBO. (a) Higher rotation target and (b) Lower rotation target. Solid lines represent the present (predicted) profiles, while dashed lines indicate the target profiles.

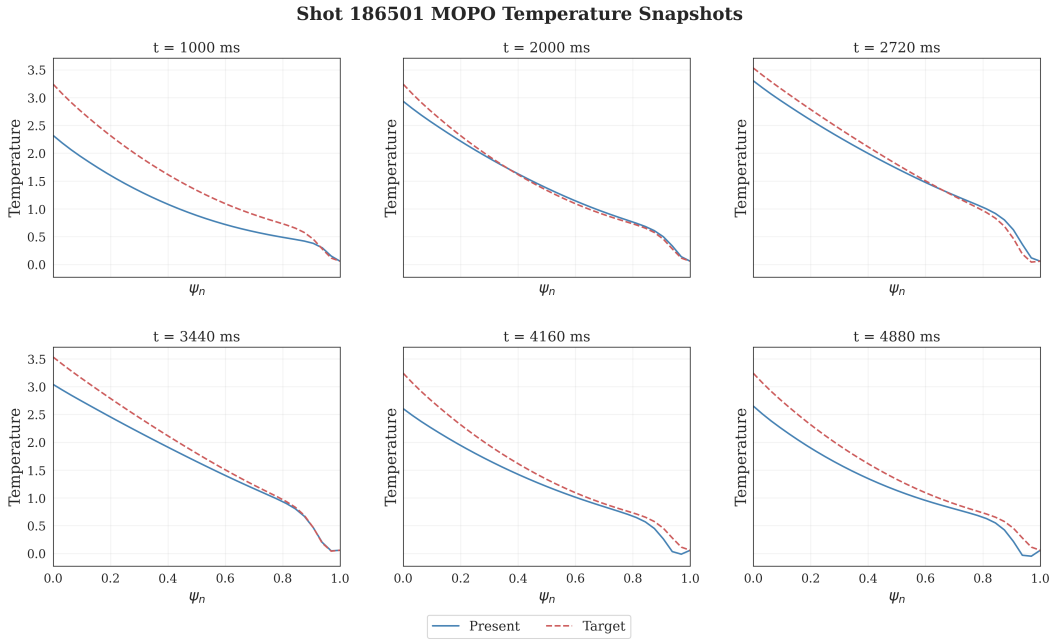


(a) Higher density target

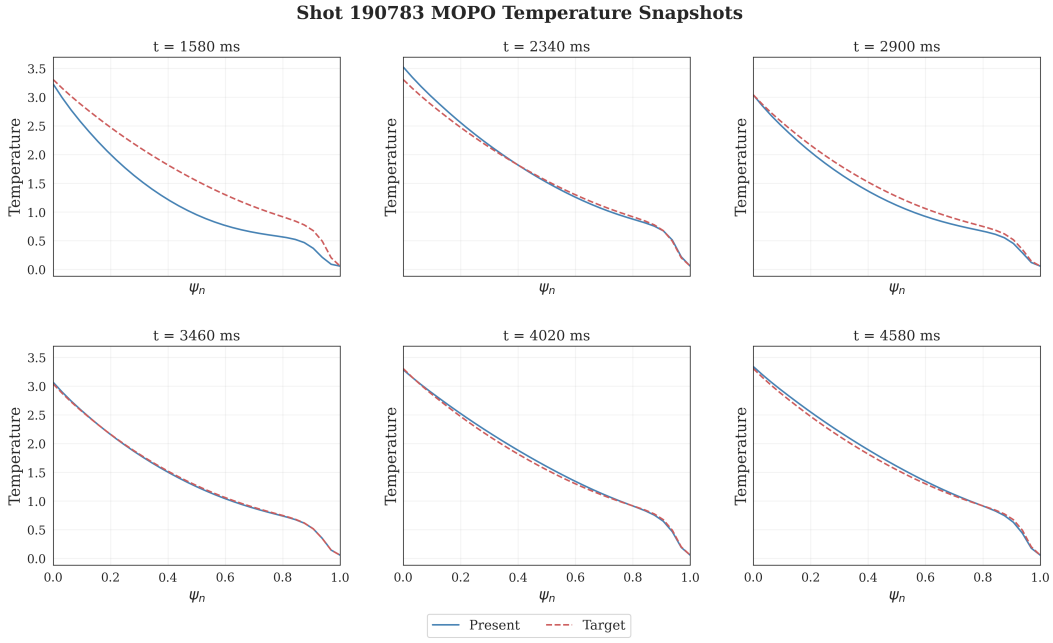


(b) Lower density target

Figure 7: Temporal evolution of full density profiles at selected time instances for two representative shots using COMBO. (a) Higher density target and (b) Lower density target. Solid lines represent the present (predicted) profiles, while dashed lines indicate the target profiles.

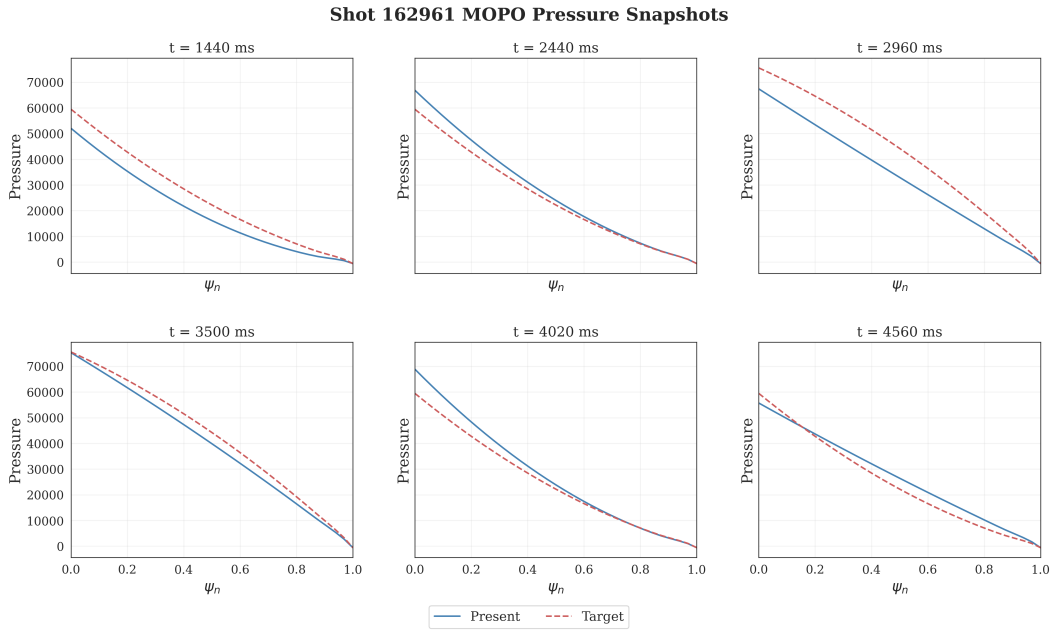


(a) Higher temperature target

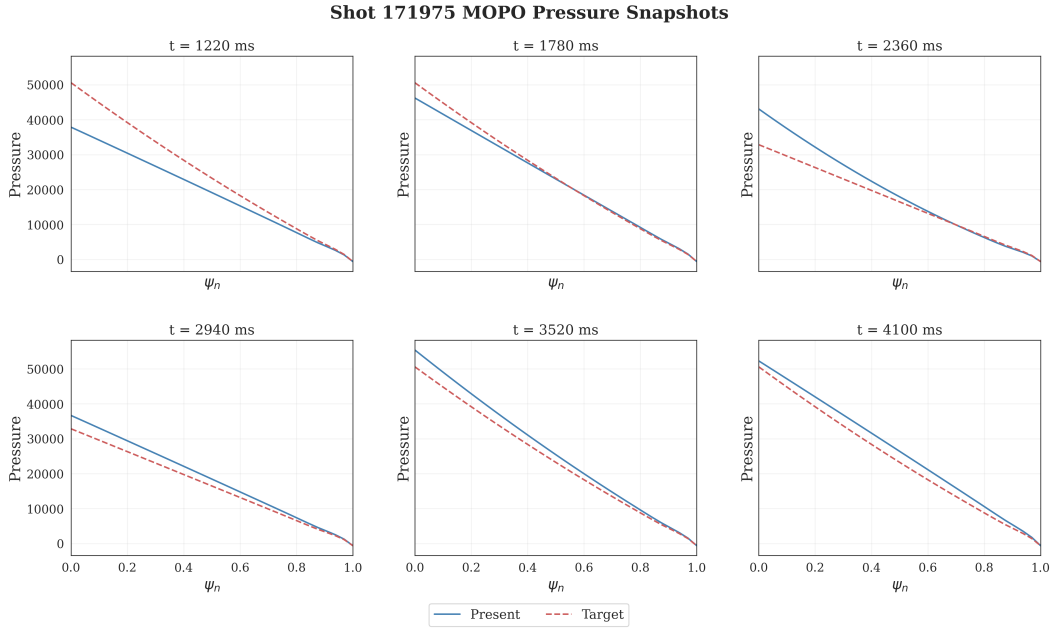


(b) Lower temperature target

Figure 8: Temporal evolution of full temperature profiles at selected time instances for two representative shots using MOPO. (a) Higher temperature target and (b) Lower temperature target. Solid lines represent the present (predicted) profiles, while dashed lines indicate the target profiles.



(a) Higher pressure target



(b) Lower pressure target

Figure 9: Temporal evolution of full pressure profiles at selected time instances for two representative shots using MOPO. (a) Higher pressure target and (b) Lower pressure target. Solid lines represent the present (predicted) profiles, while dashed lines indicate the target profiles.

B RPNN

B.1 RPNN Training

The benchmark uses two learned dynamics models: a reference dynamics model trained on historical DIII-D experimental trajectories, and a separate dynamics model trained on the synthesized benchmark trajectories for model-based baselines. Both models use the same RPNN ensemble architecture described in Appendix B.2; they differ only in the data used for training. Each model predicts transitions over the full dynamics state–actuator space, which is larger than the policy observation and action space. The corresponding signal usage is summarized in Table 7: variables marked as Actuators and States are used for dynamics modeling, while only a task-dependent subset of the state variables is exposed to the policy as observations.

We report held-out predictive fidelity using per-variable explained variance (EV). The reference dynamics model, trained on historical experimental data and used as the closed-loop evaluation environment, is summarized in Table 5. The dynamics model trained on synthesized benchmark trajectories, used by model-based baselines for learned rollouts, is summarized in Table 6.

Table 5: Per-variable explained variance (EV) of the reference dynamics model trained on historical experimental data. Higher values indicate better agreement with held-out real trajectories.

Variable	EV	Variable	EV
β_N	0.8229	itemp_pca2	0.6044
dssdenest	0.6658	itemp_pca3	0.4058
l_i	0.5374	itemp_pca4	0.2922
n1rms	0.4520	dens_pca1	0.4584
V_{loop}	0.5187	dens_pca2	0.3938
W_{MHD}	0.8144	dens_pca3	0.3578
temp_pca1	0.5810	dens_pca4	0.3938
temp_pca2	0.4225	rotation_pca1	0.4473
temp_pca3	0.3494	rotation_pca2	0.4016
temp_pca4	0.3396	rotation_pca3	0.3004
itemp_pca1	0.5878	rotation_pca4	0.3521
pres_pca1	0.8853	q_pca1	0.3945
pres_pca2	0.7107	q_pca2	0.4840

Table 6: Per-variable explained variance (EV) of the dynamics model trained on synthesized benchmark trajectories, evaluated on held-out data. Higher values indicate better predictive accuracy for model-based baselines.

Variable	EV	Variable	EV
β_N	0.9647	itemp_pca2	0.9193
dssdenest	0.8945	itemp_pca3	0.8776
l_i	0.9515	itemp_pca4	0.7848
n1rms	0.8491	dens_pca1	0.8534
V_{loop}	0.8675	dens_pca2	0.7224
W_{MHD}	0.9678	dens_pca3	0.7117
temp_pca1	0.8764	dens_pca4	0.8425
temp_pca2	0.7221	rotation_pca1	0.9133
temp_pca3	0.6276	rotation_pca2	0.8727
temp_pca4	0.8003	rotation_pca3	0.8242
itemp_pca1	0.9313	rotation_pca4	0.8100
pres_pca1	0.9838	q_pca1	0.9365
pres_pca2	0.9712	q_pca2	0.9428

Table 7: Plasma signals and how they are used as state and actuator variables for dynamics modeling. Policy observation and action space variables are also shown. Profile dimensionality depends on the task: rotation, density, and temperature use 4 PCA components, while pressure uses 2 PCA components.

Signal Group	Signals	Actuator	State	Action	Observation
Scalar States	β_N (Normalized Plasma Pressure)	✗	✓	✗	✗
	l_i (Internal Inductance)	✗	✓	✗	✗
	Line Averaged Density	✗	✓	✗	✗
	Loop Voltage	✗	✓	✗	✗
	MHD Stored Energy	✗	✓	✗	✗
Profile States	Rotation	✗	✓	✗	✓ (rot task)
	Density	✗	✓	✗	✓ (dens task)
	Ion Temperature	✗	✓	✗	✗
	Electron Temperature	✗	✓	✗	✓ (temp task)
	Pressure	✗	✓	✗	✓ (pressure task)
	Safety Factor q	✗	✓	✗	✗
Shape Variables	Elongation Upper Triangularity Bottom Triangularity a_{minor} Radial and vertical positions of magnetic axis	✓	✗	✗	✗
Neutral Beam Variables	Power Injected	✓	✗	✓	✗
	Torque Injected	✓	✗	✓	✗
Gas Puffing	GasA voltage	✓	✗	✓	✗
Electron Cyclotron Heating	ECH Total Power	✓	✗	✓	✗
Other Actuators	Current Target, Toroidal Field	✓	✗	✗	✗
Targets	Rotation Target (t), Rotation Target $(t+10)$, Error Terms (t) Error Terms $(t+10)$	✗	✗	✗	✓
Total Dimensions		**12D**	**25D**	**4D**	**20D/10D(pres task)**

B.2 Network Architecture and Training Details

Network Architecture

- **Encoder:**
  - Fully Connected (FC) layer: `input_dim` $\times$ 512
  - FC layer: 512 $\times$ 512
- **Memory Unit:**
  - Gated Recurrent Unit (GRU) block: 512 $\times$ 256
- **Decoder** (with residual connections between FC layers):
  - FC layer: 256 $\times$ 512
  - FC layers: 512 $\times$ 512 (repeated 8 times)
  - FC layer: 512 $\times$ 128
- **Output Heads:**
  - Mean head: 128 $\times$ `output_dim`

- Log-variance head: $128 \times$ `output_dim`

C Implementation Details

C.1 Hyperparameter

We separate hyperparameter selection from final evaluation. For each task and algorithm, hyperparameters were selected using the validation split, while the held-out test split was used only for the final closed-loop evaluation reported in the main results. This protocol avoids tuning directly on the test shots and provides a less biased estimate of generalization performance. Because the four profile-control tasks differ in their response channels and long-horizon dynamics, we allow a small set of task-dependent hyperparameters, but keep all other settings shared across tasks or fixed by the released configuration files.

Table 8 summarizes the common training, evaluation, network, and model-rollout settings used across the benchmark. These settings define the default experimental budget and architecture choices. Method-specific deviations are limited to the hyperparameters listed in Table 9.

For hyperparameter tuning, we used an Optuna-based tuner implemented in the released codebase. The tuner reads each algorithm's search space from the configuration file, launches training trials with sampled hyperparameters, and maximizes the validation reward extracted from the training logs. The default sampler is Optuna's TPE sampler, with support for grid search when specified in the configuration. Trials can be run in parallel across multiple GPUs, and the tuner records the best trial, selected parameters, log directory, and detailed trial results.

Table 10 reports the task-dependent hyperparameters selected by this validation procedure. These values were fixed before evaluating on the test split. Parameters not shown in the table are either shared across all four tasks, listed in Table 8, or specified in the released per-method configuration files.

Table 8: Common experimental settings used in the benchmark. We report the main shared settings here and provide complete per-method configurations in the released codebase.

Category	Setting	Value	Notes
Training			
Optimization budget	Gradient updates	1000 epochs, 1000 steps/epoch	Used by most offline RL baselines.
Mini-batch size	Default batch size	256	GCIL and PPO use method-specific batch sizes.
Discount factor	gamma	0.99	Used by most offline RL baselines; PPO use separate settings.
Target update	tau	0.005	Used by actor-critic offline RL baselines.
Evaluation			
Test set	Held-out shots	300 shots	Fixed test split shared across all tasks.
Randomization	Seeds per shot	10	Used for closed-loop policy evaluation.
Metric	Tracking error	RMSE ↓	Reported with standard error across rollout instances.
Evaluation episodes	Default value	5	PPO uses 3 evaluation episodes.
Policy and value networks			
Default MLP	Hidden dimensions	[256,256]	Used by several actor-critic and model-based baselines.
Larger MLP	Hidden dimensions	[256,256,256]	Used by COMBO, CQL, EDAC, GCIL, and MPPI.
PPO architecture	Policy/value networks	[256,256]	PPO uses separate policy and value networks.
Model-based rollouts			
Rollout batch size	rollout_batch_size	50000	Used by model-based offline RL baselines.
Model retention	model_retain_epochs	5	Controls the retained model-generated replay buffer.
Rollout schedule	rollout_freq	1000	Default rollout frequency for model-based methods.
Dynamics ensemble	Number of models	25	Used for learned dynamics-model rollouts and closed-loop evaluation.

Table 9: Algorithms evaluated in the benchmark and the hyperparameters tuned for each method. BAMCTS is implemented as `bambrl` in the codebase.

Category	Algorithm	Tuned hyperparameters
Imitation Learning	GCIL	batch size
Model-free offline RL	TD3BC	α
	CQL	CQL weight, temperature
	IQL	expectile, temperature
	EDAC	number of critics, η
	MCQ	λ, sampled actions
Model-based offline RL	PPO	clip range
	COMBO	rollout length, CQL weight
	MOPO	rollout length, penalty coef.
	MOBILE	rollout length, penalty coef.
	RAMBO	rollout length, adv. weight
	BAMCTS	rollout length, penalty coef., `use_ba`, `search_alpha` $\in \{0.5, 0.8\}$

Table 10: Task-dependent hyperparameters. All other hyperparameters are shared across the four benchmark tasks or specified in the released configuration files.

Method	Hyperparameter	Rotation	Density	Temperature	Pressure
GCIL	`batch_size`	64	512	64	256
PPO	`clip_range`	0.148	—	—	—
TD3BC	`alpha`	1.5	0.15	2.0	5.0
CQL	`cql_weight`	2.0	2.0	5.0	0.1
	`temperature`	1.0	2.0	5.0	0.1
IQL	`expectile`	0.67	0.69	0.52	0.55
	`temperature`	0.21	4.10	0.67	0.82
EDAC	`num_critics`	10	10	50	20
	`eta`	0.1	2.0	2.0	0.5
MCQ	`lmbda`	0.771114	0.750130	0.860632	0.711199
	`num_sampled_actions`	20	10	10	20
COMBO	`cql_weight`	1	3	5	3.0
	`rollout_length`	10	7	10	5
MOPO	`rollout_length`	7	5	7	7
	`penalty_coef`	0.5	5.0	1	1
MOBILE	`rollout_length`	1	—	—	—
	`penalty_coef`	1.5	—	—	0.5
RAMBO	`rollout_length`	5	2	5	4
	`adv_weight`	0.000001	0.000408	0.000095	0.000023
BAMCTS	`rollout_length`	5	2	1	5
	`penalty_coef`	1.72	0.92	1.29	1.64
	`use_ba`	`False`	`False`	`True`	`False`
	`search_alpha`	0.50	0.50	0.50	0.50

C.2 Experiments compute resources

All experiments were conducted on an Ubuntu 22.04 Linux server equipped with two Intel Xeon Platinum 8457C processors, providing 96 physical CPU cores (192 threads) in total, 503 GiB RAM, and 10 NVIDIA GeForce RTX 4090 GPUs. Each GPU provides approximately 46-49 GiB of memory. Unless otherwise specified, each training run used a single GPU. Training time varies across algorithm classes. Model-based methods generally require longer training due to learned-model rollouts, with representative runs ranging from several to about 30 GPU-hours (MCQ: 3.7h, MOPO: 11h, BAMCTS: 12.0h, MOBILE: 17h, RAMBO: 24.5h, COMBO: 1.2 days), whereas model-free and imitation-learning methods are comparatively faster, typically completing within minutes to 18 GPU-hours (GCIL: 43 min, IQL: 1.8h, CQL: 18h, EDAC: 8h, TD3-BC: 6h).

D Broader Impacts

RL4F is an open-source benchmark for offline reinforcement learning in nuclear-fusion plasma profile control, enabling researchers to develop data-driven control algorithms and compare them under standardized tasks, datasets, dynamics-model environments, and evaluation protocols. By supporting reproducible offline evaluation, RL4F can help reduce the need for costly and safety-critical trial-and-error on real tokamak devices during early-stage controller development. Progress on this problem may contribute to more effective regulation of plasma profiles related to confinement, fueling, heating, and overall plasma performance, which is an important step toward stable and efficient fusion operation. More broadly, the benchmark provides a challenging real-world testbed for offline RL research, with nonlinear dynamics, long-horizon control, partial observability, and limited data coverage. As with any simulation-based benchmark in a safety-critical domain, performance in RL4F should be interpreted as evidence from a controlled offline evaluation setting rather than as a substitute for real-device validation; any potential deployment on physical tokamaks would require additional safety constraints, expert review, and device-specific testing.