Explainable physics-based constraints on reinforcement learning for accelerator controls

Jonathan Colen,[1, 2] Malachi Schram,[1, 3, 4] Kishansingh Rajput,[3, 5] and Armen Kasparian[3]

[1]*Joint Institute on Advanced Computing for Environmental Studies,*
Old Dominion University, Norfolk, Virginia 23539 USA
[2]*Hampton Roads Biomedical Research Consortium, Portsmouth, Virginia 23703 USA*
[3]*Data Science Department, Thomas Jefferson National Accelerator Facility, Newport News, VA 23606 USA*
[4]*Computer Science Department, Old Dominion University, Norfolk, Virginia 23539 USA*
[5]*Department of Computer Science, University of Houston, Houston, TX 77204, USA*

(Dated: March 4, 2025)

We present a reinforcement learning (RL) framework for controlling particle accelerator experiments that builds explainable physics-based constraints on agent behavior. The goal is to increase transparency and trust by letting users verify that the agent's decision-making process incorporates suitable physics. Our algorithm uses a learnable surrogate function for physical observables, such as energy, and uses them to fine-tune how actions are chosen. This surrogate can be represented by a neural network or by an interpretable sparse dictionary model. We test our algorithm on a range of particle accelerator controls environments designed to emulate the Continuous Electron Beam Accelerator Facility (CEBAF) at Jefferson Lab. By examining the mathematical form of the learned constraint function, we are able to confirm the agent has learned to use the established physics of each environment. In addition, we find that the introduction of a physics-based surrogate enables our reinforcement learning algorithms to reliably converge for difficult high-dimensional accelerator controls environments.

INTRODUCTION

Large-scale scientific systems are difficult to operate, requiring simultaneous control of many apparatus to achieve experimental objectives. To be successful, an operator must understand both the scientific principles of the device as well as the subtle ways that those principles and machine idiosyncrasies collectively influence operation. Data-driven techniques such as reinforcement learning [1] are promising tools to address these challenges, with demonstrated success at playing games [2, 3], solving engineering problems [4, 5] and even scientific tasks [6–9]. However, these tools remain a black box. While deep neural networks can make accurate predictions, less is known about how to interpret their behavior and understand the rules that govern their operation. This limited explainability hinders trust and is a barrier to applying these tools, especially in environments where unexpected behavior and failures are expensive.

Physics-informed learning methods introduce physical constraints to improve performance and guide predictions to obey established governing principles [9–11]. However, they require that the system adheres closely to the imposed model and can break down for problems whose laws are not firmly established or in experimental systems where hardware, implementations, and noise may obscure the physical rules or even the relevant variables [12, 13]. Symbolic and sparse regression aim to infer these laws directly from data [14–16] and have been incorporated in machine learning pipelines for model discovery [12, 17–22]. Recent work has explored how to integrate such methods into reinforcement learning schemes by building explainable surrogates of the learned policy, environment, and reward functions [23]. However, these surrogates may be difficult to interpret, particularly for problems where the state-action-reward relationship involves additional details beyond the underlying fundamental physics.

In this work, we present a reinforcement learning (RL) framework that uses learnable and explainable constraints to improve control of particle accelerator experiment environments. We use trainable surrogate models to identify the dynamical rules for relevant physical observables such as energy. These models can be deep neural networks or sparse dictionary models, which provide the additional advantage of learning an interpretable mathematical form for these observables. The constraint models train alongside a traditional actor-critic-style model and may influence the chosen actions. We test our algorithms on surrogate implementations of linear accelerators (linacs) in Jefferson Lab's Continuous Electron Beam Accelerator Facility (CEBAF) [24]. For simpler problems, such as controlling a single cryomodule, our framework achieves comparable performance to a standard model-free RL algorithm. For the more challenging task of controlling the entire CEBAF north linac, our framework achieves superior performance at minimizing operational hazards while maintaining a target energy gain. Our learnable and explainable constraint models also provide a window inside the black box, allowing operators to verify that agent decision-making is based on a correct understanding of the underlying physics. Our algorithm represents a promising step towards the use of RL to control complex experimental apparatus and enable scientific discovery.

PREVIOUS WORK

Machine learning (ML) has been applied to a wide range of problems in physics and accelerator applications. Several studies have applied deep learning models to detect anomalies and operational faults at different experimental sites [25–28]. Other efforts have aimed to tune beam properties to reach desired experimental conditions [6, 7, 29–31]. A recent analysis [32] compared the performance of Bayesian optimization [29], and genetic algorithms [33] against RL techniques and found that deep differentiable RL [34] achieved excellent performance on CEBAF beam optimization. A drawback of this approach is that it requires a fully differentiable environment. Moreover, it remains a black box and gives operators little insight into the rules governing predicted operational behavior.

In physics applications, a path to explainability comes from the knowledge that systems are constrained to obey physical laws. Many studies have examined how to discover these laws using ML techniques [14]. Sparse model regression solves this task by selecting a simple and parsimonious model from a dictionary of candidate expressions [15]. This method has been integrated with deep learning architectures to derive predictive coordinates that are well-suited to sparse modeling [17]. Other techniques use information-theoretic frameworks, such as the information bottleneck [35], to distill reduced order models from data [36–40]. The development of these physics-inspired ML frameworks has enabled physical model discovery in fields such as materials [19–22, 41, 42] and biology [12, 43–45]. These techniques have begun to be extended to RL. A recent study integrated sparse model regression with model-based RL, learning parsimonious models of the system and agent from episodic observations [23]. However, the learned surrogates, which were sufficiently accurate for learning purposes, did not yield ready interpretations or comparisons to known physics.

METHODS

CEBAF environment description

CEBAF is the primary accelerator at Thomas Jefferson National Accelerator Facility (JLab) [24]. It accelerates electrons using a pair of superconducting radiofrequency (SRF) linear accelerators containing a series of individually-tunable SRF cavities. During an experiment, the operator must set the gradients in each cavity in order to obtain the target energy gain. At the same time, they aim to avoid potential problems, such as a high heat load or the creation of Fast Shut Down (FSD) trips. To accomplish these goals, one has to consider both the relevant physics as well as the unique operating characteristics of each cavity.

Problem	Num. Cavities	E_{target} (MeV)	δE (MeV)
8D	8	20.08	0.40
16D	16	50.00	0.60
32D	32	120.00	0.80
North linac	197	1050.00	2.00

TABLE I. CEBAF optimization problems

To learn to control CEBAF and set SRF cavity gradients, we used a computational surrogate environment for CEBAF which has been described previously [32]. In this environment, operating agents observe a state vector containing the current gradients set for each SRF cavity, and take actions to change those gradients to new values. During operation, the cavities dissipate heat into the system which depends on a number of factors and is captured by the following equation [46]

$$H = \sum_i \frac{G_i^2 \, \ell_i}{\omega_i \, Q_i(G_i)} \qquad (1)$$

Here i runs over all cavities, G_i is the cavity gradient, and the parameters ℓ_i, ω_i, Q_i are the length, impedance, and quality factor of each cavity.

CEBAF contains a number of legacy cryomodules which can trigger arc faults during operation. When an arc fault is detected, the cavity powers down and the electron beam production halts to prevent damage. A previous study characterized the rate of these FSD trips for each cavity (trip rate) using the following statistical model [46]

$$T = \sum_i \exp \left[A + B_i(G_i - F_i) \right] \qquad (2)$$

Here A, B_i are regression parameters and F_i is a fault gradient fit from historical data.

The configuration of cavity gradients also determines the total energy gain within the linac. During experiments, an operator aims to keep this energy gain within a narrow range centered at a target energy given by experimental requirements. The total energy gain is given by the equation

$$E = \sum_i G_i \ell_i \qquad (3)$$

We examined four CEBAF optimization problems in this study. Three are test problems for controlling one, two, or four cryomodules. The final problem tasks the agent with controlling the full CEBAF north linac. The problem sizes and energy targets are summarized in Table I.

Reinforcement learning algorithms

In this study, we examined the ability of reinforcement learning (RL) algorithms to operate and control the CE-

---

Algorithm 1: TD3 + learnable constraints (LC-TD3)

Initialize critics $Q_{\theta_1}, Q_{\theta_2}$ and policy network π_ϕ
Initialize target networks $\theta_{i'} \leftarrow \theta_i$, $\phi' \leftarrow \phi$
Initialize replay buffer $\mathcal{B}$
Initialize learnable surrogate network O_ξ and
  constraint function $C(o)$
for e in $1 \ldots N_e$ **do**
  Observe state s and select action $a \sim \pi_\phi$
  Execute a in environment
  Observe next state s', reward r, terminal signal d,
    and environmental observables o
  Store (s, a, r, s', d, o) in replay buffer $\mathcal{B}$
  if *time to update* **then**
    Sample batch of transitions $b \sim \mathcal{B}$
    $a' \leftarrow \pi_{\phi'}(s') + \epsilon \mathcal{N}(0, \sigma)$
    $y_i \leftarrow r + \gamma \min_i Q_{\theta_i'}(s', a')$
    Update surrogate O_ξ with gradient descent
      using $\frac{1}{|b|} \nabla_\xi \sum (O_\xi(s, a) - o)^2$
    Update Q functions with gradient descent
      using $\frac{1}{|b|} \nabla_{\theta_i} \sum (Q_{\theta_i}(s, a) - y_i)^2$
    Update policy π with gradient ascent using
      $\frac{1}{|b|} \nabla_\phi \sum (Q_{\phi_1}(s, \pi_\phi(s)) - \beta C(O_\xi(s, \pi_\phi(s))))$
    Update target networks:
    $\theta_i' \leftarrow \tau \theta_i' + (1 - \tau)\theta_i$
    $\phi' \leftarrow \tau \phi' + (1 - \tau)\phi$
  end
end

Algorithm 2: TD3 + sparse learnable constraints (Sparse LC-TD3)

Initialize critics $Q_{\theta_1}, Q_{\theta_2}$ and policy network π_ϕ
Initialize target networks $\theta_{i'} \leftarrow \theta_i$, $\phi' \leftarrow \phi$
Initialize replay buffer $\mathcal{B}$
Initialize library function $L(s, a)$, weight vector $\mathbf{w}$, and
  constraint function $C(o)$
for e in $1 \ldots N_e$ **do**
  Observe state s and select action $a \sim \pi_\phi$
  Execute a in environment
  Observe next state s', reward r, terminal signal d,
    and environmental observables o
  Store (s, a, r, s', d, o) in replay buffer $\mathcal{B}$
  if *time to update* **then**
    Sample batch of transitions $b \sim \mathcal{B}$
    $a' \leftarrow \pi_{\phi'}(s') + \epsilon \mathcal{N}(0, \sigma)$
    $y_i \leftarrow r + \gamma \min_i Q_{\theta_i'}(s', a')$
    Update weight vector $\mathbf{w}$ with gradient descent
      using $\frac{1}{|b|} \nabla_\mathbf{w} \sum (L(s, a)\mathbf{w} - o)^2$
    Update Q functions with gradient descent
      using $\frac{1}{|b|} \nabla_{\theta_i} \sum (Q_{\theta_i}(s, a) - y_i)^2$
    Update policy π with gradient ascent using
      $\frac{1}{|b|} \nabla_\phi \sum (Q_{\phi_1}(s, \pi_\phi(s)) - \beta C(L(s, \pi_\phi(s))\mathbf{w}))$
    Update target networks:
    $\theta_i' \leftarrow \tau \theta_i' + (1 - \tau)\theta_i$
    $\phi' \leftarrow \tau \phi' + (1 - \tau)\phi$
  end
end

BAF accelerator. In a standard RL problem, an agent operates within an environment and obtains rewards based on how well it performs some predetermined task. At each time step, the agent observes a state s which is a vector of measurable quantities within the environment. Based on that state, the agent selects a suitable action a. The environment uses that action to transition to the next state s' and gives the agent a reward r. The goal of the agent is to learn to select the action that maximizes the cumulative reward received for both current and future actions.

A wide variety of techniques have been proposed for RL problems [1]. Here, we considered the Twin-Delayed Deep Deterministic Policy Gradient (TD3) algorithm [47], which uses one policy model to select actions $\pi(s)$ and a pair of Q-functions $Q(s, a)$ that evaluate the expected future reward resulting from a given action. During training, the agent aims to explore the environment such that the Q-functions faithfully represent the reward space and adjusts the policy to select actions that maximize $Q(s, \pi(s))$. The complete algorithm is provided for reference in Alg. S1.

From the problem description given previously, we modified the RL problem to include constraints $C(o)$ that operate on a set of physical observables o that are returned by the environment alongside the reward r and next state s'. For CEBAF, these observables are the energy gain and the constraint objective is the energy

target $|E_\text{target} - \sum_i G_i \ell_i| < \delta E$. The RL agent must learn to configure the cavity gradients to minimize operational hazards such as heat load and FSD trip rate while satisfying this energy constraint. With the standard TD3 algorithm, one could modify the reward to promote adherence to the constraint function. However, recent work has found that traditional TD3 struggles to adapt to high-dimensional accelerator environments with hard energy constraints [32].

We modified the TD3 algorithm by introducing a learnable surrogate function that maps state-action pairs to the relevant physical observables o. We considered two cases: the first uses a deep neural network to learn the surrogate function, while the second uses a sparse dictionary model inspired by the SINDy algorithm [15]. The latter model builds a library L of mathematical terms, typically low order polynomials of the state and action components, and aims to represent the target observables as a linear combination of these elements. The coefficients for each library element are stored in a weight vector $\mathbf{w}$ such that the predicted observables are given by $o' = L(s, a)\mathbf{w}$. During training, the weights $\mathbf{w}$ are tuned so that the surrogate recovers a correct physical model of the system. The surrogate also helps train the policy π, which learns to select actions that maximize the expected reward $Q(s, \pi(s))$ and satisfy the *expected* constraints $C[L(s, \pi(s))\mathbf{w}]$. The full algorithms are outlined in Algorithms 1-2.
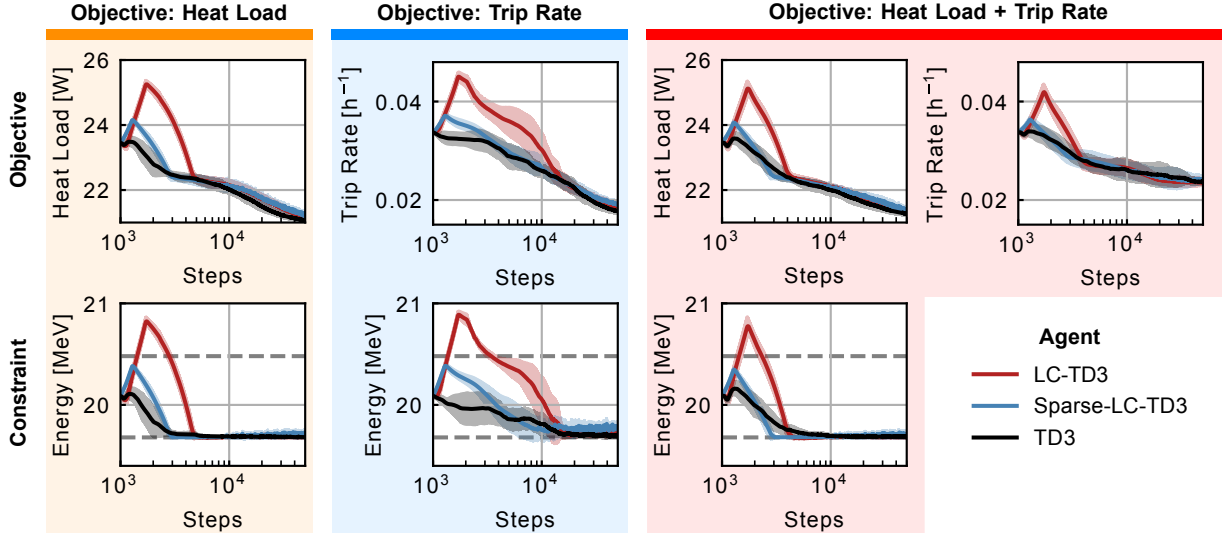
FIG. 1. **Training behavior for RL agents on CEBAF 8D problem**. (*Top*) Reward objective at each step for standard TD3 with energy penalty, LC-TD3 (Alg. 1) and Sparse LC-TD3 (Alg. 2). Columns denote agents trained to minimize heat load, trip rate, or both simultaneously. (*Bottom*) Energy per episode for each agent. Dashed lines denote the bounds of the energy constraint. A sparse dictionary surrogate function converges to the energy target faster than a deep neural network. Plots show mean and standard deviation over $N = 8$ trials per agent per objective.

For each CEBAF optimization environment in Table I, we compared LC-TD3 and Sparse-LC-TD3 (Alg. 1-2) to standard TD3 (Alg. S1) for three objectives: minimizing heat load, minimizing trip rate, and minimizing heat load and trip rate simultaneously. The reward functions were

$$R \in \{-H, -T, -0.5(H + T)\} \tag{4}$$

The energy constraints were chosen following [32]

$$C(E) = \begin{cases} 0 & |E - E_{\text{target}}| < \delta E \\ -5 \times |E - E_{\text{max}}| & E > E_{\text{target}} + \delta E \\ -5 \times |E - E_{\text{min}}| & E < E_{\text{target}} - \delta E \end{cases} \tag{5}$$

The discontinuous energy constraint $C(E)$ enables precise control by defining strict boundaries on the energy gain while applying linear penalties when the energy gain is far from the target. This helps ensure the agent learns policies that maintain the target energy levels required for optimal accelerator operation. For the TD3 agents, the energy constraint was integrated into the reward by including a penalty term. For the LC-TD3 agents, the constraint function was applied to the energy predicted by the learnable surrogate model.

RESULTS

CEBAF test environments

As an initial step, we trained our agents on a test environment containing 8 cavities, which was a surrogate

model for the 1L10 cryo-module. The training performance of each algorithm is shown in Fig. 1. We found that the standard TD3 agent rapidly converged to a solution whose energy gain lay at the lower bound of the allowed energy band. Our Sparse-LC-TD3 agent was able to achieve a similar reward after a longer training cycle, but was quicker to converge than the same algorithm using a neural network surrogate. This occured because the neural network model takes time to learn to accurately approximate the energy. For both LC-TD3 agents, the reward decreased initially, as the policy was guided by an inaccurate energy surrogate model. Once the learned energy surrogate became sufficiently accurate, the agent was able to make informed decisions and reach a high reward. When examining the predictions of the energy surrogate, we found that the sparse dictionary model reached the target energy band more quickly than the neural network model (Fig. 1 bottom).

The Sparse-LC-TD3 agent learned a surrogate model for energy parameterized by coefficients for a set of library functions. While the library contained higher-order polynomial terms, we found that the majority of learned coefficients were nearly zero. The identified equation for the energy function contained only linear terms:

$$E = 20.08 + 0.69\,G_0 + 1.48\,G_1 + 1.55\,G_2 + 0.74\,G_3 \\ + 0.25\,G_4 + 1.13\,G_5 + 0.90\,G_6 + 1.29\,G_7 \tag{6}$$

This result was expected as the equation for total energy gain (3) contains only terms linear in the cavity gradients. The identified coefficients accounted for the
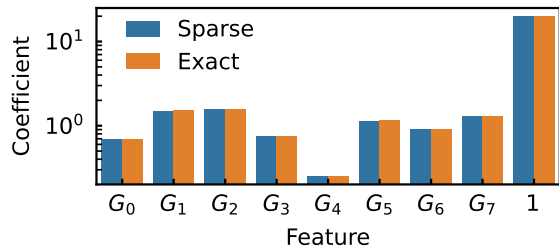
FIG. 2. **Learning a sparse interpretable model of single-cryomodule energy gain.** Comparison of the learned coefficients to their exact values set in the environment. Coefficient values were extracted from a Sparse-LC-TD3 agent trained on the CEBAF-8D problem. Plot shows coefficients with magnitude greater than a threshold $\tau = 0.05$.

cavity lengths as well as some features unique to each cavity, such as the maximum and minimum gradient settings allowed by the CEBAF optimization environment. In Fig. 2, we plotted the identified coefficients next to the ground-truth values from the environment and verified that the machine-learned surrogate learned the correct physics.

To assess performance in intermediate-scale problems, we next tested each algorithm on higher-dimensional surrogate environments with 16-dimensional and 32-dimensional state-action spaces. These represented two- and four-cryomodule environments, respectively. In Fig. 3, we plotted the heat load and trip rate selected by the agent for in these environments for each reward function. While TD3 produced more optimal behavior in the one-cryomodule case (Fig. 3a), this difference disappeared in the two- and four-cryomodule cases (Fig. 3b-c). All algorithms achieved similar performance on their primary objective in each of these cases. We did observe that the learnable constraint models achieved more optimal performance on secondary objectives. That is, Sparse-LC-TD3 and LC-TD3 models trained to minimize heat load selected configurations with lower trip rates than standard TD3 models, and (Sparse-) LC-TD3 models trained to minimize trip rate achieved lower heat loads as well. We also found that the learnable constraint models failed to stay within the target energy range for 1-2 trials for each optimization objective, although the majority of trials did converge properly. The complete results are reported in Table S2-S4.

As in the 8D problem, we examined the learned surrogate by comparing the learned coefficients to their exact values set in the environment. Once again, the majority of identified coefficients were nearly zero and the sparse dictionary model was represented by a linear equation. The non-zero terms exhibited strong agreement with the ground truth values, see Fig. S5. This indicated that the sparse surrogate model learned the correct physics governing the energy gain in the accelerator environment.
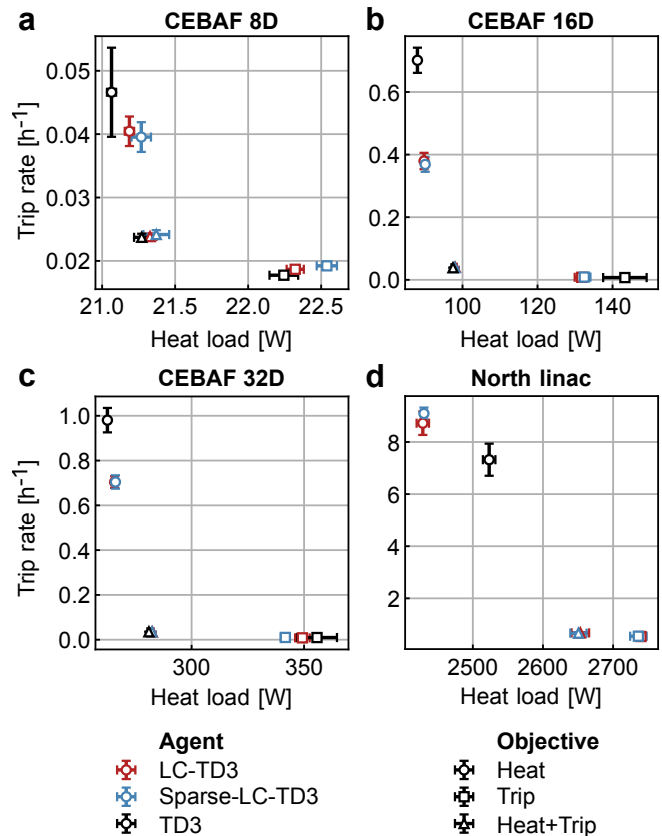

FIG. 3. **Training performance for CEBAF environments**. Heat load and trip rate for agents' selected configurations at the end of training for each CEBAF optimization problem. Each training algorithm had $N = 8$ total trials for each of the three objectives. We plot the mean and standard deviation for trials which converged to the target energy range. Full results are reported in Table II for North linac and Table S2-S4 for the test environments.

CEBAF North linac environment

After characterizing our algorithms on small-scale test problems, we turned to the more challenging task of controlling the full CEBAF north linac. This contains 197 SRF cavities whose gradients must be tuned simultaneously. Recent work showed that the TD3 algorithm can struggle to produce configurations near the target energy for this high-dimensional problem [32].

We trained RL agents using the TD3 algorithm and both LC-TD3 algorithms for 8 independent trials of 50,000 steps for each objective. The performance for each agent and objective are shown in Fig. 3d and Table II. We found that our learnable physics-based surrogates enabled agents to select cavity configurations that minimized heat load or FSD trip rate while remaining within the target energy range. The LC-TD3 agent proved slightly more consistent than the Sparse-LC-TD3 agent. The former converged to a suitable solution in all trials,
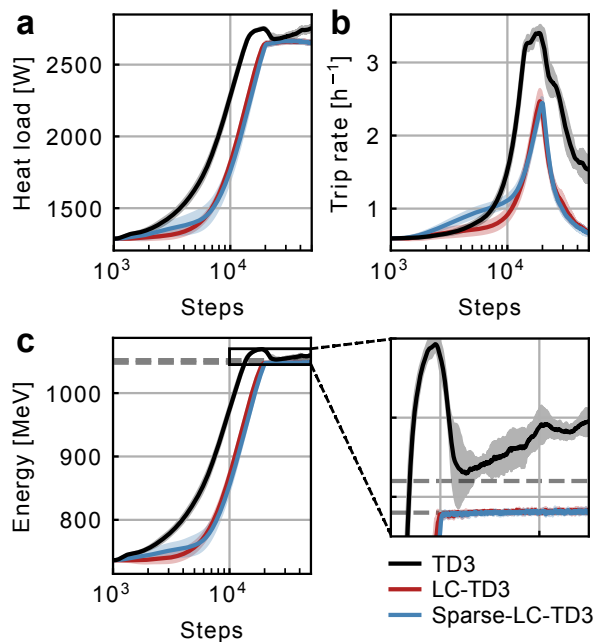
FIG. 4. **Training behavior for RL agents on CEBAF North linac problem**. For the predicted configuration at each step, we plot the heat load (**a**), trip rate (**b**), and energy gain (**c**) for each RL agent. Inset zooms in on energy gain near the end of training, showing the failure of TD3 to converge to the target energy range. Plots show mean and standard deviation over $N = 8$ agents trained using the combined Heat-Trip objective.

	Agent	Heat (W)	Trip (h^{-1})	Conv. Rate
Heat	LC-TD3	**2429 (9)**	8.72 (0.45)	**100 %**
	Sparse-LC-TD3	**2429 (5)**	9.11 (0.21)	88 %
	TD3	2559 (45)	**7.08 (0.84)**	50 %
Multi	LC-TD3	2653 (12)	**0.67 (0.03)**	**100 %**
	Sparse-LC-TD3	**2650 (11)**	**0.67 (0.03)**	88 %
	TD3	2753 (21)	1.55 (0.18)	0 %
Trip	LC-TD3	2740 (6)	**0.53 (0.03)**	**100 %**
	Sparse-LC-TD3	**2736 (12)**	0.55 (0.05)	**100 %**
	TD3	2784 (20)	1.09 (0.34)	0 %

TABLE II. End-of-training performance for each RL agent and objective in the CEBAF North linac environment. Left-most column indicates the training objective. Mean and standard deviation computed over $N = 8$ trials. Rightmost column reports the percentage of trials that converged to a configuration producing an energy gain within the allowed range.

while the Sparse-LC-TD3 agent failed to reach the target energy in one trial for two of the objectives. Both LC-TD3 algorithms outperformed standard TD3, which failed to reach the energy target for two objectives and converged only half of the time for its best-performing task.

In Fig. 4, we plotted the heat, trip, and energy for the agents' selected configurations at each step. While the LC-TD3 algorithms reached and remained within the target energy band, the standard TD3 algorithm had trouble adhering to the energy constraints. We hypothesize that this occured due to the high-dimensional nature of the problem. Because the volume of the configuration space grows exponentially with the problem dimension, a TD3 agent may require a large number of action samples to characterize the reward landscape near the target energy range. On the other hand, the LC-TD3 and Sparse-LC-TD3 models learned functions that bound the allowed subvolume of configuration space, enabling more efficient sampling and convergence to an optimal solution.

We examined this further by evaluating the local accuracy of the critic and learned surrogate functions. To do this, we randomly sampled the action space in the local neighborhood of the trained agents' selected configura-

tions. By varying the neighborhood size δ from small to large ($\delta \in [0.05, 5]$), we produced configurations with energy gains near the boundary of the allowed range. Both LC-TD3 agents more accurately estimated the reward than the TD3 agent. The accuracy was most comparable near the target energy, where TD3 approached a 1% error rate. However, the LC-TD3 critics remained accurate outside the energy range, while the TD3 critic performed significantly worse (Fig. 5 top). Because the TD3 critic was unable to characterize how the energy constraint affected the high-dimensional reward landscape, the policy could not learn to select optimal actions compatible with that constraint. By leveraging a learnable energy surrogate model, the LC-TD3 agents could more effectively explore and assess optimal configurations near the bounds of the target energy range.

We also evaluated the performance of the learned energy surrogates using a similar approach, and found that the sparse dictionary model more accurately estimated the energy gain of new configurations far from the agents' prediction (Fig. 5 bottom). This may be beneficial for applications where the experimental or operational objectives may change unexpectedly. Here, the sparse dictionary model's generalization performance could allow it to assess the physical behavior of a new configuration without requiring costly retraining.

The sparse model agent provides an additional key advantage: interpretability. While the neural network LC-TD3 model could achieve similar performance in minimizing heat load and trip rate and converges slightly more consistently, it gave the operator no insight on how it makes predictions. On the other hand, the sparse dictionary model learned a mathematical equation for the energy gain. As in the single-cryomodule case, this equation was a linear combination of terms linear in the cavity gradients G_i. In Fig. 6, we compared the inferred values with the ground truths taken from the environment

FIG. 6. **Learning an accurate model of CEBAF north linac energy gain**. Comparison of ground truth and machine-learned coefficient values for the CEBAF north linac environment. Values are averaged over all trials of the Sparse-LC-TD3 agent.
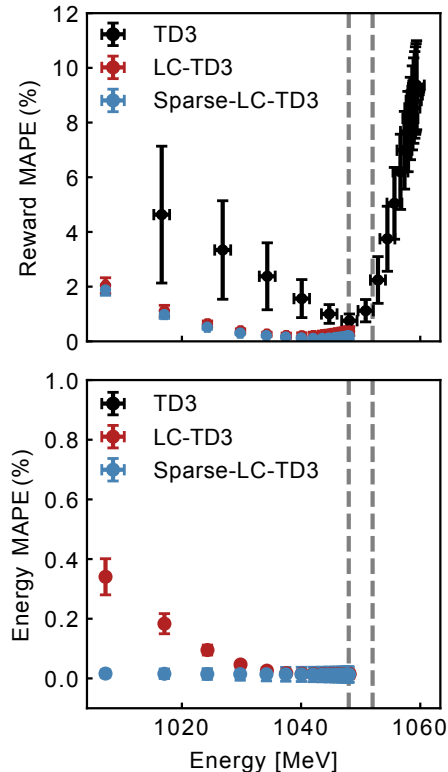
FIG. 5. **Learnable physics surrogate functions enhance performance for high-dimensional action space.** (*top*) Relative errors in reward estimation for each RL agent trained on the CEBAF North linac optimization problem. (*bottom*) Relative errors in the learned surrogates for LC-TD3 agents. Each point is the average error for randomly-sampled actions a distance δ from the agents' selected configuration at the end of training, with δ spaced logarithmically in the range $[0.05, 5]$. Errorbars denote standard deviation over $N = 8$ agents trained using the combined Heat-Trip objective.

and observed excellent agreement. Thus, an accelerator operator can independently verify that the agent is making decisions based on a correct model of the accelerator physics.

Multi-objective CEBAF optimization

As a final test of our algorithms' performance, we applied our learnable constraint agents to the conditional multi-objective CEBAF optimization problem considered in [32]. In this case, rather than specifying a fixed priority for each objective, we used a tunable weight vector α to set the relative importance of heat load and FSD trip rate minimization. As in [32], we adapted each RL algorithm by using a conditional policy $\pi(s|\alpha)$ and a Q-function $Q_i(s, a)$ that predicts a vector of expected return for each objective. During policy training, the dot product $\alpha \cdot Q$ de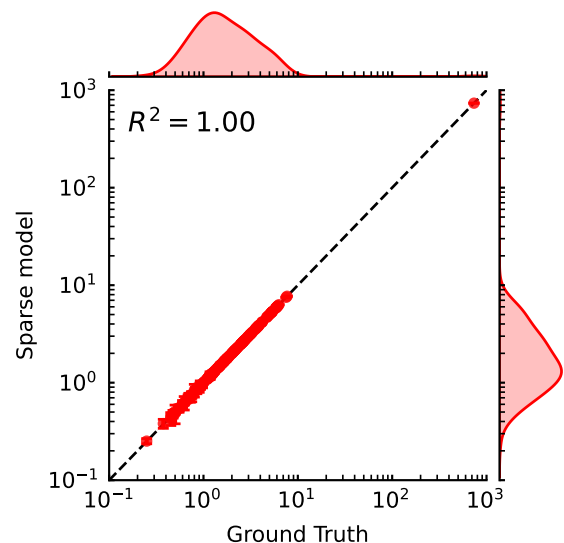fines a dynamic scalar optimization objective. This allows the agent to select from Pareto-optimal policies by changing the weight vector α. Crucially, the training of the critic and learnable constraint functions are unchanged when considering the multi-objective problem.

We trained RL agents using each algorithm and evaluated the quality of predicted configurations at 500 evenly-spaced weight vectors. We trained agents for 100,000 steps for the 8D, 16D, and 32D problem and 500,000 steps for the North linac to give agents more time to converge on the high-dimensional problem. To quantify relative performance, we used the normalized hypervolume metric [48] as done in [32], see Supplement for details. Briefly, the normalized hypervolume describes the size of the objective space captured by the agent's Pareto front. We computed hypervolume with respect to predicted configurations within the target energy range for each problem.

The conditional policy predictions and normalized hypervolumes for each problem are plotted in Fig. 7 and reported in Table III. For the 8D and 16D problems, the algorithms were nearly indistinguishable. For the 32D problem, the LC-TD3 and Sparse LC-TD3 algo-

Agent	8D	16D	32D	North linac
TD3	74.59	74.65	71.14	36.34
LC-TD3	74.40	74.86	**74.05**	0.00
Sparse-LC-TD3	**74.67**	**75.12**	73.89	**60.42**

TABLE III. Pareto front coverage, represented using the normalized hypervolume metric. Reference and ideal points for hypervolume calculation are listed in Table S1.
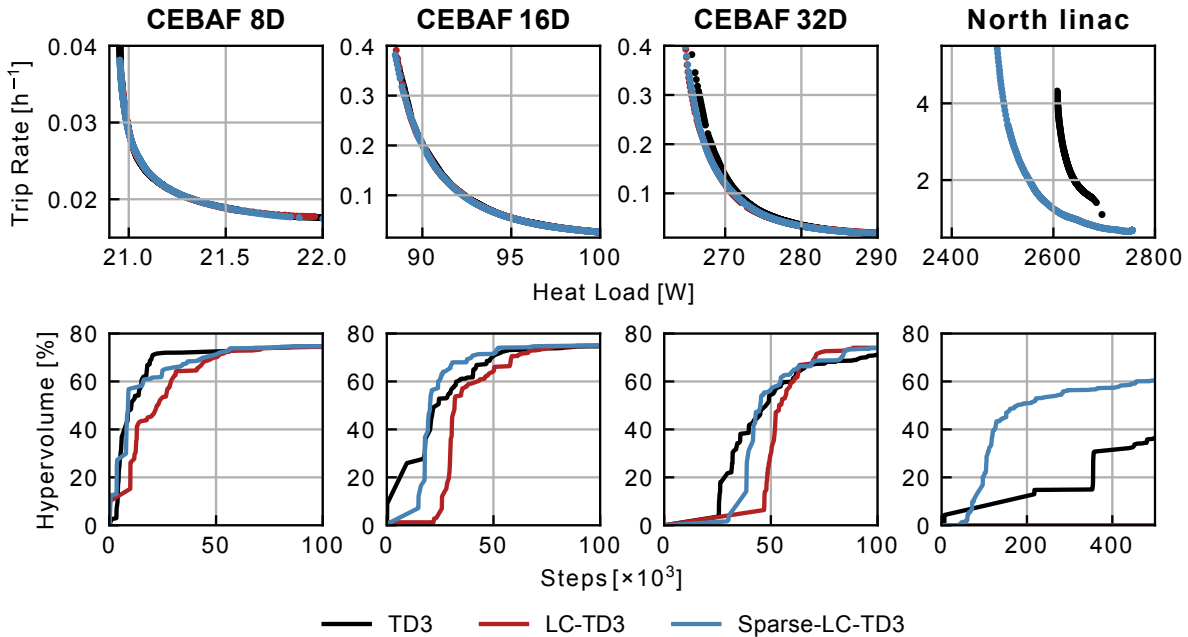
FIG. 7. **Learnable surrogate enables high-dimensional multi-objective optimization.** (*top*) Optimal pareto front produced by each RL agent trained using a multi-objective RL approach. (*bottom*) Pareto coverage over time, measured using the normalized hypervolume metric (see Supplement). Columns indicate the CEBAF problem being optimized.

rithms achieved 3.9% and 4.1% improvements over standard TD3. In the North linac case, the standard TD3 model struggled to generated predictions within the target energy range. Less than half of its predicted configurations satisfied the energy constraint, resulting in a less-populated Pareto front (see Fig. 7 top). LC-TD3 also failed to satisfy the energy constraint. This likely occurred because the surrogate model, represented by a neural network, did not accurately characterize the broad high-dimensional energy landscape needed to generate valid configurations at different objective priorities. Even in the single-objective case, we found that LC-TD3 models struggled to capture the relevant physics outside a narrow region near the selected configuration (Fig. 5).

Sparse LC-TD3 achieved the highest performance for the North linac problem, generating a continuum of configurations within the target energy range at different prioritization levels (Fig. 7). The learned constraint represented by a sparse dictionary model better characterized the energy over the high-dimensional state-action space, enabling the agent to make informed decisions. This sparse dictionary model did not require a differentiable environment. Instead, it learned a parsimonious representation of the relevant physics from observations that in turn guided its policy optimization.

We note that for the North linac, the agent's energy model required significant time to stabilize and this additional warm-up period slowed the overall optimization. The predicted cavity configurations did not begin

improving until after the energy model converged and the only started to plateau after 250,000 training steps (Fig. 8). The RL agent studied in [32] benefited from an accurate and fully differentiable surrogate environment from the start of training. As a result, it achieved faster convergence and generated predictions with lower heat loads and higher Pareto coverage. A focus of future work will be close this gap by speeding up the surrogate training process, potentially via library pruning, coefficient regularization, or alternative optimizers.

CONCLUSION

We presented a RL framework that learns physical constraints from environmental observables in order to determine optimal control configurations for particle accelerator environments. We demonstrated this approach on surrogate models of the CEBAF facility at Jefferson Lab and found that it outperformed a traditional RL algorithm on a high-dimensional experimental optimization task. Our procedure blends model-free and model-based RL techniques. The agent does learn a model of relevant physical observables, such as energy, which plays a crucial role in shaping the policy. However, the agent does not attempt to predict the complete dynamics of the full environment, which can be much more complicated and are less likely to be represented by a simple mathematical rule (see Fig. S2-S4 for an example using the classic
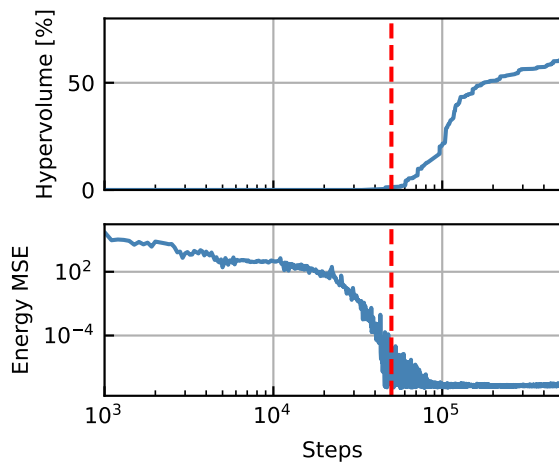
FIG. 8. **Sparse-LC-TD3 surrogate accuracy and predictive performance for North linac problem.** (*Top*) Pareto coverage at each episode step. (*Bottom*) Error rate of energy surrogate at each episode step. Red line denotes step 50,000 after which the energy surrogate began to stabilize and the predictions improved.

pendulum control problem). These results corroborate and extend recent work showing how using a differentiable physics-based surrogate environment improves RL performance on accelerator tasks by allowing gradient back-propagation [32]. Here, we showed how this surrogate can be learned from the data during agent training, bypassing the time-consuming task of building differentiable environments for complex particle accelerators. The surrogate also makes the agent more interpretable. An operator can probe the learned equation and verify it against known physics of the system. This grey-box machine learning approach is a promising path towards discovery [17, 22, 49], characterization [12, 13, 36, 37, 50], and control [8, 23, 51] of complex experimental systems whose governing physics may be obscured or unknown.

ACKNOWLEDGMENTS

[1] R. S. Sutton and A. Barto, *Reinforcement learning: an introduction*, second edition ed., Adaptive computation and machine learning (The MIT Press, Cambridge, Massachusetts London, England, 2020).

[2] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, Science **362**, 1140 (2018), publisher: American Association for the Advancement of Science.

[3] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, Nature **620**, 982 (2023), publisher: Nature Publishing Group.

[4] J. Manuel Davila Delgado and L. Oyedele, Advanced Engineering Informatics **54**, 101787 (2022).

[5] J. Kober, J. A. Bagnell, and J. Peters, The International Journal of Robotics Research **32**, 1238 (2013), publisher: SAGE Publications Ltd STM.

[6] V. Kain, S. Hirlander, B. Goddard, F. M. Velotti, G. Z. Della Porta, N. Bruchon, and G. Valentino, Phys. Rev. Accel. Beams **23**, 124801 (2020), publisher: American Physical Society.

[7] J. Kaiser, C. Xu, A. Eichler, A. Santamaria Garcia, O. Stein, E. Bründermann, W. Kuropka, H. Dinter, F. Mayet, T. Vinatier, F. Burkart, and H. Schlarb, Scientific Reports **14**, 15733 (2024), publisher: Nature Publishing Group.

[8] M. J. Falk, V. Alizadehyazdi, H. Jaeger, and A. Murugan, Physical Review Research **3**, 033291 (2021), publisher: American Physical Society.

[9] C. Banerjee, K. Nguyen, C. Fookes, and M. Raissi, "A Survey on Physics Informed Reinforcement Learning: Review and Open Problems," (2023).

[10] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Journal of Computational Physics **378**, 686 (2019).

[11] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, Nature Reviews Physics **3**, 422 (2021), number: 6 Publisher: Nature Publishing Group.

[12] M. S. Schmitt, J. Colen, S. Sala, J. Devany, S. Seetharaman, A. Caillier, M. L. Gardel, P. W. Oakes, and V. Vitelli, Cell **187**, 481 (2024), publisher: Elsevier.

[13] D. S. Seara, J. Colen, M. Fruchart, Y. Avni, D. Martin, and V. Vitelli, "Sociohydrodynamics: data-driven modelling of social behavior," (2024), arXiv:2312.17627 [cond-mat, physics:nlin, physics:physics].

[14] M. Schmidt and H. Lipson, Science **324**, 81 (2009), publisher: American Association for the Advancement of Science.

[15] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Proceedings of the National Academy of Sciences **113**, 3932 (2016), _eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1517384113.

[16] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, Annual Review of Fluid Mechanics **52**, 477 (2020).

[17] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, Proceedings of the National Academy of Sciences **116**, 22445 (2019), publisher: Proceedings of the National Academy of Sciences.

[18] K. Champion, P. Zheng, A. Y. Aravkin, S. L. Brunton, and J. N. Kutz, IEEE Access **8**, 169259 (2020).

[19] R. Supekar, B. Song, A. Hastewell, G. P. T. Choi, A. Mietke, and J. Dunkel, Proceedings of the National

Academy of Sciences **120**, e2206994120 (2023), publisher: Proceedings of the National Academy of Sciences.

[20] C. Joshi, S. Ray, L. M. Lemma, M. Varghese, G. Sharp, Z. Dogic, A. Baskaran, and M. F. Hagan, Physical Review Letters **129**, 258001 (2022), publisher: American Physical Society.

[21] M. Golden, R. O. Grigoriev, J. Nambisan, and A. Fernandez-Nieves, Science Advances **9**, eabq6120 (2023), publisher: American Association for the Advancement of Science.

[22] J. Colen, A. Poncet, D. Bartolo, and V. Vitelli, Physical Review Letters **133**, 107301 (2024), publisher: American Physical Society.

[23] N. Zolman, U. Fasel, J. N. Kutz, and S. L. Brunton, "SINDy-RL: Interpretable and Efficient Model-Based Reinforcement Learning," (2024), arXiv:2403.09110 [cs, eess, math].

[24] P. A. Adderley, S. Ahmed, T. Allison, R. Bachimanchi, K. Baggett, M. BastaniNejad, B. Bevins, M. Bevins, M. Bickley, R. M. Bodenstein, S. A. Bogacz, M. Bruker, A. Burrill, L. Cardman, J. Creel, Y.-C. Chao, G. Cheng, G. Ciovati, S. Chattopadhyay, J. Clark, W. A. Clemens, G. Croke, E. Daly, G. K. Davis, J. Delayen, S. U. De Silva, M. Diaz, R. Dickson, L. Doolittle, D. Douglas, M. Drury, E. Feldl, J. Fischer, A. Freyberger, V. Ganni, R. L. Geng, C. Ginsburg, J. Gomez, J. Grames, J. Gubeli, J. Guo, F. Hannon, J. Hansknecht, L. Harwood, J. Henry, C. Hernandez-Garcia, T. Hiatt, D. Higinbotham, S. Higgins, A. S. Hofler, J. Hogan, C. Hovater, A. Hutton, C. Jones, K. Jordan, M. Joyce, R. Kazimi, M. Keesee, M. J. Kelley, C. Keppel, A. Kimber, L. King, P. Kjeldsen, P. Kneisel, J. Kowal, G. A. Krafft, G. Lahti, T. Larrieu, R. Lauze, C. Leemann, R. Legg, R. Li, F. Lin, D. Machie, J. Mammosser, K. Macha, K. Mahoney, F. Marhauser, B. Mastracci, J. Matalevich, J. McCarter, M. McCaughan, L. Merminga, R. Michaud, V. Morozov, C. Mounts, J. Musson, R. Nelson, W. Oren, R. B. Overton, G. Palacios-Serrano, H.-K. Park, L. Phillips, S. Philip, F. Pilat, T. Plawski, M. Poelker, P. Powers, T. Powers, J. Preble, T. Reilly, R. Rimmer, C. Reece, H. Robertson, Y. Roblin, C. Rode, T. Satogata, D. J. Seidman, A. Seryi, A. Shabalina, I. Shin, C. Slominski, R. Slominski, M. Spata, D. Spell, J. Spradlin, M. Stirbet, M. L. Stutzman, S. Suhring, K. Surles-Law, R. Suleiman, C. Tennant, H. Tian, D. Turner, M. Tiefenback, O. Trofimova, A.-M. Valente, H. Wang, Y. Wang, K. White, C. Whitlatch, T. Whitlatch, M. Wiseman, M. J. Wissman, G. Wu, S. Yang, B. Yunn, S. Zhang, and Y. Zhang, Phys. Rev. Accel. Beams **27**, 084802 (2024), publisher: American Physical Society.

[25] D. Marcato, G. Arena, D. Bortolato, F. Gelain, V. Martinelli, E. Munaron, M. Roetta, G. Savarese, and G. A. Susto, in *2021 IEEE Conference on Control Technology and Applications (CCTA)* (2021) pp. 240–246, iSSN: 2768-0770.

[26] Y. Alanazi, M. Schram, K. Rajput, S. Goldenberg, L. Vidyaratne, C. Pappas, M. I. Radaideh, D. Lu, P. Ramuhalli, and S. Cousineau, Machine Learning with Applications **13**, 100484 (2023).

[27] J. P. Edelen and N. M. Cook, "Anomaly Detection in Particle Accelerators using Autoencoders," (2021), arXiv:2112.07793 [physics].

[28] W. Blokland, K. Rajput, M. Schram, T. Jeske, P. Ramuhalli, C. Peters, Y. Yucesan, and A. Zhukov, Phys-

[29] R. Roussel, A. Hanuka, and A. Edelen, Physical Review Accelerators and Beams **24**, 062801 (2021), publisher: American Physical Society.

[30] A. Scheinker, X. Pang, and L. Rybarcyk, Physical Review Special Topics - Accelerators and Beams **16**, 102803 (2013), publisher: American Physical Society.

[31] A. Scheinker, S. Hirlaender, F. M. Velotti, S. Gessner, G. Z. D. Porta, V. Kain, B. Goddard, and R. Ramjiawan, AIP Advances **10** (2020), 10.1063/5.0003423.

[32] K. Rajput, M. Schram, A. Edelen, J. Colen, A. Kasparian, R. Roussel, A. Carpenter, H. Zhang, and J. Benesch, "Harnessing the Power of Gradient-Based Simulations for Multi-Objective Optimization in Particle Accelerators," (2024), arXiv:2411.04817.

[33] B. Terzić, A. S. Hofler, C. J. Reeves, S. A. Khan, G. A. Krafft, J. Benesch, A. Freyberger, and D. Ranjan, Physical Review Special Topics - Accelerators and Beams **17**, 101003 (2014), publisher: American Physical Society.

[34] T. Jaisson, "Deep differentiable reinforcement learning and optimal trading," (2022), arXiv:2112.02944 [q-fin].

[35] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," (2000), arXiv:physics/0004057.

[36] K. A. Murphy and D. S. Bassett, Phys. Rev. Lett. **132**, 197201 (2024), publisher: American Physical Society.

[37] K. A. Murphy and D. S. Bassett, Proceedings of the National Academy of Sciences **121**, e2312988121 (2024), publisher: Proceedings of the National Academy of Sciences.

[38] M. S. Schmitt, M. Koch-Janusz, M. Fruchart, D. S. Seara, M. Rust, and V. Vitelli, "Information theory for data-driven model reduction in physics and biology," (2024).

[39] M. Koch-Janusz and Z. Ringel, Nature Physics **14**, 578 (2018), number: 6 Publisher: Nature Publishing Group.

[40] D. E. Gökmen, Z. Ringel, S. D. Huber, and M. Koch-Janusz, Physical Review Letters **127**, 240603 (2021), publisher: American Physical Society.

[41] S. Schoenholz, E. Cubuk, D. Sussman, E. Kaxiras, and A. Liu, Nature Physics **12**, 469 (2016), publisher: Nature Publishing Group.

[42] E. Zhang, M. Dao, G. E. Karniadakis, and S. Suresh, Science Advances **8**, eabk0644 (2022), _eprint: https://www.science.org/doi/pdf/10.1126/sciadv.abk0644.

[43] N. M. Mangan, S. L. Brunton, J. L. Proctor, and J. N. Kutz, IEEE Transactions on Molecular, Biological and Multi-Scale Communications **2**, 52 (2016).

[44] C. J. Soelistyo, G. Vallardi, G. Charras, and A. R. Lowe, Nature Machine Intelligence **4**, 636 (2022), publisher: Nature Publishing Group.

[45] N. Romeo, A. Hastewell, A. Mietke, and J. Dunkel, eLife **10**, e68679 (2021), publisher: eLife Sciences Publications, Ltd.

[46] J. Benesch, "A longitudinal study of field emission in CEBAF's SRF cavities 1995-2015," (2015), arXiv:1502.06877 [physics].

[47] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," (2018), arXiv:1802.09477 [cs, stat].

[48] A. P. Guerreiro, C. M. Fonseca, and L. Paquete, ACM Comput. Surv. **54**, 119:1 (2021).

[49] M. Lefebvre, J. Colen, N. Claussen, F. Brauns, M. Raich, N. Mitchell, M. Fruchart, V. Vitelli, and S. J. Stre-

ichan, "Learning a conserved mechanism for early neu- roectoderm morphogenesis," (2024), arXiv:2405.18382 [physics].

[50] J. M. Hanlan, S. Dillavou, A. J. Liu, and D. J. Durian, "Cornerstones are the Key Stones: Using Interpretable Machine Learning to Probe the Clogging Process in 2D Granular Hoppers," (2024), arXiv:2407.05491 [cond-mat].

[51] C. Floyd, A. R. Dinner, and S. Vaikuntanathan, "Tai-loring interactions between active nematic defects with reinforcement learning," (2024), arXiv:2411.09588 [cond-mat].

[52] J. Blank and K. Deb, IEEE Access **8**, 89497 (2020).

[53] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI Gym," (2016), arXiv:1606.01540 [cs].

SUPPLEMENTARY INFORMATION

Algorithm S1: Twin-delayed deep deterministic policy gradient (TD3)

Initialize critics $Q_{\theta_1}, Q_{\theta_2}$ and policy network π_ϕ
Initialize target networks $\theta_{i'} \leftarrow \theta_i$, $\phi' \leftarrow \phi$
Initialize replay buffer $\mathcal{B}$
for e *in* $1 \ldots N_e$ **do**
    Observe state s and select action $a \sim \pi_\phi$
    Execute a in environment
    Observe next state s', reward r, and terminal signal d
    Store (s, a, r, s', d) in replay buffer $\mathcal{B}$
    if *time to update* **then**
        Sample batch of transitions $b \sim \mathcal{B}$
        $a' \leftarrow \pi_{\phi'}(s') + \epsilon \mathcal{N}(0, \sigma)$
        $y_i \leftarrow r + \gamma \min_i Q_{\theta_i'}(s', a')$
        Update Q functions with gradient descent
          using $\frac{1}{|b|} \nabla_{\theta_i} \sum (Q_{\theta_i}(s, a) - y_i)^2$
        Update policy π with gradient ascent using
          $\frac{1}{|b|} \nabla_\phi \sum Q_{\phi_1}(s, \pi_\phi(s))$
        Update target networks:
        $\theta_i' \leftarrow \tau \theta_i' + (1 - \tau) \theta_i$
        $\phi' \leftarrow \tau \phi' + (1 - \tau) \phi$
    end
end

Quantifying Pareto front coverage for multi-objective RL

To quantify the performance of RL agents in the multi-objective problem, we used the normalized hypervolume or S-metric [48] approach as in [32]. For this metric, an ideal and reference point are chosen for each objective value. These represent an upper and lower bound respectively for the region of objective space. The hypervolume H is the volume of objective space delineated by the Pareto front. The normalized hypervolume reported in Table III is given by

$$NH = H/A \times 100 \tag{S1}$$

Here A is the total volume of the objective region defined by the reference and ideal points. Figure S1 visualizes this calculation for the multi-objective problem of minimizing heat load and trip rate. The normalized hypervolumes reported in this paper were computed using the Pymoo package [52], with respect to the ideal and reference points given in Table S1. For the CEBAF 8D, 16D, and 32D problems, the reference and ideal points are identical to those used in [32].

For the North linac, we selected new upper and lower bounds to enable a clearer comparison between Sparse-LC-TD3 and standard TD3, as the latter produced configurations above the original reference point leading to
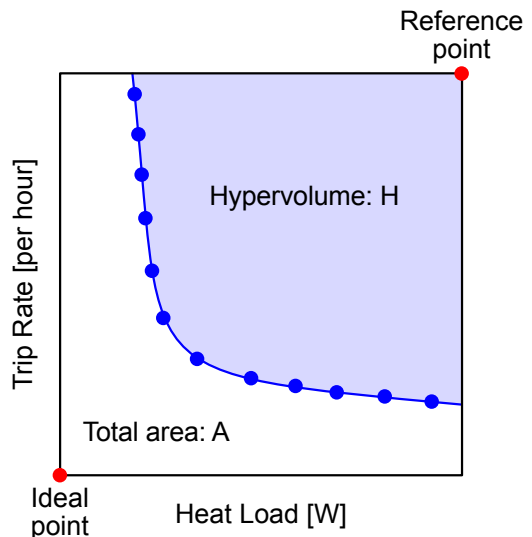

FIG. S1. Hypervolume calculation for a 2D Pareto front using chosen reference an ideal points.

Problem	Ref (H, T)	Ideal (H, T)
8D	(22.4, 0.05)	(20.9, 0.015)
16D	(100, 0.40)	(88, 0.015)
32D	(290, 0.40)	(262, 0.01)
North linac	(2800, 5.5)	(2380, 0.5)

TABLE S1. Reference and ideal points used for multi-objective hypervolume calculation.

a normalized hypervolume of 0. For the DDRL agent results reported in [32], we calculated a normalized hypervolume of 77.79 with respect to the bounds in Table S1.

Benchmark on OpenAI-Pendulum

To demonstrate our algorithm's performance on a more standard reinforcement learning environment, we consider here the Pendulum-v1 environment from OpenAI-Gym [53]. The system contains a free pendulum subject to gravity. The agent observes a state $s = (x, y, \omega)$ where (x, y) is the position of the pendulum end and ω is the angular velocity. The objective is for the agent to continuously apply a torque τ such that the pendulum swings upright and remains inverted. We add the additional constraint that the kinetic energy remain below some specific value $\frac{1}{2} I \omega^2 \leq T_{\max}$.

We trained RL agents using Algorithms 1-2 for 200 episodes with $T_{\max} = 2$ and plotted the reward after each episode in Fig. S2. As a comparison, we also plot the reward for a TD3 agent (Alg. S1) with no energy penalty ($T_{\max} \rightarrow \infty$). After training, the two LC-TD3 agents achieve comparable rewards to the standard TD3 agent.

In Fig. S3, we examined the agent's learned behavior and compared it to the known physical behavior of the pendulum environment. The reward function was given exactly by

$$R = -\theta^2 - \alpha\omega^2 - \beta\tau^2 \tag{S2}$$

The critic's learned Q-function was more complicated and depended nonlinearly on the pendulum position, the direction of motion, and the direction of the applied torque. It was difficult to represent this function using a simple equation. The energy function was comparatively simple and the learned surrogate is accurate to the exact values given by the environment.

For the Sparse-LC-TD3 agent, we compared the learned surrogate model to the exact equation for energy within the environment. Recall that o is returned by the environment after performing an action, so the model predicts the new kinetic energy after the angular velocity updates. The ground truth equation is

$$E(t + \Delta t) = \frac{1}{2}I\left[\omega + \left(\frac{3g}{2\ell}x + \frac{3}{m\ell^2}\tau\right)\Delta t\right]^2 \tag{S3}$$

Here I is the moment of inertia, g is the acceleration due to gravity, m is the pendulum mass, ℓ is the pendulum length, and Δt is the simulation time step. The term in brackets is the angular velocity after one time step, which can be obtained by inspecting the open-source environment code or by manual derivation. In Fig. S4, we compared the numerical values for each coefficient in this equation to the corresponding learned values in the weight vector $\mathbf{w}$. The agent learned coefficients that closely align with the ground truth. Examining the constraint surrogate in this way allows one to ensure that the policy is shaped by an accurate representation of the relevant physics of the system.
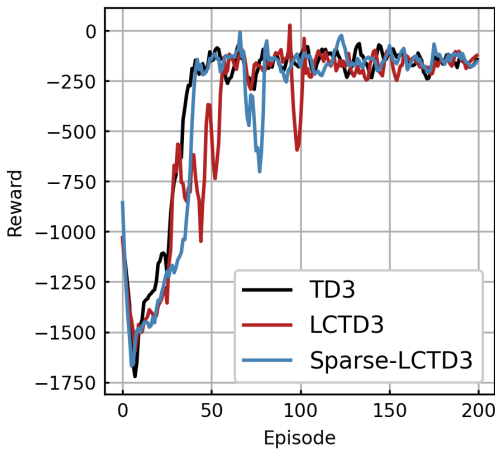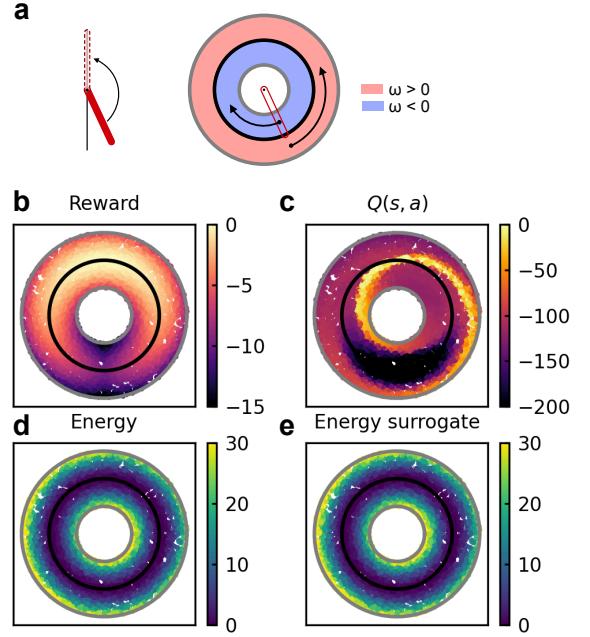

FIG. S3. (a) The Pendulum-v1 objective is to swing the pendulum upright. We plot the state space such that the angular coordinate shows the pendulum position and the radial coordinate denotes the angular velocity. (b) Ground truth reward landscape for the pendulum environment. (c) Learned critic $Q(s, a)$ is a nonlinear function of state and action variables. (d) Ground truth energy function. (e) Energy surrogate model. All plots show ground truth and predictions for $N = 5000$ randomly sampled state-action pairs.
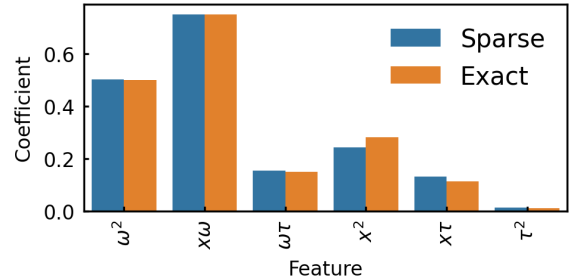

FIG. S2. Training curves for TD3, LC-TD3, and Sparse-LC-TD3 agents trained on a Pendulum-v1 environment

FIG. S4. Sparse-LC-TD3 energy surrogate coefficients for Pendulum-v1 environment compared to the exact result of Eq. S3

CEBAF test environment results

Objective	Agent	Heat load (W)	Trip rate (h^{-1})	Convergence
Heat	LC-TD3	21.18 (0.04)	0.0410 (0.0026)	88 %
	Sparse-LC-TD3	21.27 (0.07)	0.0395 (0.0023)	100 %
	TD3	21.06 (0.03)	0.0466 (0.0070)	100 %
Multi	LC-TD3	21.32 (0.04)	0.0236 (0.0006)	75 %
	Sparse-LC-TD3	21.37 (0.09)	0.0242 (0.0007)	100 %
	TD3	21.27 (0.05)	0.0237 (0.0006)	100 %
Trip	LC-TD3	22.32 (0.06)	0.0187 (0.0004)	100 %
	Sparse-LC-TD3	22.54 (0.07)	0.0192 (0.0004)	100 %
	TD3	22.24 (0.10)	0.0177 (0.0004)	100 %

TABLE S2. End-of-training performance for each RL agent and objective in the single-cryomodule 8D optimization problem. Mean and standard deviation computed over $N = 8$ trials. Convergence denotes the percentage of trials that converged to a configuration producing an energy gain within the allowed range.

Objective	Agent	Heat load (W)	Trip rate (h^{-1})	Convergence
Heat	LC-TD3	89.7 (0.1)	0.3796 (0.0261)	100 %
	Sparse-LC-TD3	90.0 (0.3)	0.3651 (0.0212)	75 %
	TD3	88.0 (0.2)	0.7013 (0.0402)	100 %
Multi	LC-TD3	98.0 (0.4)	0.0398 (0.0014)	88 %
	Sparse-LC-TD3	98.0 (0.2)	0.0393 (0.0008)	100 %
	TD3	97.5 (0.4)	0.0386 (0.0013)	100 %
Trip	LC-TD3	131.8 (1.9)	0.0081 (0.0002)	100 %
	Sparse-LC-TD3	132.5 (1.6)	0.0088 (0.0002)	100 %
	TD3	143.3 (5.8)	0.0072 (0.0004)	100 %

TABLE S3. End-of-training performance for each RL agent and objective in the two-cryomodule 16D optimization problem. Mean and standard deviation computed over $N = 8$ trials. Convergence denotes the percentage of trials that converged to a configuration producing an energy gain within the allowed range.

Objective	Agent	Heat load (W)	Trip rate (h^{-1})	Convergence
Heat	LC-TD3	265.7 (0.5)	0.7017 (0.0249)	88 %
	Sparse-LC-TD3	266.0 (0.8)	0.7086 (0.0301)	88 %
	TD3	262.6 (0.2)	0.9804 (0.0545)	100 %
Multi	LC-TD3	282.5 (0.6)	0.0344 (0.0012)	100 %
	Sparse-LC-TD3	282.3 (0.6)	0.0353 (0.0014)	88 %
	TD3	281.0 (0.7)	0.0352 (0.0012)	100 %
Trip	LC-TD3	349.2 (3.2)	0.0083 (0.0005)	100 %
	Sparse-LC-TD3	341.7 (2.0)	0.0100 (0.0003)	100 %
	TD3	355.8 (8.9)	0.0096 (0.0015)	100 %

TABLE S4. End-of-training performance for each RL agent and objective in the four-cryomodule 32D optimization problem. Mean and standard deviation computed over $N = 8$ trials. Convergence denotes the percentage of trials that converged to a configuration producing an energy gain within the allowed range.
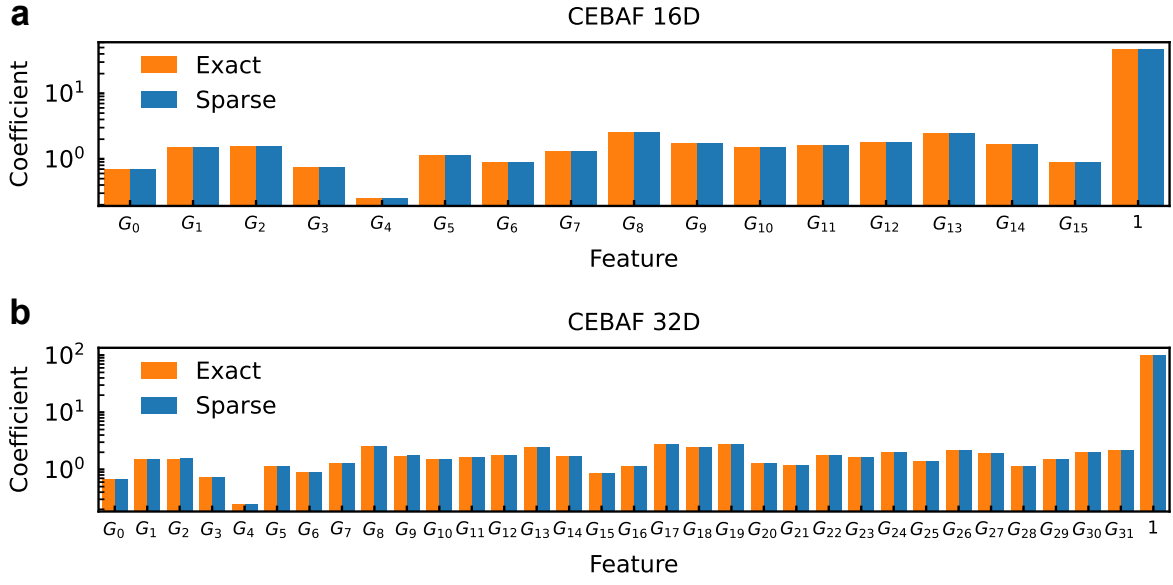
FIG. S5. Comparison of average learned coefficients compared to their values set in intermediate-scale CEBAF environments. (**a**) Results for CEBAF 16D environment. (**b**) Results for CEBAF 32D environment. Plots show all coefficients with magnitudes above a threshold $\tau = 0.05$. Coefficient values are averaged over all trials of the Sparse LC-TD3 agent.