Deep Reinforcement Learning for Methane Slip Reduction in Hybrid-Powered Liquefied Natural Gas Marine Vessels

Ahmed Abdalla[a,*], Patrick Kirchen[b] & Bhushan Gopaluni[a]

[a] *Chemical & Biological Engineering Department, University of British Columbia, Vancouver, BC V6T 1Z3, Canada.*

[b] *Mechanical Engineering Department, University of British Columbia, Vancouver, BC V6T 1Z4, Canada.*

* *Corresponding author. email: ahmedoa@mail.ubc.ca*

Abstract: Hybrid-powered liquefied natural gas (LNG) vessels can reduce the greenhouse gas (GHG) emissions of marine transport through CO_2 reductions. These reductions can be offset by CH_4 emissions, generally highest at low engine loads due to methane slip. Efficient operation of LNG engines is therefore mandatory to achieve the required GHG emissions reduction. Using a hybrid LNG-battery powertrain provides additional degrees of freedom to modify the operation and mitigate GHG emissions; however, the optimal power allocation between the LNG engine and the battery system must be developed specific to the operation of the vessel. This paper studies the feasibility of using the twin delayed deep deterministic policy gradient (TD3), soft actor-critic (SAC), and proximal policy optimization (PPO) deep reinforcement learning (DRL) agents as part of the energy management system (EMS) of a hybrid-powered LNG vessel. The objective of the EMS is to reduce cumulative GHG emissions from sailing trips by optimally allocating power between the engine and battery of the vessel under study. The TD3 and SAC agents performed better than the PPO agent in terms of GHG emissions reduction and significantly better in terms of battery health maintenance. The PPO agent was excluded from further analysis due to its poor performance. The GHG reduction efficiency of both the TD3 and SAC agents, alongside the existing control strategy, was evaluated by normalizing emission reductions against the expected maximum reductions estimated using sequential least squares programming offline optimization. The reductions were calculated relative to a baseline of sailing trips without hybridization. The TD3 agent demonstrated improvements in the GHG reduction efficiencies by 19.80% on the training dataset and 18.64% on the test dataset compared to the existing control strategy, while the SAC agent achieved improvements of 11.46% and 7.61%, respectively. These results demonstrate that TD3 and SAC outperform the existing control strategy, with their generalized policies enabling online decision-making across various power demand profiles. The achieved reductions in GHG emissions were mainly driven by minimizing methane slip.

Key words: Real-time optimal control, Hybrid electric propulsion system, Reinforcement learning, Energy management, Methane slip reduction

1 Introduction

International shipping accounts for about 3% of global greenhouse gas (GHG) emissions which is equivalent to the aviation industry [1]. Shipping emissions are estimated to increase by 23% by 2035 compared to 2015 emissions if no additional policy measures are established, and could account for 10% of global GHG emissions by 2050[1,2]. The International Maritime Organization (IMO) has set targets in the 2023 IMO strategy to reduce shipping GHG emissions. The strategy outlines specific checkpoints to reach net-zero GHG emissions by 2050, which include an annual GHG emissions reduction of 20-30% and 70-80% by 2030 and 2040, respectively [3]. Both reduction checkpoints are relative to the estimated emissions from 2008.

There is a potential to reduce emissions from maritime transportation by 75-85% using currently available technologies such as hybridization, alternative fuels, and optimization of operations [4]. Such significant reduction efforts are required to ensure absolute reductions in GHG emissions from maritime transportation, as reduction efforts are offset by the fast growth of the maritime industry [4]. Hybridization enables the efficient use of multiple technologies such as batteries and combustion engines while lowering environmental impacts [4,5]. Alternative fuels such as methanol and ethanol, liquified natural gas (LNG) and hydrogen are considered potential green options for the maritime industry [6]. LNG is a near-term solution to reduce tank to wake NO_x, SO_x, PM, and CO_2 emissions, with renewable LNG providing a longer-term solution of well to wake GHG emission reductions. For both LNG and renewable LNG, unintended methane emissions can potentially lead to an increase in GHG emissions. Higher methane emissions at low engine loads, caused by methane slip due to factors such as incomplete combustion and low flame speeds, offset any carbon dioxide emission reductions achieved by using LNG as a fuel [4,7–9]. Effective operation of an LNG engine is therefore mandatory to ensure an adequate GHG emissions reduction (e.g., by minimizing low load operation). Operational measures such as energy management have the potential to reduce shipping carbon dioxide emissions by 1-10% [4]. Energy management systems (EMS) are considered important to achieve higher energy efficiencies for hybrid-powered vessels by optimally controlling the vessel's power sources based on the unique advantages of each power source [5,10,11].

EMS for hybrid-powered vehicles can be classified into three categories: rule-based, optimization-based, and learning-based [11]. Rule-based EMSs are control systems developed using operational guidelines or modes that are established using human expertise or mathematical models. The primary goal of optimization-based EMSs is to find the optimal control sequence of an energy management task by converting the task to a mathematical optimization problem and solving the problem subject to certain constraints. Learning-based methods are used to derive optimal control strategies by learning from available data with no requirement for an explicit model [11].

EMS for hybrid-powered LNG vessels have used rule-based and optimization-based approaches, though there are not yet any learning-based examples. Yanbiao Feng et al. [12] used a surrogate-based global optimization search algorithm to simultaneously optimize the power train component sizes and the propulsion control of an LNG-fueled hybrid powered electric ship. The performance was compared against reference diesel-mechanical and LNG-mechanical systems where the ship is mechanically propelled without an energy storage system. Ailong Fan et al. [13] used fuzzy logic control with particle swarm optimization to control the power distribution in a hybrid-powered diesel-LNG ship. The objective was to reduce fuel consumption and carbon dioxide emissions; however, methane slip was not considered. Similarly, in [14,15], rule-based approaches were used to develop EMSs for hybrid-powered LNG ships to reduce carbon dioxide emissions but, again, methane slip was not considered. Bo Pang et al. [16] used genetic algorithm global optimization to determine the optimal sizes of the LNG engines and battery energy storage system (BESS) to retrofit a ship's diesel-electric propulsion system. The objective of the optimization was to reduce the BESS and LNG engines investment costs, fuel costs, BESS degradation costs, and GHG emissions. The authors also propose an extended Kalman filter (EKF) based model predictive control (MPC) algorithm for real-time optimal power control of the ship's LNG engines and BESS. Existing EMSs for hybrid-powered LNG vessels mainly utilize rule-based control strategies, which are known for their poor dynamic response to changes, and optimization approaches that rely on highly accurate forecast models. There is a gap in research exploring control strategies that offer improved dynamic response without depending on forecast models. Learning-based methods address this gap by providing adaptive control without the need for forecast models.

Reinforcement Learning (RL), a subset of learning-based methods, is a model-free machine learning method that consists of two components: an agent and an environment. The agent interacts with the environment and learns an optimal control strategy that maximizes a reward signal received from the environment [11]. The use of RL methods to optimally control the EMS of hybrid-powered vessels has been studied, with the optimization objectives embedded within the RL agent's reward function. Peng Wu et al. [17] used a double-Q agent to control the power load distribution of a hybrid fuel cell and battery-powered ship by training the double-Q agent using historical sailing power profiles. The objective was to reduce the reduce the operational costs of the ship which included fuel and degradation costs. In Ref. [18,19], the work was extended to the continuous state space using a double deep Q-network and to the continuous state and action spaces using a twin delayed deep deterministic policy gradient (TD3). The three RL methods used were found to achieve near-optimal cost performance when compared to benchmark cost reductions achieved by deterministic dynamic programming (DDP). Chengya Shang et al. [20] proposed a deep Q network to control the power load distribution of a diesel hybrid-powered ship with the objective of reducing operational costs. The performance of the deep Q network was comparable to the benchmark cost reductions achieved by mixed integer quadratic programming. Wongwan Jung & Daejun Chang [21] also proposed a deep Q network to control the power load distribution of a liquid hydrogen hybrid-powered ship. The proposed RL method, however, led to an increase in operational costs when tested on power profiles not used for training the RL agent, while using dynamic programming outputs as a benchmark. The studies implemented have shown the feasibility of using RL to optimally control the power load distribution of hybrid-powered ships by embedding the optimization objective in the RL agent's reward function. Tiewei Song et al. [22] proposed a hybrid penalized proximal policy optimization (HP3O) based EMS that utilizes a continuous actor network to control a diesel-electric ship's generator power and cruising speed, and a discrete actor network that determines the activity status of the generators. The objective was to minimize the operational costs of the ship while maintaining operational constraints. The problem was formulated as a constrained Markov decision process to handle constraints related to power and velocity limits. The solution obtained using HP3O was comparable to the optimal solution obtained using mixed integer linear programming. These studies highlighted the viability of using RL to optimize power load distribution in hybrid-powered vessels by structuring the agent's reward function around cost reduction targets. However, the emphasis was largely on cutting operational costs, which may not directly lead to improved environmental efficiency [17].

2 Problem Statement

This paper explores the feasibility of using online deep RL (DRL) methods, namely TD3, soft actor-critic (SAC) and proximal policy optimization (PPO) to control the power load distribution of a hybrid-powered LNG vessel. The objective of the DRL agents is to minimize the cumulative GHG emissions generated during typical sailing trips while considering the effects of methane slip as well as CO_2 emissions. Due to its non-linear load dependence, methane slip can have an outsized impact on the total GHG emissions for a given vessel application. To the best of the authors' knowledge, the application of DRL to control the EMS of hybrid-powered LNG vessels to reduce total exhaust stack GHG emissions has not been explored in existing literature. DRL methods offer the advantage of implementing an online model-free control system that is able to learn and adapt based on the data provided. However, the performance of DRL methods is subject to the quality and quantity of training data. Real-world vessel operation data was available to train and test the DRL agents used in this study and was provided by an industry partner. In contrast, for online optimization-based methods such as MPC, where even though solutions close to global optima can be determined, the performance of the control system is quite sensitive to inaccuracies in the provided model and disturbances in the environment under control. Knowledge of future states of a given environment is also required to take a control action, whereas DRL methods only require knowledge of the current state of the environment [11]. This article is organized as follows. Section 3 describes the characteristics of the vessel under study. Section 4 defines the three DRL methods used in this study. Section 5 details the characteristics of the RL environment. The results and conclusions are presented in section 6 and section 7, respectively.

3 Candidate Vessel

The vessel considered in this work is a hybrid-powered, LNG, roll-on roll-off ferry that operates in the Canadian Salish Sea. The characteristics of the vessel are summarized in Table 1. Typical power demand profiles of the vessel's sailings are illustrated in Figure 1. The red line represents the average power demand, and the blue lines represent the 198 individual sailing trips used in this study. Each trip services the same ports and variations in the power demand are due to externalities such as marine traffic, tides and currents, vessel loading, weather, crew choices, ad hoc schedule changes, and/or auxiliary power requirements. The vessel is operated at low loads at the start and end of the trips and is operated at a high load for most of the duration of the trips. Figure 1 illustrates the high variability in the power demand profiles, both in terms of the power demand value and length of the trips. These variabilities demonstrate the need for methods which consider the instantaneous operation of the vessel, and not just nominal operation.

Table 1: Characteristics of the hybrid-powered LNG vessel under study.

Characteristic	Value
Vessel type	Roll-on roll-off ferry
Build year	2021
Installed engine power	2 × 4770 kW
Fueling	Low pressure dual fuel (LNG + pilot diesel) or diesel
Battery Capacity	2034 kWh
Maximum battery charge rate	2000 kW
Maximum battery discharge rate	800 kW

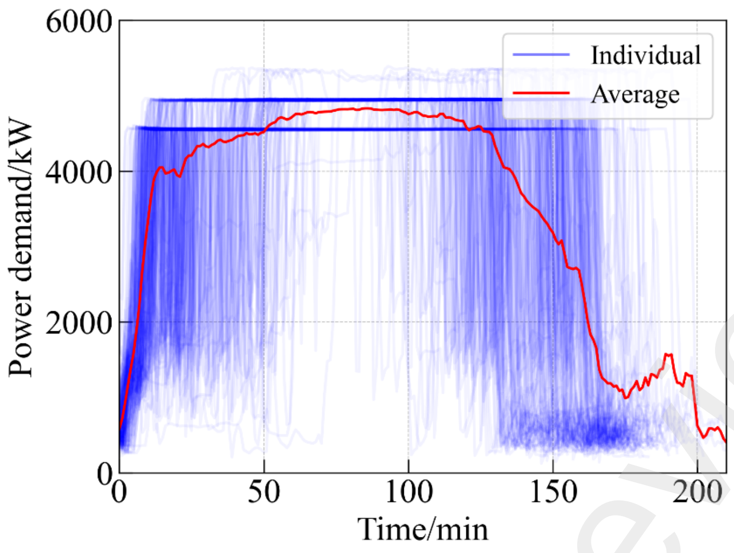


Figure 1: Typical power demand profiles of the hybrid-powered LNG vessel under study. The red line represents the average power demand whereas the blue lines represent the 198 individual sailing trips.

A schematic of the vessel's hybrid system and proposed EMS is depicted in Figure 2. The vessel's hybrid system is composed of two LNG-fueled engine generators, a battery and two electric propulsion motors. The vessel is generally operated using a single engine, as this operational mode has been determined to result in lower cumulative GHG emissions by increasing the average engine load [7,8]. The purpose of the EMS proposed in this paper is to control the delivered power between the engine and the battery while maintaining battery operational constraints. The control objective is to reduce the cumulative GHG emissions of sailing trips, including GHG contributions from both CO_2 and CH_4. The EMS provides the suggested engine power at the next time step, given inputs of the battery state of charge (SOC), engine power, and vessel power demand at the current time step.

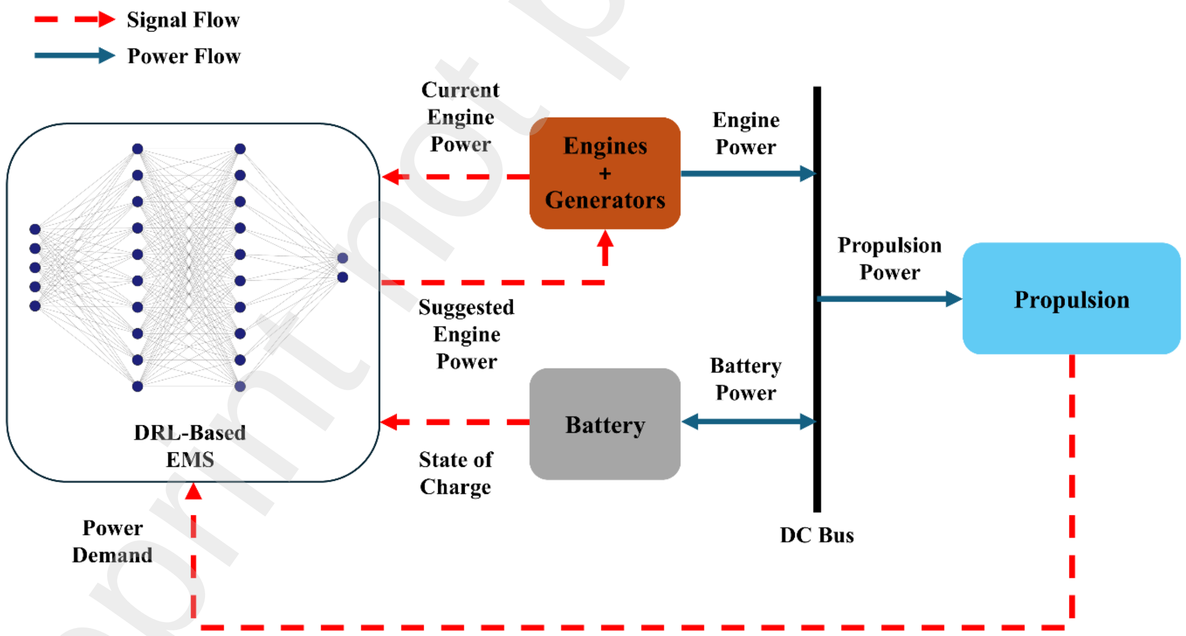


Figure 2: Schematic of the hybrid-powered setup and the energy management system. The powertrain consists of two LNG-fueled engines and associated generators, a battery, and two electric propulsion motors.

3.1 Emission Factors

The instantaneous CO_2 and CH_4 emissions from the engine were estimated as a function of engine load based on experimentally measured emission factors [8]. The emission factors were calculated using 1-minute averages of emission rates over fixed engine loads. Figure 3 depicts the GHG emission factors of the vessel in kilogram carbon dioxide equivalent per hour (kg $CO_{2,eq}$/hr) against the vessel's engine load. The $CO_{2,eq}$ emission rates include CO_2 and CH_4, with the latter considered using a 20-year global warming potential (GWP20) of 85.5 (midpoint of reported 84-87 [23]). The correlation between the emission rates in $CO_{2,eq}$ and engine load is non-linear. A linear correlation would have been observed if CO_2 emissions were solely considered. The total $CO_{2,eq}$ emission rate peaks at an engine load of 32% due to the combination of high CH_4 emissions (due to methane slip at low load), and increased CO_2 emissions relative to very low load operation. This peak emission load is the region that must be avoided to minimize total $CO_{2,eq}$ emissions for a given sailing. The minimum emission rate at operational loads between 20-100% is at an engine load of 74%. The non-linear nature of the total GHG emission rates from the hybrid-powered LNG ferry indicates that an increase in engine load is not necessarily associated with an increase in GHG emission rates. This phenomenon is the subject of exploitation in the proposed EMS.

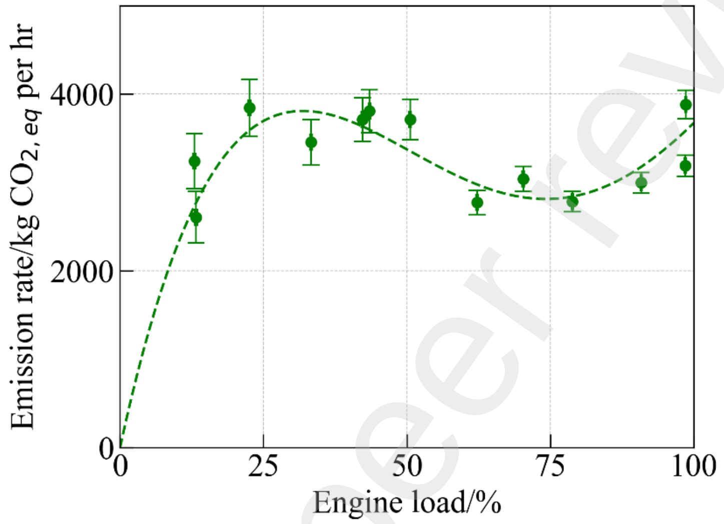


Figure 3: Measured GHG emission rates in kg carbon dioxide equivalent per hour versus the vessel's engine load. The emission rates peak at an engine load of 32%. The minimum emission rate at operational loads between 20-100% is at an engine load of 74%.

Equation (1) describes the instantaneous GHG emission rate as a function of engine load fitted to a fourth-degree polynomial.

$$g(L) = \sum_{i=0}^{4} \alpha_i L^i \#(1)$$

Where $g(L)$ is the instantaneous GHG emission rate in kg $CO_{2,eq}$/hr, α_i are the polynomial coefficients, and L is the engine load.

3.2 Battery State of Charge

A model of the vessel's battery is required to simulate the SOC dynamics so that the SOC can be considered in the RL environment (described in Section 4). A constant battery charge and discharge efficiency of 92% was used to simulate the SOC dynamics as per equation (2).

$$B_{soc,t+1} = B_{soc,t} + \frac{1}{Q_{cap}} \int_{t}^{t+1} \dot{Q} dt \#(2)$$

Where $B_{soc,t}$ and $B_{soc,t+1}$ are the battery SOC at time t and t+1, respectively, Q_{cap} is the battery capacity corrected for its state of health (SOH), and $\dot{Q}$ is the charge rate corrected for efficiency.

The efficiency value of 92% was taken as the mid-point value of the overall Li-ion battery efficiency (85-99%[24]). The battery SOH, obtained from the vessel's historical data, is approximately 82.75%. A total of 21 random sailing SOC profiles were used to assess the performance of the constant efficiency model. The average R^2 and mean squared error (MSE) of the model over the 21 SOC profiles were found to be 0.90 and 0.001, respectively. Figure 4 illustrates a sample plot that compares the actual and predicted SOC profiles using the constant efficiency model.

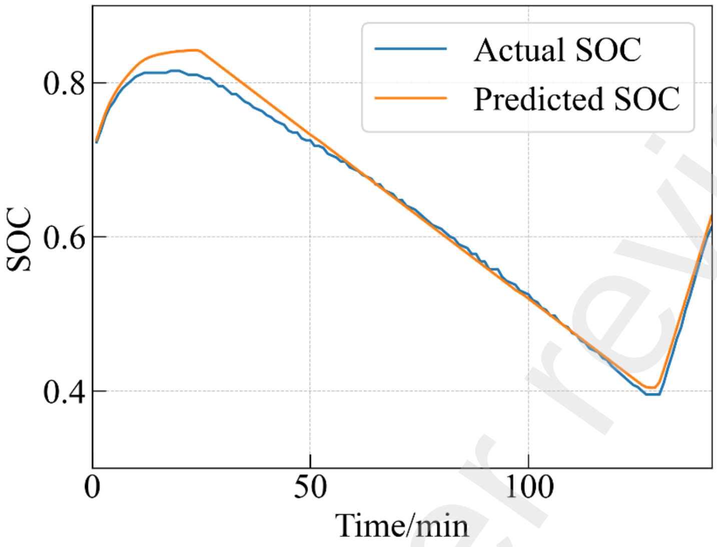


Figure 4: A sample SOC profile plot that compares the actual SOC to the predicted SOC of a sailing trip using the constant efficiency model. The R^2 and MSE for this plot are 0.99 and 0.0004, respectively.

4 Deep Reinforcement Learning

RL is a subfield of machine learning in which an agent learns how to map situations to actions in order to maximize a reward signal. The agent learns by interacting with an environment of a finite task (episodic) or an infinite task (continuous) through a trial-and-error approach [25]. At every time step t, the RL agent is provided with a state s_t from the environment. The agent then selects an action a_t based on a policy $\pi(a_t|s_t)$ and receives a reward r_t. The policy is a mapping from state s_t to an action a_t and can be either deterministic or stochastic. The environment then transitions to a new state s_{t+1} according to the dynamics of the environment or a model, with a reward function $\mathcal{R}(s,a)$ and state transition probability $\wp(s_{t+1} \mid s_t,a_t)$ [26]. At each state, the agent aims to maximize the expectation of the long-term return described in Equation (3), which is the sum of the discounted reward received by the agent over the future [25].

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \#(3)$$

Where G_t is the return received k time steps in the future discounted by a rate of $\gamma \in (0,1]$.

The state-value function defined in equation (4) is the expected return when starting in state s and thereafter following a policy π. The action-value function defined in Equation (5) is the expected return when starting in state s, taking action a, and then following policy π thereafter [25].

$$V_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid s_t = s] \#(4)$$

$$Q_\pi(s,a) \doteq \mathbb{E}_\pi[G_t \mid s_t = s, a_t = a] \#(5)$$

Where $V_\pi(s)$ is the state-value function, $Q_\pi(s,a)$ is the action-value function (Q-function) and $\mathbb{E}_\pi$ denotes the expectation under policy π.

The objective while training an RL agent is to find an optimal policy π_* that maximizes the total return received by the agent. The state-value function and action-value function associated with an optimal policy are called the optimal state-value and optimal action-value functions and are defined in Equations (6) and (7), respectively [25].

$$V_*(s) \doteq \max_{\pi} V_\pi(s) \#(6)$$

$$Q_*(s,a) \doteq \max_{\pi} Q_\pi(s,a) \#(7)$$

Where $V_*(s)$ is the optimal state-value function and $Q_*(s,a)$ is the optimal action-value function. Equations (6) and (7) hold for all states under an optimal policy.

DRL is an integration between deep learning and RL in which a deep neural network is used to estimate components of RL such as the value function, policy, state transition function, or reward function [26]. Deep neural networks have demonstrated their capability as universal function approximators, making them suitable for approximating value functions and policies in complex tasks with inputs of high dimensionality [27]. Actor-critic methods are a subclass of policy gradient methods that learn approximations to both the policy and value functions. The actor refers to the policy that is learned whereas the critic refers to the value function that is learned. The actor is used by the RL agent to take actions and the critic evaluates the actions taken [25]. The three RL methods utilized in this paper, namely, TD3, SAC, and PPO are online DRL actor-critic methods. The online nature means that the RL agents make decisions at each time step. Brief descriptions of the three DRL methods are outlined in this section.

4.1 Twin Delayed Deep Deterministic Policy Gradient

TD3 is an actor-critic method developed by Scott Fujimoto et al. [28] as an extension to the deep deterministic policy gradient (DDPG) algorithm [29]. TD3 addresses the Q-value overestimation bias and function approximation errors in actor-critic methods. TD3 is composed of one actor network, π_ϕ, and two critic networks, Q_{θ_1} and Q_{θ_2}, where ϕ, θ_1, and θ_2 represent the parameters of the neural networks. TD3 is also composed of one target actor network, $\pi_{\phi'}$, and two target critic networks, $Q_{\theta'_1}$ and $Q_{\theta'_2}$, where ϕ', θ'_1, and θ'_2 represent the parameters of the target networks [28].

The actor network is updated based on the gradient defined in Equation (8) as per the deterministic policy gradient algorithm [30].

$$\nabla J(\phi) = \mathbb{E}_{s\sim\rho_\pi}\left[\nabla_a Q_\pi(s,a)|_{a=\pi(s)} \nabla_\phi \pi_\phi(s)\right] \#(8)$$

Where $J(\phi)$ is the objective function and p_π is the state marginal distribution.

TD3 uses a single target update value y, as defined in Equation (9), that is calculated using the minimum of the two Q values estimated by the target critic networks. The use of the minimum Q value leads to a reduction in the overestimation bias [28]. The target update y is used in the loss function that is used to update the critic parameters θ_i.

$$y \leftarrow r + \gamma\, min_{i=1,2} Q_{\theta'_i}(s',\tilde{a}) \#(9)$$

The action $\tilde{a}$ is the action selected by the target actor network plus a small amount of random noise ε as defined in Equation (10) and s' is the next state. The added noise leads to the smoothing of the value estimate by bootstrapping off of similar state-action value estimates [28].

$$\tilde{a} \leftarrow \pi_{\phi'}(s') + \varepsilon,$$

$$\varepsilon \sim clip(\mathcal{N}(0,\sigma), -c, c) \#(11)$$

Where $\mathcal{N}(0,\sigma)$ is a gaussian distribution with mean 0 and standard deviation σ, and c is the noise clipping factor.

The update of the actor network in TD3 is delayed with respect to the critic networks. The delayed actor network update leads to the use of a value estimate with a lower variance that results in policy updates of higher quality [28].

4.2 Soft Actor-Critic

SAC is an RL algorithm developed by Tuomas Haarnoja et al. [31] that combines the advantages of policy optimization and entropy maximization through the use of a stochastic actor and a maximum entropy objective. SAC was found to exceed DDPG in both efficiency and final performance. RL with maximum entropy aims to optimize policies for the dual objective of maximizing both the expected return and the expected entropy of the policy. The maximum entropy objective defined in Equation (12) allows the policy to explore more and identify different modes of near-optimal behavior, and improves the learning speed of SAC compared to other state-of-the-art methods [31].

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t,a_t)\sim p_\pi}\left[r(s_t,a_t) + \alpha\mathcal{H}(\pi(\cdot\,|\,s_t))\right] \#(12)$$

Where $J(\pi)$ is the maximum entropy objective, $r(s_t,a_t)$ is the reward, $\mathcal{H}(\pi(\cdot\,|\,s_t))$ is the entropy term, and α is the temperature parameter that controls the importance of the entropy term against the reward.

SAC is composed of one actor network π_ϕ, two critic networks Q_{θ_1} and Q_{θ_2} and two target critic networks $Q_{\theta'_1}$ and $Q_{\theta'_2}$ [32]. The soft Q-function, defined in Equation (13), is used in the SAC algorithm instead of the traditional Q-function [33].

$$Q^*_{soft} = r_t + E_{(s_{t+1},\ldots)\sim p_\pi}\left[\sum_{l=1}^{\infty}\left(\gamma^l r_{t+l} + \alpha\mathcal{H}(\pi(\cdot\,|\,s_{t+l}))\right)\right] \#(13)$$

Where Q^*_{soft} is the soft Q-function.

4.3 Proximal Policy Optimization

PPO is an RL algorithm developed by John Schulman et al. [34] that has some benefits from the Trust Region Policy Optimization (TRPO) method but with much better empirical sample complexity and ease of implementation. TRPO maximizes a surrogate objective defined in Equation (14). Maximization of equation (14) leads to excessively large policy updates [34].

$$L^{CPI}(\theta) = \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t\,|\,s_t)}{\pi_{\theta_{old}}(a_t\,|\,s_t)}\hat{A}_t\right] = \hat{\mathbb{E}}\left[r_t(\theta)\hat{A}_t\right] \#(14)$$

Where $L^{CPI}(\theta)$ is the surrogate objective, CPI is conservative policy iteration, θ and θ_{old} are the new and old policy parameters, $r_t(\theta)$ is the probability ratio $\frac{\pi_\theta(a_t\,|\,s_t)}{\pi_{\theta_{old}}(a_t\,|\,s_t)}$ and $\hat{A}_t$ is the advantage function. The advantage function can be mathematically defined as the difference between the action-value and state-value. The advantage function describes how much better it is to take a certain action at a state, compared to randomly selecting an action according to the policy being followed [35].

PPO uses a modified surrogate objective, defined in Equation (15), that clips the probability ratio $r_t(\theta)$ to keep the ratio within a certain interval $[1-\epsilon, 1+\epsilon]$. Clipping the probability ratio prevents the policy updates from deviating too far from the previous policy [34].

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min\left(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right] \#(15)$$

Where L^{CLIP} is the clipped surrogate objective and ϵ is the clipping factor.

To further enhance stability and encourage exploration, PPO's objective function can be augmented with additional terms. These include the value function error term, which ensures better alignment between the estimated and actual returns, and the entropy bonus, which encourages exploration by promoting more diverse actions. The resulting objective function balances policy improvement, value estimation, and exploration [34], and is defined in Equation (15).

$$L^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t\left[L^{CLIP}(\theta) - c_1 L^{VF} + c_2 S[\pi_\theta](s_t)\right] \#(16)$$

Where $L^{CLIP+VF+S}(\theta)$ is the modified objective, L^{VF} is the value function error term, $S[\pi_\theta](s_t)$ is the entropy bonus term, and c_1 and c_2 are coefficients.

5 Deep Reinforcement Learning Energy Management System

TD3, SAC and PPO were evaluated as DRL control agents to suggest EMS strategies for measured instantaneous environments. The objective of the DRL agents is to select the optimal engine power for a given specific state of the vessel, as illustrated in Figure 2. It should be noted that selecting the engine power also defines the share of electrical power supplied to/from the battery, as the combination of these must satisfy the total vessel power demand. The selection of the optimal engine power by the DRL agents is done on the basis of minimizing cumulative sailing trip GHG emissions and adhering to the battery's operational constraints. The state of the vessel at a given time step is defined in the RL environment's state space, the action is defined in the action space, and the reward function is designed to incentivize the agent to take actions that minimize cumulative GHG emissions while adhering to the battery's operational constraints.

5.1 Data Preparation

Data signals that were collected during the vessel's sailing trips included travel time, engine power, battery power and battery SOC. The vessel's power demand, which is required to train the DRL agents, was calculated as the sum of the engine power and battery power as per Equation (17).

$$P_{tot,t} = P_{eng,t} + P_{bat,t}\#(17)$$

Where $P_{tot,t}$ is the total power demand, $P_{eng,t}$ is the engine power, and $P_{bat,t}$ is the battery power at time t.

The data was collected at a 1-hertz frequency for a total of 198 sailing trips. Each data signal was converted into a 1-minute average to reduce noise and match the time resolution of the estimated GHG emission factors. A time step of one minute was therefore also used in the RL environment. The power demand and engine power were normalized by their respective maximum values. The data was divided into a training dataset, composed of 178 sailing trips, and a test dataset of 20 sailing trips. The test dataset was used to test the performance of the DRL agents on sailing trips the agents had not been trained on. The training dataset was then augmented to create more training examples by randomly sampling a set of 2 sailing trips iteratively without repetition and averaging their profiles. A total of 1000 augmented sailing trip profiles were generated adding up to a total of 1178 training examples. In the context of this paper, one sailing profile represents one finite episode in the RL environment. Sailing trip profiles from the training dataset were randomly sampled during each DRL agent training iteration.

All code used in this study was written in Python. The *stable_baselines3* library was used to set up the TD3, SAC, and PPO agents. The RL environment was set up using OpenAI's *gymnasium* library. The *Optuna* library was used for hyperparameter tuning.

5.2 State Space

The state space used in the RL environment is a 4-dimensional continuous array as defined in Equation (18). The state space represents the input to the DRL agents during training and testing.

$$\boldsymbol{S_t} = [t_{norm}, P_{dem,t}, L_t, B_{soc,t}]^T\#(18)$$

Where $\boldsymbol{S_t}$ is the state space array at time t, t_{norm} is the normalized time with respect to the maximum trip duration, $P_{dem,t}$ is the normalized power demand, L_t is the engine load (normalized engine power), and $B_{soc,t}$ is the battery SOC at time t. The maximum trip duration is known during the training of the RL agents. In terms of real-life applications, the maximum trip duration would be the estimated time of arrival of the vessel in minutes. The state space defined in (18) describes all the necessary information at each time step that is relevant for the agent to select an optimal action.

During the training of the DRL agents, the initial states of the episodes were set as the initial states of the actual sailing trips, except for the battery SOC, which was randomized between 0.75-0.85 to increase the exploration of the agents. Subsequent time and power demand values were taken from the actual sailing trip profiles as the DRL agents were required to learn policies based on the historical power demand profiles. Subsequent engine load values were dependent on the actions taken by the agent and subsequent SOC values were calculated using Equation (2).

5.3 Action Space

The action space used in the RL environment is a 1-dimensional continuous array as defined in Equation (19). The action space represents the action taken by the agent given a certain state.

$$\boldsymbol{A_t} = [L_{t+1}]\#(19)$$

Where $\boldsymbol{A_t}$ is the action space array at time t and L_{t+1} is the engine load at time $t + 1$. The agent selects the engine load at the next time step. The battery power at time t+1 is calculated using Equation (17). The action selected by the agent dictates whether the battery is operated in charge or discharge mode.

5.4 Reward Function

The reward function was designed to support the agent in learning a policy that minimizes cumulative GHG emissions while working within the battery's operational constraints. The reward function consists of multiple components, each rewarding the agent for a specific behavior.

The first part of the reward function consisted of a reward, with values between 0-1, that incentivizes state transitions that lower cumulative GHG emissions between successive time steps. The first part of the reward function is defined in Equation (20).

$$r_{ghg,t+1} = \exp\left(\frac{-\int_t^{t+1} g(L)\,dt}{\beta_1}\right)\#(20)$$

Where $r_{ghg,t+1}$ is the time step GHG reward, $\int_t^{t+1} g(L)dt$ is the amount of GHG emissions released within one time step, and β_1 is a factor used to scale the reward.

The second part of the reward function consisted of battery constraint violation penalties. A reward of -1 was provided to the agent if any of the constraints were violated. The constraints included the maximum battery charge and discharge rates, and the SOC levels required to maintain battery health (SOC = [0.30,0.85]). The episode was truncated early if the battery was over or undercharged (i.e., if SOC $\notin$ [0,1]) to enforce a hard constraint. The rewards placed on constraint violations are defined in Equations (21)-(23). Negative power values in the range defined in Equation (21) represent battery charging.

$$r_{P_{bat,t+1}} = \begin{cases} -1,\ if\ P_{bat,t+1} \notin [-2000, 800]\ kW \\ 0, \qquad\qquad\qquad\qquad\qquad else. \end{cases}\#(21)$$

$$r_{B_{SOC,t+1}} = \begin{cases} -1,\ if\ B_{SOC,t+1} \notin [0.30, 0.85] \\ 0, \qquad\qquad\qquad\qquad else. \end{cases}\#(22)$$

$$r_{B_{chrg,t+1}} = \begin{cases} -1\ and\ truncate,\ if\ B_{SOC,t+1} \notin [0.00, 1.00] \\ 0, \qquad\qquad\qquad\qquad\qquad else. \end{cases}\#(23)$$

Where $r_{P_{bat,t+1}}$ is the battery power penalty, $r_{B_{SOC,t+1}}$ is the battery SOC penalty, and $r_{B_{chrg,t+1}}$ is the overcharge/undercharge penalty.

The final part of the reward function consisted of a terminal (final) state battery charging penalty defined in Equation (24). The terminal reward function penalized the agent based on its deviation from the target final SOC (0.85). If the SOC is not at 0.85 by the end of the episode, the battery is charged at a rate of 2000 kW (41.9% engine load), and the associated GHG emissions are calculated accordingly.

$$r_{term} = clip\left(\beta_2 - \frac{g(41.9\%)\Delta t_{chrg}}{\beta_3}, -\beta_2,\ +\beta_2\right)\#(24)$$

Where r_{term} is the terminal reward, $g(41.9\%)$ is the GHG emission rate at an engine load of 2000 kW, Δt_{chrg} is the battery charging time, β_2 is a factor used to set the limits for the reward/penalty, and β_3 is a factor used to scale the terminal reward.

The selection of β_2 should ensure that the terminal reward does not overshadow time step rewards to a great extent. The maximum terminal reward of $+\beta_2$ is achieved if the final SOC is 0.85. Final SOC levels higher than 0.85 do not result in a higher reward so the agent does not receive an incentive to charge the battery above a final SOC of 0.85.

The total reward for each state transition is defined in Equation (25).

$$R_{t+1} = \begin{cases} r_{ghg,t+1} + r_{P_{bat,t+1}} + r_{B_{SOC,t+1}} + r_{B_{chrg,t+1}},\ t_{norm} < 1 \\ r_{ghg,t+1} + r_{P_{bat,t+1}} + r_{B_{SOC,t+1}} + r_{B_{chrg,\ t+1}} + r_{term},\ t_{norm} = 1 \end{cases}\#(25)$$

Where R_{t+1} is the total reward at the end of each time step.

5.5 Hyperparameter Tuning

The hyperparameters of the DRL agents were tuned using the *Optuna* library with a total run time of 8 hours or 100 trials. The objective was the maximization of the final total reward received by the agent after 100,000 time steps. The tree-structured parzen estimator was used as the parameter sampler during the hyperparameter tuning process. The TD3, SAC, and PPO agents were tested using both the Huber loss function and the mean squared error (MSE) loss function for critic loss calculation. TD3 performed best with the Huber loss function, while SAC and PPO showed better performance with MSE. The final hyperparameters of the TD3, SAC, and PPO agents are summarized in Table 2, Table 3, and Table 4, respectively. Normal action noise with linear decay was used during the training of the TD3 agent, and the parameters of the noise were also subjected to hyperparameter tuning.

Table 2: Hyperparameters of the TD3 agent tuned using the Optuna library.

Hyperparameter	Value
Discount factor, γ	0.99
Learning rate, α	9.94×10^{-4}
Batch size	128
Buffer size	100,000
Polyak coefficient, τ	0.02
Training frequency	1 step
Action noise initial standard deviation, σ_{init}	0.37
Action noise final standard deviation, σ_{fin}	0.07
Noise decay steps	10,000
Actor network size	[500, 400]
Critic network size	[500, 400]

Table 3: Hyperparameters of the SAC agent tuned using the Optuna library.

Hyperparameter	Value
Discount factor, γ	1
Learning rate, α	9.98×10^{-3}
Batch size	1024
Buffer size	100,000
Polyak coefficient, τ	0.005
Training frequency	4 steps
Actor network size	[400, 300]
Critic network size	[400. 300]

Table 4: Hyperparameters of the PPO agent tuned using the Optuna library.

Hyperparameter	Value
Discount factor, γ	0.9
Learning rate, α	3.64×10^{-4}
Batch size	32
Number of update steps	512
Entropy coefficient	1.07×10^{-3}
Clip range, ϵ	0.3
Number of epochs	20
Generalized advantage estimator trade-off, λ	0.99
Max gradient clipping value	1
Value function coefficient	0.75
Actor network size	[64, 64]
Critic network size	[64, 64]

6 DRL Agents Training Performance Assessment

Each DRL agent was trained for a total of 100,000 timesteps using five different random seeds, and learning curves were generated to assess the learning performance of the agents. The learning curves of the DRL agents are illustrated in Figure 5. The learning curves indicate the performance of the DRL agents as training progresses in terms of the mean episode reward and the mean episode length, which are both expected to increase as training progresses until convergence. An increasing mean episode reward as training progresses indicates that the DRL agents are learning policies that maximize GHG reductions and minimize operational constraints. An increasing mean episode length indicates that the DRL agents are learning policies that minimize episode truncations as per Equation 23. The mean episode length is expected to increase up to a value of 169, which is the average duration of a complete sailing trip. The shaded regions represent the standard error over the values obtained from the five different training trials. The performances of TD3 and SAC were recorded against training episodes, whereas the performance of PPO was recorded against training time steps due to the characteristics of the *stable_baselines3* logger.

Figure 5 (a) and (b) show that the TD3 agent learns a policy that accumulates a greater mean episode reward (~55) compared to the SAC (~42) and PPO (~24) agents. Over the five training iterations, the TD3 and SAC agents consistently learned policies that complied with the vessel's operational constraints. However, for four of the five iterations, the PPO agent learned policies that violated the vessel's SOC constraint (i.e., SOC $\notin [0,1]$). Figure 5 (b) shows that the PPO agent failed to learn a stable policy that converges to a maximal reward. Figure 5 (c) illustrates that the SAC agent learns a policy that minimizes episode truncations during training faster than the TD3 agent. Figure 5 (d) indicates that the PPO agent learns a policy that minimizes episode truncations; however, adherence to constraints is inconsistent across episodes. The minimal shaded regions in the SAC learning curves are due to the agent's use of a stochastic policy with entropy maximization which is known to stabilize training [31].

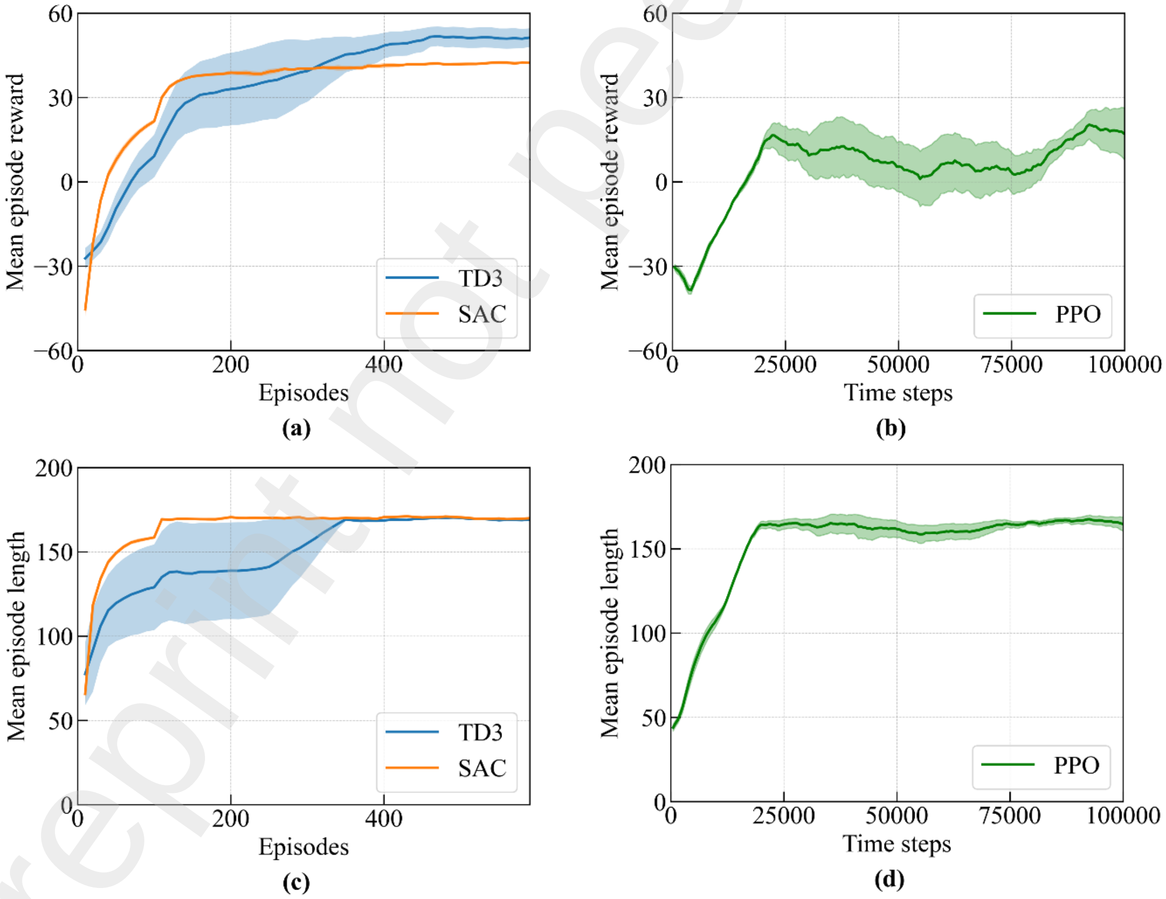


Figure 5: Learning curves of the TD3 and SAC agents after 100,000 time steps of training. (a) and (b) represent the mean episode reward calculated as training progresses, while (c) and (d) represent the mean episode length.

6.1 GHG Reduction and Battery Health Performance of the DRL Agents

The total GHG emissions, which include emissions during sailing plus any emissions to recharge the battery to a final SOC level of 0.85, were calculated for the engine profiles generated by the DRL agents for measured vessel power demands. The GHG emissions reductions achieved by the DRL agents for the hybrid power system (natural gas engine + battery) were compared to a baseline in which the same sailing trips were simulated with no battery use (i.e., no hybridization). Equation 26 was used to calculate the GHG reduction percentages for each sailing trip. The GHG reductions relative to the baseline were used to assess the performance of the DRL agents. The assessment was performed using the 178 training dataset examples and the 20 test dataset examples. For this application phase, SAC and PPO were configured to select actions deterministically, while they were kept in their original stochastic form during training. TD3 employs a deterministic policy which was used in both the training and application phases.

$$E_{red\%} = \frac{E_{NH} - E_{RL}}{E_{NH}} \times 100\%\#(27)$$

Where $E_{red\%}$ is the emissions reduction %, E_{NH} is the amount of GHG emissions associated with a sailing trip simulated with no hybridization in kg $CO_{2,eq}$, and E_{RL} is the amount of GHG emissions associated with a sailing trip simulated with a DRL control agent in kg $CO_{2,eq}$.

Table 5 summarizes the performance of the DRL agents in terms of GHG reductions. The results show that TD3 is the best performing DRL agent in terms of GHG reduction performance followed by SAC and then PPO. The comparable results between the training and test datasets show that the learned policies can be effectively applied to new power demand profiles that the agents have not been trained on.

Table 5: GHG reduction performance of the TD3, SAC, and PPO agents averaged over 178 training examples and 20 test examples.

Algorithm	GHG Reduction (%)	
	Training Examples	Test Examples
TD3	14.62	13.42
SAC	12.19	11.42
PPO	10.79	9.41

The DRL agents were also evaluated based on their ability to maintain the vessel's battery health by operating within the SOC limits of 0.30 to 0.85. Sailings controlled by the TD3 and SAC agents agent were outside of these limits for less than one minute per trip for the training and test datasets, respectively. In contrast, the PPO agent-controlled trips average just above 21 minutes per trip outside these limits for the training and test datasets, respectively. These results demonstrate that the TD3 and SAC agents effectively manage the vessel's power distribution while maintaining battery health, whereas the PPO agent shows poor performance in this regard. The times of operation outside the battery health SOC limits are summarized in Table 6.

Table 6: Average time of operation outside the battery health SOC limits for the sailings controlled by the TD3, SAC, and PPO agents.

Algorithm	Time of Operation (mins)	
	Training Examples	Test Examples
TD3	0.69	1.00
SAC	0.80	0.70
PPO	21.61	23.35

Figure 6 illustrates an example of the power load distribution for a sample sailing trip from the test dataset for the TD3 and SAC agents. PPO's results are excluded due to its inferior GHG reduction performance and battery operational constraint compliance. The power load distribution set by the DRL agents in simulation was compared to the actual power load distribution (i.e., not controlled by DRL) that occurred during the course of the sailing trips. Figure 6 (a) and (c) show the TD3 agent's power load distribution and SOC profiles compared to actual operational scenarios, while Figure 6 (b) and (d) show the same comparisons for the SAC agent. The sailing trip (i.e., power demand) considered in Figure 6 is identical for both the TD3 and SAC agents. For the trip duration from minutes 20 to 140, the engine load and battery power profiles selected by the DRL agents closely resemble the actual operational scenarios. The major differences are observed at the beginning (minutes 0-20) and the end (minutes 140+) of the sailing trip. This behavior is consistent across most sailing trips in both the training and test datasets. To assess the generality of this behavior, the absolute deviation of the engine load selected by the DRL agents from the actual engine loads was analyzed. Figure 7 shows the engine load deviations for both the TD3 and SAC agents across all sailing trips in the training and test datasets. As illustrated in Figure 7, the most significant variations occur at the beginning and end of the sailing trips.

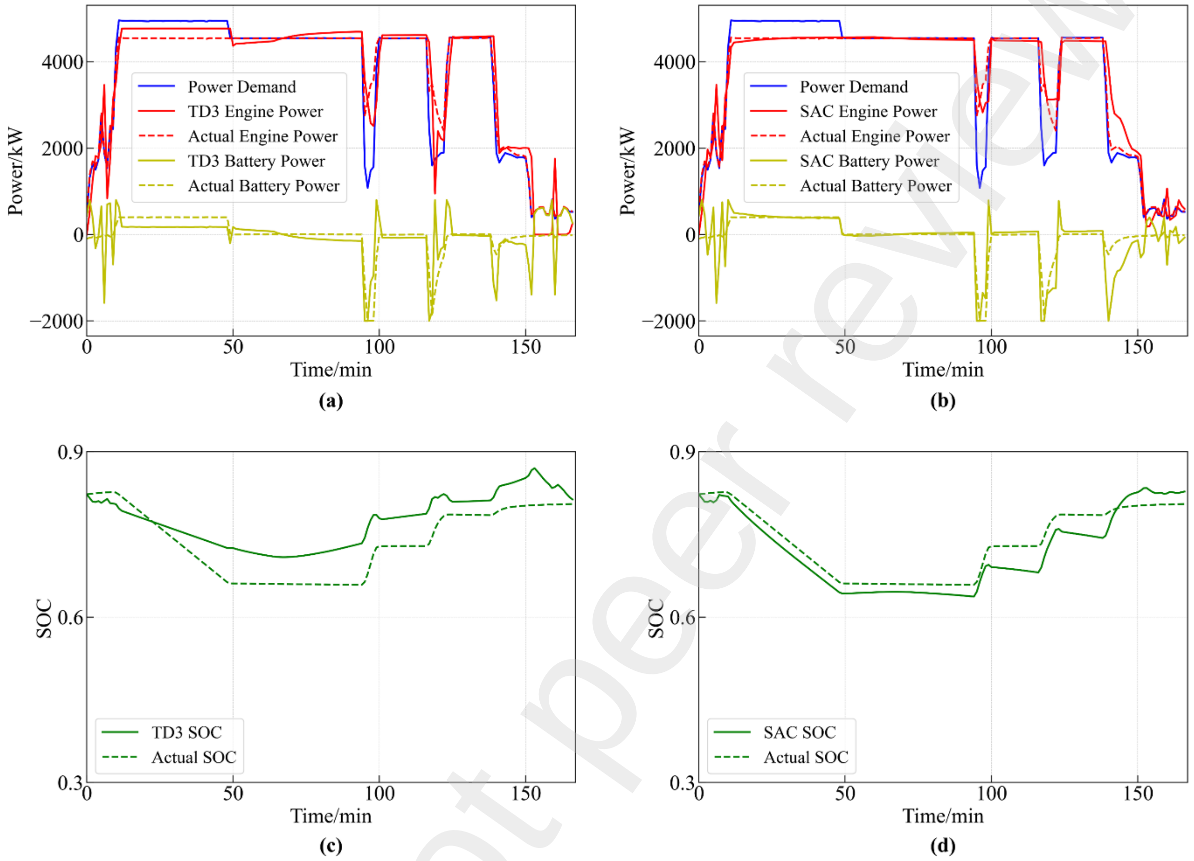


Figure 6: Comparison plots of power and SOC profiles for a sample sailing trip from the test dataset for TD3, (a) and (c), and SAC, (b) and (d).

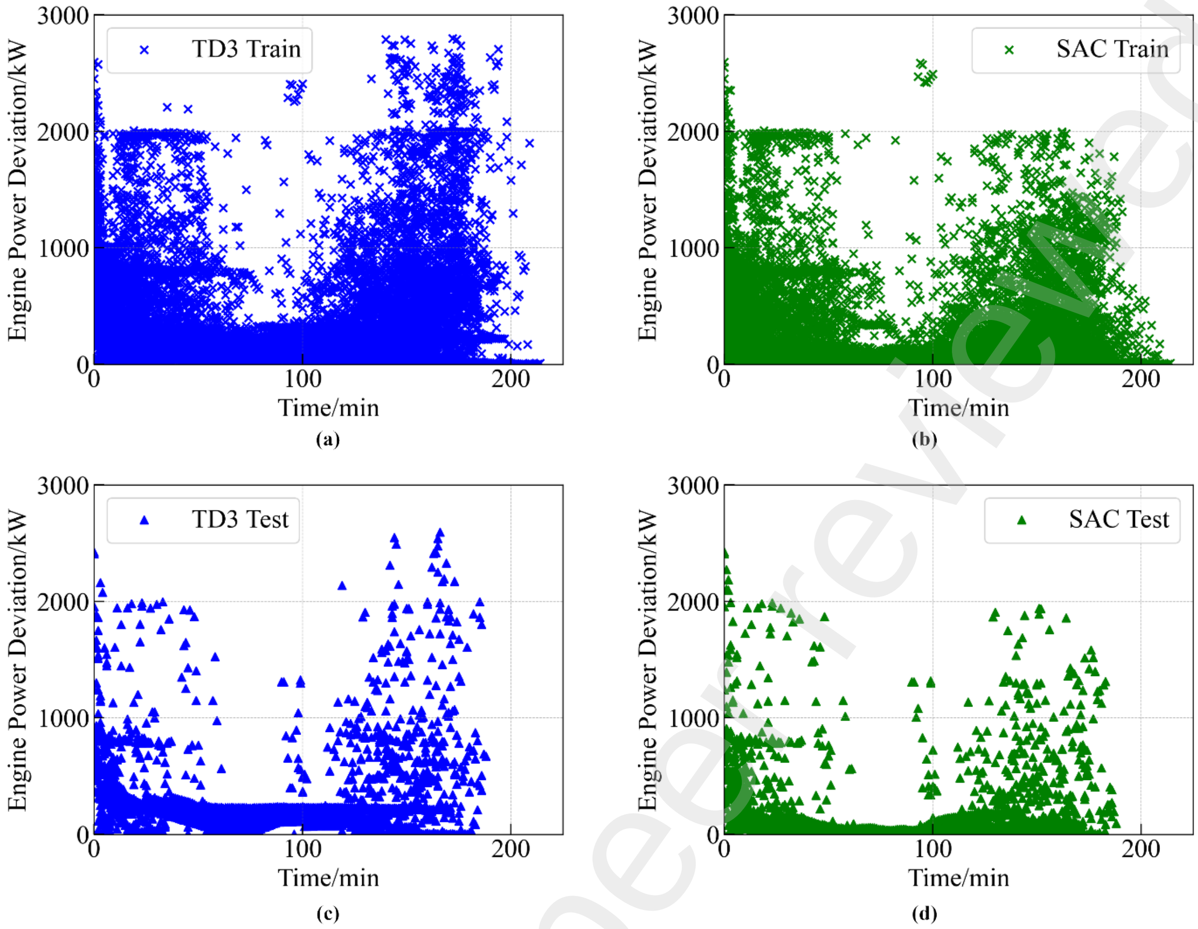


Figure 7: Deviations of the DRL selected engine loads from the actual operational engine loads. (a) and (b) represent the training dataset absolute engine load deviations for the TD3 and SAC agents, respectively. (c) and (d) represent the test dataset absolute engine load deviations for the TD3 and SAC agents, respectively. Each data point on the plots represents a data sample from a one-minute time step.

6.2 GHG Emissions Reduction Performance Compared to Offline Optimization

Sequential least squares programming (SLSQP) was used to estimate the maximum possible theoretical GHG emissions reductions that can be achieved by optimizing the engine load profile of the sailing trips through offline optimization. The solutions of the offline optimization problem represent hypothetical references used to establish best-case scenarios of power load distribution. Unlike online optimization using RL, SLSQP requires prior knowledge of the entire power demand profile for each sailing trip. The optimization task was set up as a constrained optimization problem, with constraints set on the engine power, battery SOC, and battery power limits. The objective function included the cumulative GHG emissions from each episode plus the emissions released when recharging the battery to a final SOC level of 0.85. Each sailing trip was optimized individually. The constrained optimization problem was solved using the *SciPy* library.

Optimal engine load profiles were generated using SLSQP. The optimization process took a total of 10.35 hours to determine the optimal engine load profiles for all 198 sailing trips. Figure 8 illustrates an example of an optimal solution to the optimization problem for the same sailing trip used in Figure 6. The GHG emissions reductions achieved by the TD3 and SAC agents across all 198 trips were normalized against the reductions achieved by the SLSQP optimizer. Reductions based on the actual control strategy (i.e., peak shaving + load leveling) were also normalized to assess the improvement in GHG reductions by the TD3 and SAC agents. The normalized reduction reflects the GHG reduction efficiency of each strategy and is defined in Equation 28.

$$E_{eff\%} = \frac{E_{NH} - E_{CS}}{E_{NH} - E_{SLSQP}} \times 100\%\#(29)$$

Where $E_{eff\%}$ is the GHG reduction efficiency, E_{CS} is the amount of GHG emissions associated with a sailing trip following a particular control strategy (actual/RL/SLSQP) in kg $CO_{2,eq}$, and E_{SLSQP} is the amount of GHG emissions associated with a sailing trip simulated with offline optimization using SLSQP in kg $CO_{2,eq}$.

The average GHG emissions reductions achieved by the different control strategies and their respective reduction efficiencies are summarized in Table 7. The TD3 agent showed reduction efficiency improvements of 19.80% and 18.64%, for the training and test datasets respectively, compared to the actual control strategy while the SAC agent showed improvements of 11.46% and 7.61%. The results indicate that both the TD3 and SAC agents provide more efficient control strategies than the actual approach, with the TD3 agent outperforming the SAC agent.

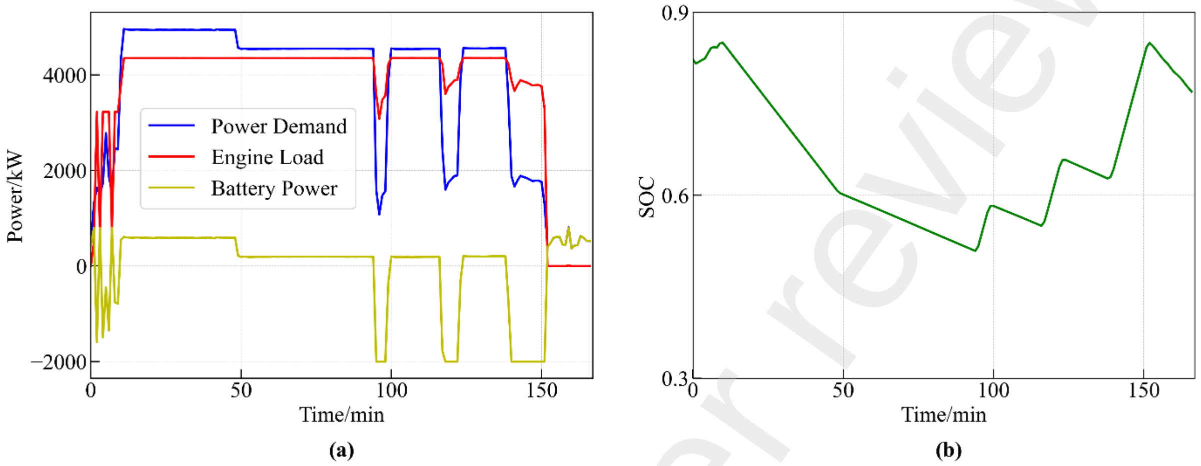


Figure 8: Power (a) and SOC (b) profiles obtained from the SLSQP optimization for a sample sailing trip from the test dataset.

Table 7: Average GHG emissions reductions achieved by the actual sailing operations, the TD3 and SAC agents (online optimization), and SLSQP (offline optimization). The emissions reductions were calculated against emissions from the same sailing trips simulated with no battery use.

Strategy	**Training Dataset**		**Test Dataset**	
	Average Emissions Reduction (kg $CO_{2,eq}$)	**Reduction Efficiency (%)**	**Average Emissions Reduction (kg $CO_{2,eq}$)**	**Reduction Efficiency (%)**
Actual Conditions	1258	61.37	1143	60.90
TD3 (Online)	1664	81.17	1493	79.54
SAC (Online)	1493	72.83	1286	68.51
SLSQP (Offline)	2050	100	1877	100

The GHG emissions reductions were further divided into CH_4 and CO_2 reductions in Table 8. The TD3 and SAC agents as well as the SLSQP optimizer led to an increase in the total CO_2 emissions by <2% while significantly reducing CH_4 emissions. This increase in CO_2 emissions is attributed to the more complete oxidation of CH_4 to CO_2. The preferential reduction in CH_4 emissions is driven by the tendency to avoid engine loads with high GHG emission rates that are driven by the non-linear nature of CH_4 emissions with engine load as described in Figure 3, and by the high GWP of CH_4. Notably, all GHG reductions achieved were the result of minimizing methane slip.

Table 8: Average CH_4 and CO_2 reductions achieved by the TD3 and SAC agents as well as the SLSQP optimizer, compared against the same sailing trips with no battery use.

Dataset & Algorithm	Methane Emissions Reduction (%)	Carbon Dioxide Emissions Reduction (%)
Training Dataset (TD3)	26.84	-1.31
Test Dataset (TD3)	25.02	-1.44
Training Dataset (SAC)	22.61	-1.54
Test Dataset (SAC)	21.42	-1.63
Training Dataset (SLSQP)	33.34	-1.49
Test Dataset (SLSQP)	31.88	-1.64

7 Conclusion and Future Work

The adoption of hybrid-powered LNG powertrains offers significant potential for reducing maritime GHG emissions, but its success depends on an effective EMS to optimize powertrain performance. This study developed a DRL-based EMS to minimize GHG emissions by controlling power distribution between the engine and battery of a hybrid LNG vessel while adhering to battery operational constraints.

Three DRL algorithms—TD3, SAC, and PPO—were assessed for feasibility in this context. TD3 and SAC emerged as the best-performing algorithms, demonstrating superior GHG reduction and compliance with battery constraints. The TD3 agent achieved GHG reduction efficiencies of 81.17% and 79.54% on training and test datasets, representing improvements of 19.80% and 18.64% over the vessel's actual control strategy. Similarly, the SAC agent achieved efficiencies of 72.83% and 68.51%, with improvements of 11.46% and 7.61%. In contrast, the PPO agent exhibited inferior GHG reduction performance and frequent violations of the SOC constraint. The GHG reductions that were achieved were primarily driven by reductions in methane slip through the optimization of the vessel's engine load. The 'model-free' nature of the DRL algorithms enabled training with real-world data, enhancing their practicality for dynamic maritime energy management scenarios.

Future work would incorporate vessel velocity optimization to further support the GHG reduction objective. The findings demonstrate that TD3 and SAC are viable tools for operators seeking to enhance environmental sustainability and operational efficiency in hybrid-powered LNG vessels, paving the way for broader adoption of intelligent EMS in maritime transport.

Acknowledgements

This project was undertaken with the financial support of the Government of Canada.

CRediT Authorship Contribution Statement

Ahmed Abdalla: Writing – original draft, Writing – review & editing, Formal analysis, Conceptualization, Software, Methodology, Visualization. **Patrick Kirchen:** Writing – review & editing, Supervision, Conceptualization, Investigation, Project administration. **Bhushan Gopaluni:** Supervision, Methodology, Conceptualization, Resources, Funding acquisition.

Competing Interests

The authors declare no competing interests.

References

[1] Transport & Environment [Internet]. 2023. Climate impact of shipping. Available from: https://www.transportenvironment.org/challenges/ships/greenhouse-gases/

[2] Forum IT. Decarbonising Maritime Transport: Pathways to zero-carbon shipping by 2035 [Internet]. Paris: OECD; 2018 Mar [cited 2024 Feb 15]. Available from: https://www.oecd-ilibrary.org/transport/decarbonising-maritime-transport_b1a7632c-en

[3] 2023 IMO Strategy on Reduction of GHG Emissions from Ships [Internet]. [cited 2024 Feb 14]. Available from: https://www.imo.org/en/OurWork/Environment/Pages/2023-IMO-Strategy-on-Reduction-of-GHG-Emissions-from-Ships.aspx

[4] Bouman EA, Lindstad E, Rialland AI, Strømman AH. State-of-the-art technologies, measures, and potential for reducing GHG emissions from shipping – A review. Transp Res Part Transp Environ. 2017 May 1;52:408–21.

[5] Inal OB, Charpentier JF, Deniz C. Hybrid power and propulsion systems for ships: Current status and future challenges. Renew Sustain Energy Rev. 2022 Mar 1;156:111965.

[6] Deniz C, Zincir B. Environmental and economical assessment of alternative marine fuels. J Clean Prod. 2016 Feb 1;113:438–49.

[7] Sommer DE, Yeremi M, Son J, Corbin JC, Gagné S, Lobo P, et al. Characterization and Reduction of In-Use CH4 Emissions from a Dual Fuel Marine Engine Using Wavelength Modulation Spectroscopy. Environ Sci Technol. 2019 Mar 5;53(5):2892–9.

[8] Rochussen J, Jaeger NSB, Penner H, Khan A, Kirchen P. Development and demonstration of strategies for GHG and methane slip reduction from dual-fuel natural gas coastal vessels. Fuel. 2023 Oct 1;349:128433.

[9] Balcombe P, Heggo DA, Harrison M. Total Methane and CO2 Emissions from Liquefied Natural Gas Carrier Ships: The First Primary Measurements. Environ Sci Technol. 2022 Jul 5;56(13):9632–40.

[10] Yuan Y, Wang J, Yan X, Shen B, Long T. A review of multi-energy hybrid power system for ships. Renew Sustain Energy Rev. 2020 Oct 1;132:110081.

[11] Cha M, Enshaei H, Nguyen H, Jayasinghe SG. Towards a future electric ferry using optimisation-based power management strategy in fuel cell and battery vehicle application — A review. Renew Sustain Energy Rev. 2023 Sep 1;183:113470.

[12] Feng Y, Zhu H, Dong Z. Simultaneous and Global Optimizations of LNG-Fueled Hybrid Electric Ship for Substantial Fuel Cost, CO2, and Methane Emission Reduction. IEEE Trans Transp Electrification. 2023 Jun;9(2):2282–95.

[13] Fan A, Li Y, Fang S, Li Y, Qiu H. Energy management strategies and comprehensive evaluation of parallel hybrid ship based on improved fuzzy logic control. IEEE Trans Transp Electrification. 2023;1–1.

[14] Roslan SB, Tay ZY, Konovessis D, Ang JH, Menon NV. Rule-Based Control Studies of LNG–Battery Hybrid Tugboat. J Mar Sci Eng. 2023 Jul;11(7):1307.

[15] Energies | Free Full-Text | State Transitions Logical Design for Hybrid Energy Generation with Renewable Energy Sources in LNG Ship [Internet]. [cited 2024 Mar 15]. Available from: https://www.mdpi.com/1996-1073/14/22/7803

[16] Pang B, Liu S, Zhu H, Feng Y, Dong Z. Real-time optimal control of an LNG-fueled hybrid electric ship considering battery degradations. Energy. 2024 Jun 1;296:131170.

[17] Wu P, Partridge J, Bucknall R. Cost-effective reinforcement learning energy management for plug-in hybrid fuel cell and battery ships. Appl Energy. 2020 Oct 1;275:115258.

[18] Wu P, Partridge J, Anderlini E, Liu Y, Bucknall R. Near-optimal energy management for plug-in hybrid fuel cell and battery propulsion using deep reinforcement learning. Int J Hydrog Energy. 2021 Nov 18;46(80):40022–40.

[19] Wu P, Partridge J, Anderlini E, Liu Y, Bucknall R. An Intelligent Energy Management Framework for Hybrid-Electric Propulsion Systems Using Deep Reinforcement Learning [Internet]. arXiv; 2021 [cited 2024 Feb 15]. Available from: http://arxiv.org/abs/2108.00256

[20] Shang C, Fu L, Bao X, Xu X, Zhang Y, Xiao H. Energy optimal dispatching of ship's integrated power system based on deep reinforcement learning. Electr Power Syst Res. 2022 Jul 1;208:107885.

[21] Jung W, Chang D. Deep Reinforcement Learning-Based Energy Management for Liquid Hydrogen-Fueled Hybrid Electric Ship Propulsion System. J Mar Sci Eng. 2023 Oct;11(10):2007.

[22] Song T, Fu L, Zhong L, Fan Y, Shang Q. HP3O algorithm-based all electric ship energy management strategy integrating demand-side adjustment. Energy. 2024 May 15;295:130968.

[23] Vallero DA. Chapter 8 - Air pollution biogeochemistry. In: Vallero DA, editor. Air Pollution Calculations [Internet]. Elsevier; 2019 [cited 2024 Feb 20]. p. 175–206. Available from: https://www.sciencedirect.com/science/article/pii/B9780128149348000089

[24] Wu P, Bucknall R. Marine propulsion using battery power. 2016.

[25] Sutton RS, Barto AG. Reinforcement Learning, second edition: An Introduction. MIT Press; 2018. 549 p.

[26] Li Y. Deep Reinforcement Learning: An Overview [Internet]. arXiv; 2018 [cited 2024 Feb 21]. Available from: http://arxiv.org/abs/1701.07274

[27] Wang X, Wang S, Liang X, Zhao D, Huang J, Xu X, et al. Deep Reinforcement Learning: A Survey. IEEE Trans Neural Netw Learn Syst. 2022;1–15.

[28] Fujimoto S, van Hoof H, Meger D. Addressing Function Approximation Error in Actor-Critic Methods [Internet]. arXiv; 2018 [cited 2024 Feb 21]. Available from: http://arxiv.org/abs/1802.09477

[29] Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, et al. Continuous control with deep reinforcement learning [Internet]. arXiv; 2019 [cited 2024 Feb 21]. Available from: http://arxiv.org/abs/1509.02971

[30] Silver D, Lever G, Heess N, Degris T, Wierstra D, Riedmiller M. Deterministic Policy Gradient Algorithms. In: Proceedings of the 31st International Conference on Machine Learning [Internet]. PMLR; 2014 [cited 2024 Feb 21]. p. 387–95. Available from: https://proceedings.mlr.press/v32/silver14.html

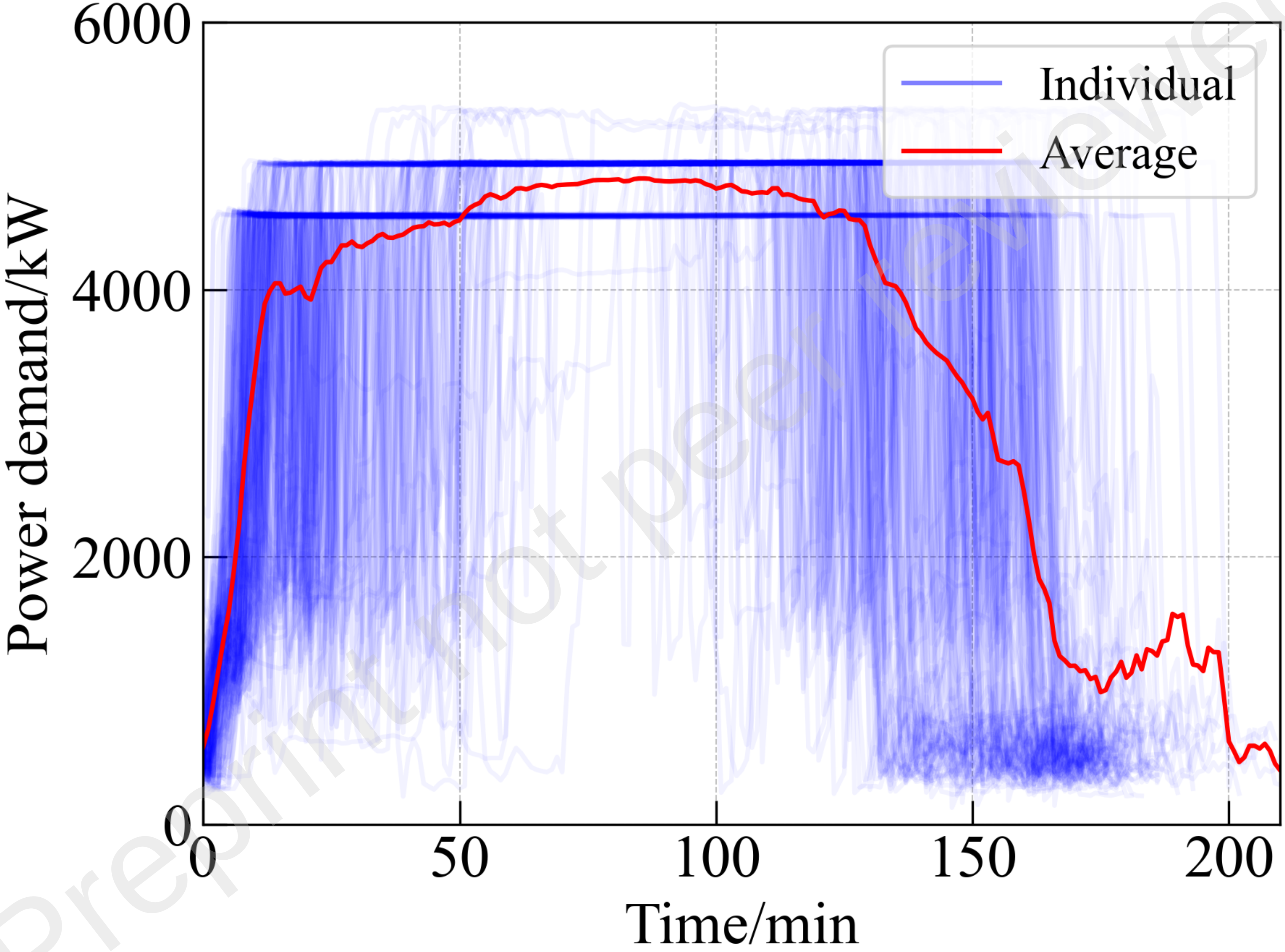
[31] Haarnoja T, Zhou A, Abbeel P, Levine S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor [Internet]. arXiv; 2018 [cited 2024 Feb 22]. Available from: http://arxiv.org/abs/1801.01290

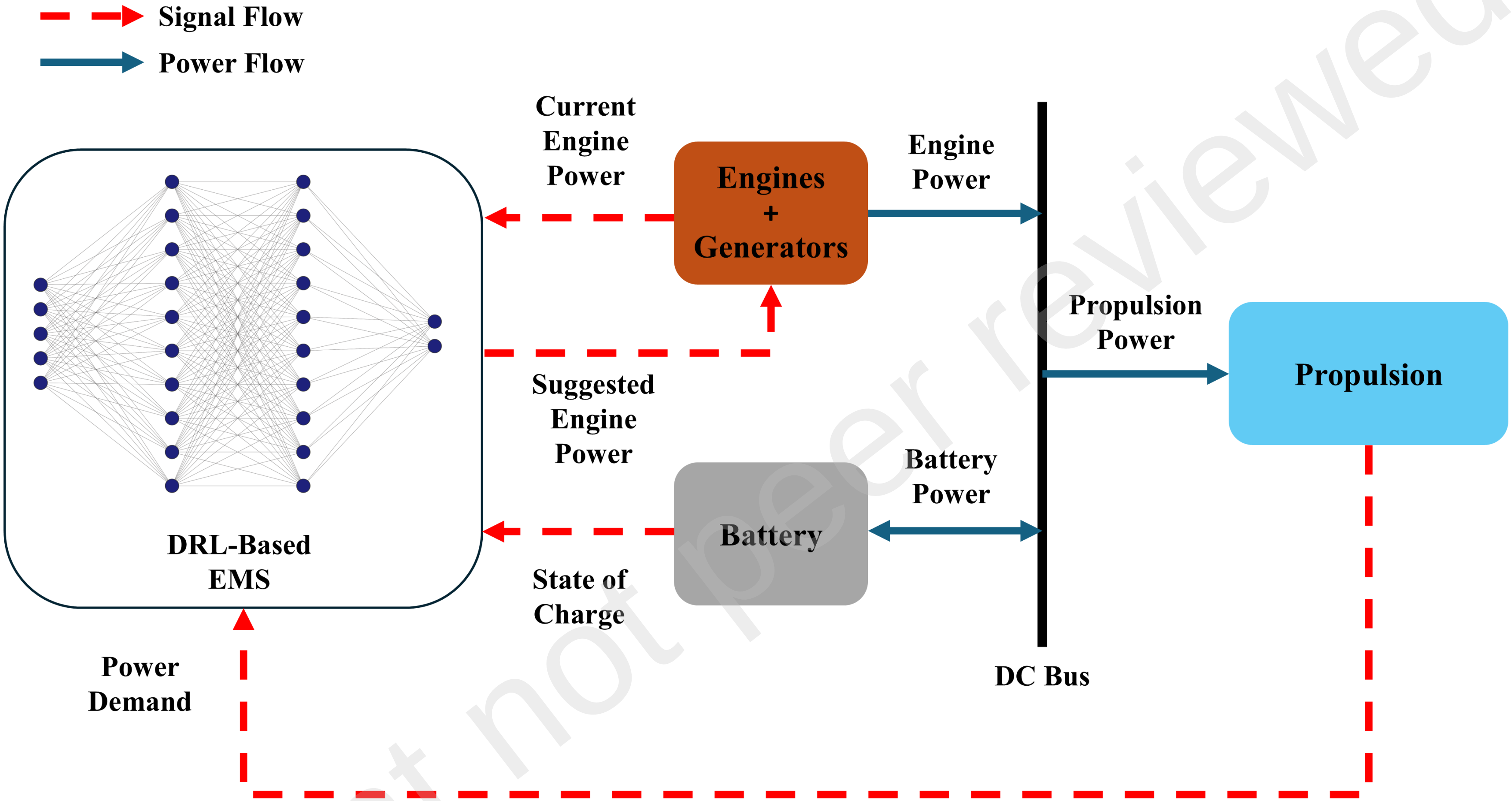
[32] Haarnoja T, Zhou A, Hartikainen K, Tucker G, Ha S, Tan J, et al. Soft Actor-Critic Algorithms and Applications [Internet]. arXiv; 2019 [cited 2024 Feb 22]. Available from: http://arxiv.org/abs/1812.05905

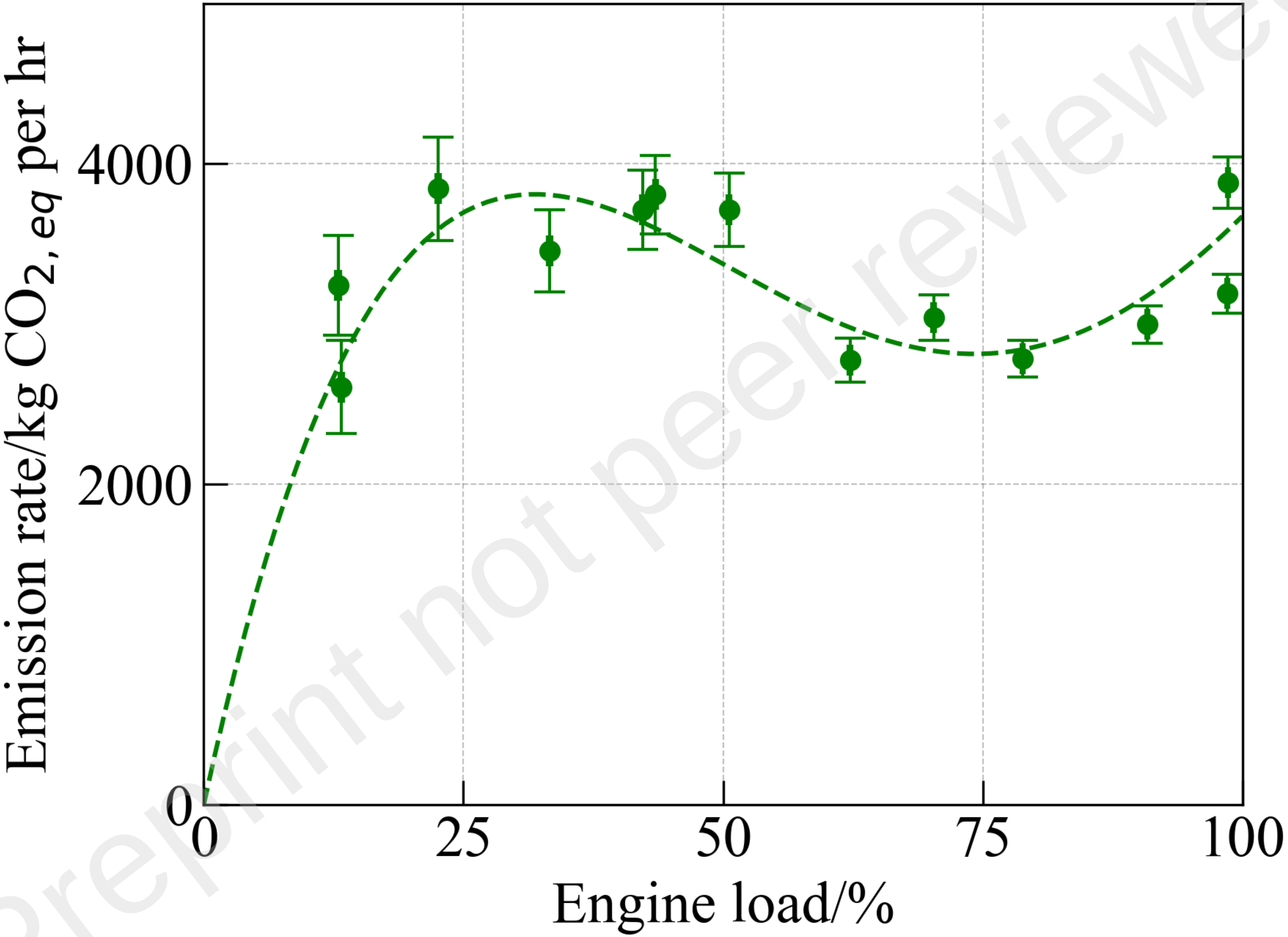
[33] Haarnoja T, Tang H, Abbeel P, Levine S. Reinforcement Learning with Deep Energy-Based Policies [Internet]. arXiv; 2017 [cited 2024 Feb 22]. Available from: http://arxiv.org/abs/1702.08165

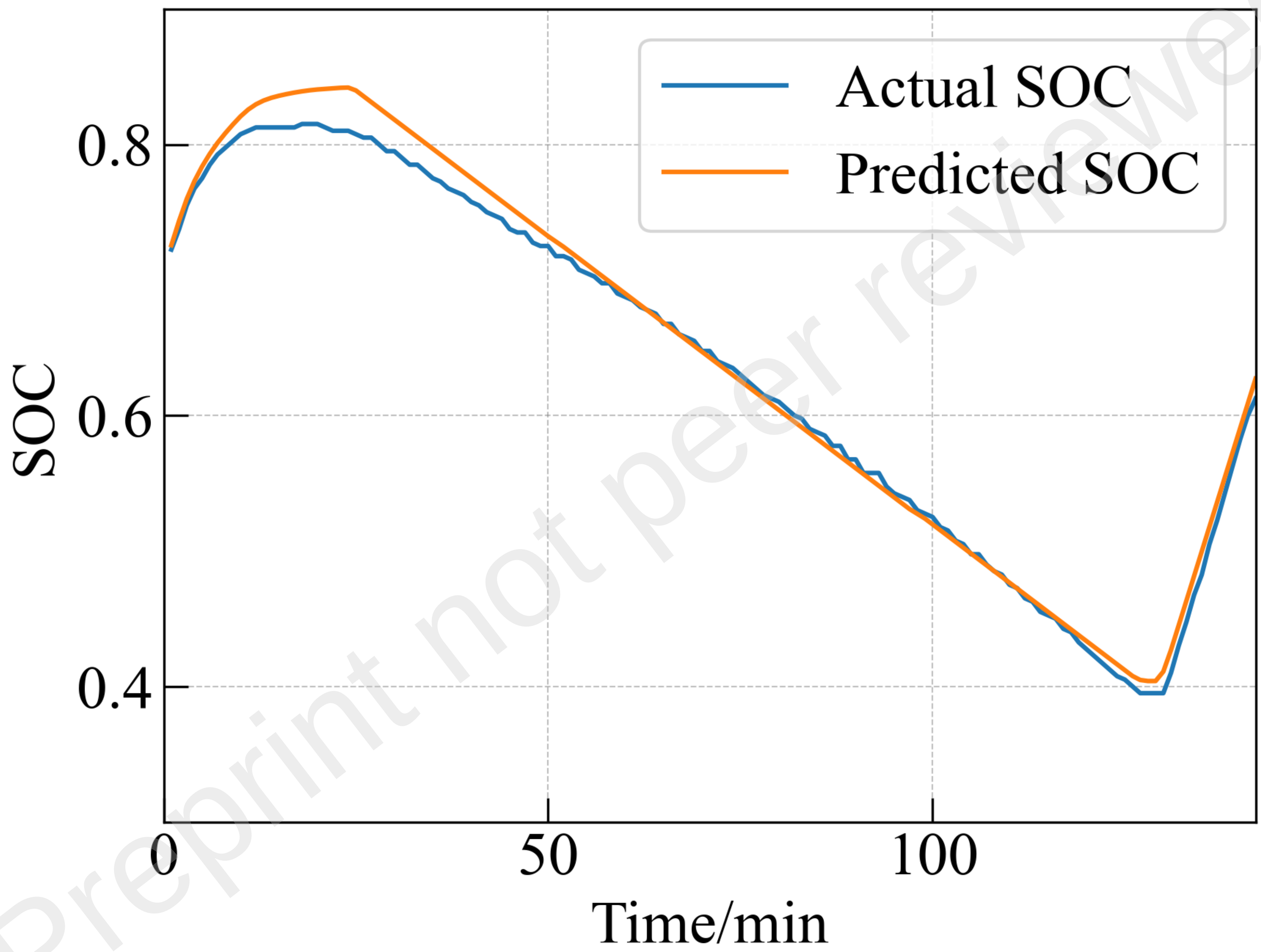
[34] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal Policy Optimization Algorithms [Internet]. arXiv; 2017 [cited 2024 Feb 22]. Available from: http://arxiv.org/abs/1707.06347

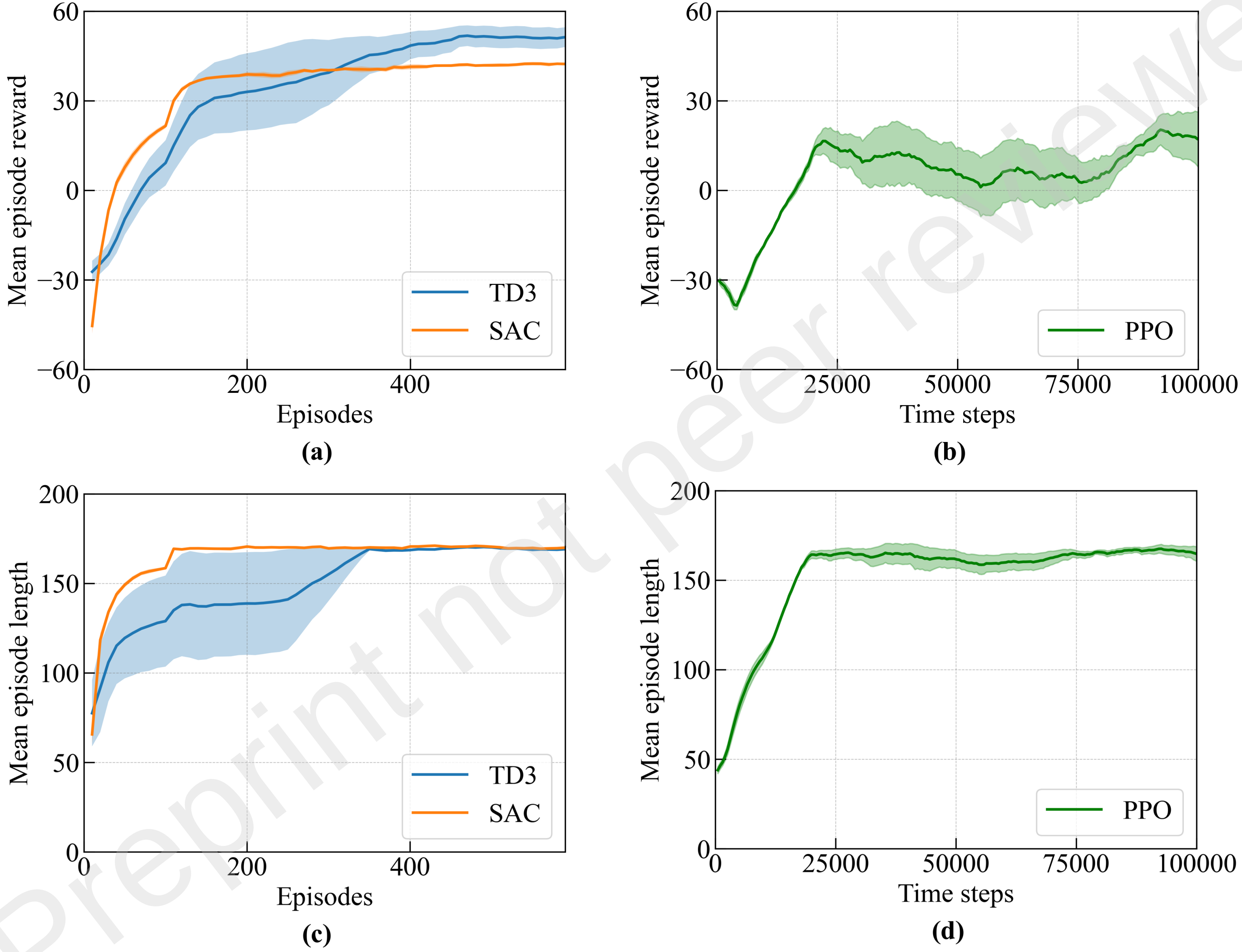
[35] Schulman J, Levine S, Moritz P, Jordan MI, Abbeel P. Trust Region Policy Optimization [Internet]. arXiv; 2017 [cited 2024 Nov 22]. Available from: http://arxiv.org/abs/1502.05477


6000
4000
2000
0
Power demand/kW
0
50
100
150
200
Time/min
Individual
Average


Signal Flow
Power Flow
Current Engine Power
Engines + Generators
Engine Power
Propulsion Power
Propulsion
Suggested Engine Power
Battery Power
Battery
DRL-Based EMS
State of Charge
DC Bus
Power Demand

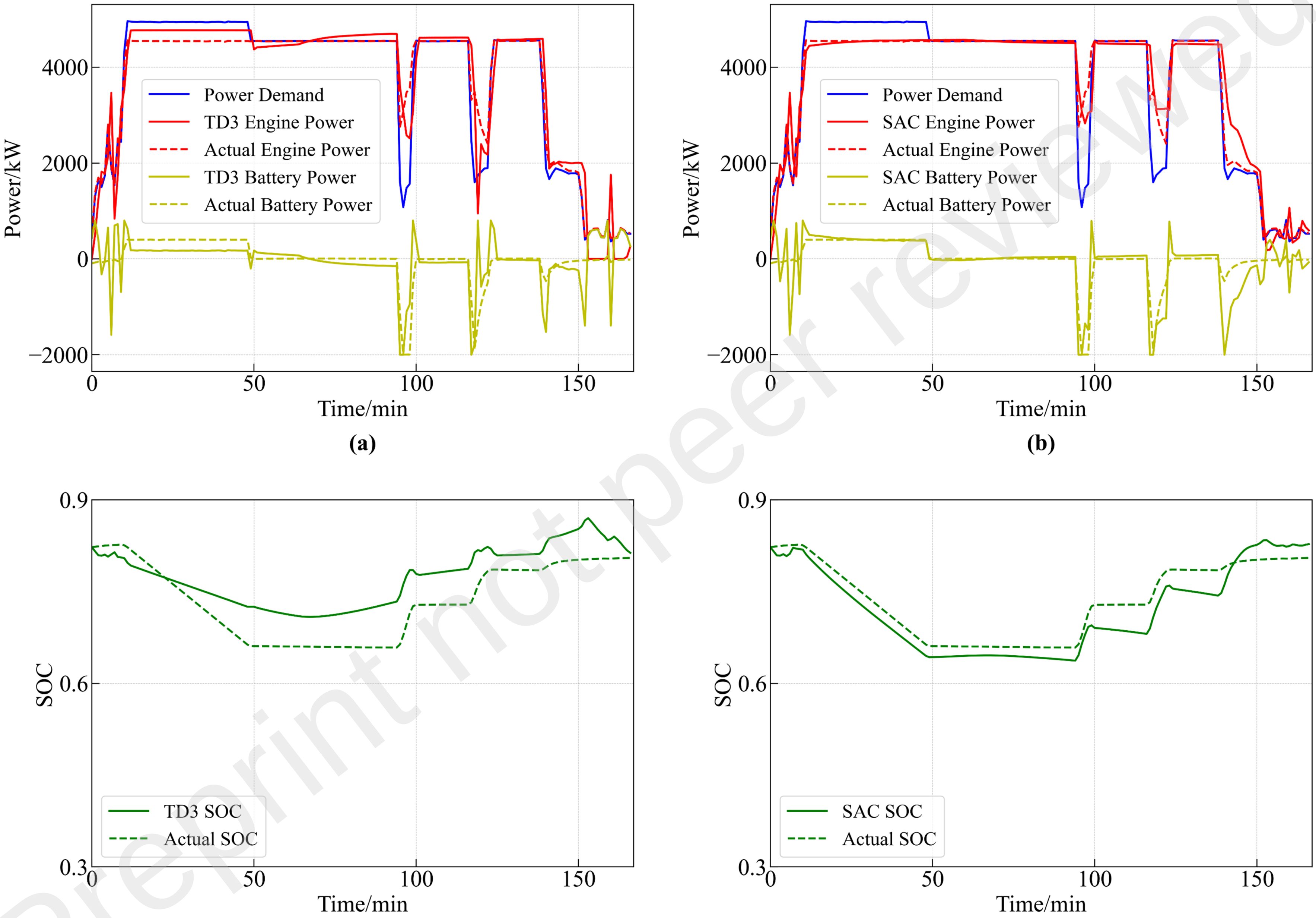

Emission rate/kg $CO_{2,eq}$ per hr
4000
2000
0
0
25
50
75
100
Engine load/%


Actual SOC
Predicted SOC
SOC
0.8
0.6
0.4
0
50
100
Time/min

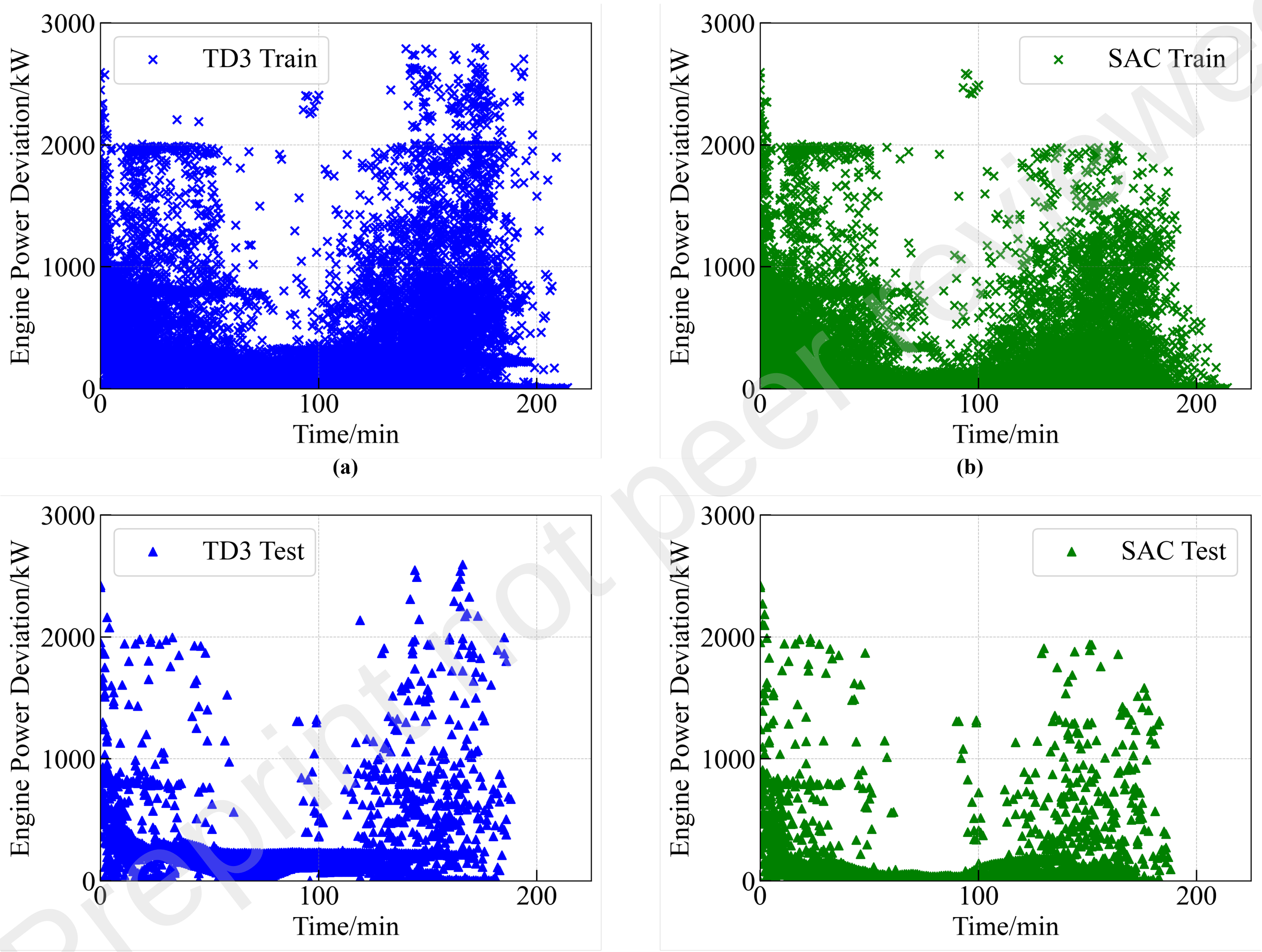

Mean episode reward
60
30
0
−30
−60
0
200
400
Episodes
TD3
SAC
Mean episode reward
60
30
0
−30
−60
0
25000
50000
75000
100000
Time steps
PPO
Mean episode length
200
150
100
50
0
0
200
400
Episodes
TD3
SAC
Mean episode length
200
150
100
50
0
0
25000
50000
75000
100000
Time steps
PPO

(a) **(b)**

(c) **(d)**


Power/kW
4000
2000
0
−2000
0
50
100
150
Time/min
Power Demand
TD3 Engine Power
Actual Engine Power
TD3 Battery Power
Actual Battery Power
Power Demand
SAC Engine Power
Actual Engine Power
SAC Battery Power
Actual Battery Power
SOC
0.9
0.6
0.3
TD3 SOC
Actual SOC
SAC SOC
Actual SOC

(a) (b) (c) (d)

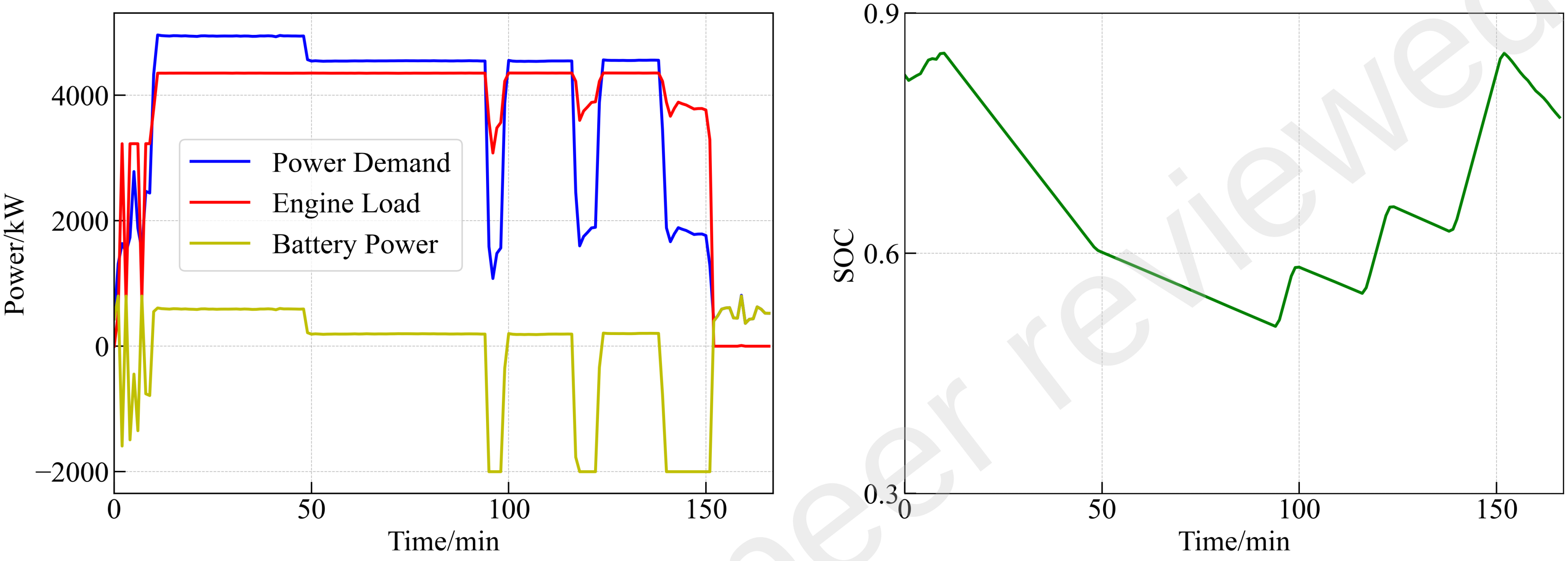

TD3 Train
SAC Train
TD3 Test
SAC Test
Engine Power Deviation/kW
Time/min
0
1000
2000
3000
0
100
200

(a)

(b)

(c)

(d)


Power/kW
4000
2000
0
−2000
0
50
100
150
Time/min
Power Demand
Engine Load
Battery Power
SOC
0.9
0.6
0.3
0
50
100
150
Time/min

(a)

(b)

Figure 1: Typical power demand profiles of the hybrid-powered LNG vessel under study. The red line represents the average power demand whereas the blue lines represent the 198 individual sailing trips.

Figure 2: Schematic of the hybrid-powered setup and the energy management system. The powertrain consists of two LNG-fueled engines and associated generators, a battery, and two electric propulsion motors.

Figure 3: Measured GHG emission rates in kg carbon dioxide equivalent per hour versus the vessel's engine load. The emission rates peak at an engine load of 32%. The minimum emission rate at operational loads between 20-100% is at an engine load of 74%.

Figure 4: A sample SOC profile plot that compares the actual SOC to the predicted SOC of a sailing trip using the constant efficiency model. The R^2 and MSE for this plot are 0.99 and 0.0004, respectively.

Figure 5: Learning curves of the TD3 and SAC agents after 100,000 time steps of training. (a) and (b) represent the mean episode reward calculated as training progresses, while (c) and (d) represent the mean episode length.

Figure 6: Comparison plots of power and SOC profiles for a sample sailing trip from the test dataset for TD3, (a) and (c), and SAC, (b) and (d).

Figure 7: Deviations of the DRL selected engine loads from the actual operational engine loads. (a) and (b) represent the training dataset absolute engine load deviations for the TD3 and SAC agents, respectively. (c) and (d) represent the test dataset absolute engine load deviations for the TD3 and SAC agents, respectively. Each data point on the plots represents a data sample from a one-minute time step.

Figure 8: Power (a) and SOC (b) profiles obtained from the SLSQP optimization for a sample sailing trip from the test dataset.