drones

MDPI

Article

Energy-Aware Hierarchical Reinforcement Learning Based on the Predictive Energy Consumption Algorithm for Search and Rescue Aerial Robots in Unknown Environments

M. Ramezani and M. A. Amiri Atashgah *

College of Interdisciplinary Science and Technologies, University of Tehran, Tehran 14174-66191, Iran; ramezani.mahya@ut.ac.ir
* Correspondence: atashgah@ut.ac.ir

Abstract: Aerial robots (drones) offer critical advantages in missions where human participation is impeded due to hazardous conditions. Among these, search and rescue missions in disaster-stricken areas are particularly challenging due to the dynamic and unpredictable nature of the environment, often compounded by the lack of reliable environmental models and limited ground system communication. In such scenarios, autonomous aerial robots' operation becomes essential. This paper introduces a novel hierarchical reinforcement learning-based algorithm to address the critical limitation of the aerial robot's battery life. Central to our approach is the integration of a long short-term memory (LSTM) model, designed for precise battery consumption prediction. This model is incorporated into our HRL framework, empowering a high-level controller to set feasible and energy-efficient goals for a low-level controller. By optimizing battery usage, our algorithm enhances the aerial robot's ability to deliver rescue packs to multiple survivors without the frequent need for recharging. Furthermore, we augment our HRL approach with hindsight experience replay at the low level to improve its sample efficiency.

Keywords: hierarchical reinforcement learning; long short-term memory networks; search and rescue mission; energy-efficient path planning

Citation: Ramezani, M.; Amiri Atashgah, M.A. Energy-Aware Hierarchical Reinforcement Learning Based on the Predictive Energy Consumption Algorithm for Search and Rescue Aerial Robots in Unknown Environments. *Drones* **2024**, *8*, 283. https://doi.org/10.3390/drones8070283

Academic Editor: Abdessattar Abdelkefi

Received: 26 May 2024
Revised: 12 June 2024
Accepted: 13 June 2024
Published: 23 June 2024

1. Introduction

Aerial robots have become increasingly indispensable in various applications. One of these applications is in search and rescue (SAR) operations, especially in post-disaster scenarios where human intervention is often fraught with danger or impracticality. These missions are typically conducted in environments with unpredictable environmental conditions [1,2]. The foremost objective of SAR operations is the rapid location of targets and the execution of critical follow-up actions, such as relaying information and delivering essential supplies, all within a constrained timeframe. Employing aerial robots in SAR missions offers numerous benefits, including their swift deployment capabilities, cost-effective maintenance, exceptional mobility, and the capacity to operate in areas where manual intervention limits risks or requires rapid decision-making processes [3]. Challenges include navigation among numerous obstacles, efficiently locating and assisting survivors, potential damage to ground system infrastructure, and limitations due to the aerial robot's battery capacity [4].

Path planning is a crucial component in these contexts, entailing the formulation of an optimal trajectory from the origin to the destination while adhering to operational constraints and mission objectives. Traditional path planning methods, such as grid-based and graph-based algorithms (such as A* [5], artificial potential fields [6], and Dijkstra's algorithm [7]), have demonstrated efficacy in stable scenarios [8]. However, they often struggle in the unpredictable and dynamic terrains characteristic of disaster zones, primarily

due to limitations in real-time adaptability and autonomous decision-making in the face of complex obstacles and environmental uncertainties.

With the advent of advancements in artificial intelligence (AI), particularly in machine learning (ML) and reinforcement learning (RL) [9], new solutions have emerged to address these challenges. RL, with its adaptability and learning-based approach [10], is particularly suitable for dynamic and uncertain post-disaster environments. RL systems have shown promise in dynamically adapting to changing terrains and unforeseen obstacles [11]. Deep reinforcement learning (DRL) [12], which incorporates deep neural networks into RL, has been explored for its effectiveness in complex and multifaceted scenarios [13,14]. Specifically, DRL's application in aerial robot path planning has led to remarkable improvements in multi-objective environments, such as navigating through disaster-stricken areas [15].

Aerial robots are constrained by limited flight duration, primarily due to their reliance on battery power. Consequently, strategic planning of flight routes and scheduling recharge stops become essential to ensure successful mission completion. Energy-aware navigation frameworks for aerial robots are designed to address this challenge. They focus on providing efficient route planning that not only circumvents obstacles but also optimizes battery usage, thereby enhancing the operational range and effectiveness of aerial robots in various applications [16]. Bouhamed et al. [17] developed a framework utilizing the deep deterministic policy gradient (DDPG) algorithm for aerial robot navigation. This framework is designed to efficiently guide aerial robots to designated target positions while maintaining communication with ground stations. Moreover, Imanberdiyev et al. [18] proposed a methodology that focuses on monitoring critical aerial robot parameters, including battery level, rotor condition, and sensor readings, for enhanced route planning. The approach involves dynamically adjusting the aerial robot's flight path as necessary to facilitate battery charging when needed. An autonomous aerial robot path planning framework utilizing the DDPG approach was developed to train aerial robots by the authors of [19]. This framework enables aerial robots to effectively navigate through or above obstacles to reach predetermined targets.

Despite these advancements, DRL models often struggle to manage multiple objectives simultaneously, such as evading various targets and optimizing different objectives. Managing several goals simultaneously often leads to complex and larger state spaces and inefficiencies due to the conflicting nature of objectives. To overcome these challenges, there is a need for distinct models for each objective [20,21]. Other frameworks, such as meta-learning-based DRL and modular hierarchical DRL architectures, are suggested to address the computational demands and recalibration needs in complex, multi-objective scenarios [22].

Hierarchical reinforcement learning (HRL) can overcome these challenges of the reinforcement learning [23]. Inspired by human thinking on solving complex problems, HRL not only breaks down the problem into sub-problems that are easier to handle but has the ability to train multiple policies that are connected at different levels of temporal abstraction. HRL offers a structured approach for tasks involving multiple objectives, by segmenting decision-making into different layers [24]. Its application in aerial robot navigation has included coordinating multi-objective missions, exemplified in recent studies where HRL has been employed to optimize task allocation and path planning [25,26].

In addition, despite HRL's advantages in managing complex tasks, HRL faces several critical problems, including sparse rewards, long time horizons, and the effective transfer of policies across different tasks. However, the most significant challenge is ensuring that high-level policies assign feasible and well-defined tasks to low-level policies [27]. Inconsistent or poorly defined tasks can lead to inefficiencies and failure in task execution. Algorithms like hierarchical proximal policy optimization (HiPPO) aim to address this by jointly training all levels of the hierarchy and allowing continuous adaptation of skills even in new tasks [28]. However, HiPPO itself faces challenges, including the complexity of simultaneously training multiple levels of policies, the potential for increased computa-

tional demands, and the difficulty in maintaining stability and convergence during the training process.

To address this challenge, this paper proposes a novel method for aerial robot path planning in SAR missions. Our work introduces a novel integration of adaptive long short-term memory (LSTM) networks for real-time battery consumption prediction within a hierarchical reinforcement learning framework. This integration offers several unique advantages:

- The LSTM model assists the high-level policy in selecting feasible goals for the low-level policy by predicting the battery requirements for each goal, ensuring that the chosen goals remain within the robot's energy constraints.
- Our model incorporates a bidirectional LSTM framework for accurate battery consumption prediction. This dual-layer structure processes data sequences in both forward and backward directions, enhancing the model's ability to understand context and sequence dynamics, thus providing more accurate predictions of energy requirements for each target.
- By forecasting battery consumption for each target, the LSTM model informs the high-level policy (HLC) within the HRL framework, enabling more informed and energy-efficient goal selection. This model dynamically adjusts flight paths based on real-time battery predictions, enhancing the robot's effectiveness in complex, multi-objective missions. By providing energy consumption forecasts, our framework ensures mission completion without energy depletion, increases the overall success rate of SAR operations, and avoids mid-mission energy shortages, thereby enhancing mission success rates.
- The use of hindsight experience replay at the LLC level improves learning efficiency and robustness in changing environments. This accelerates the convergence of learning algorithms, enabling quicker adaptation to dynamic environments and reducing the training time required for effective path planning.
- The proposed framework includes an adaptive mechanism that gradually reduces reliance on LSTM predictions as the HLC learns from environmental interactions. This transition enhances the HLC's autonomy and efficiency in mission planning, optimizing energy usage based on real-time learning and experiences.

This study aims to investigate energy-aware path planning for aerial robots tasked with delivering supplies to multiple targets while avoiding obstacles. Path planning is conducted over two hierarchical levels. An LSTM network first predicts the battery consumption for each target, informing the high-level HRL policy's goal selection. The selected goal then guides the low-level navigational decisions, enabling the aerial robot to identify efficient paths while avoiding obstacles. Additionally, the LSTM's predictions assist the RL algorithm in making energy-efficient decisions by forecasting battery consumption for upcoming states. To address non-stationarity at the low level, we employ hindsight experience replay in the low-level policy. In the HRL framework, a spectrum of RL algorithms is strategically deployed to bolster the functionality of both the high-level controller (HLC) and the low-level controller (LLC). Among these, the soft actor-critic (SAC) [29,30] algorithm is known for its unparalleled adaptability and efficiency, rendering it an optimal choice for navigating the complexities of aerial robot navigation within high-dimensional continuous action spaces. The SAC algorithm is also celebrated for its incorporation of entropy regularization, a feature that inherently promotes an exploratory stance. This aspect is crucial as it guarantees a comprehensive exploration of the state space, thereby facilitating the development of versatile strategies for aerial robot navigation. The off-policy nature of SAC significantly enhances learning efficiency by leveraging past experiences, a critical advantage in scenarios where real-time responsiveness is paramount. Therefore, we incorporate the SAC into our HRL framework. We evaluate our algorithm's efficiency in an aerial robot tasked with delivering food to multiple survivors, focusing on energy efficiency and optimizing time in paths that require obstacle avoidance.

The rest of the paper is organized as follows: Section 2 contains the works that relate to our paper, Section 3 presents the preliminary and problem formulation, and Section 4 introduces the proposed HRL method. In Section 5, the experiments and results are shown. Section 6 brings the final remarks and proposes future work.

2. Background

2.1. Hierarchical Reinforcement Learning

HRL has suitable structures to address long-horizon tasks [31,32]. Primarily, these structures are conceptualized either where multiple policies are hierarchically organized or where singular policies are sequentially stacked. In classic architectures, a high-level policy is typically trained over a set of predefined low-level policies, as seen in [33]. One example is the Options framework [34], which involves learning to select and execute an 'option' until its termination condition is met. It essentially functions as a switch over these options. However, this framework often necessitates prior domain-specific knowledge for the design of options. Addressing this, the option-critic approach [35] presents a methodology for jointly training a high-level policy.

Recent advancements in the Options framework include the incorporation of regularization to facilitate the learning of multiple sub-policies [36]. Conversely, another structure, exemplified by the FeUdal framework [37] and hierarchical reinforcement learning with off-policy (HIRO) algorithms [29], primarily utilizes goal-conditioned rewards.

In these models, the high-level policy operates as a manager or planner, directing the low-level policy toward the achievement of the provided sub-goals. The advantage of these methods is their generic representation, obviating the need for domain-specific customization. HIRO employs the state in the raw form to construct a parameterized reward function. Subsequent works [38,39] have extended the concept of sub-goals to latent spaces. This goal-conditioned approach has been effectively adapted to real-world robotic applications [40] and has seen extensive exploration in domains beyond traditional HRL [41,42]. However, the challenges associated with these algorithms, as highlighted in [43,44], suggest opportunities for further refinement and development.

2.2. Hierarchical Reinforcement Learning in the Aerial Robot's Battery Life

Battery life is a critical limitation for aerial robots, particularly in SAR missions where prolonged operations are essential for mission success. HRL offers promising solutions for managing battery life more effectively through its structured approach involving high-level controllers (HLC) and low-level controllers (LLC). HRL frameworks dynamically adapt to real-time changes in battery levels and environmental conditions, ensuring optimal energy usage throughout the mission [45]. This hierarchical structure allows for the separate optimization of different mission aspects, such as navigation and battery management, leading to more efficient learning processes and enhanced overall performance.

By decomposing tasks, HRL accelerates the learning process, as sub-tasks can be learned independently and in parallel, facilitating faster convergence to optimal policies [46]. This is particularly beneficial for real-time applications where rapid decision-making is crucial. Additionally, the hierarchical structure enhances adaptability to changing environments and mission requirements, allowing aerial robots to modify or relearn specific sub-tasks without necessitating an overhaul of the entire system, thereby improving robustness [47].

Despite these advantages, several challenges arise when utilizing HRL for battery management in aerial robot path planning. Synchronizing the actions of the HLC and LLC to ensure cohesive and efficient battery management can be complex, especially in dynamic environments [48]. Reliable prediction of battery consumption is crucial for making informed decisions at both hierarchical levels as inaccurate predictions can lead to suboptimal paths and mission failures. Furthermore, the credit assignment problem in HRL complicates the learning process as feedback from actions taken by lower-level agents affects overall mission outcomes and battery usage. This delayed feedback makes it

difficult to accurately attribute outcomes to specific actions or decisions, requiring careful coordination and fine-tuning [49].

Preventing battery depletion during missions is a primary challenge in HRL for aerial robots. Ensuring that the UAV does not run out of battery mid-mission is essential for the success of SAR operations. Proposed solutions include incorporating energy constraints into the HRL framework where the HLC sets goals within the UAV's energy capacity, and the LLC optimizes actions to achieve these goals efficiently [50]. Dynamic recharging strategies and energy-aware planning algorithms have also been suggested to proactively manage battery levels [51].

However, challenges remain in accurately managing battery consumption, particularly during the early training episodes, which can lead to suboptimal performance in real-world missions. Integrating a battery consumption prediction method, such as using LSTM networks, can address this issue. By predicting battery usage based on real-time data and historical patterns, the LSTM model helps the HLC to set feasible goals [52]. This integration not only enhances the overall efficiency and reliability of the HRL framework but also increases the success rate and accelerates the convergence rate by ensuring that the UAV operates within its energy constraints throughout the mission.

3. Problem Definition

This research addresses the multifaceted challenge of optimal path planning for aerial robots in the critical context of post-disaster missions. The objective is to deploy an aerial robot to deliver vital supplies to survivors whose locations are predetermined yet situated in an uncertain post-disaster environment. The aerial robot is equipped with LIDAR for obstacle detection, GPS for positioning, and IMU sensors to track velocity.

The goal is to design a path for the aerial robot that satisfies time efficiency, energy consumption, and collision avoidance constraints. Furthermore, each successful delivery results in a decrease in the aerial robot's payload weight, a factor that introduces variables affecting the aerial robot's flight dynamics and energy efficiency. The problem formulation is presented as follows:

Given a set of targets $\mathbb{T} = \{\approx_1, \approx_2, ..., \approx_n\}$, the aerial robot must determine a path $\mathbb{P} = \{\imath_1, \imath_2, ..., \imath_n\}$, where $\imath_i$ denotes the trajectory to target $\approx_i$ such that the total operational cost C is minimized. The cost encompasses the combined metrics of time $\tau(\mathbb{P})$, energy $E(\mathbb{P})$, and the collision avoidance $C(\mathbb{P})$, represented as a weighted sum in the objective function:

$$\min_{P} \{\omega_1 \cdot E(\mathbb{P}) + \omega_2 \cdot \tau(\mathbb{P}) + \omega_3 \cdot C(\mathbb{P})\} \quad (1)$$

subject to the following constraints:

- Collision avoidance: Each trajectory segment $\imath_i$ must adhere to safe navigational practices, as determined by onboard LIDAR and GPS data, to avoid collisions.
- Energy consumption: The energy function $E(\mathbb{P})$ considers the variable weight due to payload changes, as well as aerial robot-specific power consumption profiles.
- Time efficiency: The time function $\tau(\mathbb{P})$ is a measure of the temporal efficiency of each trajectory, with a lower τ indicating a more desirable path.
- Safety: The safety function $C(\mathbb{P})$ is a measure of the aerial robot's adherence to collision avoidance throughout the mission, with higher values indicating safer operations.
- Battery constraint: The trajectory must be completed without the need for recharging, imposing a natural limit on the length and complexity of the path.

The optimization problem is not a convex optimization problem due to the non-linear nature of the energy and safety functions and the discrete nature of collision avoidance constraints.

To achieve this, drawing from the insights gathered through a thorough literature survey and recognizing the inherent advantages of hierarchical approaches, we propose to address the problem by leveraging goal-conditioned HRL. One of the main challenges in the context of HRL is based on the high level assigning goals for the low level, which

is primarily due to the potential for the HLC to assign unattainable goals [53]—because they are implausible or beyond the capabilities of the LLC. In instances where the HLC requests an unattainable goal, the LLC tends to persist in its efforts until a timeout is reached, incurring a notable cost in terms of interactions with the environment. This issue is particularly noticeable during the initial stages of training as the HLC may frequently select unattainable goals without accounting for the associated resource costs [54]. Based on our mission, a pivotal challenge for unfeasible goals emerges in the context of battery-level feasibility for goal attainment. This challenge is intricately linked to the HLC capacity to set realistic and achievable goals, given the aerial robot's limited battery life. While the aerial robot's targets are predetermined and accessible, ensuring that the aerial robot can reach these targets without exhausting its battery reserves is critical. This necessitates the integration of a battery consumption model within the HLC, capable of accurately estimating the power requirements for reaching each delivery target and returning to the recharge station.

To overcome these challenges in this paper, we propose an approach using goal-conditioned HRL, integrated with an external pretrained LSTM-based battery consumption prediction for energy-aware decision-making. Given that LSTM has the potential to predict the battery consumption model of the aerial robot based on current states for subsequent states, it aids the LLC in making better decisions for the next action. This is based on the battery consumption predictions, enabling more energy-efficient decision-making.

Additionally, goal-conditioned HRL systems face the challenge of non-stationarity at the low level. This challenge arises as the high-level controller adjusts goals, which can shift the landscape that the low-level controller must navigate, affecting its ability to maintain consistent policy performance [55]. To mitigate it, the framework utilizes the hindsight experience replay (HER), which addresses the low-level policy's non-stationarity arising from goal changes dictated by the HLC. This adaptation assists the HLC in refining its decision-making process, based on previous failures to achieve feasible goals, thereby enhancing its ability to set more attainable goals for the low level. The schematic algorithm of the proposed method is illustrated in Figure 1.

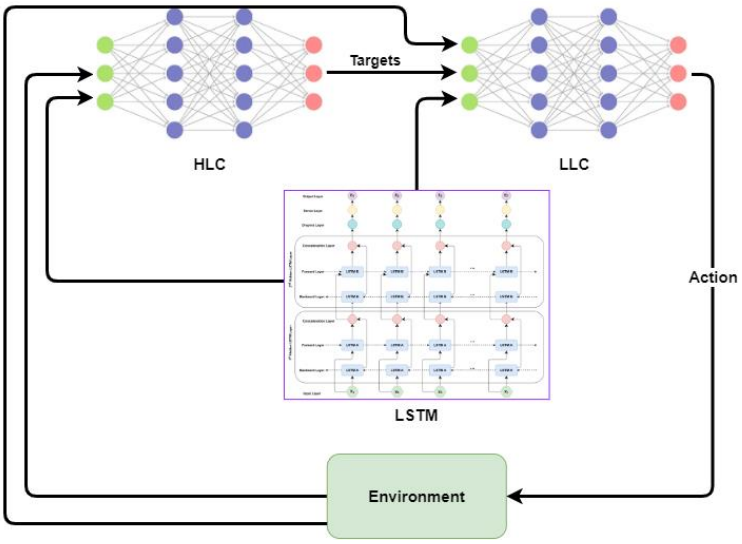

Figure 1. The block diagram of the method.

3.1. Goal-Conditioned Markov Decision-Making

In formulating the RL problem, we should formalize it by leveraging the principles of Markov decision processes (MDPs) [56]. By adhering to the constructs of MDPs, we can articulate the RL problem with precision and clarity, thereby facilitating its systematic analysis and resolution within the domain of decision-making under uncertainty. In HRL

settings, the MDP [13] is typically augmented with a set of goals G. This augmentation is pivotal to aligning the MDP with goal-oriented tasks.

The goal set G changes the standard MDP into a goal-conditioned MDP. This new framework mixes objectives with the state space directly. It leads to a focused decision-making process. Now, the agent's actions respond to both the environment and specific goals. In this version, we consider a tuple (S, A, G, P, R, γ) where S represents the state space; A signifies the action space; G encompasses the set of goals that guide the agent's policy; $P : S \times A \times G \to S$ is the transition probability function, dictating the state evolution; $R : S \times A \times G \to R$ is the goal-dependent reward function; and γ is the discount factor, indicating the value of future rewards.

3.2. Soft Actor-Critic Algorithm

In this section, we introduce the foundational principles and mathematical underpinnings of the SAC algorithm, a cornerstone of our proposed HRL framework. SAC, an advanced off-policy algorithm in the domain of deep reinforcement learning, excels in managing complex and high-dimensional control tasks. It is particularly lauded for its ability to strike a harmonious balance between exploration and exploitation, achieved through the incorporation of entropy into the reward optimization process. This attribute is crucial for navigating continuous action spaces with a high degree of efficiency and sample economy. Moreover, adding an entropy term to the objective function helps ensure that policies do not become overly deterministic too quickly. This encourages exploration and helps avoid premature convergence to suboptimal policies, enhancing stability. The entropy term promotes sufficient exploration of the policy space, which is critical for navigating the complex and non-convex optimization landscape effectively [57,58]. The SAC objective function for the policy π is given by:

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t, a_t) \sim \rho_\pi} [r(s_t, a_t) + \alpha H(\pi(\cdot \mid s_t))] \tag{2}$$

where ρ_π denotes the state-action distribution under policy π, $r(s_t, a_t)$ is the reward function, α stands for the temperature parameter controlling the trade-off between reward and entropy, and H signifies the entropy of the policy.

The core components of SAC, instrumental in operationalizing its objectives, are the action-value function $Q(s_t, a_t)$, the state-value function $V(s_t)$, and the policy $\pi(a_t \mid s_t)$, governed by which are defined as follows:

$$\begin{aligned} Q(s_t, a_t) &= r(s_t, a_t) + \gamma \mathbb{E}_{(s_{t+1} \sim p)} [V(s_{t+1})] \\ V(s_t) &= \mathbb{E}_{at \sim \pi} \Big[Q(s_t, a_t) \Big[\alpha \log_{\pi(a_t \mid s_t)} \Big] \end{aligned} \tag{3}$$

The policy is updated by minimizing the expected KL divergence to a target policy, effectively adjusting the policy toward actions that maximize the expected sum of rewards and entropy. The optimization of these components is achieved through iterative updates, leveraging stochastic gradient descent and experience replay for efficient learning.

SAC's introduction of the entropy term into the optimization objective ensures a more exploration strategy than traditional reinforcement learning algorithms. By dynamically adjusting the policy to encourage exploration, SAC mitigates the risk of premature convergence to suboptimal policies and enhances the algorithm's adaptability to diverse and unpredictable environments.

In operationalizing SAC, neural networks $Q_\theta(s_t, a_t)$ and $\pi_\phi(a_t \mid s_t)$ are employed to represent the action-state value function $Q(s_t, a_t)$ and policy $\pi(a_t \mid s_t)$, respectively. The iterative refinement of these entities is facilitated through minibatch sampling from the experience replay buffer. Furthermore, the introduction of a target network for both $Q(s_t, a_t)$ and $\pi(a_t \mid s_t)$ enables soft updates, markedly enhancing the stability of the learning process.

The optimization of SAC's framework is governed by two distinct loss functions, tailored to refine the critic and actor networks. Critic network and actor network losses are given by:

$$\begin{aligned} J_Q(\theta) &= \mathbb{E}(s_t, a_t)_{\sim D}\left[\frac{1}{2}\left(Q_\theta(s_t, a_t) - \hat{Q}(s_t, a_t)\right)^2\right] \\ J_\pi(\phi) &= \mathbb{E}_{s_t \sim D, a_t \sim \pi_\phi}\left[\log \pi_\phi(a_t \mid s_t) - \frac{1}{\alpha} Q_\theta(s_t, a_t)\right] \end{aligned} \tag{4}$$

in which

$$\hat{Q}(s_t, a_t) = r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim p}\left[V_\psi(s_{t+1})\right] \tag{5}$$

4. Methodology

In this section, we outline the methodology. Initially, we investigate the LSTM battery prediction, followed by the elucidation of our HRL framework and its integration into the system.

4.1. LSTM Battery Prediction

Predicting battery consumption accurately is critical for optimizing the aerial robot's path planning and mission execution. LSTM networks, known for their ability to capture long-term dependencies and temporal patterns in sequential data [59], are particularly effective for this task. Recent advancements in LSTM applications, such as those demonstrated by [60,61], highlight their effectiveness in energy consumption prediction.

Our proposed model for battery consumption prediction incorporates a bidirectional LSTM (Bi-LSTM) framework, as illustrated in Figure 2. A Bi-LSTM layer is a neural network architecture that extends the traditional LSTM networks by introducing a dual structure to process data sequences in both forward and backward directions simultaneously [62]. This bidirectional approach allows the network to capture temporal dependencies from both past (backward) and future (forward) states, thereby enhancing its ability to understand context and sequence dynamics.

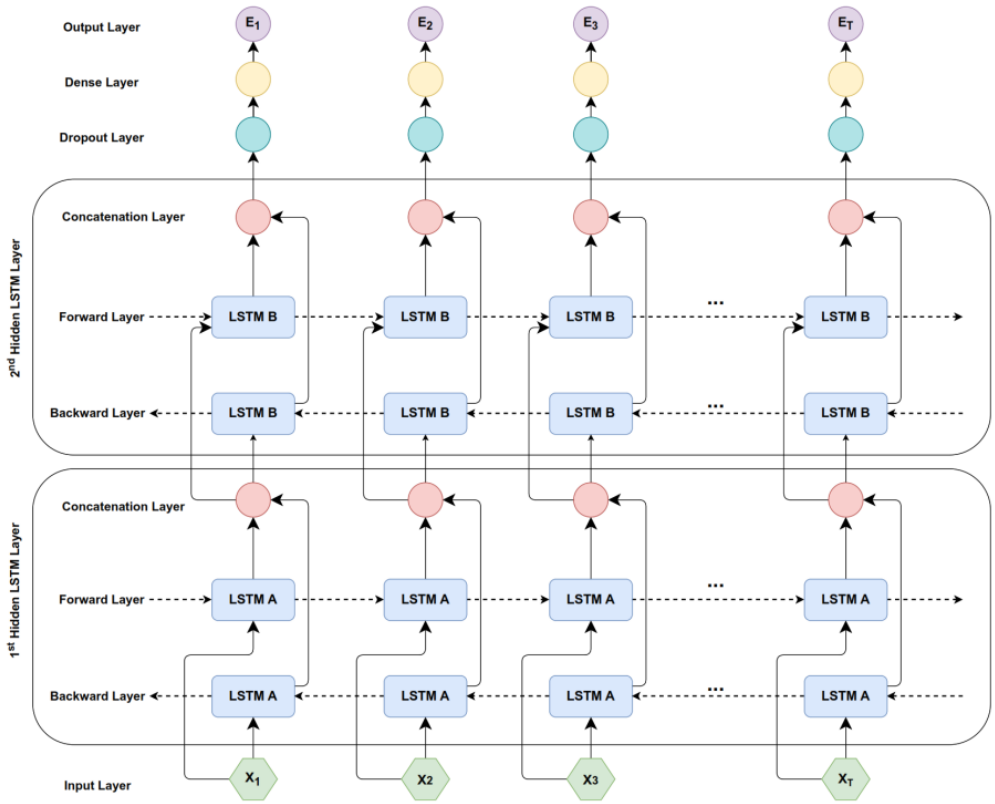

Figure 2. The proposed LSTM architecture for aerial robot energy consumption prediction.

Each Bi-LSTM comprises two separate LSTMs, each processing the data in opposite directions—one forward and one backward—with their respective parameters and hidden states, which are then concatenated at each time step that leads to a more robust representation of temporal sequences. Bi-LSTM layers are particularly effective in tasks where the context of the entire sequence is crucial for accurate predictions [63]. The incorporation of two Bi-LSTM layers in our LSTM architecture enhances the model's capacity to capture and interpret complex temporal dependencies in sequential data. The first Bi-LSTM layer effectively captures immediate, short-term temporal patterns, while the subsequent layer, receiving the output of the first, can discern more abstract, higher-level temporal relationships in the data. This ability makes the model particularly suited for complex sequential modeling tasks where understanding both past and future contexts is vital [64].

The input layer includes aerial robot-specific operational parameters including aerial robot velocity (v), payload weight (P_W), historical battery measurements (B_{h_i}), current battery level (B_c), and the position of the aerial robot. These parameters heavily influence battery consumption in aerial robot operations. It is important to note that our analysis does not consider flight and environmental conditions, like motor efficiency and wind, and it is assumed that the altitude is constant. Additionally, the model's scope is limited to cruising phases of aerial robot operation, excluding aspects of takeoff and landing maneuvers.

Following Bi-LSTM layers, a dropout layer is designed to counteract overfitting. The dropout rate serves as a hyperparameter, adjustable during the model's validation phase for optimal performance tuning. After the data pass through the dropout layer, they are then processed by a dense layer. This dense layer serves to further interpret the features extracted by the bidirectional LSTM layers and to consolidate the information into a form that is suitable for output prediction. This dense layer uses a ReLU activation function and has 128 cells.

The output layer uses a linear activation function to predict the upcoming battery level and determines the battery requirements necessary for reaching each target. Each of the LSTM layers contains 128 hidden cells. The input data are processed in time windows of a length of 20, and they are matched with the T_C, which ensures the model's predictions are synchronized with the HLC updates.

Regarding the mathematical formulation of the proposed LSTM model, we define an input vector X_t in $\mathbb{R}^N$ for each time instant t, which includes N key inputs. The aerial robot's energy consumption at a time t is represented as $E_t \in \mathbb{R}$. Our goal is to develop a predictive function $F(\cdot)$ that can accurately predict energy consumption across a specified time window. The core of our analysis involves minimizing the total squared differences between the actual (E_t) and predicted ($\hat{E}_t = F(X_t)$) energy consumption. This objective, aimed at enhancing the accuracy of energy usage predictions for aerial robot operations, is formalized by optimizing the function $\sum_{t \in T} \left(E_t - \hat{E}_t\right)^2$, ensuring F accurately predicts E_t from X_t.

For this problem, the mean squared error (MSE) is employed as the loss function and the Adam optimizer is selected as an optimizer. The hyperparameters are shown in Table 1. The hyperparameters were chosen based on the minimum averaged MSE as these parameters were identified to yield optimal results during the 5-fold cross-validation phase.

Table 1. Hyperparameters for the LSTM model.

Parameter	Value
Hidden units per layer	128
Learning rate	0.001
Batch size	128
Number of epochs	50
Dropout rate	0.5
Optimizer	Adam
Input features	Battery level, velocity, distance, time duration, position

4.2. Proposed Hierarchical Reinforcement Learning Framework

4.2.1. Goal-Conditioned Hierarchical Reinforcement Learning

This framework employs a two-level policy structure where the HLC, represented as π_{HLC}, determines goals based on the environmental state and the LLC, represented as π_{LLC}, scores actions to achieve these goals. The HLC is responsible for setting feasible goals for the LLC. One important difference between the HLC and LLC learning process is that the LLC generates transitions $(g_t, s_t, a_t, r_t, s_{t+1})$ at every single time step through the primitive aerial robot actions, while the transitions of HLC $(G_t, s_t, p_t, r_t, G_{t+T_c}, s_{t+T_c})$ are produced over a slower time scale through a sequence of goal selections.

The HLC is tasked with the strategic selection of intermediate objectives, referred to as feasible goals (G_t), based on the agent's current environmental state (s_t). These goals are set within the domain of the goal space G, which, in our study, includes locations of multiple targets.

Transition dynamics in the HRL framework are critical. They define how the agent moves from one goal to the next. Mathematically, this transition is represented as:

$$g_{t+1} = \pi_{HLC}(s_{t+1}),\ for\ t \equiv 0\ (mod\ T_c) \tag{6}$$

Equation (6) denotes the transition mechanism from one high-level goal to the next, occurring at intervals of c time steps. Here, c represents the frequency of updates from the HLC, indicating how often the high-level goals are reassessed and potentially altered based on the evolving state of the environment.

Between these transition points, the high-level goal adaptation function, $\eta\left(s_t, g_t, s_{\{t+1\}}\right)$ comes into play, defined as:

$$\eta(s_t, g_t, s_{t+1}) = \Delta f(s_t, g_t) - \Delta f(s_{t+1}, g_t) \tag{7}$$

where Δf represents the differential mapping from the state-goal pairs to the goal space G.

The LLC π_{LLC} undertakes the execution of these high-level objectives by generating a sequence of actions a_t in response to the current state s_t and high-level goal g_t. The efficacy of these actions is evaluated based on the feedback from the environment, encapsulated in the reward signal $r(s_t, a_t)$. The LLC performs immediate goal-conditioned actions a_t based on the current state s_t and the goal G_t, with actions optimized at every time step. The low-level policy aims to achieve the set goals within the c-step timeframe, guided by an intrinsic reward function. This process will continue until the target is reached or one of three scenarios occurs: either an obstacle collision occurs or the aerial robot is unable to reach the target within a predefined maximum number of time steps or the aerial robot reaches the required charge for the goal.

HLC and the LLC receive rewards separately while interacting with the environment. The HLC receive extrinsic reward from the environment for choosing the best goal based on its states for the LLC. The LLC receive intrinsic rewards conditioned by the goal specified by the HLC. The HLC optimizes policy π_{HLC} to select goals G_t based on the state s_T, aiming to maximize a combination of expected cumulative rewards and policy entropy. The reward $r_{t_{HLC}}$ for the HLC incorporates factors like mission success and the feasibility of goals considering the predicted battery consumption. The reward function for π_{HLC} is formulated to accumulate rewards over a fixed interval of c time steps. Equation (8) describes the reward function for the HLC. It accumulates the rewards over c time steps, with γ representing the discount factor, a measure of the importance of future rewards.

$$R_{HLC}(s_t, g_t) = \sum_{i=0}^{c-1} \gamma^i r(s_{t+i}, a_{t+i}) \tag{8}$$

The primary goal of the high-level policy is to optimize the expected cumulative reward, which is expressed as:

$$G(\pi_{HLC}) = E\left[\sum_{t=0}^{\infty} \gamma^t R_{HLC}(s_t, g_t)\right] \tag{9}$$

In contrast, the low-level policy aims to achieve the set goals within the c-step timeframe, guided by an intrinsic reward function. This function is designed to maximize the expected return related to the achievement of the intermediate goals:

$$G(\pi_{LLC}) = E\left[\sum_{t=0}^{c} \gamma^t R_{LLC}(s_t, g_t, s_{t+1})\right] \tag{10}$$

4.2.2. Soft Actor-Critic for Proposed Hierarchical Reinforcement Learning Augmented with LSTM Battery Prediction

In this framework, we employ the off-policy RL algorithm SAC to train policies at both the HLC and LLC. The SAC optimizes the policy for selecting targets that maximize mission success and efficiency at the HLC layer and that are operatively focused. Also, the LLC employs SAC to optimize the aerial robot's navigation toward the chosen targets and manage specific tasks efficiently.

The HLC is responsible for making decisions on selecting reachable targets, considering states and the predicted battery for each target from the LSTM. The LSTM model predicts battery consumption for different targets to help the HLC policy choose feasible goals for the LLC. The LSTM's predictions are integrated into the HLC framework as part of the state information, aiding the HRL model in making more informed decisions, especially during the early stages of training when it has not yet learned to effectively predict battery usage.

The critic loss function for the HLC using SAC is defined as:

$$L_Q^{HLC}(\theta_{HLC}) = \mathbb{E}_{(G_t, s_t, p_t, r_{t_{HLC}}, G_{t+T_c}, s_{t+T_c} \sim D_{HLC})}\left[\frac{1}{2}\left(Q_{\theta_{HLC}}(s_t, G_t) - \hat{Q}_t^{HLC}\right)^2\right] \tag{11}$$

where $Q_t^{HLC} = r_t^{HLC} + \gamma^{HLC}\mathbb{E}\left[V_{\theta^-_{HLC}}(s_{t+T_c})\right]$, and D_{HLC} is the replay buffer for HLC, containing transitions over a slower time scale through sequences of goal selections. Moreover, the actor loss function for the HLC is given by:

$$L_\pi^{HLC}(\phi_{HLC}) = \mathbb{E}_{(s_t \sim D_{HLC})}\left[\left[log\left(\pi_{\phi_{HLC}}(G_t \mid s_t)\right) - \frac{1}{\alpha} Q_{\theta_{HLC}}(s_t, G_t)\right]\right] \tag{12}$$

The LLC is tasked with implementing objectives delineated by the HLC, operating autonomously to fulfill these specified goals. It is presumed that the goals set by the HLC are optimal, guiding the LLC's policy to adapt its actions toward efficient goal achievement. The LLC's primary responsibility includes navigating the aerial robot toward its objectives while managing its battery efficiently and navigating around obstacles, embodying a direct application of the high-level strategy to operational actions. The HLC objectives for the LLC persist until the LLC either fulfills these objectives or encounters conditions that necessitate an alternative approach, as mentioned before.

We enhance the decision-making process of the LLC by integrating LSTM-based predictions of battery consumption into the actor network's objective function. This integration allows the LLC to make more informed decisions that not only aim to achieve operational objectives but also optimize for energy efficiency. The key idea is to leverage the LSTM's capability to predict the battery consumption associated with different actions in given states, thus enabling the aerial robot to prefer actions that are energy efficient. The LLC optimization incorporates LSTM predictions to adjust actions for energy efficiency and LLC Objective Function integrates the LSTM's prediction of battery consumption, adjusting the

SAC objective to balance mission success, policy entropy, and predicted battery efficiency. Thus, the LLC actor loss function is defined as:

$$L_{\pi}^{LLC}(\phi_{LLC}) = \mathbb{E}_{(g_t,s_t)\sim D_{LLC}}\Big[\Big[\alpha log\big(.\,\pi_{\phi_{LLC}}\,(a_t \mid s_t, g_t)\big) - Q_{\theta_{LLC}}(s_t, a_t, g_t) + \lambda \cdot LSTM_{cost_{(a_t,s_t)}}\Big]\Big] \quad (13)$$

where $LSTM_{cost_{(a_t,s_t)}}$ represents the predicted battery consumption cost for acting a_t in state s_t, predicted by the LSTM. λ is a weighting factor that balances the importance of energy efficiency (battery usage optimization) against other objectives within the LLC's loss function. This formulation explicitly integrates LSTM-based predictions into the LLC's objective function, allowing for the aerial robot to make informed decisions that optimize both operational objectives and energy efficiency.

Critic loss for LLC is defined as follows:

$$L_{Q}^{LLC}(\theta_{LLC}) = \mathbb{E}_{(s_t,a_t,g_t,r_t,s_{t+1})\sim D_{LLC}}\Big[\big(r_t + \gamma\mathbb{E}\big[V_{\theta^-{}_{LLC}}(s_{t+1}, g_t)\big] - Q_{\theta_{LLC}}(s_t, a_t, g_t)\big)^2\Big] \quad (14)$$

The architecture and hyperparameters of the proposed hierarchical reinforcement learning model are detailed in Table 2.

Table 2. Proposed hierarchical reinforcement learning hyperparameters.

Parameter	Value
Learning rate (HLC, LLC)	0.0001, 0.0003
Actor/Critic L2 regularization factor	0.00001
Batch size	64
Replay buffer size	1,000,000
Discount factor (γ)	0.99
Target update interval	5
τ	0.005
Actor network	2 layers (256 units each)
Critic network	2 layers (256 units each)
Temperature parameter (α)	Start with 0.2
Activation function	ReLU

4.2.3. Switching Mechanism in the HLC

In our proposed framework, the HLC initially leverages LSTM predictions for battery consumption to inform its goal-setting and decision-making processes. This reliance on LSTM outputs facilitates early-stage learning, enabling the HLC to make informed decisions with limited experience. As the HLC interacts more with the environment, it starts learning from the environment experiences. It gradually understands the dynamics of battery consumption in relation to various actions and environmental conditions. With increasing knowledge, the reliance on the LSTM estimator decreases.

Therefore, as training progresses, our framework incorporates an adaptive switching mechanism that dynamically adjusts the reliance on LSTM predictions based on their accuracy compared with the model's internal decision-making processes. The HLC gradually transitions to a more autonomous decision-making model when its internally generated predictions or decisions consistently exhibit lower errors compared with the LSTM's predictions. This transition underscores the HLC's capability to internalize and surpass the LSTM model's predictive accuracy through continuous learning, ultimately achieving enhanced autonomy and efficiency in mission planning and execution.

This shift is governed by comparing the temporal difference (TD) error, δ_{TD}, against the LSTM prediction error $\delta_{LSTM} = \frac{1}{N}\sum_{i=1}^{N} B_{est}^{i} - B_{C}^{i}$ where N is the number of instances or episodes considered for the evaluation, B_{est}^{i} is the LSTM model's predicted battery level for the i^{th} instance, and B_{C}^{i} is the corresponding currently observed battery level. The policy switch occurs when $\delta_{TD} < \delta_{LSTM}$ and x episodes have passed.

The TD error is defined as $\delta_{TD} = r_t + \gamma V(s_{t+1}) - Q(s_t, a_t)$. Here, r_t is the reward at a time t; $V(s_{t+1})$ is the value function of the next state, estimated by the critic network,

representing the expected return; $Q(s_t, a_t)$ is the action-value function output by the critic network for the current state-action pair, indicating the expected return of taking action in the state; and γ is the discount factor, weighing the importance of future rewards. This adaptive mechanism ensures that the HLC enhances its autonomy and efficiency in mission planning, optimizing energy usage based on real-time learning and experiences.

4.3. Proposed HRL Modeling

4.3.1. HLC Network

The state space of the HLC is designed to navigational and operational metrics, essential for optimizing aerial robot mission strategies. We define the state space S_{HLC} as:

$$S_{HLC} = [D, \theta, B_c, P_W, v, B_{est}] \tag{15}$$

where D quantifies the Euclidean distances to each mission target and the recharge station from the aerial robot's current coordinates, enhancing route planning efficacy, and θ represents the bearings from the aerial robot to these points of interest, crucial for directional guidance. In addition, the current battery level B_c, and the payload weight P_W,, directly influence the aerial robot's flight dynamics and energy consumption. The aerial robot's velocity vector $v = (v_x, v_y, v_z)$ and the predicted battery requirements B_{est} for reaching each target, as predicted by the LSTM model, complete the state space definition, facilitating informed, energy-efficient decision-making.

The action space A_{HLC} for the HLC is defined as:

$$A_{HLC} = [\mathbb{T}, \Delta B_{HLC}] \tag{16}$$

where $\mathbb{T}$ denotes the selection action for the next target point, and ΔB_{HLC} specifies the anticipated battery charge required to reach the selected target. This action space enables the HLC to dynamically choose between progressing toward mission targets or recharging, contingent upon the current operational state and energy requirements. In addition, to enhance the reliability of our proposed framework, we incorporate a safety margin into the battery consumption prediction model. This safety margin accounts for uncertainties, such as unexpected maneuvers to avoid obstacles, which might impact the aerial robot's energy consumption. After the initial 1000 episodes of training, this safety margin is manually adjusted based on observed performance and environmental interactions.

The reward function for the HLC, $R_{HLC} = \sum_{i=1}^{4} R_i$, is designed to encapsulate this layer's objectives, promoting actions that enhance mission success while conserving energy and ensuring safety:

***Target Achievement Reward* (R_1)**: The primary component of the HLC's reward function is designed to incentivize the selection of reachable targets and efficient resource utilization:

$$R_1 = R_{H_1} - (w_1 \cdot D + w_2 \cdot B_{est}) - D_t \tag{17}$$

where R_{H_1} represents the base reward for targeting a new goal. D_e is the Euclidean distance to the selected target, and B_{est} denotes the predicted battery usage to reach the target. Moreover, w_1 and w_2 are scaling factors that adjust the influence of distance and battery usage on the reward. D_t is a traveled distance by the aerial robot, which helps choose the optimal path. In scenarios where a target is deemed not reachable due to goal feasibility, a high penalty of R_{H_2} is applied to discourage the selection of targets.

***Successful Delivery Reward* (R_2)**: Upon successful delivery of a payload to a target, the HLC is awarded a significant reward ($R_2 = R_{H_3}$) for each target.

***Mission Efficiency Bonus* (R_3)**: To further incentivize the completion of the mission with minimal energy expenditure, an additional bonus ($R_3 = R_{H_4}$) is awarded if all targets are served without the need for recharging.

Recharging Decision Component **(R_4)**: The HLC's decision to recharge the aerial robot's battery is also factored into the reward function:

$$R_4 = \begin{cases} R_{H_5} & \text{Aerial robot recharges when critically low on battery.} \\ -R_{H_7} & \text{Aerial robot fails to recharge when critically low on battery.} \\ -R_{H_8} & \text{aerial robot recharges unnecessarily.} \end{cases} \tag{18}$$

4.3.2. LLC Network

The state space of the LLC, denoted as S_{LLC}, encapsulates the aerial robot's operational parameters necessary for executing navigational tasks toward predefined goals. Formally, we define S_{LLC} as:

$$S_{LLC} = \left[D, \theta, v_x, v_y, v_z, L_i, B_{HLC},\ B_c\right] \tag{19}$$

where D represents the aerial robot's distance to the target from GPS coordinates; θ denotes the orientation toward the goal; v_x, v_y, v_z are the velocity components from the IMU, L_i encapsulates the lidar data for obstacle detection; B_{HLC} is the predicted battery required to reach the goal from HLC; and B_c is the current battery.

The LLC leverages environmental data acquired through LIDAR distance sensors, characterized by a scan angle range of π radians. the horizontal plane is monitored using seven sensors, with an angular resolution of $\pi/6$ radians between adjacent rays. The lidar range consider 50 m. This configuration ensures a detailed spatial awareness, facilitating the aerial robot's ability to navigate complex environments by detecting and avoiding obstacles.

The LLC's action space is meticulously defined to accommodate precise control over the aerial robot's navigational and energy management strategies. The action space, denoted as A_{LLC}, is normalized to a range of -1 to 1 for both speed adjustments and yaw angle modifications $\left[a_v, a_\psi\right]$.

The LLC is responsible for the execution of navigation and operational tasks. The intrinsic reward function for the LLC, $R_{LLC} = \sum_{i=1}^{7} R_i$, is designed to encourage efficient path execution and safe navigation:

Proximity to Target Reward **(R_1)**: This reward increases as the aerial robot moves closer to the target, encouraging the reduction in the distance to the goal, using the difference in distances between consecutive states, defined as:

$$R_1 = d(s_t) - d(s_{t+1}) \tag{20}$$

where $d(s_t)$ is the distance to the target from the current state and $d(s_{t+1})$ is the distance to the target from the next state.

Target Reach Reward **(R_2)**: A reward ($R_2 = R_{L_1}$) given when the aerial robot reaches the target where R_{L_1} is a large positive value.

Efficient Path Penalty **(R_3)**: A constant penalty ($R_3 = -R_{L_2}$) for each time step taken to encourage time efficiency in reaching the target where R_{L_2} is a small positive constant.

Energy Efficiency Reward **(R_4)**: This reward encourages energy-efficient navigation, given by

$$R_4 = \frac{(B_C - B_{HLC}) \cdot (B_{est} - B_c)}{B_{total}} \tag{21}$$

where B_{HLC} is the battery at the target from high-level, B_{est} and B_c are the predicted values from LSTM for the state and current battery level, and B_{total} is the total battery capacity.

Battery Threshold Penalty **(R_5)**: A penalty ($R_5 = R_{L_3}$) is considered if the battery level (B_C) falls below a certain threshold where R_{L_3} is a penalty reward.

Obstacle Avoidance Penalties **(R_6 and R_7)**: A safety zone penalty ($R_6 = R_{L_4}$) is applied when entering the safety zone around obstacles to encourage maintaining a safe distance. Moreover, a substantial collision penalty ($R_7 = R_{L_5}$) for colliding with obstacles is considered, with $R_{L_5} > R_{L_4}$ to reflect the higher severity of collisions.

4.4. Hindsight Experience Replay

One of the challenges in deploying HRL for aerial robot navigation is the non-stationary nature of the environment, which stems from dynamic changes in the goals set by the HLC. These changes can cause the optimal policy to shift over time, making it difficult for the LLC to consistently achieve its assigned goals.

To address this challenge, we incorporate HER into our framework. HER is particularly adept at mitigating the effects of non-stationarity by allowing the LLC to learn from both successful and unsuccessful attempts, effectively turning failures into valuable learning opportunities. HER enables the LLC to reinterpret previously unsuccessful attempts at reaching a goal as successful outcomes toward alternative goals, fostering adaptive learning from every mission scenario. This approach enhances the LLC's capability to adjust its strategy in response to changing environmental conditions and mission parameters, ensuring continuous improvement in the operational efficiency and decision-making process

To implement it, we begin with the LLC, denoted here as Algorithm LLC, and ensure the replay buffer D_{LLC} is initially empty. At the onset of each episode, an initial state s_0 and a goal g are randomly selected from their respective spaces, S and G. During the episode, spanning environment steps $t = T_c$, the LLC engages with the environment to produce transitions $(g_t, s_t, a_t, r_t, s_{t+1})$.

Upon completion of these steps, HER examines the sequence of states traversed, represented as $\zeta = \{s_0, s_1, ..., s_T\}$. Utilizing this sequence, HER proceeds to populate the replay buffer D_{LLC} with each transition (s_t, a_t, r_t, s_{t+1}), alongside the initial goal G. In a subsequent step, HER enriches the dataset by appending additional transitions $(s_t, a_t, r'_t, s_{t+1})$, each associated with a new goal $g' \in \phi$ where $\phi = [g'_1, g'_2, ..., g'_m]$ consists of m novel goals selected uniformly from the encountered states $\zeta = [s_0, s_1, ..., s_T]$.

This enhancement process allows HER to offer the LLC agent augmented rewards ${r_t}' = r(s_t, a_t, g')$ for each new goal g', irrespective of the original goal G's achievement status. Therefore, HER boosts the LLC agent's efficiency in learning and its capability to successfully attain goals, broadening the agent's exposure to a variety of potential scenarios.

When the aerial robot fails to reach its designated goal due to battery exhaustion, HER recalibrates the learning objective based on the actual operational outcome. It identifies the furthest point e_1 reached within the battery's capacity as a new achievable goal ${g_1}'$. This relabeling process includes updating the experience tuple to reflect the current battery B_c and correlating it with the traveled distance D_t. In instances where the aerial robot's mission is compromised due to an inability to successfully navigate around obstacles, HER adapts by selecting the point of failure e_2 as a new goal g'_2. For scenarios where the aerial robot does not fulfill its objective within the allocated number of steps, HER intervenes by marking the endpoint reached within the step constraint E as the revised goal g'_3.

5. Experimental Setup and Results

In our study, we conducted an experimental analysis to assess the performance of our proposed hierarchical reinforcement learning framework. Our evaluation is twofold: initially, we verified the accuracy of the pre-trained LSTM model, followed by an assessment of our HRL approach's effectiveness. We compare our model's performance against standard soft actor-critic and soft actor-critic augmented with HER models and hierarchical actor-critic (HAC) [65] to showcase the improvements our framework offers.

5.1. Simulation Environment and Setup

Simulations were performed using MATLAB 2023, leveraging a preconfigured aerial robot model from Simscape within the UAV Toolbox. This setup allowed us to accurately replicate the flight dynamics and battery consumption of aerial robots in a simulated $400 \times 400 \times 50$ m environment. The aerial robot's start point is set constant, and the recharging station is located at the starting position. Missions tasked the aerial robot with reaching randomly placed survivor positions in each trial, flying at speeds up to 20 m/s. Details on the aerial robot model can be found in Table 3.

Table 3. Parameters of the aerial robot employed in simulations.

Parameter	Value
Planar dimensions	12.5 m × 8.5 m
Battery capacity	7.6 × 3 Ah
Mass	1.2726 kg
Propeller diameter	0.254 m
Motor max torque	0.8 Nm
Motor max power	160 W

In our simulation framework, we consider the aerial robot's height constant during the mission to simplify the problem to a 2-dimensional space. This assumption allows us to focus on optimizing horizontal navigation and battery consumption without the added complexity of 3-dimensional movement.

5.2. Performance of LSTM Predictor

In this section, we detail the outcomes of our experiments with the LSTM-based model for predicting battery consumption. Our goal is to demonstrate the model's predictive accuracy using the dataset [66]. The dataset contains energy usage data for 100 commercial aerial robots; these data were documented over 195 test flights. These flights varied in terms of payload, velocity, and elevation. Data collection encompassed distinct attributes, capturing details from the aerial robot's current battery, GPS, and IMU. We selected the parameter set that minimized the average mean squared error (MSE), identifying it as the optimal hyperparameter for our model.

Table 4 provides the hyperparameters and corresponding evaluations. Based on the results of Table 4, our analysis selected the hyperparameter set with a 0.5 dropout rate, a 0.001 learning rate, and a batch size of 128 due to its predictive accuracy, as evidenced by the lowest average RMSE and average MSE. Figure 3 shows the performance and validation of the LSTM based on the corresponding optimal hyperparameters. It illustrates the performance of the proposed LSTM architecture in data prediction.

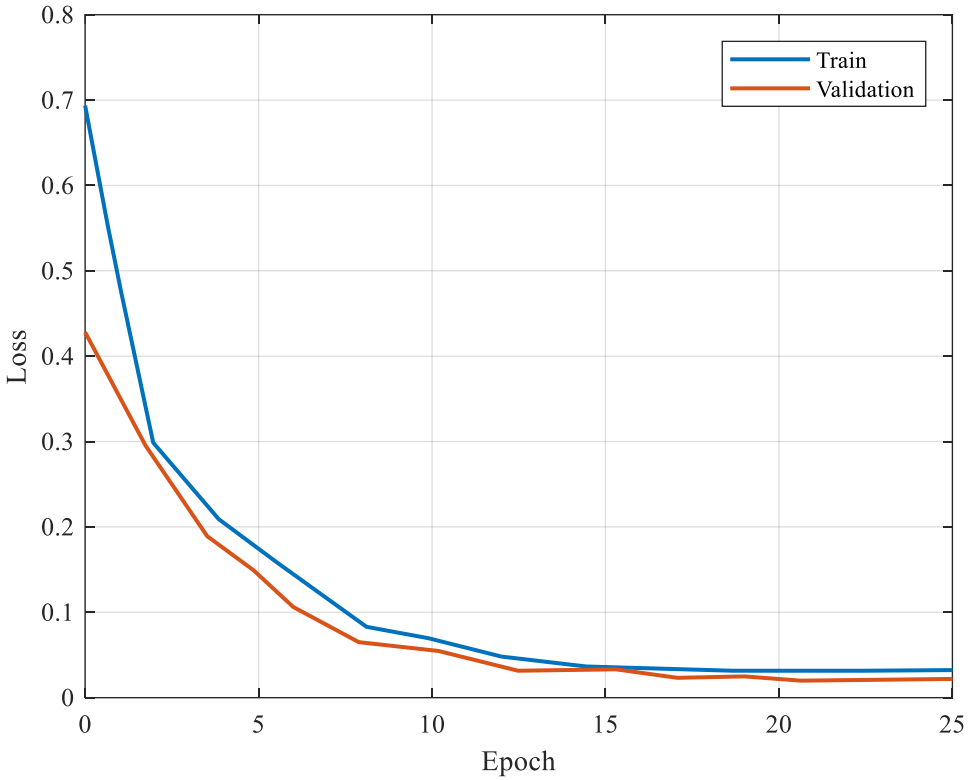

Figure 3. Validation loss curves regarding the optimized dropout and batch size.

Table 4. Analysis of hyperparameters and their impact on model performance.

Dropout	Learning Rate	Batch Size	Average MSE	Average RMSE
0.2	0.001	128	5.523	45.324
0.2	0.0001	128	5.796	47.256
0.5	0.001	128	5.404	43.195
0.5	0.0001	128	5.631	44.237
0.5	0.01	128	6.884	59.462

Figure 4 presents the LSTM model's performance in comparison to the ground truth data, focusing on the actual aerial robot battery consumption without accounting for wind effects. The evaluation of the model's accuracy was based on the MSE between the predicted and actual battery levels. The results reveal that the LSTM model achieves high precision on the testing dataset, exhibiting a minimal discrepancy of just 4.503 watts from the true energy consumption.

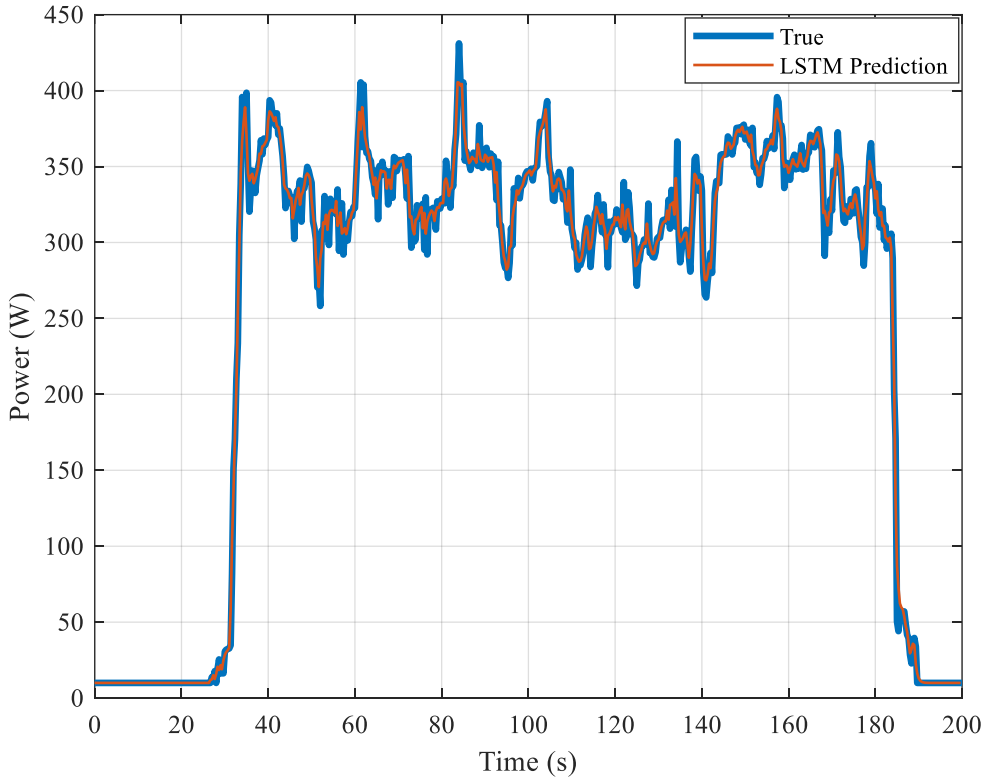

Figure 4. Comparison of the actual aerial robot battery consumption without environmental change with proposed LSTM-based architecture prediction.

5.3. Training Result

To evaluate the efficacy of our proposed algorithm in aerial robot path planning, we designed a simulation framework within a MATLAB environment, encompassing a spatial domain of 400 × 400 × 50 m. The simulation environment is dynamically configured for each training episode, with the aerial robot's initial position randomly chosen from the environment's corners. This setup is further complicated by the presence of 20 cylindrical obstacles randomly distributed across the space, each with a 7 m radius and a 50 m height, to mimic navigational challenges. Also, the aerial robot should deliver variable payloads (25, 25, 50, 75 g) to four randomly positioned targets.

The high-fidelity simulated environment generates a large and diverse set of training samples by replicating various scenarios. To enhance the robustness and generalization of

our model, we employ domain randomization techniques, systematically varying aspects such as target locations.

To ensure learning and convergence, we set the maximum number of training episodes to 10,000, with each episode capped at 500 time steps. The hierarchical decision-making process is governed by a temporal goal-setting interval (T_c) of 20 steps to strike a balance between goal feasibility and computational time.

In all simulated environments, we assume that the aerial robot's battery is sufficiently charged to complete deliveries to all targets without the need for interim recharging provided the operation is executed with energy efficiency. In all the environments, a safety margin of 0.5 around each obstacle is considered.

Our comparative analysis, illustrated in Table 5, Figures 5 and 6, positions our algorithm against SAC, SAC integrated with HER, and HAC with SAC. Uniformity across trials was maintained by considering three random seeds for each algorithm. Figure 5 illustrates the cumulative average reward, with solid curves representing the mean performance and shaded areas highlighting the variability across different seeds. This graph underscores the superiority of our algorithm, showcasing its excellence in convergence efficiency and path planning effectiveness. The distinct advantage of our algorithm is attributed to a multifaceted approach:

Table 5. Comparison of success and collision rates at 2000, 5000, and 10,000 episodes for proposed HRL, HAC, SAC, and SAC augmented HER.

Metric/Training Phase	Proposed HRL	HAC+SAC	SAC	SAC+HER
Success rate (%)—After 2000 episodes	62.5	56.8	40.1	52.4
Success rate (%)—After 5000 episodes	83.1	72.3	54.2	68.4
Success rate (%)—After 10,000 episodes	92.4	84.7	62.2	79.3
Collision rate (%)—After 2000 episodes	22.8	27.1	31.4	24.2
Collision rate (%)—After 5000 episodes	11.7	15.4	19.2	17.4
Collision rate (%)—After 10,000 episodes	5.2	8.1	13.5	9.3

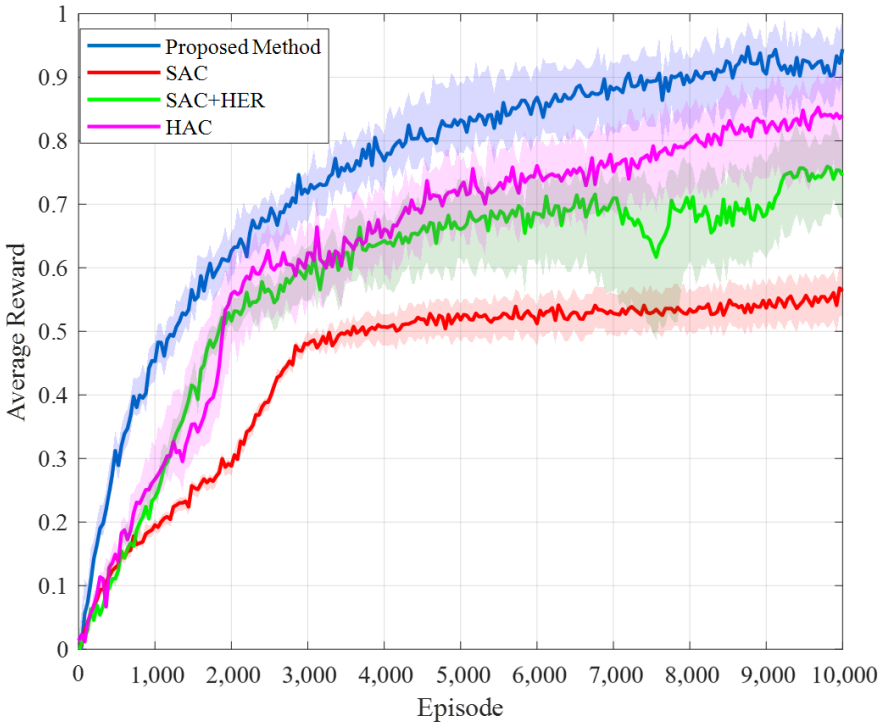

Figure 5. The comparison of the average reward of different methods.

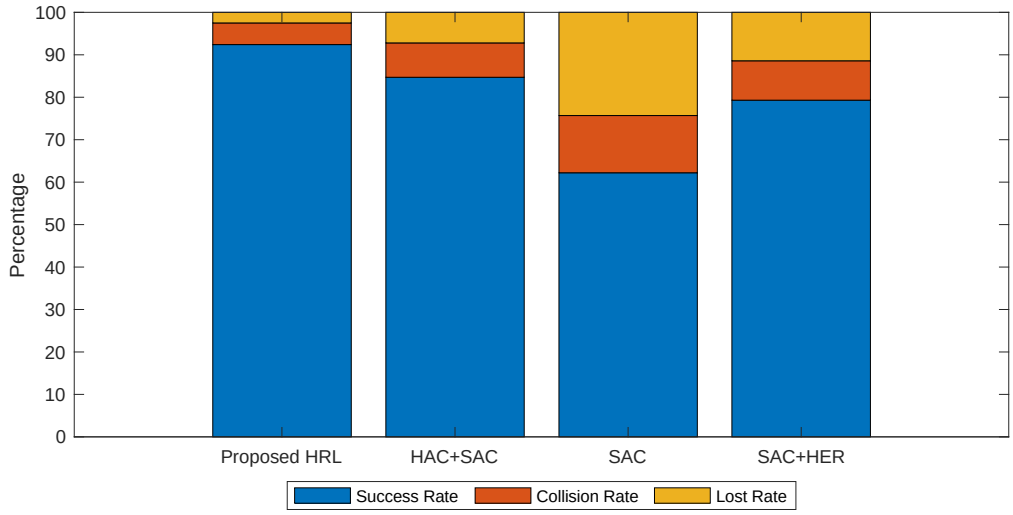

Figure 6. Comparison of success rates after 10,000 episodes across different algorithms during the training phase.

Hierarchical Task Decomposition: Simplifying the complex aerial robot path planning into a structured hierarchy of sub-tasks expedites the learning process. This strategic segmentation allows for quicker adaptation and efficient resolution of navigational challenges.

LSTM-Enhanced Goal Setting: Integrating LSTM networks enables dynamic and realistic goal setting by the high-level policy, considering the aerial robot's state and immediate environmental context. This not only accelerates convergence but also minimizes the risk of mission failure by ensuring that goals are both relevant and attainable.

Stability through Hindsight Experience Replay: The incorporation of HER plays a crucial role in enhancing the stability of the low-level controller amidst the dynamic goal adjustments from the high level. By reinterpreting past experiences under new goals, HER enables the low-level controller to maintain a consistent learning trajectory, thus significantly bolstering stability and ensuring robustness against the variability in high-level goal setting.

Adaptive Goal Flexibility: HER also aids in adapting the aerial robot's behavior to sudden changes in goals, ensuring that the learning process remains unaffected by the high-level policy's dynamic decisions. This adaptability is crucial for maintaining performance consistency across varied and unpredictable scenarios.

Training results show a higher success rate, accelerated convergence, and enhanced stability in aerial robot path planning tasks compared with established benchmarks.

Figure 7 shows the energy efficiency of our proposed method in the training environment in comparison to other methods. The LSTM-based prediction model anticipates future battery requirements, so our system can make informed decisions that optimize energy use. The hierarchical structure of our framework enables control over aerial robots' missions. This allows for mission planning that inherently prioritizes energy efficiency, from the macro selection of mission objectives to the micromanagement of in-flight maneuvers. The integration of HER into our framework enhances the aerial robot's ability to learn from past experiences, including previous energy expenditures. This learning process fine-tunes the aerial robot's decision-making, steering it toward more energy-efficient strategies over time. Moreover, by adapting to the outcomes of past missions, our system continuously refines its understanding of energy-optimal behaviors. A critical aspect of our system's design is the frequent update mechanism within the hierarchical decision-making process. By adjusting goals and paths at regular intervals based on real-time data and predictive insights from the LSTM model, the aerial robot can make corrections to its flight plan that avert inefficient energy use. This dynamic adaptability reduces the likelihood of scenarios that necessitate regular recharges, ensuring that the aerial robot's energy reserves are utilized for prolonged operational periods.

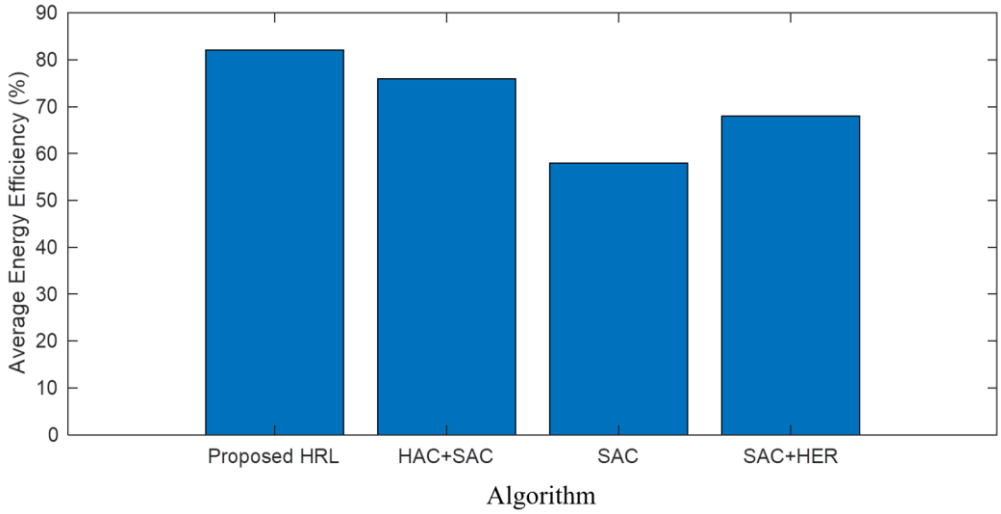

Figure 7. Comparative analysis of energy efficiency between the proposed algorithm and benchmark algorithms during the training phase.

5.4. Test Result and Discussion

After training in the training environment, we tested the proposed algorithm against benchmark algorithms and algorithms in the literature, including, SAC, SAC integrated with HER, and HAC in the following environments to evaluate the performance of the proposed algorithm. All test environments have the same dimensions as the training environment.

To evaluate the performance of each algorithm in the test environments, for each environment and algorithm, we observed the following performance metrics: the success rate, which represents the aerial robot's ability to successfully deliver to all the targets after 2000 episodes; the collision rate, which indicates instances where the aerial robot collides with an obstacle; and the average rewards. Additionally, since the primary focus of this paper is path planning, we primarily illustrate the paths generated by the proposed method. Table 6 presents the results for each environment, comparing our proposed method to other algorithms in terms of success rate, collision rate, and average rewards.

Environment 1: In this scenario, the aerial robot maneuvered through 50 cylindrical obstacles, while delivering uniform payloads of 25 g to each of the three targets and there was no need to recharge. The path generated by the proposed HRL is shown in Figure 8. The proposed HRL algorithm recorded a success rate of 90.1%. This level of performance highlights the algorithm's exceptional ability to navigate through densely populated obstacle spaces with robust obstacle avoidance and energy management capabilities. Additionally, a lower collision rate of 8.1% attests to its precision in ensuring safe navigation. The average reward metric further underscores the overall mission efficiency of the algorithm. Notably, the proposed HRL algorithm maintains a high success rate even in environments laden with a greater number of obstacles, whereas other methods exhibit a more pronounced decline in success rates. Moreover, it gives the lowest loss rate among the compared methods.

Environment 2: In this scenario, the aerial robot maneuvered through 30 cylindrical obstacles while delivering uniform payloads of 25 g to each of the three targets. As the obstacle density decreased in the second environment, the proposed HRL algorithm not only maintained but also improved its success rate to 95.3%. This improvement reflects the algorithm's adaptability to varying environmental complexities, showcasing an enhanced ability to optimize routes and manage payloads effectively. The collision rate further dropped to 3.5%, indicating an even more pronounced advantage in navigating with safety and precision. The generated path by the proposed method in Environment 2 is shown in Figure 9.

Environment 3: In this configuration, the experiment incorporated 30 cylindrical obstacles, maintaining a uniform payload and featuring five targets.

Table 6. The comparative analysis of performance across test environments 1, 2, 3, and 4 for the proposed algorithm versus SAC, HAC, and SAC augmented with HER.

Environment	Proposed HRL	SAC	HAC+SAC	SAC+HER
		SR (%)		
E 1	90.1	59.2	81.1	75.4
E 2	95.3	64.1	85.5	82.2
E 3	93.6	61.7	82.2	79.7
E 4	89.4	55.4	76.6	71.2
		CR (%)		
E 1	8.1	24.6	9.4	12.3
E 2	3.5	16.4	5.2	5.6
E 3	3.8	17.2	5.8	5.8
E 4	6.4	22.1	8.2	6.1
		AR		
E 1	26.7	15.90	24.33	21.37
E 2	28.43	18.21	25.61	23.25
E 3	46.48	30.78	41.50	39.15
E 4	62.32	38.51	56.62	54.14

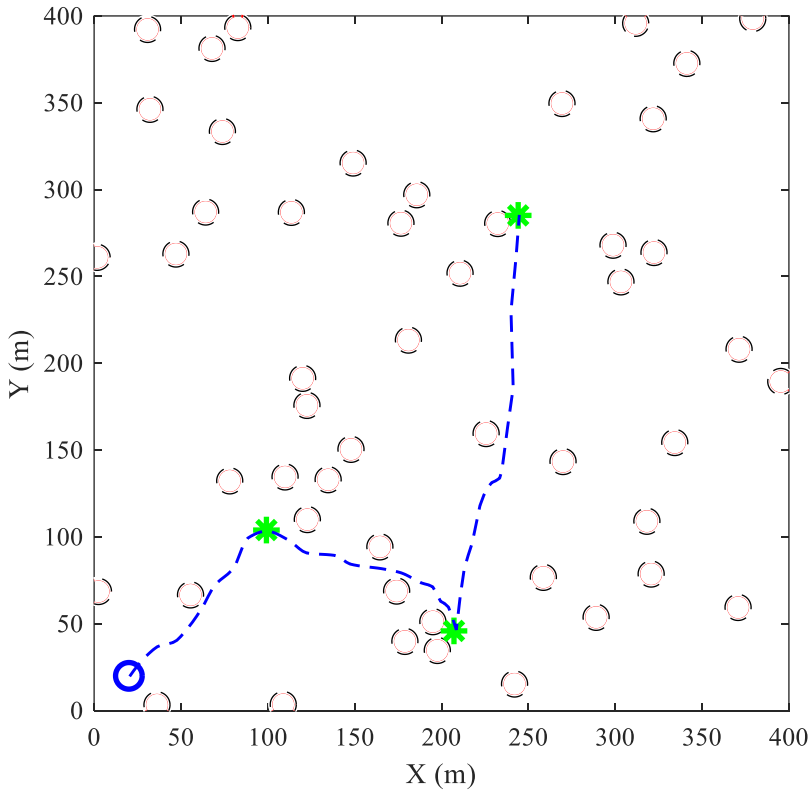

Figure 8. The generated path by the proposed method in Environment 1.

The introduction of more targets presented a more complex challenge, yet the proposed HRL algorithm continued to excel with a 93.6% success rate. This environment tested

the algorithms' capacity to manage additional mission objectives without compromising efficiency or safety. The proposed HRL's success in this scenario emphasizes its effective goal prioritization and energy utilization strategies, crucial for handling multiple objectives. The generated path by the proposed method in Environment 3 is shown in Figure 10.

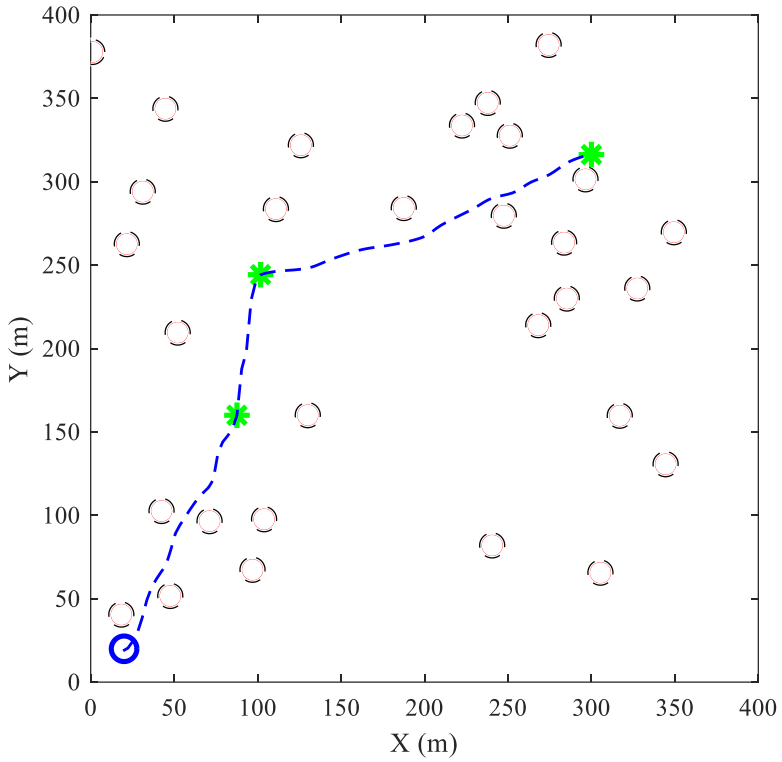

Figure 9. The generated path by the proposed method in Environment 2.

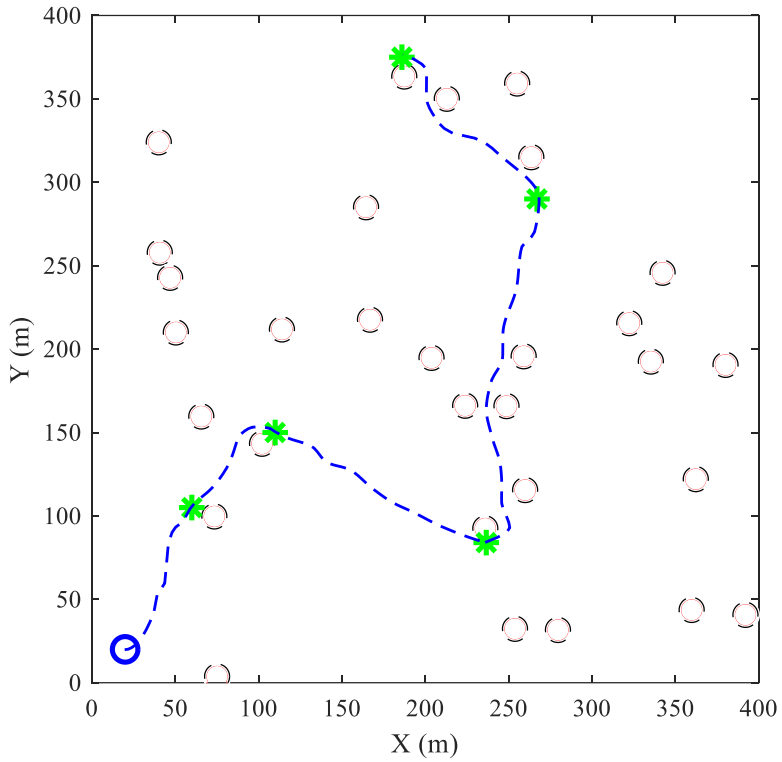

Figure 10. The generated path by the proposed method in Environment 3.

Environment 4: Similar to Environment 3, this setting encompassed 30 cylindrical obstacles with a uniform payload distribution, yet with an increased target count of 7. The proposed HRL algorithm achieved an 89.4% success rate, the highest among the compared methods, demonstrating its resilience and strategic planning prowess in highly complex scenarios. Although the success rate saw a slight decline from previous environments, the algorithm's consistent performance in terms of both collision rate and average rewards highlighted its robustness and the effectiveness of its hierarchical decision-making framework. The generated path by the proposed method in Environment 4 is shown in Figure 11.

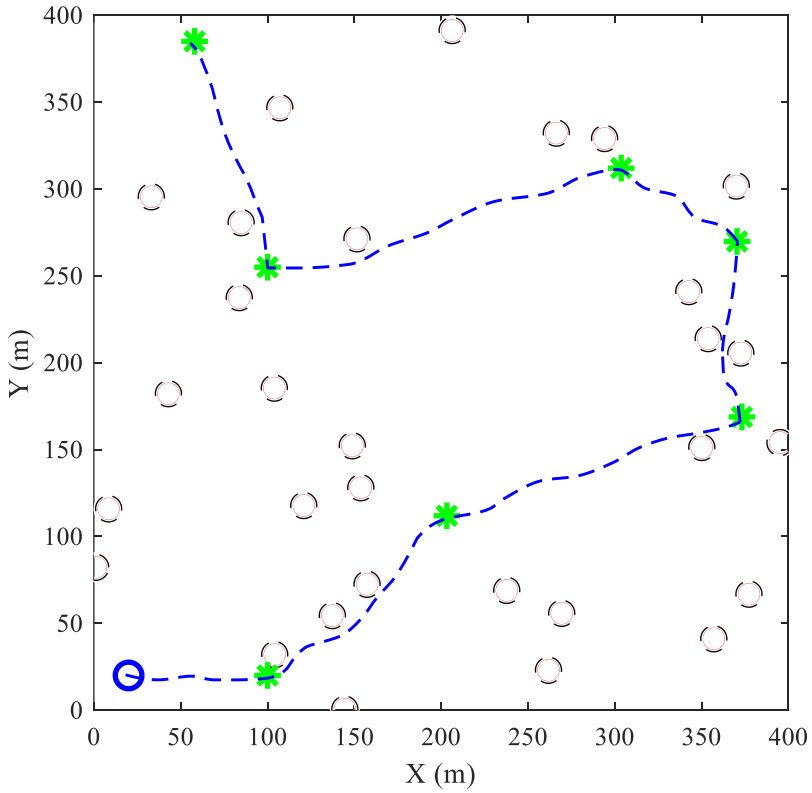

Figure 11. The generated path by the proposed method in Environment 4.

Based on Table 6, the proposed HRL method consistently achieves higher success rates across all test environments, highlighting its robustness and efficiency. Specifically, it achieves a success rate of 90.1% in Environment 1, significantly outperforming SAC (59.2%), HAC+SAC (81.1%), and SAC+HER (75.4%). This trend continues in Environment 2 with a success rate of 95.3% and in Environments 3 and 4 with success rates of 93.6% and 89.4%, respectively. These results demonstrate the method's superior adaptability and effectiveness in varied and complex scenarios.

Our method also exhibits lower collision rates compared with the benchmark algorithms. In Environment 1, the collision rate is 8.1%, compared with SAC (24.6%), HAC+SAC (9.4%), and SAC+HER (12.3%). This improvement persists across all environments, with collision rates of 3.5% in Environment 2, 3.8% in Environment 3, and 6.4% in Environment 4. These results underscore the method's effectiveness in ensuring safe navigation and obstacle avoidance.

The average rewards achieved by our method are significantly higher in all test environments. For instance, in Environment 1, the average reward is 26.7, compared with 15.90 for SAC, 24.33 for HAC+SAC, and 21.37 for SAC+HER. This trend continues with rewards of 28.43 in Environment 2, 46.48 in Environment 3, and 62.32 in Environment 4. Higher average rewards indicate more efficient path planning and energy utilization, validating the advantages of our HRL framework with LSTM-based battery prediction.

Across all test environments, the proposed HRL algorithm consistently outperformed SAC, HAC+SAC, and SAC+HER in success rate, collision avoidance, and average rewards. This superior performance can be attributed to the algorithm's integration of LSTM-based energy consumption prediction, adaptive goal setting, and an advanced learning mechanism that dynamically optimizes path planning and energy management in real time. The success rates of our algorithm diminished less compared with the other models under testing. Furthermore, our results indicate that our proposed algorithm achieved faster convergence with fewer steps required, an essential factor in time-sensitive SAR operations.

In reinforcement learning literature, achieving a success rate of approximately 90% in dynamic and obstacle-rich environments is considered highly effective, and as the results show, the effectiveness of our approach is evident [56]. The integration of LSTM for energy prediction and HER for enhanced learning efficiency significantly contributes to the high performance of our method. Our approach not only demonstrates superior adaptability and efficiency but also maintains robustness across varied scenarios, which is crucial for practical applications in search and rescue missions.

6. Conclusions

This study presents a hierarchical reinforcement learning framework, enhanced with a long short-term memory-based battery consumption prediction model, for optimizing aerial robots' operations in post-disaster scenarios. The integration of LSTM into the HRL framework has been a pivotal advancement, enabling more accurate battery usage predictions and improving decision-making processes at both high and low levels of the control hierarchy.

Our experimental results, obtained through simulations using MATLAB 2023 and a combination of Simscape and UAV Toolbox, demonstrate the superiority of our proposed framework over traditional HRL and benchmark soft actor-critic architectures. The LSTM-augmented HRL model exhibited improvements in mission success rates, energy efficiency, and adaptability to environmental changes, particularly in scenarios involving variable payloads and increased obstacle density. These enhancements are crucial for aerial robots' operations in disaster-stricken areas where efficient resource management and flexible response to unforeseen challenges are essential.

Furthermore, the proposed framework showed a marked increase in endurance and operational efficiency, with aerial robots capable of longer flight times and reduced battery consumption. This efficiency is vital in emergency scenarios where extended aerial robot operation can be critical for successful mission outcomes. Additionally, the framework demonstrated superior collision avoidance and path planning capabilities, further underscoring its applicability in complex and unpredictable environments.

Author Contributions: Conceptualization, M.R. and M.A.A.A.; Methodology, M.R.; Validation, M.R.; Formal analysis, M.R.; Investigation, M.R.; Writing—original draft, M.R.; Writing—review & editing, M.A.A.A.; Supervision, M.A.A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the main author due to ramezani.mahya@ut.ac.ir.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

Acronym	Full Form
AI	Artificial Intelligence
Bi-LSTM	Bidirectional Long Short-Term Memory
DDPG	Deep Deterministic Policy Gradient
DRL	Deep Reinforcement Learning
GPS	Global Positioning System

HER	Hindsight Experience Replay
HiPPO	Hierarchical Proximal Policy Optimization
HLC	High-Level Controller
HRL	Hierarchical Reinforcement Learning
IMU	Inertial Measurement Unit
LiDAR	Light Detection and Ranging
LLC	Low-Level Controller
LSTM	Long Short-Term Memory
MDP	Markov Decision Process
ML	Machine Learning
RL	Reinforcement Learning
SAC	Soft Actor-Critic
SAR	Search and Rescue
UAV	Unmanned Aerial Vehicle

References

1. Abtahi, S.-A.; Atashgah, M.A.; Tarvirdizadeh, B.; Shahbazi, M. Aerial Robotics in Urban Environments: Optimized Path Planning and SITL Assessments. In Proceedings of the 2023 11th RSI International Conference on Robotics and Mechatronics (ICRoM), Tehran, Iran, 19–21 December 2023; pp. 271–278.
2. Lavaei, A.; Atashgah, M.A. Optimal 3D trajectory generation in delivering missions under urban constraints for a flying robot. *Intell. Serv. Robot.* **2017**, *10*, 241–256. [CrossRef]
3. Lyu, M.; Zhao, Y.; Huang, C.; Huang, H. Unmanned aerial vehicles for search and rescue: A survey. *Remote Sens.* **2023**, *15*, 3266. [CrossRef]
4. Ajith, V.; Jolly, K. Unmanned aerial systems in search and rescue applications with their path planning: A review. *J. Phys. Conf. Ser.* **2021**, *2115*, 012020. [CrossRef]
5. Souissi, O.; Benatitallah, R.; Duvivier, D.; Artiba, A.; Belanger, N.; Feyzeau, P. Path planning: A 2013 survey. In Proceedings of the 2013 International Conference on Industrial Engineering and Systems Management (IESM), Agdal, Morocco, 28–30 October 2013; pp. 1–8.
6. Warren, C.W. Global path planning using artificial potential fields. In Proceedings of the 1989 IEEE International Conference on Robotics and Automation, Scottsdale, AZ, USA, 14–19 May 1989; pp. 316–321.
7. Husain, Z.; Al Zaabi, A.; Hildmann, H.; Saffre, F.; Ruta, D.; Isakovic, A. Search and rescue in a maze-like environment with ant and dijkstra algorithms. *Drones* **2022**, *6*, 273. [CrossRef]
8. Hayat, S.; Yanmaz, E.; Bettstetter, C.; Brown, T.X. Multi-objective drone path planning for search and rescue with quality-of-service requirements. *Auton. Robot.* **2020**, *44*, 1183–1198. [CrossRef]
9. Daud, S.M.S.M.; Yusof, M.Y.P.M.; Heo, C.C.; Khoo, L.S.; Singh, M.K.C.; Mahmood, M.S.; Nawawi, H. Applications of drone in disaster management: A scoping review. *Sci. Justice* **2022**, *62*, 30–42. [CrossRef] [PubMed]
10. Ramezani, M.; Alandihallaj, M.A.; Hein, A.M. PPO-Based Dynamic Control of Uncertain Floating Platforms in Zero-G Environment. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024.
11. Ebrahimi, D.; Sharafeddine, S.; Ho, P.-H.; Assi, C. Autonomous UAV trajectory for localizing ground objects: A reinforcement learning approach. *IEEE Trans. Mob. Comput.* **2020**, *20*, 1312–1324. [CrossRef]
12. Azar, A.T.; Koubaa, A.; Ali Mohamed, N.; Ibrahim, H.A.; Ibrahim, Z.F.; Kazim, M.; Ammar, A.; Benjdira, B.; Khamis, A.M.; Hameed, I.A. Drone deep reinforcement learning: A review. *Electronics* **2021**, *10*, 999. [CrossRef]
13. Ramezani, M.; Habibi, H.; Sanchez-Lopez, J.L.; Voos, H. UAV path planning employing MPC-reinforcement learning method considering collision avoidance. In Proceedings of the 2023 International Conference on Unmanned Aircraft Systems (ICUAS), Warsaw, Poland, 6–9 June 2023; pp. 507–514.
14. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
15. Bushnaq, O.M.; Mishra, D.; Natalizio, E.; Akyildiz, I.F. Unmanned aerial vehicles (UAVs) for disaster management. In *Nanotechnology-Based Smart Remote Sensing Networks for Disaster Prevention*; Elsevier: Amsterdam, The Netherlands, 2022; pp. 159–188.
16. AlMahamid, F.; Grolinger, K. Autonomous unmanned aerial vehicle navigation using reinforcement learning: A systematic review. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105321. [CrossRef]
17. Bouhamed, O.; Wan, X.; Ghazzai, H.; Massoud, Y. A DDPG-based Approach for Energy-aware UAV Navigation in Obstacle-constrained Environment. In Proceedings of the 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 2–16 June 2020; pp. 1–6.
18. Imanberdiyev, N.; Fu, C.; Kayacan, E.; Chen, I.-M. Autonomous navigation of UAV by using real-time model-based reinforcement learning. In Proceedings of the 2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV), Phuket, Thailand, 13–15 November 2016; pp. 1–6.

19. Bouhamed, O.; Ghazzai, H.; Besbes, H.; Massoud, Y. Autonomous UAV navigation: A DDPG-based deep reinforcement learning approach. In Proceedings of the 2020 IEEE International Symposium on circuits and systems (ISCAS), Seville, Spain, 12–14 October 2020; pp. 1–5.
20. Zhang, Z.; Wu, Z.; Zhang, H.; Wang, J. Meta-learning-based deep reinforcement learning for multiobjective optimization problems. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 7978–7991. [CrossRef]
21. Birman, Y.; Ido, Z.; Katz, G.; Shabtai, A. Hierarchical Deep Reinforcement Learning Approach for Multi-Objective Scheduling with Varying Queue Sizes. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; pp. 1–10.
22. Hutsebaut-Buysse, M.; Mets, K.; Latré, S. Hierarchical reinforcement learning: A survey and open research challenges. *Mach. Learn. Knowl. Extr.* **2022**, *4*, 172–221. [CrossRef]
23. Ramezani, M.; Alandihallaj, M.A.; Sanchez-Lopez, J.L.; Hein, A. Safe Hierarchical Reinforcement Learning for CubeSat Task Scheduling Based on Energy Consumption. *arXiv* **2023**, arXiv:2309.12004.
24. Zhao, J.; Gan, Z.; Liang, J.; Wang, C.; Yue, K.; Li, W.; Li, Y.; Li, R. Path planning research of a UAV base station searching for disaster victims' location information based on deep reinforcement learning. *Entropy* **2022**, *24*, 1767. [CrossRef] [PubMed]
25. Yu, J.; Su, Y.; Liao, Y. The path planning of mobile robot by neural networks and hierarchical reinforcement learning. *Front. Neurorobot.* **2020**, *14*, 63. [CrossRef] [PubMed]
26. Liu, Z.; Cao, Y.; Chen, J.; Li, J. A hierarchical reinforcement learning algorithm based on attention mechanism for uav autonomous navigation. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 13309–13320. [CrossRef]
27. Gürtler, N.; Büchler, D.; Martius, G. Hierarchical reinforcement learning with timed subgoals. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 21732–21743.
28. Li, A.C.; Florensa, C.; Clavera, I.; Abbeel, P. Sub-policy adaptation for hierarchical reinforcement learning. *arXiv* **2019**, arXiv:1906.05862.
29. Nachum, O.; Gu, S.S.; Lee, H.; Levine, S. Data-efficient hierarchical reinforcement learning. *Adv. Neural Inf. Process. Syst.* **2018**, *31*.
30. Wang, Y.; Shi, D.; Xue, C.; Jiang, H.; Wang, G.; Gong, P. AHAC: Actor hierarchical attention critic for multi-agent reinforcement learning. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 11–14 October 2020; pp. 3013–3020.
31. Parr, R.; Russell, S. Reinforcement learning with hierarchies of machines. In Proceedings of the NIPS'97: Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems, Denver, CO, USA, 31 July 1998.
32. Sutton, R.S.; Precup, D.; Singh, S. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artif. Intell.* **1999**, *112*, 181–211. [CrossRef]
33. Stolle, M.; Precup, D. Learning options in reinforcement learning. In Proceedings of the Abstraction, Reformulation, and Approximation: 5th International Symposium, SARA 2002, Kananaskis, AL, Canada, 2–4 August 2002; pp. 212–223.
34. Precup, D. Temporal Abstraction in Reinforcement Learning. Ph.D. Thesis, University of Massachusetts Amherst, Amherst, MA, USA, 2000.
35. Bacon, P.-L.; Harb, J.; Precup, D. The option-critic architecture. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
36. Harb, J.; Bacon, P.-L.; Klissarov, M.; Precup, D. When waiting is not an option: Learning options with a deliberation cost. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
37. Dayan, P.; Hinton, G.E. Feudal reinforcement learning. In Proceedings of the Advances in Neural Information Processing Systems 5, [NIPS Conference], San Francisco, CA, USA, 30 November–3 December 1992.
38. Jiang, Y.; Gu, S.S.; Murphy, K.P.; Finn, C. Language as an abstraction for hierarchical deep reinforcement learning. In Proceedings of the NIPS'19: 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
39. Nachum, O.; Gu, S.; Lee, H.; Levine, S. Near-optimal representation learning for hierarchical reinforcement learning. *arXiv* **2018**, arXiv:1810.01257.
40. Nachum, O.; Ahn, M.; Ponte, H.; Gu, S.; Kumar, V. Multi-agent manipulation via locomotion using hierarchical sim2real. *arXiv* **2019**, arXiv:1908.05224.
41. Mahadevan, S.; Maggioni, M. Proto-value Functions: A Laplacian Framework for Learning Representation and Control in Markov Decision Processes. *J. Mach. Learn. Res.* **2007**, *8*, 2169–2231.
42. Sutton, R.S.; Modayil, J.; Delp, M.; Degris, T.; Pilarski, P.M.; White, A.; Precup, D. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems, Taipei, Taiwan, 2–6 May 2011; Volume 2, pp. 761–768.
43. Hejna, D.; Pinto, L.; Abbeel, P. Hierarchically decoupled imitation for morphological transfer. In Proceedings of the International Conference on Machine Learning, Vienna, Austria, 12–18 July 2020; pp. 4159–4171.
44. Zhang, J.; Yu, H.; Xu, W. Hierarchical reinforcement learning by discovering intrinsic options. *arXiv* **2021**, arXiv:2101.06521.
45. Shen, H.; Zhang, Y.; Mao, J.; Yan, Z.; Wu, L. Energy management of hybrid UAV based on reinforcement learning. *Electronics* **2021**, *10*, 1929. [CrossRef]
46. Gebauer, C.; Dengler, N.; Bennewitz, M. Sensor-Based Navigation Using Hierarchical Reinforcement Learning. In Proceedings of the International Conference on Intelligent Autonomous Systems, Zagreb, Croatia, 13–16 June 2022; pp. 546–560.

47. Tallamraju, R.; Saini, N.; Bonetto, E.; Pabst, M.; Liu, Y.T.; Black, M.J.; Ahmad, A. AirCapRL: Autonomous aerial human motion capture using deep reinforcement learning. *IEEE Robot. Autom. Lett.* **2020**, *5*, 6678–6685. [CrossRef]
48. Khamidehi, B.; Sousa, E.S. Distributed deep reinforcement learning for intelligent traffic monitoring with a team of aerial robots. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–22 September 2021; pp. 341–347.
49. Faust, A.; Oslund, K.; Ramirez, O.; Francis, A.; Tapia, L.; Fiser, M.; Davidson, J. PRM-RL: Long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 5113–5120.
50. Ugurlu, H.I.; Pham, X.H.; Kayacan, E. Sim-to-real deep reinforcement learning for safe end-to-end planning of aerial robots. *Robotics* **2022**, *11*, 109. [CrossRef]
51. Bartolomei, L.; Kompis, Y.; Teixeira, L.; Chli, M. Autonomous emergency landing for multicopters using deep reinforcement learning. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 3392–3399.
52. Hou, Z.; Fei, J.; Deng, Y.; Xu, J. Data-efficient hierarchical reinforcement learning for robotic assembly control applications. *IEEE Trans. Ind. Electron.* **2020**, *68*, 11565–11575. [CrossRef]
53. Qin, Y.; Wang, Z.; Chen, C. HRL2E: Hierarchical reinforcement learning with low-level ensemble. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022; pp. 1–7.
54. Xing, L. Learning and exploiting multiple subgoals for fast exploration in hierarchical reinforcement learning. *arXiv* **2019**, arXiv:1905.05180.
55. Li, J.; Tang, C.; Tomizuka, M.; Zhan, W. Hierarchical planning through goal-conditioned offline reinforcement learning. *IEEE Robot. Autom. Lett.* **2022**, *7*, 10216–10223. [CrossRef]
56. Ramezani, M.; Sanchez-Lopez, J.L. Human-Centric Aware UAV Trajectory Planning in Search and Rescue Missions Employing Multi-Objective Reinforcement Learning with AHP and Similarity-Based Experience Replay. *arXiv* **2024**, arXiv:2402.18487.
57. Ma, J. Entropy Augmented Reinforcement Learning. *arXiv* **2022**, arXiv:2208.09322.
58. Ahmed, Z.; Le Roux, N.; Norouzi, M.; Schuurmans, D. Understanding the impact of entropy on policy optimization. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 151–160.
59. Alandihallaj, M.A.; Ramezani, M.; Hein, A.M. MBSE-Enhanced LSTM Framework for Satellite System Reliability and Failure Prediction. In Proceedings of the MODELSWARD, Rome, Italy, 21–23 February 2024; pp. 349–356.
60. Vela, A.E. Trajectory-Based State-of-Charge Prediction Using LSTM Recurrent Neural Networks. In Proceedings of the 2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC), Barcelona, Spain, 1–5 October 2023; pp. 1–7.
61. Jiang, P.; Wang, Z.; Li, X.; Wang, X.V.; Yang, B.; Zheng, J. Energy consumption prediction and optimization of industrial robots based on LSTM. *J. Manuf. Syst.* **2023**, *70*, 137–148. [CrossRef]
62. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **2005**, *18*, 602–610. [CrossRef] [PubMed]
63. Sundermeyer, M.; Schlüter, R.; Ney, H. Lstm neural networks for language modeling. In Proceedings of the Interspeech, Portland, OR, USA, 9–13 September 2012; pp. 194–197.
64. Zhang, Z.; Xu, M.; Ma, L.; Yu, B. A state-of-charge estimation method based on bidirectional lstm networks for lithium-ion batteries. In Proceedings of the 2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV), Shenzhen, China, 13–15 December 2020; pp. 211–216.
65. Levy, A.; Platt, R.; Saenko, K. Hierarchical actor-critic. *arXiv* **2017**, arXiv:1712.00948.
66. Rodrigues, T.A.; Patrikar, J.; Choudhry, A.; Feldgoise, J.; Arcot, V.; Gahlaut, A.; Lau, S.; Moon, B.; Wagner, B.; Matthews, H.S. In-flight positional and energy use data set of a DJI Matrice 100 quadcopter for small package delivery. *Sci. Data* **2021**, *8*, 155. [CrossRef]