

Explainable physics-based constraints on reinforcement learning for accelerator optimization

Jonathan Colen
Old Dominion University

Machine Learning Applications for Particle Accelerators
April 9, 2025



High dimensional accelerator optimization problem

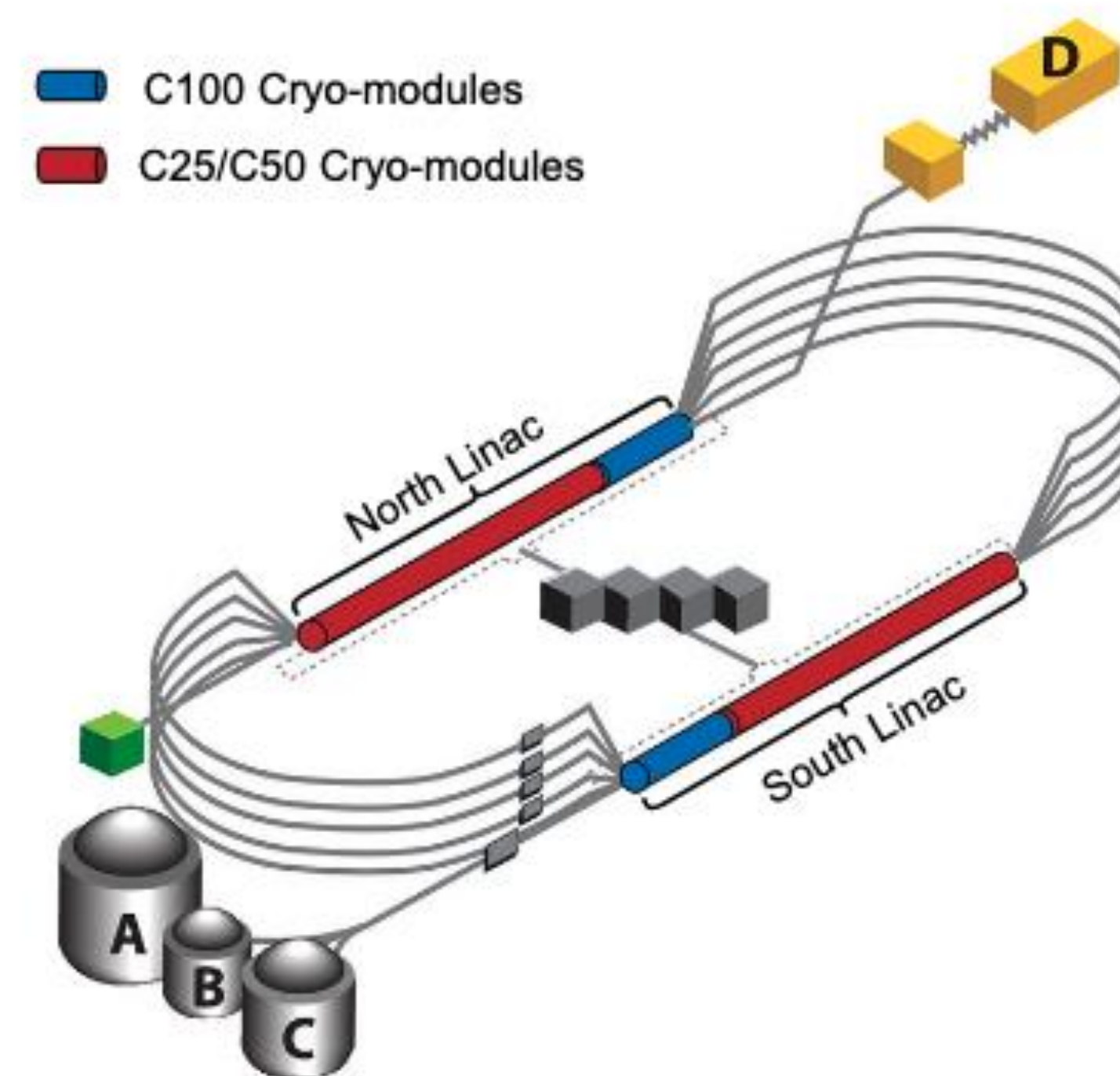
- CEBAF accelerates electrons using 200 configurable superconducting cavities
- Operators tune cavity gradients to satisfy objectives:
 - Maintain target **energy gain**

$$E = \sum_i G_i \ell_i \quad |E - E_{target}| \leq \delta E$$

- Minimize hazards: **heat load** and **FSD trip rate**

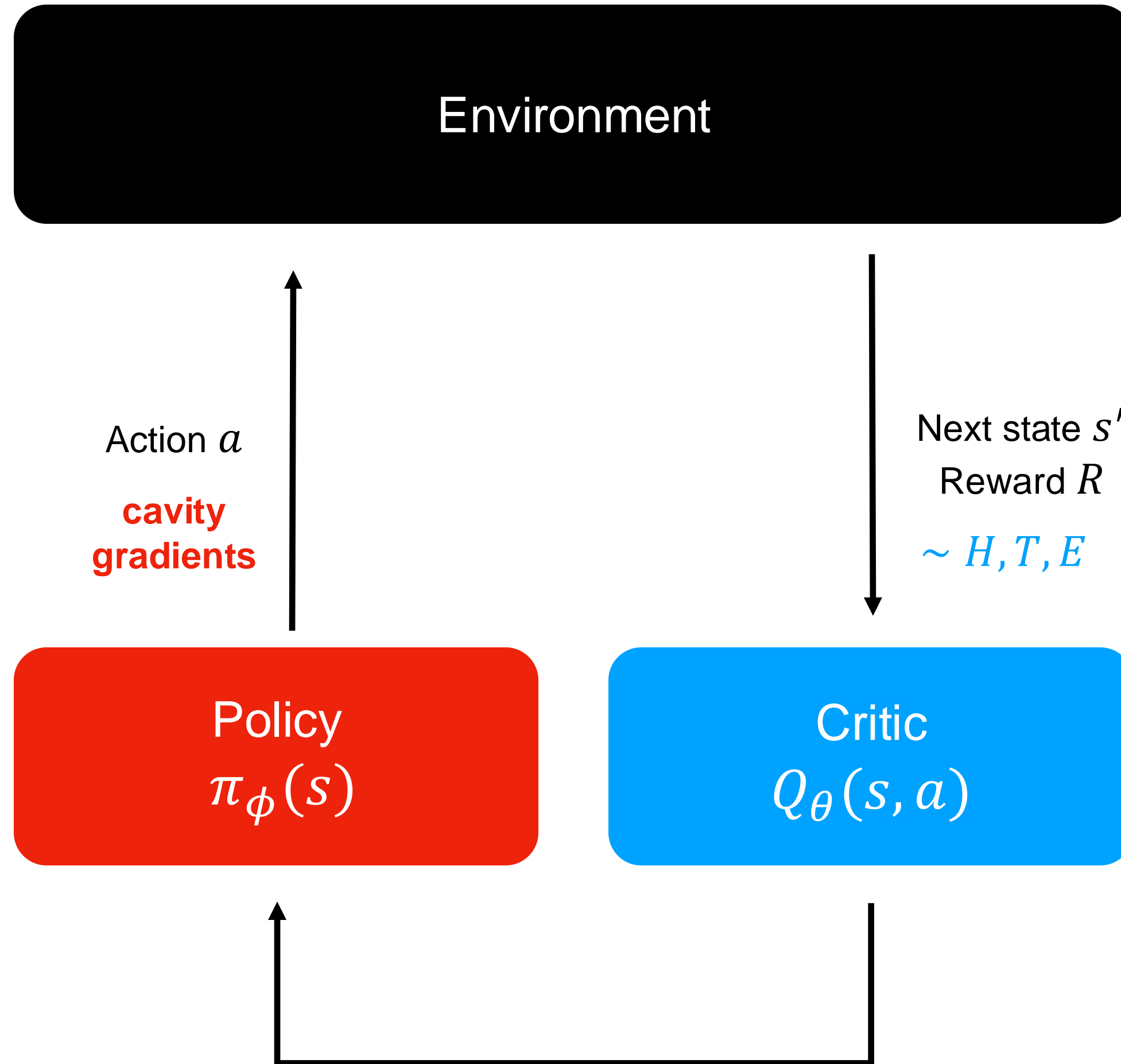
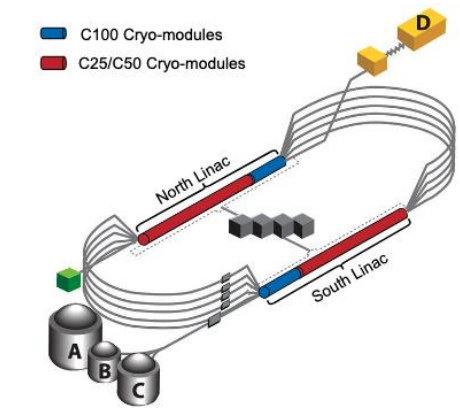
$$H = \sum_i \frac{G_i^2 \ell_i}{\omega_i Q_i(G_i)} \quad T = \sum_i \exp\{A + B_i(G_i - F_i)\}$$

Rajput et al, ML S&T (2025)

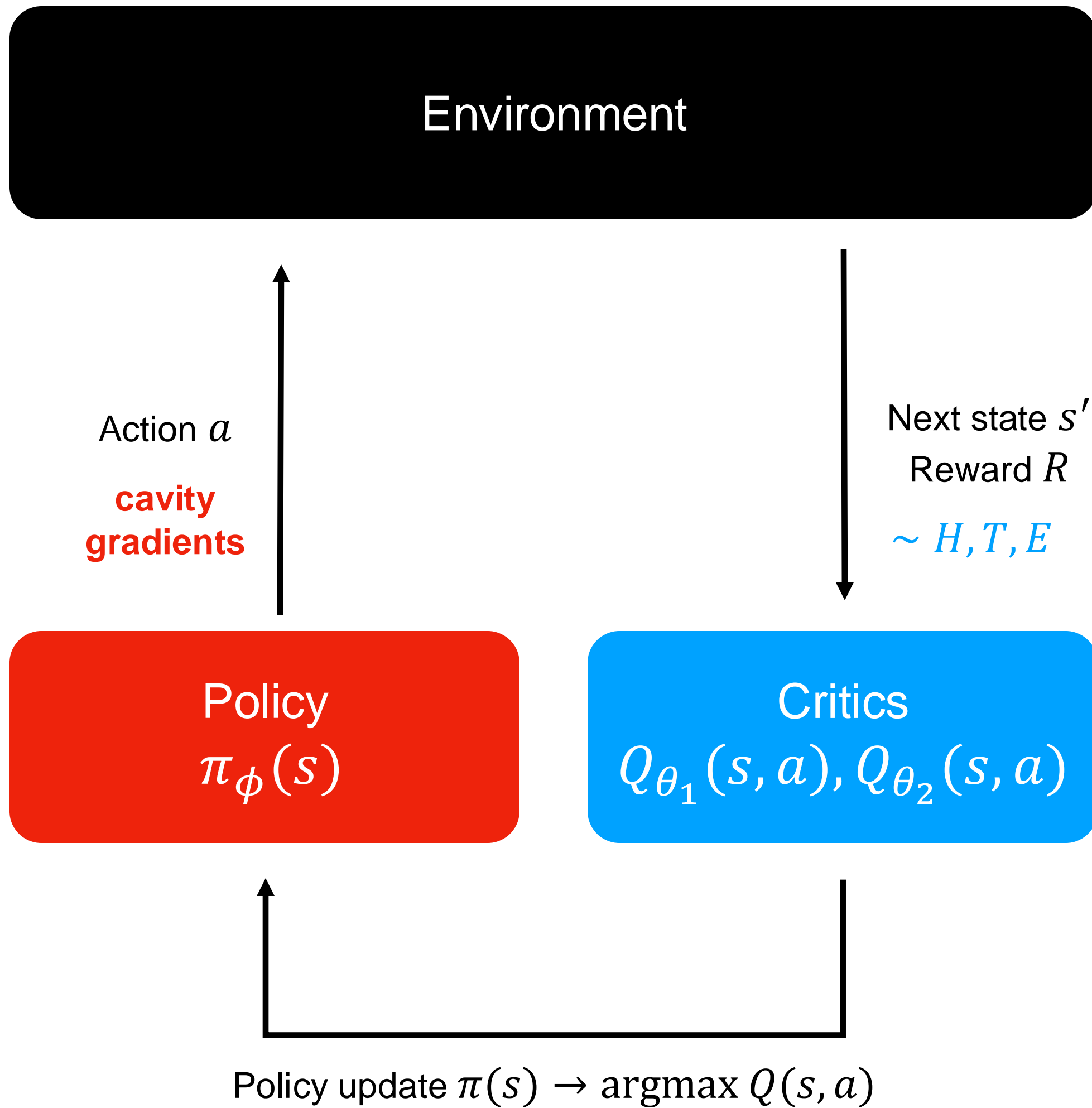
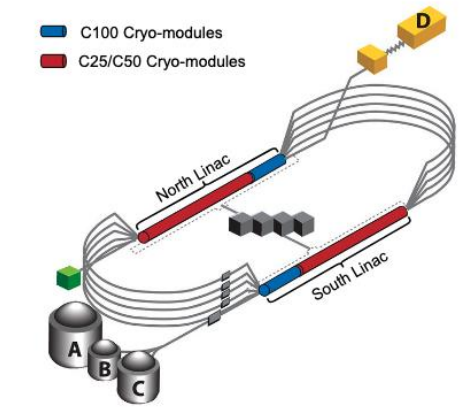


CEBAF accelerator at Jefferson Lab

Reinforcement learning for accelerator optimization



Reinforcement learning for accelerator optimization



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

```

Initialize critics  $Q_{\theta_1}, Q_{\theta_2}$  and policy network  $\pi_\phi$ 
Initialize target networks  $\theta_{i'} \leftarrow \theta_i, \phi' \leftarrow \phi$ 
Initialize replay buffer  $\mathcal{B}$ 
for  $e$  in  $1 \dots N_e$  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  $\pi$  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

```

Reinforcement learning for accelerator optimization

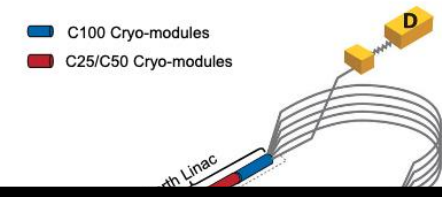
Algorithm S1: Twin-delayed deep deterministic policy gradient (TD3)

We know that TD3 **fails** for our CEBAF problem

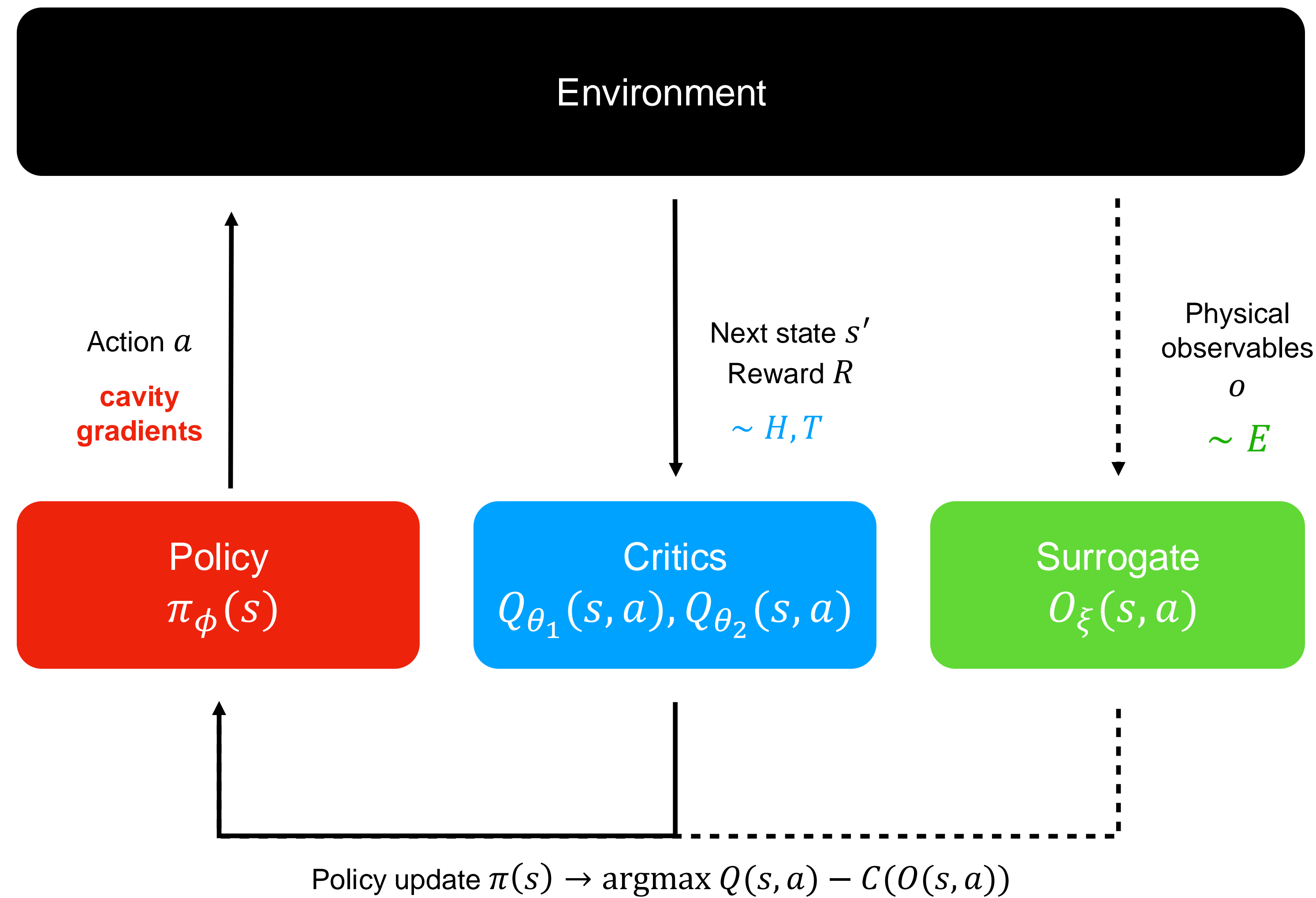
How can we **adapt** it to improve performance?

Policy update $\pi(s) \rightarrow \operatorname{argmax} Q(s, a)$

```
     $v_i \leftarrow \tau v_i + (1 - \tau) v_i$   
     $\phi' \leftarrow \tau \phi' + (1 - \tau) \phi$   
end  
end
```



Learnable constraint for physical observables



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

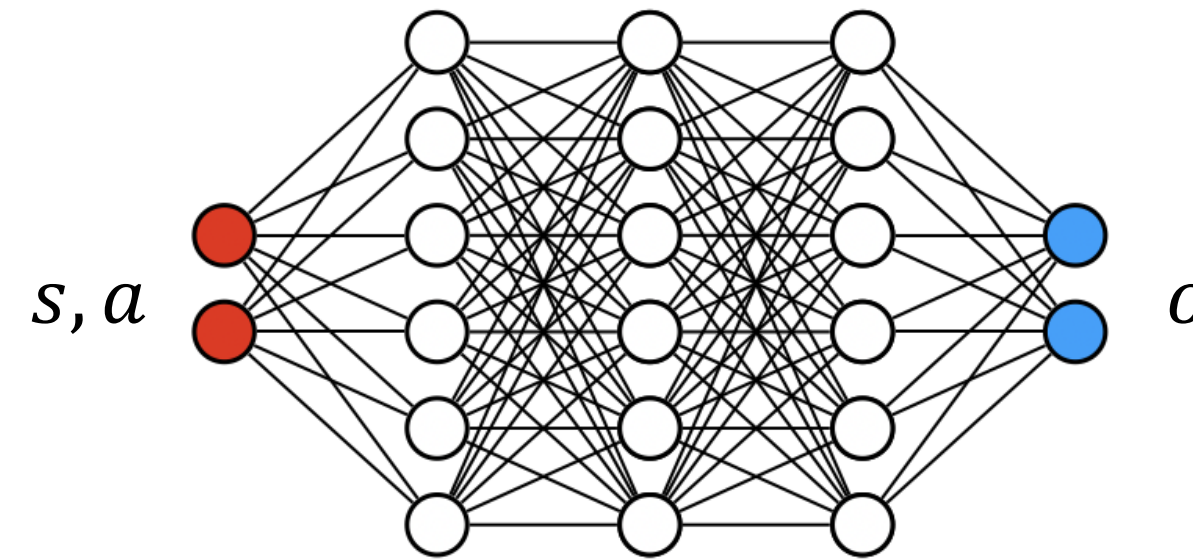
```

Initialize critics  $Q_{\theta_1}, Q_{\theta_2}$  and policy network  $\pi_\phi$ 
Initialize target networks  $\theta_{i'} \leftarrow \theta_i, \phi' \leftarrow \phi$ 
Initialize replay buffer  $\mathcal{B}$ 
Initialize learnable surrogate network  $O_\xi$  and
constraint function  $C(o)$ 
for  $e$  in  $1 \dots N_e$  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  $O_\xi$  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  $\pi$  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

```

Learnable constraint for physical observables

Surrogate
 $O_\xi(s, a)$

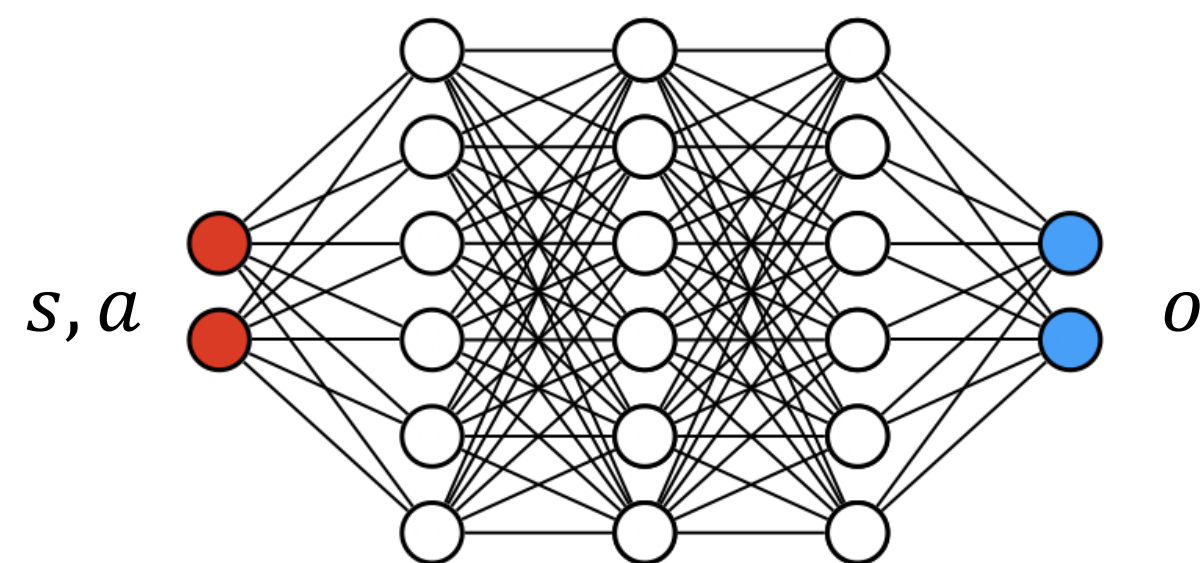


Algorithm 1: TD3 + learnable constraints (LC-TD3)

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 $1 \dots N_e$ **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

Learnable constraint for physical observables

Surrogate
 $O_\xi(s, a)$



$$\text{Minimize } \mathbb{E} \left[\left(O_\xi(s, a) - o \right)^2 \right]$$

Learn how actions affect
energy gain

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

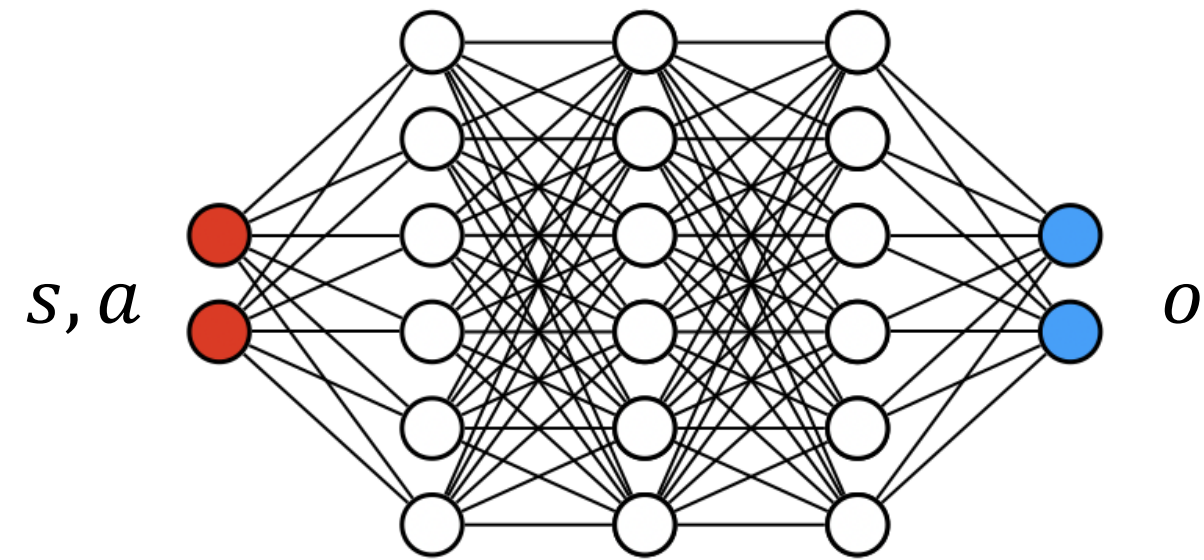
```

Initialize critics  $Q_{\theta_1}, Q_{\theta_2}$  and policy network  $\pi_\phi$ 
Initialize target networks  $\theta_{i'} \leftarrow \theta_i, \phi' \leftarrow \phi$ 
Initialize replay buffer  $\mathcal{B}$ 
Initialize learnable surrogate network  $O_\xi$  and
constraint function  $C(o)$ 
for  $e$  in  $1 \dots N_e$  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  $O_\xi$  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  $\pi$  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

```

Learnable constraint for physical observables

Surrogate
 $O_\xi(s, a)$



$$\text{Minimize } \mathbb{E} \left[\left(O_\xi(s, a) - o \right)^2 \right]$$

Learn how actions affect
energy gain

$$\text{Minimize } \mathbb{E} \left[C \left(O_\xi(s, \pi_\phi(s)) \right) \right]$$

Use surrogate to estimate
energy target

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

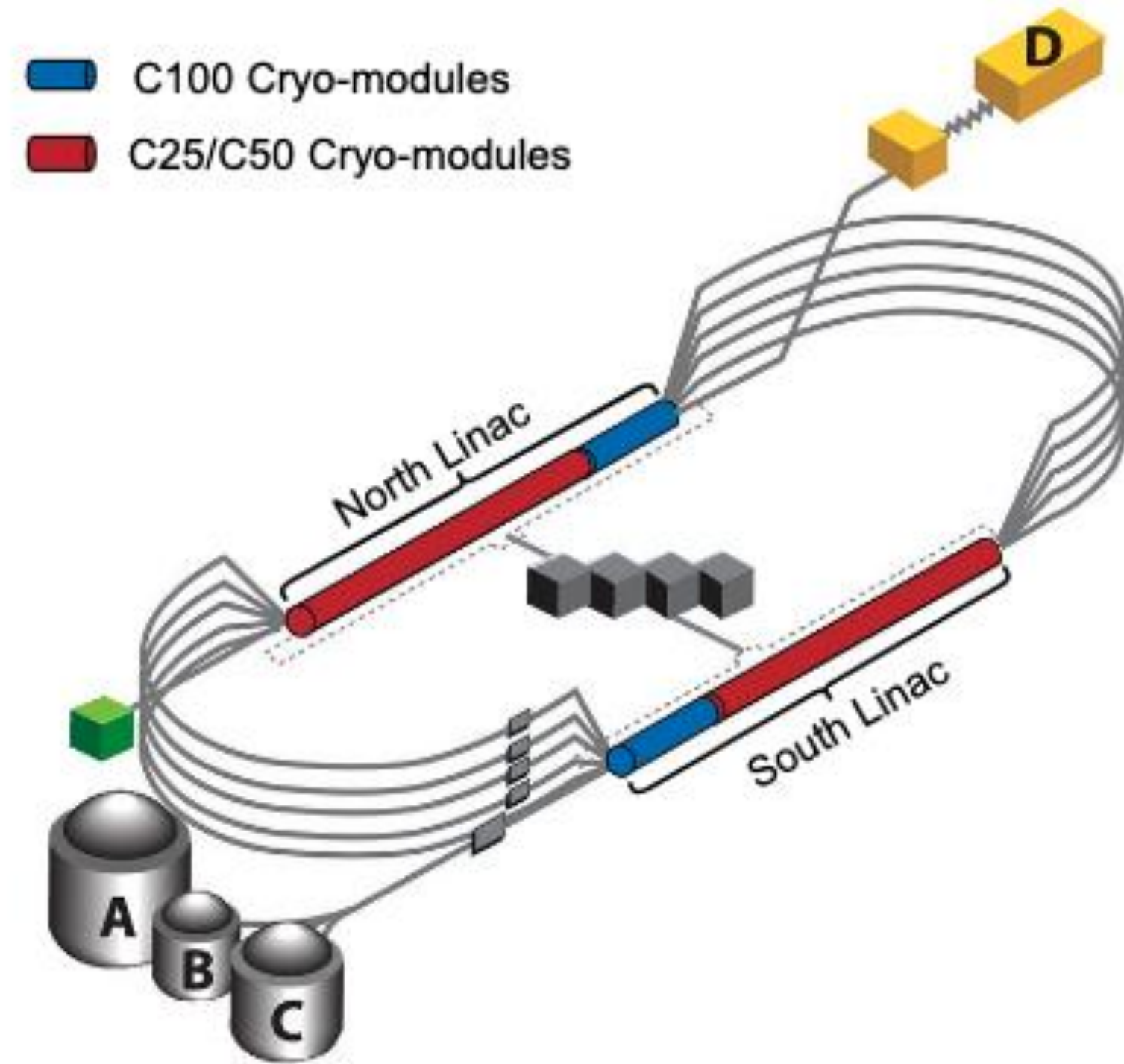
```

Initialize critics  $Q_{\theta_1}, Q_{\theta_2}$  and policy network  $\pi_\phi$ 
Initialize target networks  $\theta_{i'} \leftarrow \theta_i, \phi' \leftarrow \phi$ 
Initialize replay buffer  $\mathcal{B}$ 
Initialize learnable surrogate network  $O_\xi$  and
constraint function  $C(o)$ 
for  $e$  in  $1 \dots N_e$  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  $O_\xi$  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  $\pi$  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

```

CEBAF evaluation testbed

Rajput *et al*, ML S&T (2025)

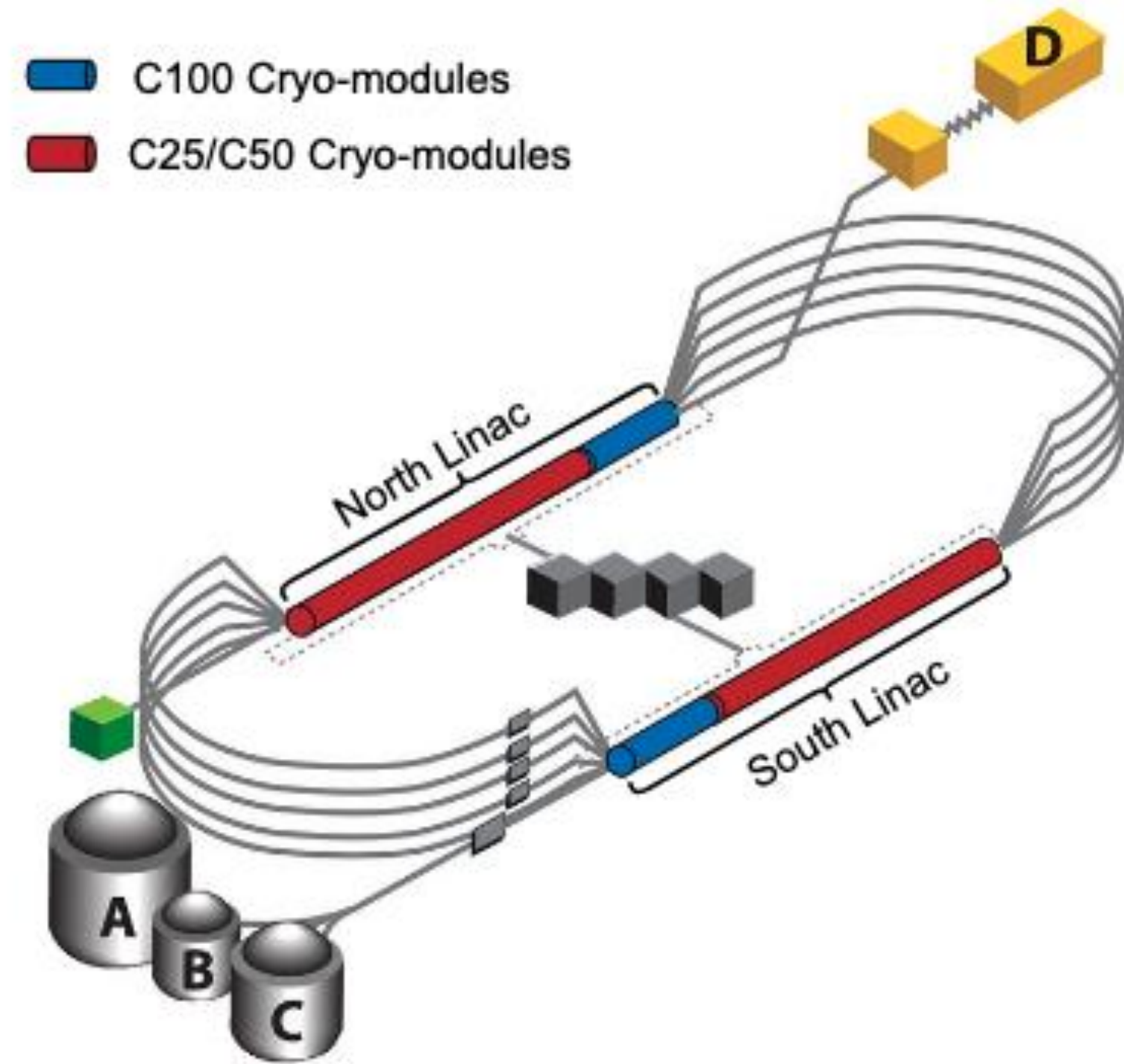


Problem	Num. Cavities	E_{target} (MeV)	δE (MeV)
8D	8	20.08	0.40
16D	16	50.00	0.60
32D	32	120.00	0.80
North linac	200	1050.00	2.00

Test algorithms on problems ranging from **low-dimensional** cryomodule optimization to **high-dimensional** linac optimization

CEBAF evaluation testbed

Rajput *et al*, ML S&T (2025)

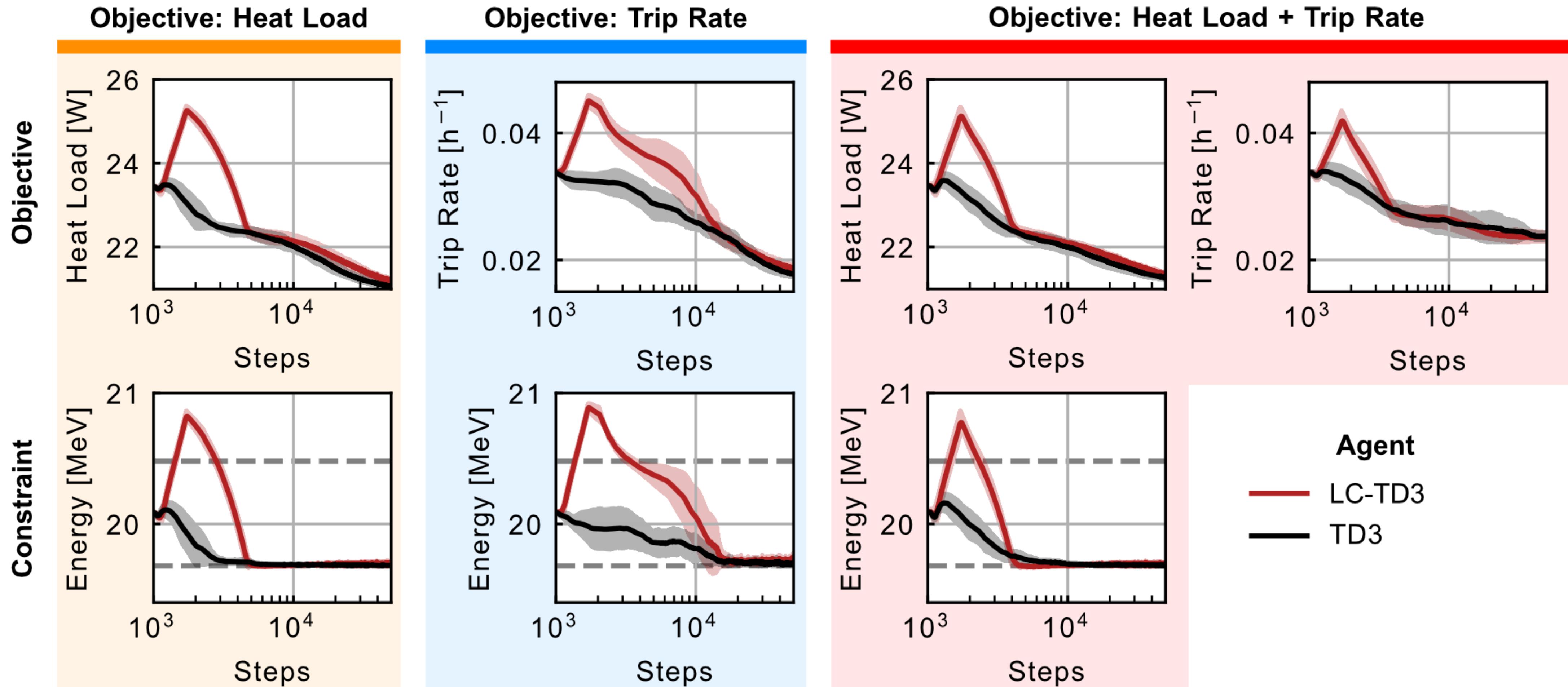


Problem	Num. Cavities	E_{target} (MeV)	δE (MeV)
8D	8	20.08	0.40
16D	16	50.00	0.60
32D	32	120.00	0.80
North linac	200	1050.00	2.00

Single-cryomodule optimization

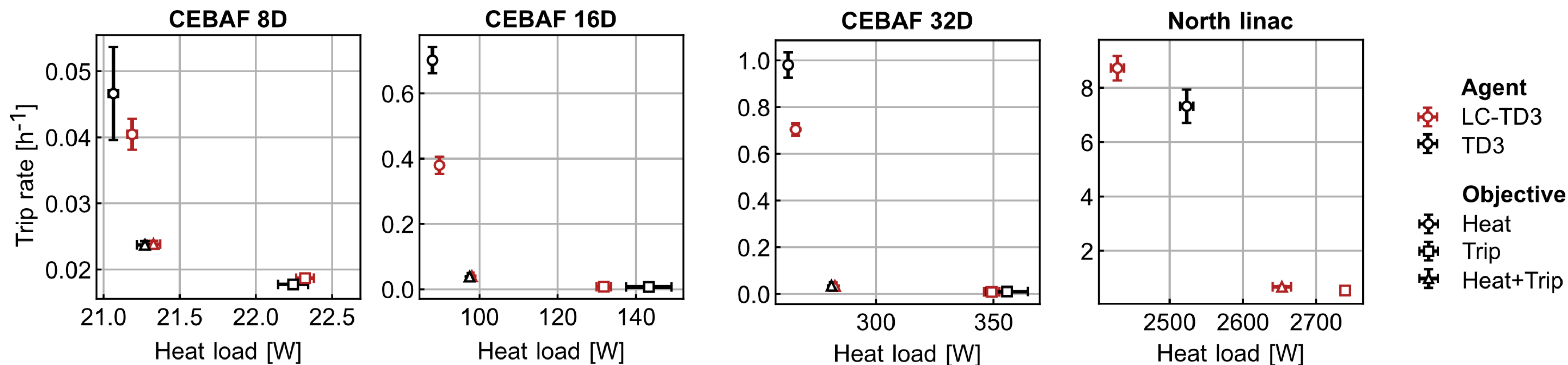
Test algorithms on problems ranging from **low-dimensional** cryomodule optimization to **high-dimensional** linac optimization

Single-cryomodule optimization example



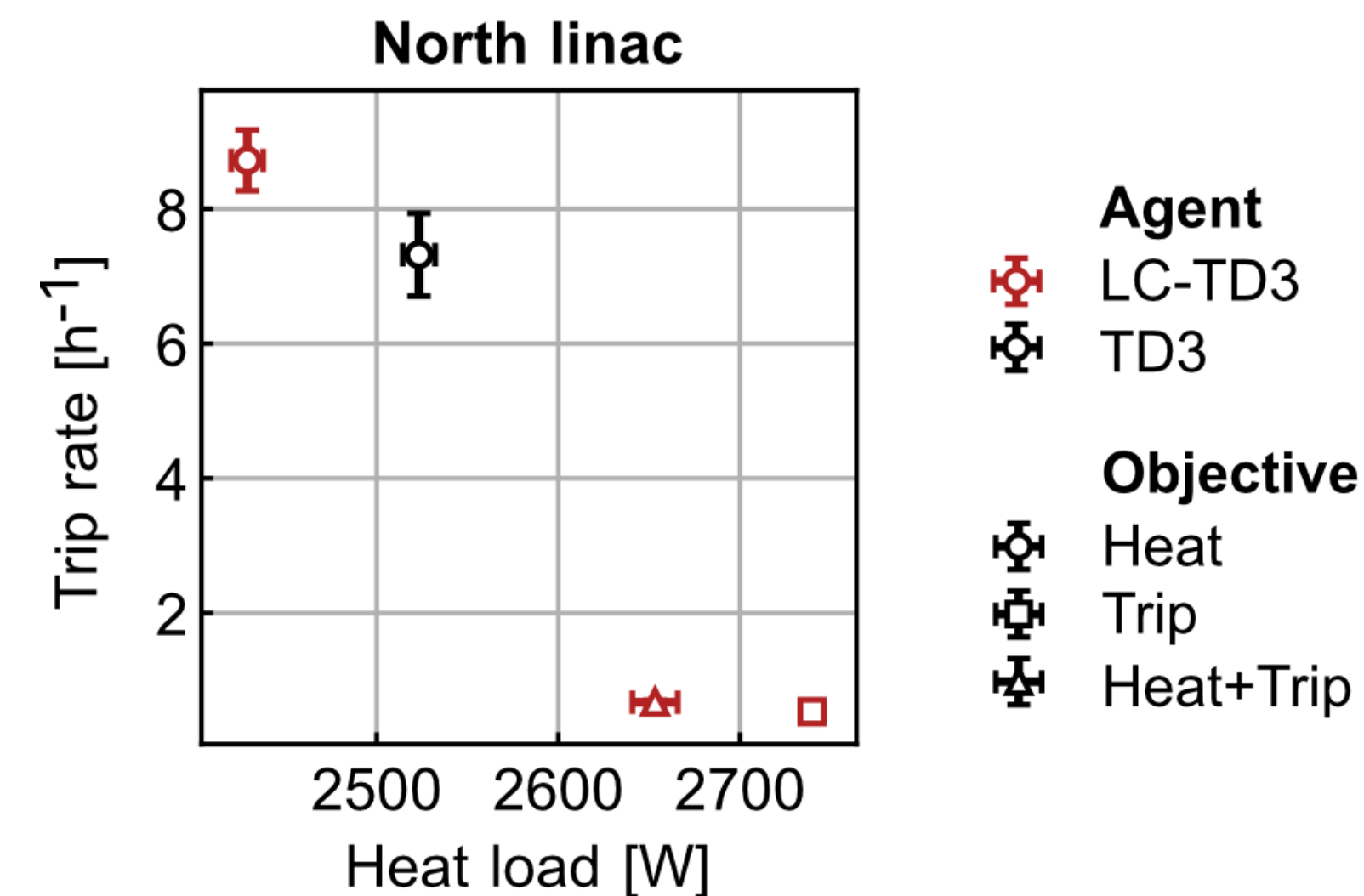
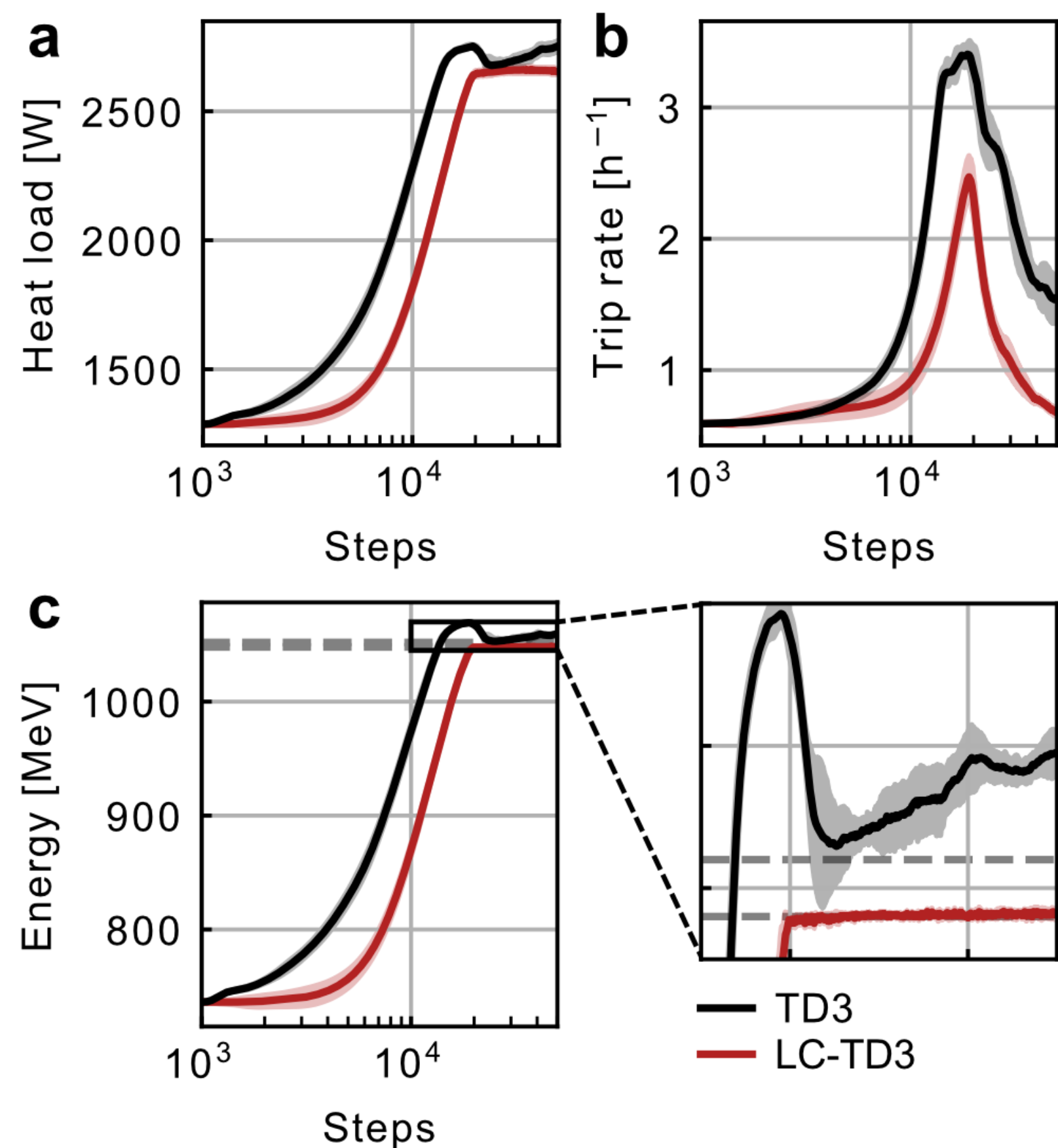
Algorithms converge to lower bound of target energy range

Optimization performance on higher-dimensional problems



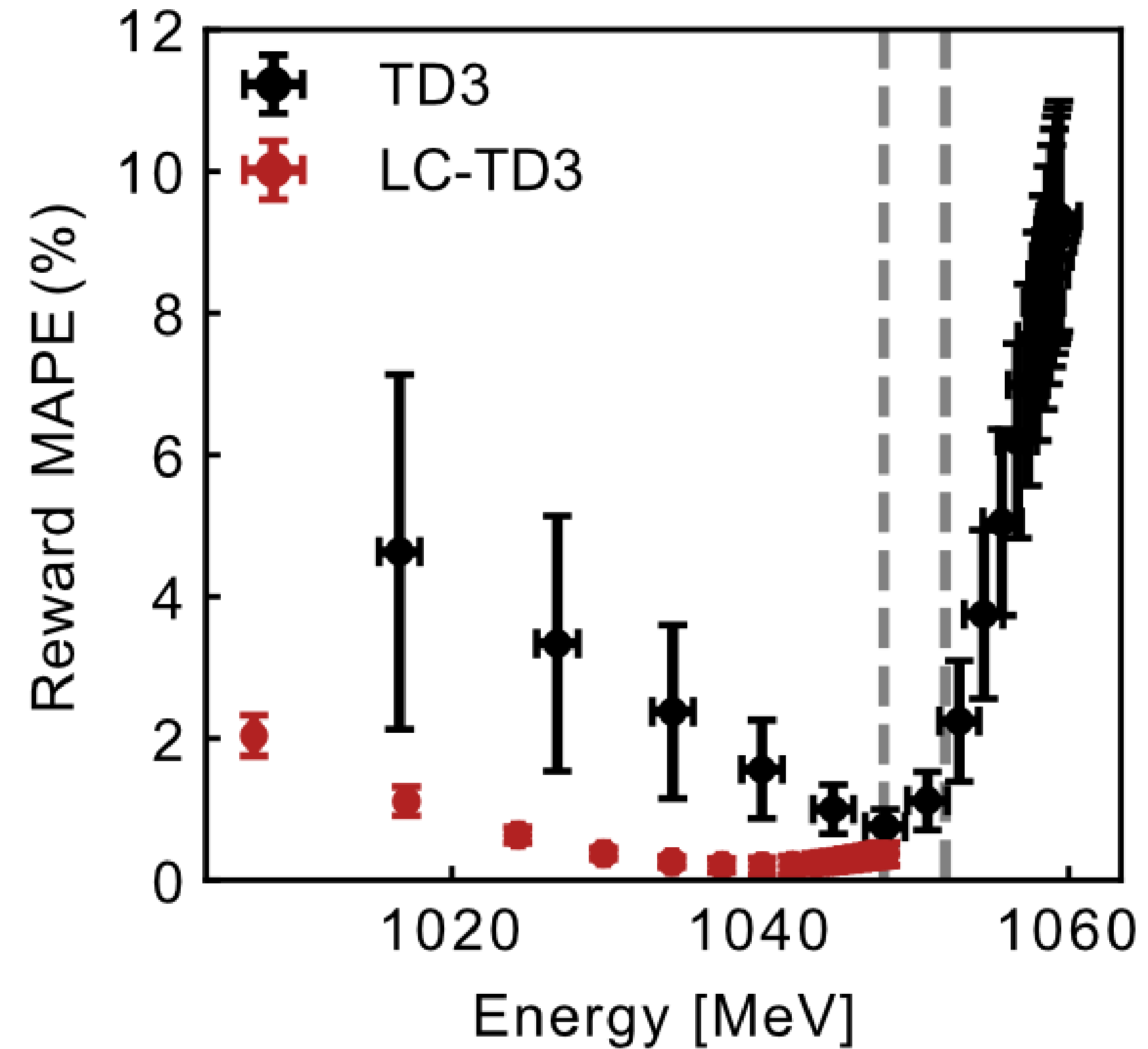
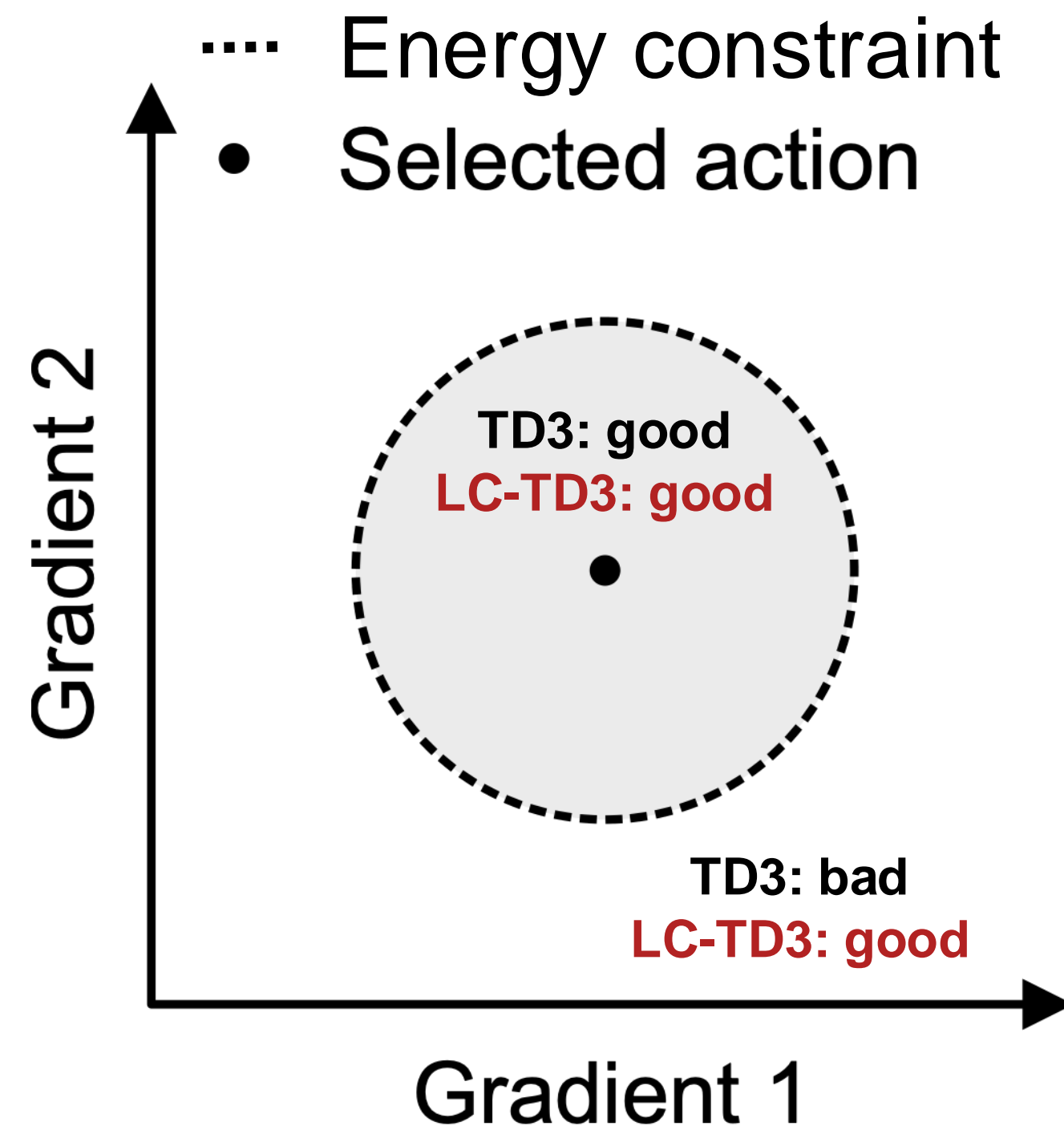
Learnable constraint enables superior performance on North linac optimization

Optimization performance on higher-dimensional problems



Learnable constraint aids convergence to target energy

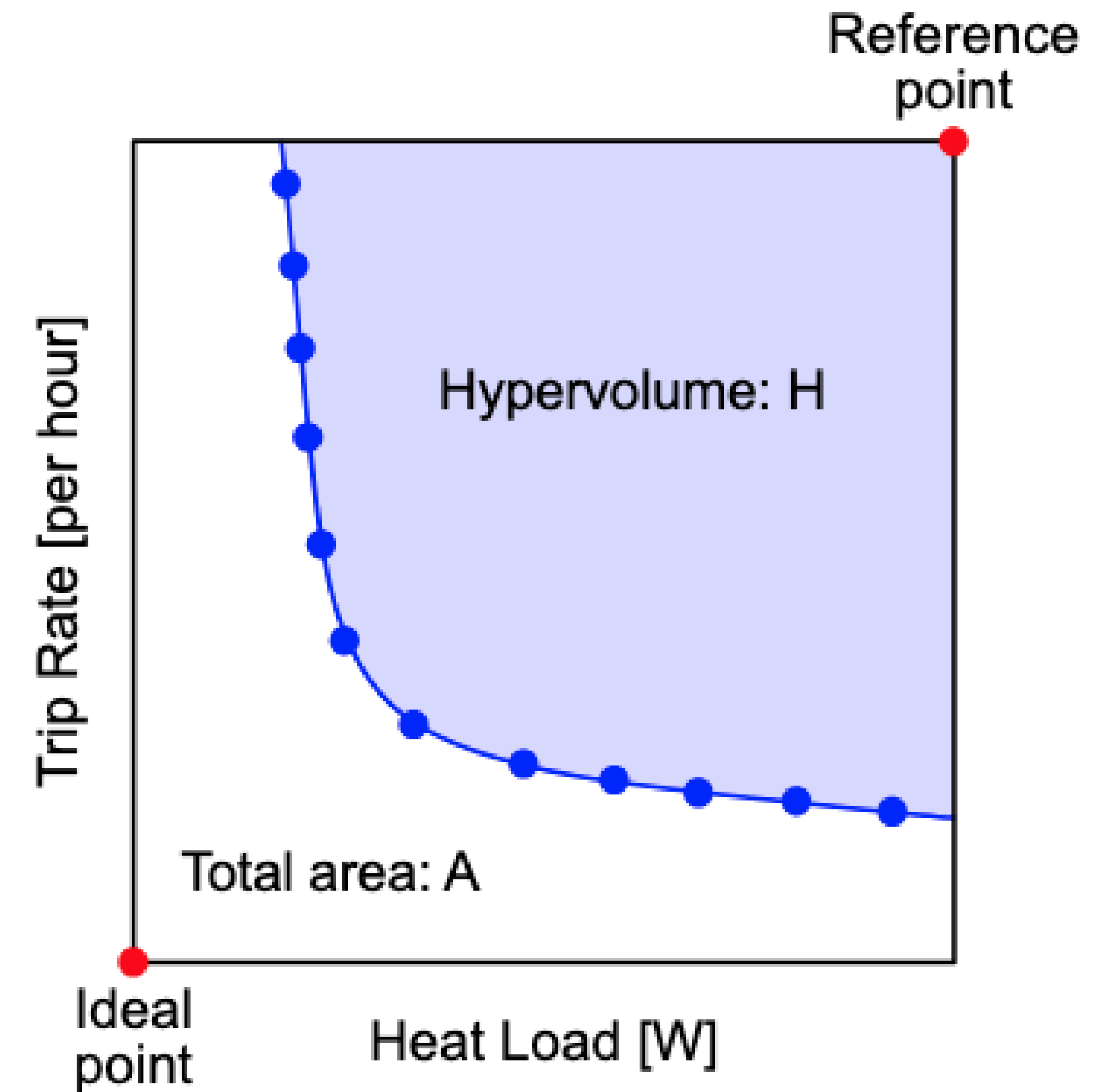
Energy surrogate helps critic capture North linac reward landscape



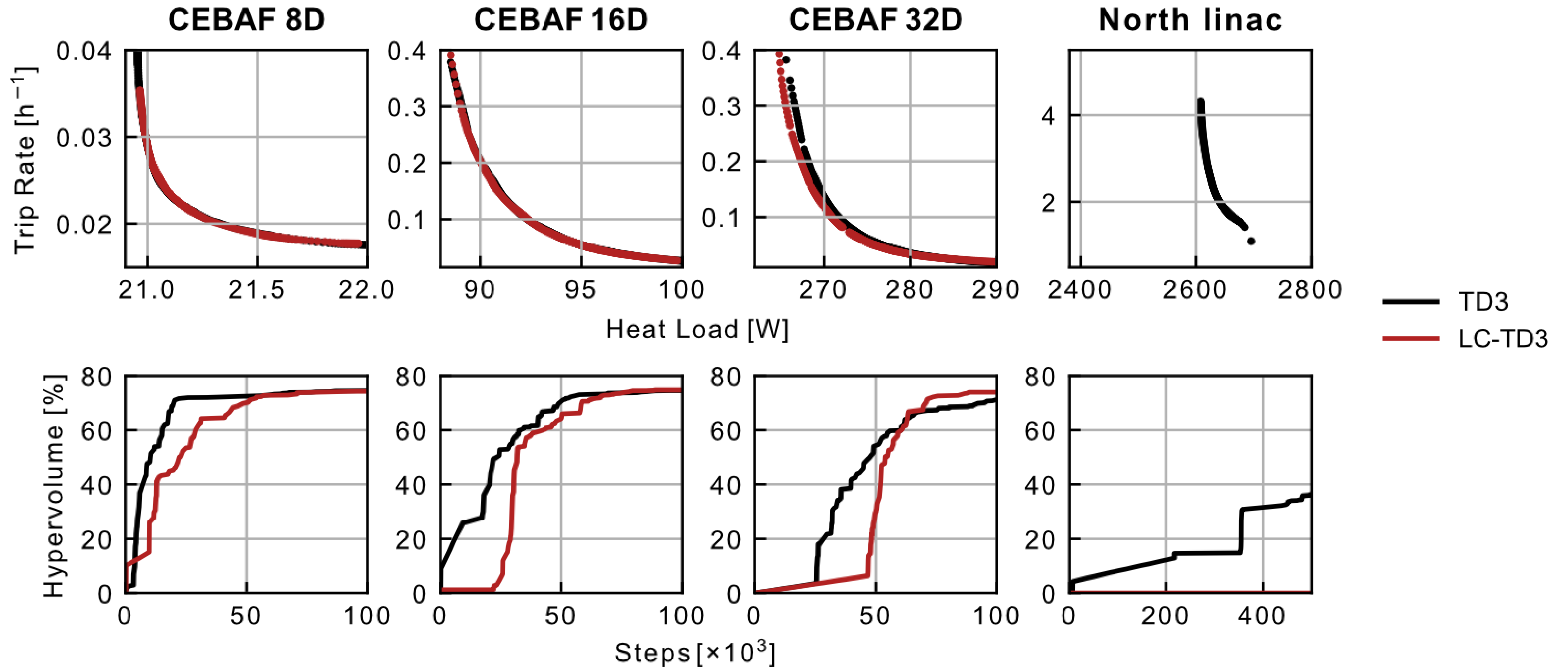
LC-TD3 accurately predicts reward near the bounds of the target energy range

Multi-objective CEBAF optimization

- Conditional policy $\pi(s; \alpha)$
- α defines the priority of heat load or trip rate minimization
- Sweeping $\alpha: 0 \rightarrow 1$ produces a Pareto front of optimal solutions
- Solution quality determined by **normalized hypervolume** metric



Multi-objective CEBAF optimization



LC-TD3 struggles with high-dimensional multi-objective optimization

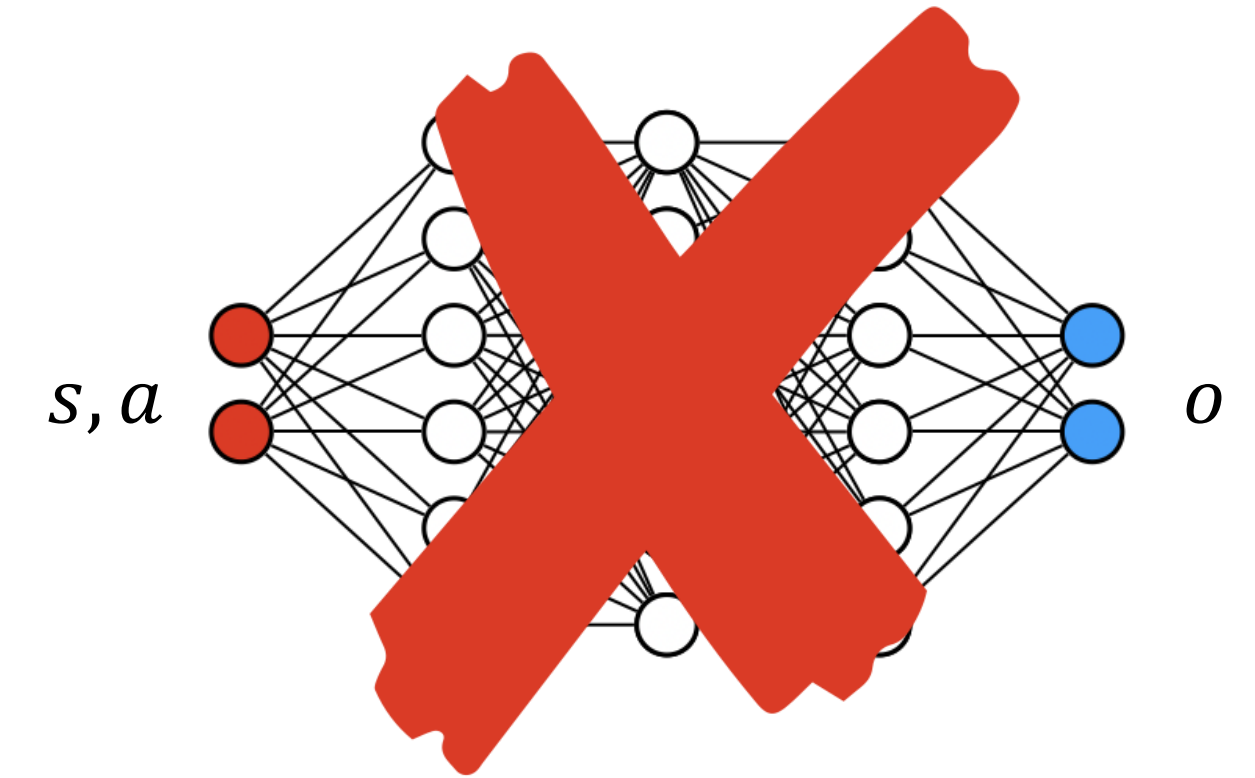
ArXiv:2502.20247

Interpretable surrogate model for physical observables

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 $1 \dots N_e$ **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

Surrogate
 $O_\xi(s, a)$

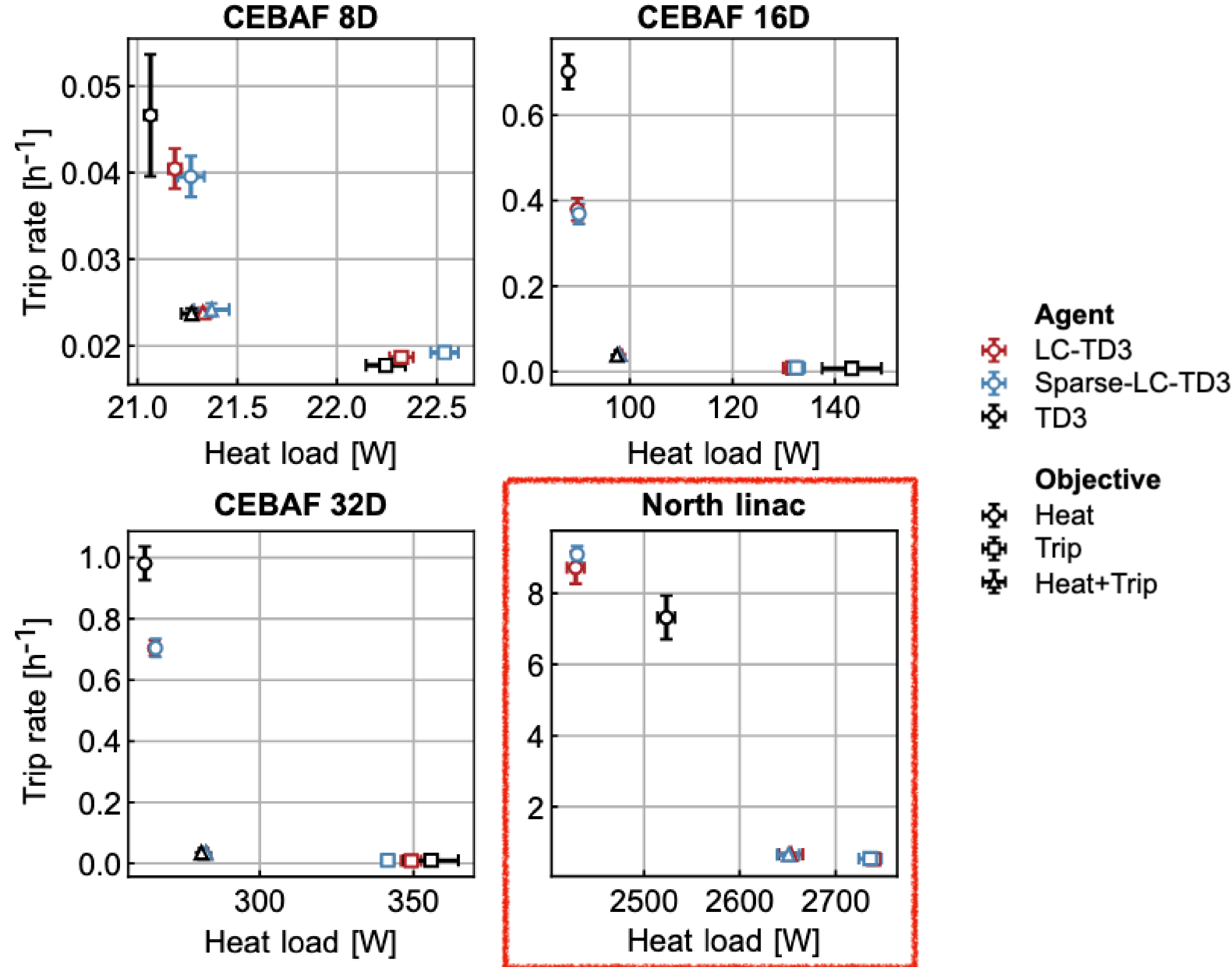


$$(s, a) \rightarrow L(s, a)\mathbf{w} = \begin{pmatrix} f_1(s_1, a_1) & f_2(s_1, a_1) & f_3(s_1, a_1) & \dots \\ f_1(s_2, a_2) & f_2(s_2, a_2) & f_3(s_2, a_3) & \dots \\ f_1(s_3, a_3) & f_2(s_3, a_3) & f_3(s_3, a_3) & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \end{pmatrix} = \begin{pmatrix} o_1 \\ o_2 \\ o_3 \\ \vdots \end{pmatrix}$$

Brunton, Proctor, and Kutz. PNAS (2016)

Sparse dictionary model builds surrogate **equation** for physical observables from a **library** of candidate functions

Sparse energy surrogate for CEBAF optimization

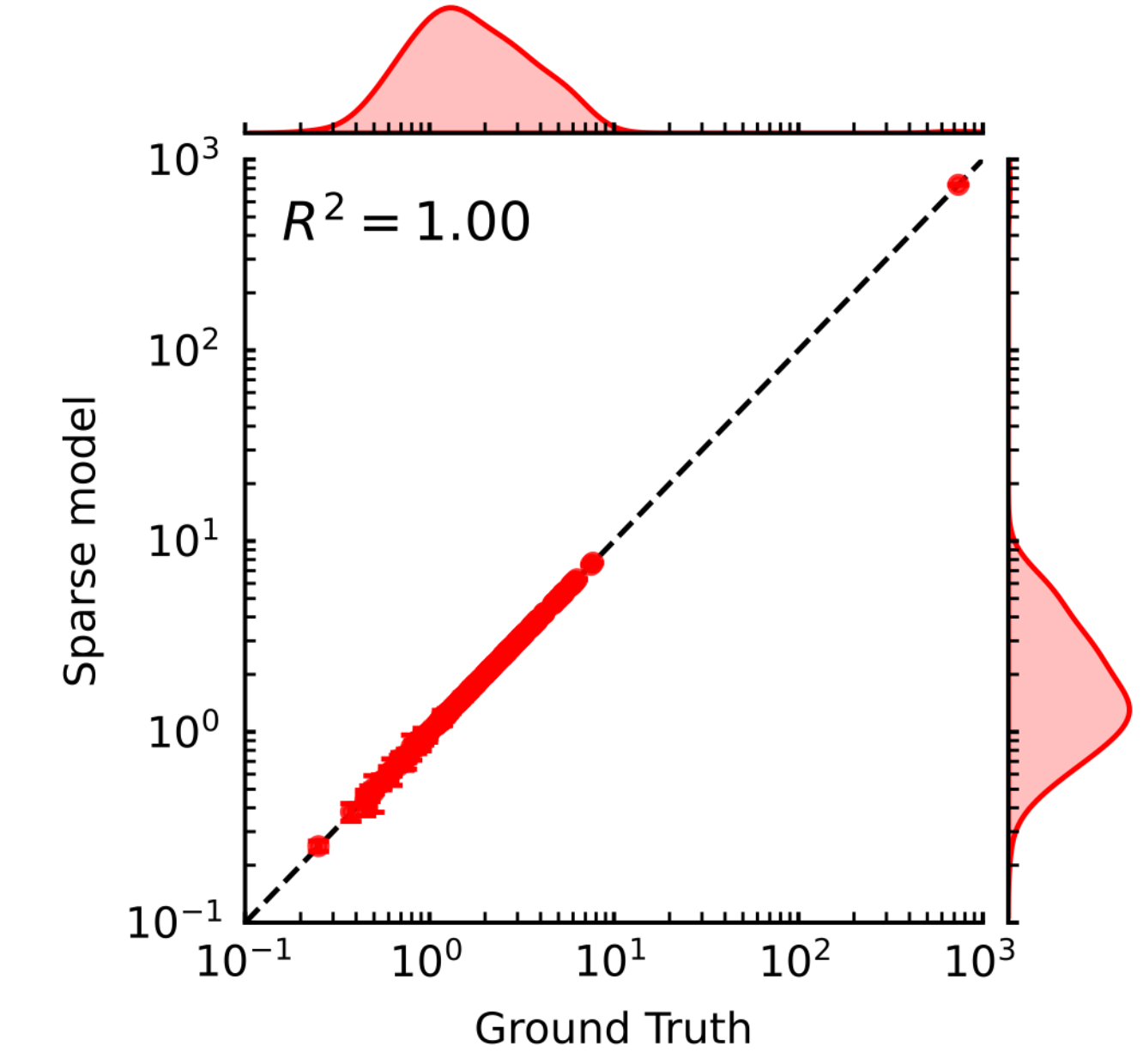
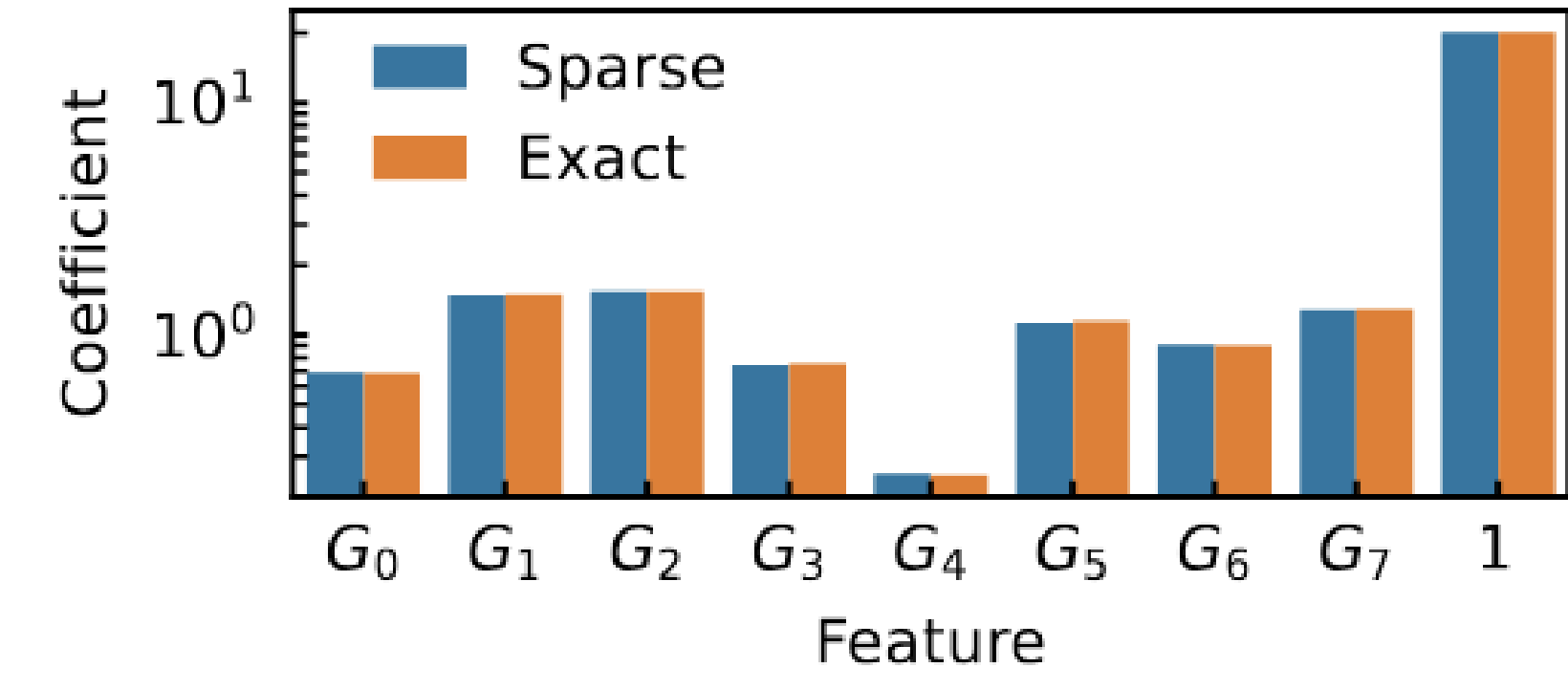
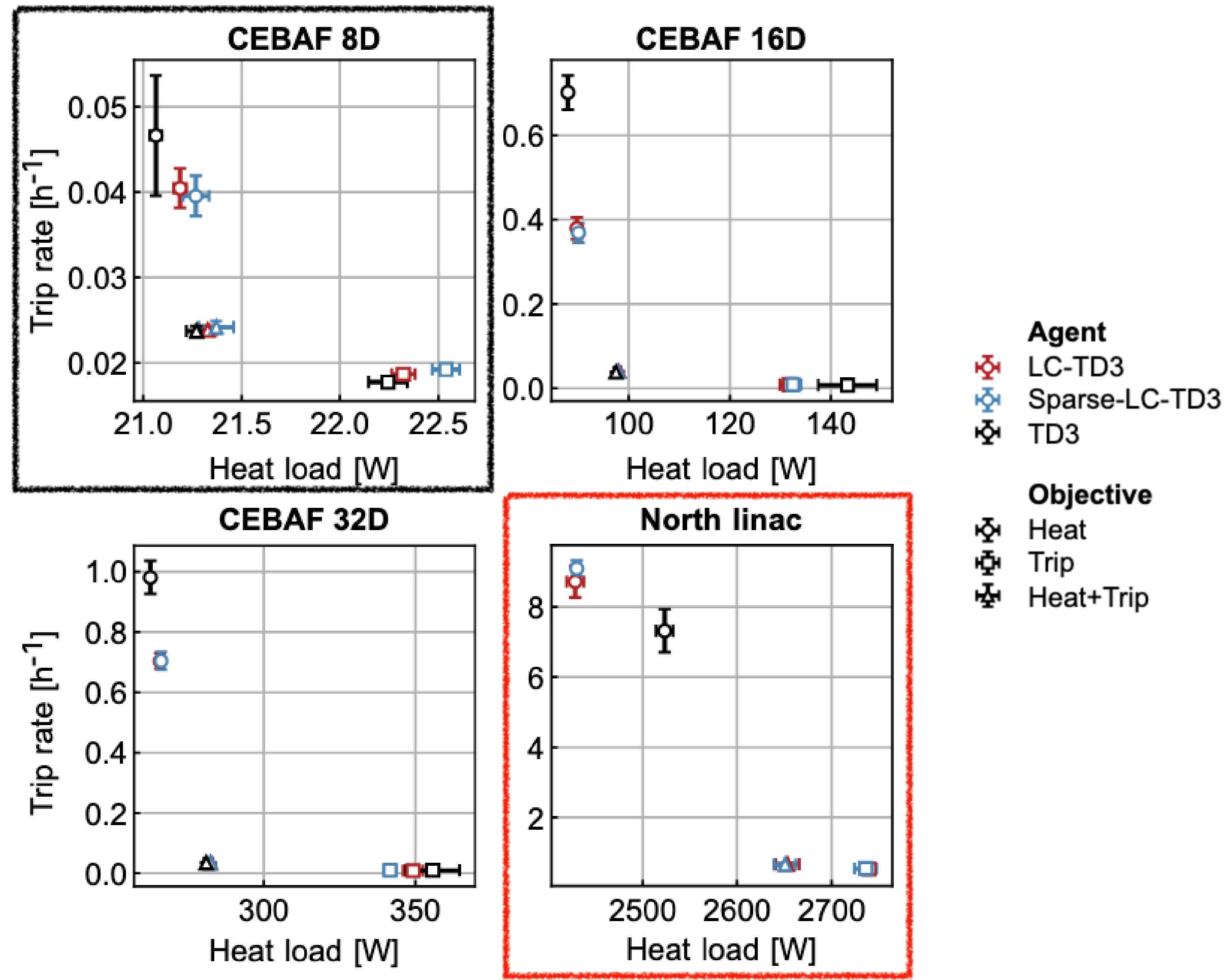


	Agent	Heat (W)	Trip (h^{-1})	Conv. Rate
Heat	LCTD3	2429 (9)	8.72 (0.45)	100 %
	Sparse-LCTD3	2429 (5)	9.11 (0.21)	88 %
	TD3	2559 (45)	7.08 (0.84)	50 %
Multi	LCTD3	2653 (12)	0.67 (0.03)	100 %
	Sparse-LCTD3	2650 (11)	0.67 (0.03)	88 %
	TD3	2753 (21)	1.55 (0.18)	0 %
Trip	LCTD3	2740 (6)	0.53 (0.03)	100 %
	Sparse-LCTD3	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.

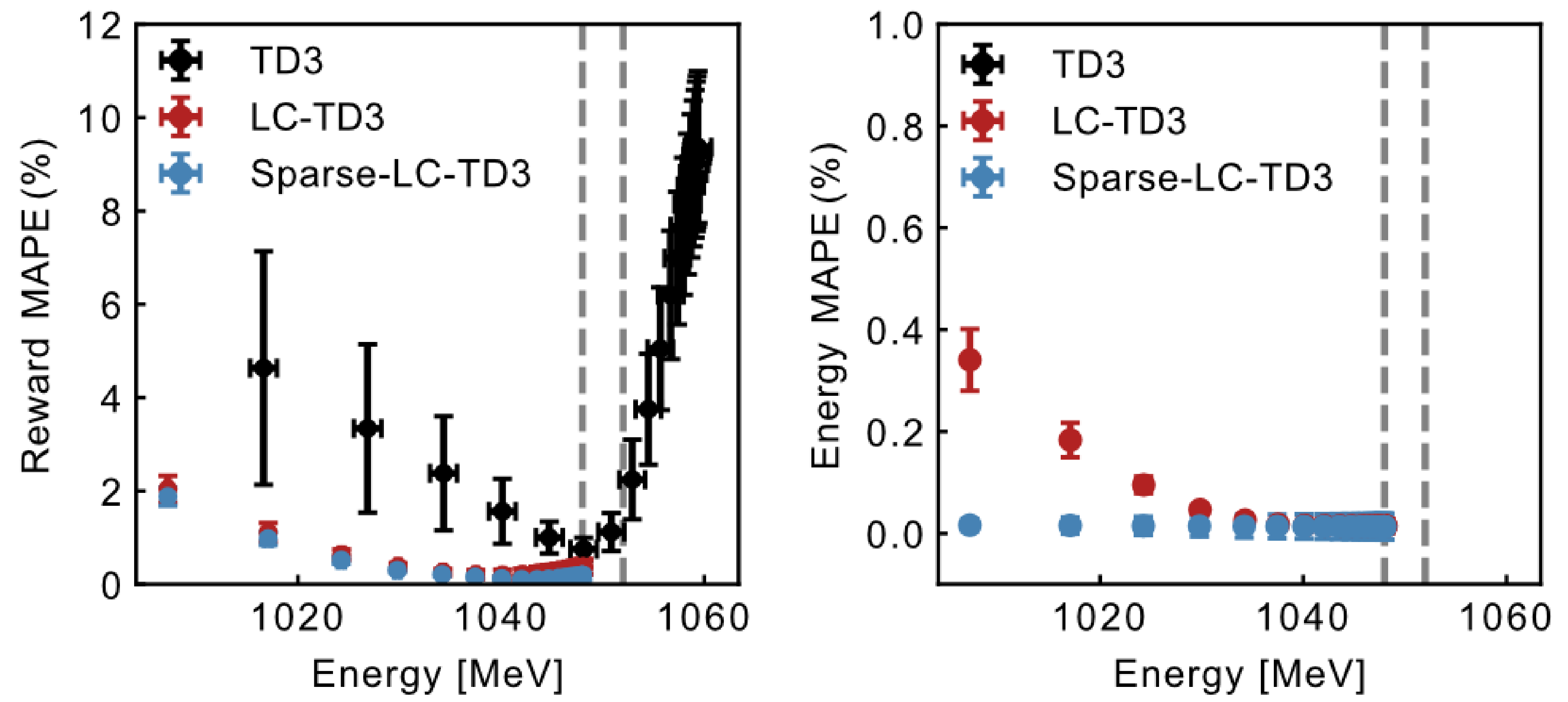
Agent maintains valid predictions on high-dimensional North linac problem

Sparse energy surrogate for CEBAF optimization



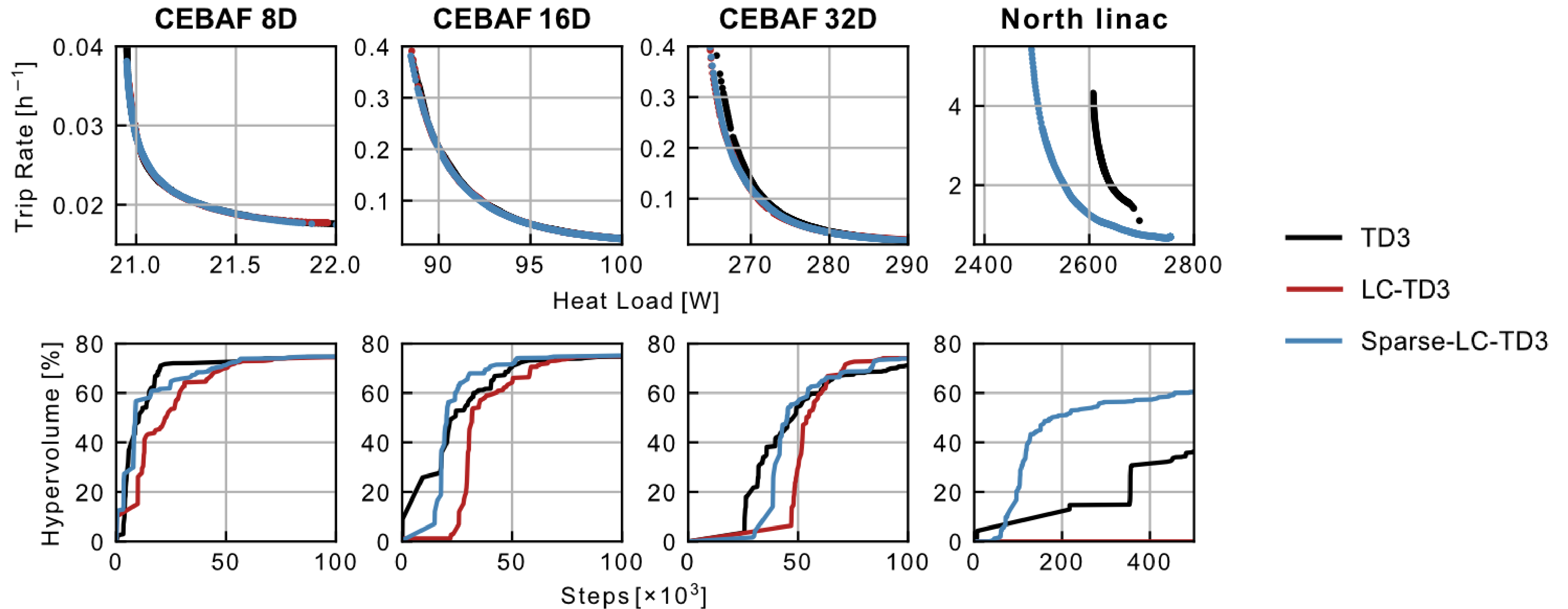
Surrogate equation is **interpretable** and allows *post hoc* verification

Sparse energy surrogate for CEBAF optimization



Sparse energy equation provides broad characterization of energy landscape

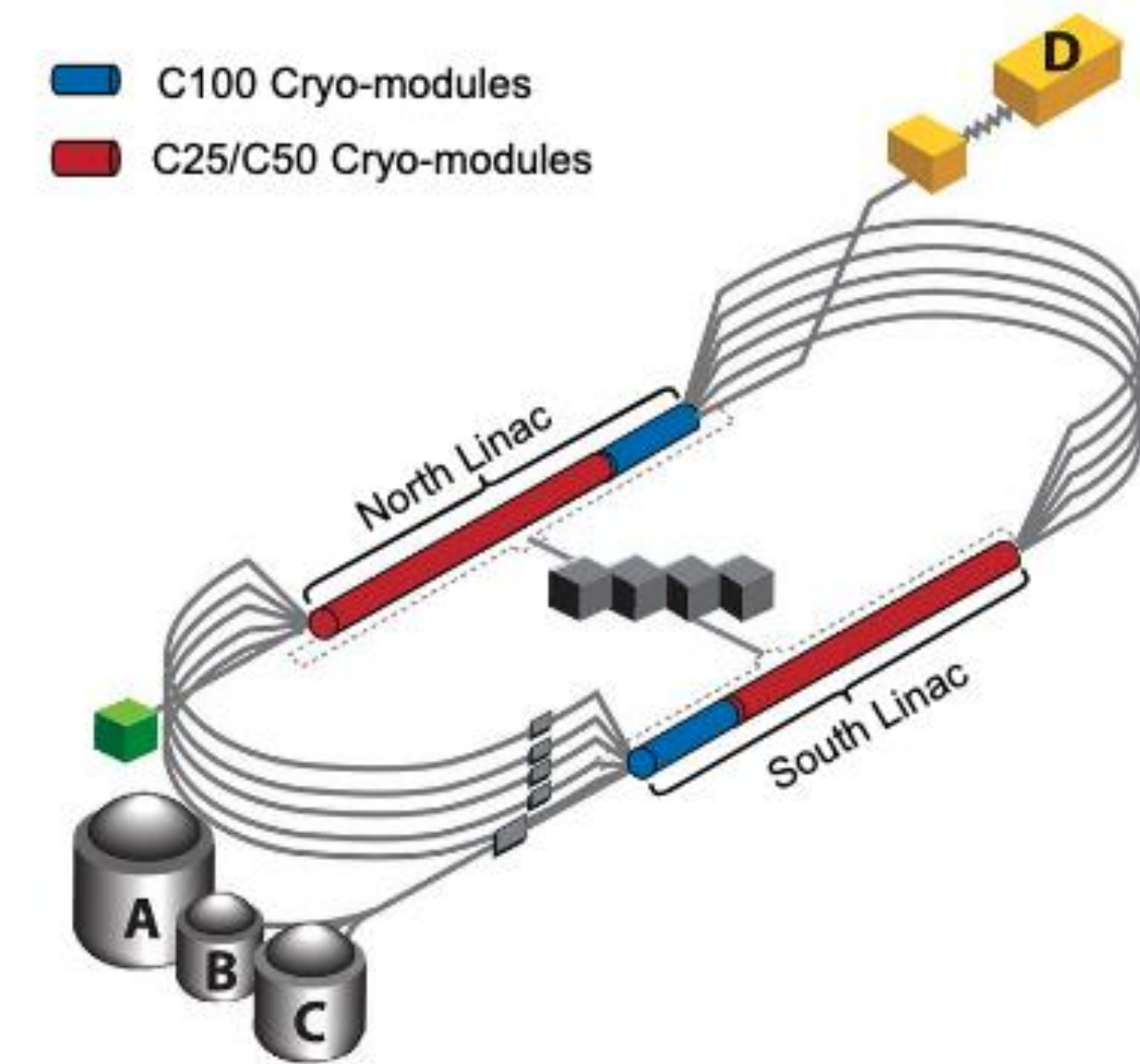
Multi-objective CEBAF optimization



Sparse LC-TD3 achieves superior performance on all problems

Conclusions

- Model-free RL struggles with high-dimensional accelerator optimization
- Learning a physics-based constraint surrogate enables gradient backpropagation
- Sparse dictionary equations accurately capture energy landscape for multi-objective problems
- Grey-box RL approach combines predictive power with operator interpretability



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.

Thank you!



Malachi Schram



Kishansingh Rajput



Armen Kasparian



U.S. DEPARTMENT
of ENERGY



HAMPTON ROADS
BIOMEDICAL RESEARCH
CONSORTIUMSM

Questions?