scientific reports

Check for updates

OPEN

Agile DQN: adaptive deep recurrent attention reinforcement learning for autonomous UAV obstacle avoidance

Fadi AlMahamid & Katarina Grolinger✉

Unmanned Aerial Vehicle (UAV) obstacle avoidance in 3D environments demands sophisticated handling of high-dimensional inputs and effective state representations. Current Deep Reinforcement Learning (DRL) algorithms struggle to prioritize salient aspects of state representations and manage extensive state and action spaces, particularly in partially observable environments. Addressing these challenges, this paper proposes Agile DQN (AG-DQN), a novel algorithm that dynamically focuses on key visual features and robust Q-value estimation to enhance learning. The AG-DQN architecture synergizes several components—Glimpse Network, LSTM Recurrent Network, Emission Network, and Q-Network—to dynamically and selectively process crucial visual features, optimizing decision-making without processing the entire state. AG-DQN's adaptive temporal attention strategy also adjusts to environmental changes, maintaining a balance between recent and past observations. Experimental results demonstrate AG-DQN's improved performance over existing DRL methods, highlighting its potential in advancing autonomous UAV navigation and robotics.

Keywords Autonomous unmanned aerial vehicles, Deep reinforcement learning, Autonomous visual navigation, Attention models, Deep learning, Deep Q-networks, Algorithms, Obstacle avoidance

Autonomous navigation in intricate indoor spaces, populated with diverse obstacles, presents significant challenges in robotics and autonomous systems[1]. These challenges are particularly pertinent for Unmanned Aerial Vehicles (UAVs) due to their operation in three-dimensional spaces, which complicates environmental perception and decision-making[2]. UAVs must navigate highly dynamic conditions, such as varied obstacles and varied lighting, requiring real-time and robust decision-making capabilities. Additionally, the high dimensionality of the input state and the need for dynamic representations, coupled with the requirement for real-time decision-making, introduce further layers of complexity[3].

In the realm of Autonomous UAV navigation, there is a range of tasks, including path planning, attitude control, and obstacle avoidance. Path planning involves determining the optimal trajectory from a start to an endpoint, focusing on efficiency and strategic route selection. Attitude control is critical for maintaining flight stability and involves adjusting the UAV's orientation—roll, pitch, and yaw—in real-time. In contrast, obstacle avoidance requires the UAV to detect and navigate around obstacles using real-time sensory data, often from front-facing cameras. Recognizing the diversity of the environments and the necessity of avoiding collisions, this paper focuses on obstacle avoidance as the primary evaluation context.

Despite the adaptability of Deep Reinforcement Learning (DRL) to learn from environmental interactions for obstacle avoidance[4,5], conventional DRL algorithms such as DQN[6,7], DRQN[8], and DARQN[9] show limitations of efficient state management and representation in highly dynamic environments, pointing to a clear gap in the capability of existing DRL algorithms.

DRQN and DARQN, while integrating memory and attention mechanisms to enhance state representation, treat environmental features uniformly. This approach can lead to inefficiencies, especially in environments where the relevance of features changes dynamically. The static nature of their attention configurations also limits adaptability, which is essential for dynamic environments, such as autonomous UAV obstacle avoidance in three-dimensional environments.

To address these gaps, this paper introduces Agile DQN (AG-DQN), a novel DRL algorithm that advances beyond traditional DRL algorithms by incorporating a dynamic multi-glimpse strategy. Unlike conventional

Department of Electrical and Computer Engineering, Western University, London N6A 5B9, ON, Canada. ✉email: kgroling@uwo.ca

methods that uniformly process the entire environmental state for action selection, AG-DQN selectively and adaptively processes only the most relevant parts of the state (image). This approach addresses critical limitations in practical UAV navigation scenarios, where static attention mechanisms and inefficient processing impede responsiveness to rapid and unpredictable environmental changes, such as dynamic obstacles, varying illumination, and complex spatial layouts. AG-DQN theoretically enhances computational efficiency, accelerates decision-making, and improves adaptability in complex, real-world environments by dynamically predicting and attending to essential visual regions. Consequently, AG-DQN not only manages high-dimensional input representations effectively but also practically enhances UAV robustness and efficiency.

Additionally, AG-DQN incorporates a dynamic temporal attention strategy to handle temporal dependencies among observations, which is important as the sequence of interactions plays a significant role in the DRL agent's decision-making. AG-DQN leverages a dynamic temporal attention strategy that continuously adapts to changing environments, enabling the agent to balance the focus on recent observations and maintain a broader perspective of past experiences.

AG-DQN's success stems from a synergy of several components optimizing the agent's decision-making. The Glimpse Network extracts and synthesizes spatial and temporal information from multiple glimpses, generating a context vector that encapsulates the agent's understanding of the environment. The Recurrent Network considers this context vector alongside the agent's history of interactions. The Emission Network predicts future glimpse locations, fostering a dynamic interaction with the environment. Lastly, the Q-Network serves as the decision-making hub, enabling action selection based on the integrated information. This integration amplifies the agent's learning and navigation capabilities, making AG-DQN a promising solution for complex UAV navigation tasks.

The proposed AG-DQN algorithm is evaluated for obstacle avoidance, emphasizing its reliance on images captured by the UAV's front-facing camera. This approach differentiates AG-DQN from many DRL algorithms evaluated in abstract environments like Atari games. The contributions of AG-DQN as a novel DRL algorithm can be summarized as follows:

- **Dynamic Multi-Glimpse Strategy:** AG-DQN improves traditional DRL approaches by selectively processing state components that are critical for immediate decision-making, thus addressing the inefficiencies of full-state processing.
- **Adaptive Temporal Attention:** This mechanism enables the agent to maintain a balance between recent and past observations, which is crucial for improved decision-making in dynamic environments.
- **Empirical Validation of Improved Performance:** AG-DQN is empirically validated through comparative evaluations of autonomous UAV obstacle avoidance in complex 3D indoor environments, achieving better performance while leveraging only 33% of the input image compared to existing methods that require processing 100% of the input image,

The remainder of this paper is organized as follows: "Background" introduces core RL concepts relevant to AG-DQN, "Related work" discusses the related work, "Problem formulation" formulates the navigation problem, "Agile DQN" describes AG-DQN architecture, "Evaluation" describes experiments and discusses findings. Finally, "Conclusion" concludes the paper.

Background

This section provides an overview of the Enriched Deep Recurrent Attention Model (EDRAM), a key inspiration for Agile-DQN's multi-glimpse strategy. It also examines a variety of Deep Q-Network (DQN) extensions applied to improve Agile-DQN's reward.

Enriched deep recurrent attention model

The Enriched Deep Recurrent Attention Model (EDRAM)[10], an advanced version of the Deep Recurrent Attention Model (DRAM)[11], is optimized for recognizing multiple objects in complex visual environments. EDRAM enhances efficiency in object recognition tasks by processing strategic sections of the input data, known as 'glimpses'. This approach is particularly effective in scenarios with cluttered or occluded scenes.

EDRAM consists of four main components:

Glimpse Network: This component extracts relevant information from localized regions within an input image, converting these 'glimpses' into a fixed-size vector representation. It focuses on processing pertinent data within specific image areas.

Recurrent Network: Leveraging Long Short-Term Memory (LSTM) units, this network processes temporal information, blending new insights from the current glimpse with previously processed data.

Emission Network: dynamically predicts the next glimpse location based on the hidden state from the Recurrent Network, guiding the model's adaptive focus on various segments of the input space.

Classification Network: This network translates the accumulated knowledge into a probability distribution over class labels, enabling informed object recognition decisions.

Collectively, these components enhance EDRAM's performance in object recognition tasks, especially in visually complex environments, making it a promising approach for applications like UAV navigation.

DQN extensions

Applying various DQN extensions analogous to those employed in the *Rainbow DQN* algorithm[12] has demonstrated effectiveness in improving the reward. While these DQN extensions are not the main contribution of this paper, they are briefly reviewed here to contextualize the experimental evaluations conducted in "Evaluation", where their specific impacts on AG-DQN are analyzed.

Double Q-Learning: The standard DQN algorithm suffers from overestimation bias because it uses the same network for selecting and evaluating the best action during the maximization step in the Q-learning update rule (i.e., minimizing the difference between the current Q-value estimate and the target Q-value). Double DQN[7] tackles the overestimation problem by employing two separate neural networks: the first for action selection and the second for action evaluation during the maximization step.

Prioritized Experience Replay: Experience replay (ER)[13] is employed with conventional DQN to enhance learning by storing and reusing past experiences using a replay buffer. By uniformly sampling experiences from the buffer, DQN breaks the correlation between consecutive samples, promoting stable learning and improving convergence. In contrast, Prioritized Experience Replay (PER)[14] advances upon ER by assigning a priority score to each experience based on the magnitude of its temporal-difference (TD) error. This prioritization of experiences allows the agent to focus on more informative samples during training, ultimately resulting in accelerated convergence and superior performance.

Dueling Networks: The dueling network[15] is a DQN architecture enhancement that aims to improve learning by employing separate streams for the state-value functions $\boldsymbol{V(s)}$, which measures how good it is for the agent to be in state $\boldsymbol{s}$, and the advantage-value function $\boldsymbol{A(s,a)}$ function, which captures how beneficial an action is compared to other actions at a given state $\boldsymbol{s}$. The dueling network improves approximation and stability during the learning process by decoupling the estimation of the state-value and advantage-value. The two streams are combined in the final layer to produce the Q-values $\boldsymbol{Q(s,a)}$ representing the expected return of taking action $\boldsymbol{a}$ while in state $\boldsymbol{s}$. As explained in Equation 1, the combination is achieved by adding the state-value function and the advantage-value function while subtracting the mean of the advantage-value function to maintain identifiability.

$$Q(s,a) = V(s) + \left(A(s,a) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s,a)\right) \tag{1}$$

Distributional RL: Traditional Q-learning focuses on learning the state-action value function, $\boldsymbol{Q(s,a)}$, which estimates the expected cumulative reward for taking action $\boldsymbol{a}$ in state $\boldsymbol{s}$ while following the optimal policy. In contrast, Distributional RL aims to learn the entire distribution of potential returns for each state-action pair, denoted as $\boldsymbol{z(s,a)}$, where $\boldsymbol{z}$ is a vector with $N_{atoms} \in \mathbb{N}^+$ discrete atoms representing different possible return values. Rainbow DQN incorporates Distributional RL using the Categorical DQN (C51) algorithm[16]. C51 approximates the distribution of returns by employing a fixed set of categorical support points. By learning the distribution of returns rather than a single expected value, Distributional RL enables the agent to more accurately capture the uncertainty in its environment, thereby improving its decision-making.

Noisy Nets: Conventional DQNs balance exploration and exploitation using the epsilon-greedy strategy, which might lead to inefficient exploration due to random action selection and a fixed exploration schedule. Additionally, because the agent's action selection only considers the expected cumulative reward, it may underestimate the uncertainty in the environment and miss to account for the complete range of potential outcomes. On the other hand, Noisy Nets[17] address these issues by injecting noise directly into the neural network's weights, promoting more intelligent exploration, enabling the agent to adjust its exploration rate, and handling the environment uncertainty.

Related work

The advent of sophisticated Deep Reinforcement Learning (DRL) techniques has significantly propelled the field of autonomous UAV navigation forward, introducing innovative solutions to navigate complex and dynamic environments efficiently. Among these challenges, path planning, attitude control, and obstacle avoidance are critical areas requiring advanced computational strategies and precise, adaptable solutions.

This literature review aims to contextualize AG-DQN within this evolving landscape, especially focusing on obstacle avoidance. More specifically, the review focuses on attention-based models due to their recent successes in obstacle avoidance and ability to selectively process environmental states through dynamic attention mechanisms.

Our analysis within the domain of UAV navigation for obstacle avoidance underscores AG-DQN's unique position and the practical implications of its attention-based processing strategy. Therefore, this review of related work focuses on two primary categories: (1) attention-based DRL methods employed for autonomous UAV navigation through the lens of obstacle avoidance and (2) the use of temporal and attention mechanisms in DRL algorithms.

Attention based DRL for UAV navigation

In recent years, significant advancements have been made in autonomous navigation using deep reinforcement learning. Research studies have proposed diverse methodologies, with many incorporating attention mechanisms into reinforcement learning architectures to boost performance[18–22].

Hierarchical reinforcement learning models have been used in the works of Liu et al.[23] and Chen et al.[24] to address complex navigation tasks. These models integrate high-level decision-making with low-level control.

Attention mechanisms are also integrated into deep reinforcement learning systems by Huang et al.[25] and Wei et al.[26]. Huang et al. used a multi-view perception module to filter redundant information from multi-camera sensing. Wei et al. introduced an attention mechanism within the actor-critic technique to improve path planning in UAV crowdsensing systems.

Several studies, including those by Chen et al.[27], Josef et al.[28], and Chen et al.[29], explored navigation in crowded environments and harsh terrains. Chen et al.[27] proposed a Crowd-Robot Interaction (CRI) model using a self-attention mechanism. Josef et al.[28] used a deep reinforcement learning approach with self-attention modules to improve the explainability of the learned policy. Chen et al.[29] concentrated on crowd-aware robot navigation using graph convolutional networks with attention learned from the human gaze.

Other studies, such as those by Mousavi et al.[30], Mezghan et al.[31], and Mayo et al.[32], implemented different types of attention mechanisms for autonomous UAV navigation tasks. Mousavi et al. used a soft attention mechanism with a DQN model to highlight task-relevant locations in input frames. Mezghan et al. introduced a memory-augmented, attention-based model for image-goal navigation. Mayo et al. developed a visual navigation model that uses spatial attention to guide the agent towards a goal location based on a single image.

Chipka et al.[33] and Shi et al.[34] extend the exploration of attention mechanisms in reinforcement learning. Chipka et al. proposed a computer vision-based attention generator using DQN to discern relevant visual input features. Shi et al.[34] presented a unique method for path planning that incorporates an attention mechanism to focus selectively on specific waypoints, enabling dynamic task prioritization.

While the abovementioned studies have significantly contributed to autonomous navigation using reinforcement learning, AG-DQN introduces a unique dynamic attention mechanism. This mechanism adaptively and selectively processes essential features within the image, enabling more agile and improved navigation in complex 3D environments, which sets AG-DQN apart as a promising solution to this challenging problem.

Temporal and attention mechanisms in RL algorithms

DRQN[8] and DARQN[9] are foundational DRL algorithms that embed temporal dependencies and attention mechanisms to focus on significant state features. DRQN, with its recurrent layers (specifically LSTM), models temporal dependencies to capture sequential information in partially observable environments. However, DRQN processes all state aspects uniformly, lacking selective emphasis on significant features, which could hinder its performance in complex scenarios.

In contrast, DARQN[9] incorporates attention mechanisms into its architectural design to selectively process and improve the estimated Q-values. Nevertheless, it is essential to recognize that DARQN's attention mechanism may struggle with complex problem domains that require intricate feature extraction and dynamic adaptation, as its capacity to adjust to varying or complex contexts is limited.

In particular, the DARQN's attention strategy operates under a static framework, wherein the positions and quantity of attention points remain consistent throughout the learning process, limiting the attention mechanism's potential. Furthermore, although DARQN's attention mechanism enhances state representation, it necessitates processing the entire environmental state.

Addressing the shortcomings in DRQN and DARQN, the proposed AG-DQN incorporates elements from both methods. Specifically, it adopts the temporal processing characteristic of DRQN and DARQN while introducing a dynamic attention model inspired by EDRAM[10] into Q-learning. This combination creates a comprehensive reinforcement learning algorithm by employing an attention mechanism that adaptively focuses on the most pertinent information within the state, providing a dynamic and detailed representation of the input state.

Recent advancements in Deep Q-Learning have also explored alternative innovations to enhance sample efficiency and stabilize learning. Preference-guided stochastic exploration methods have improved exploration efficiency in large action spaces[35], while self-punishment and reward backfill mechanisms have provided novel ways to stabilize Q-learning in challenging environments[36]. However, these approaches primarily target exploration strategies and reward stabilization, and do not specifically address the central challenges of dynamic visual attention and partial observability, which AG-DQN directly tackles.

Problem formulation

This research addresses the challenge of autonomous visual navigation in complex indoor environments using a Partially Observable Markov Decision Process (POMDP). A POMDP provides a structured mathematical framework suitable for scenarios where an agent makes decisions based on incomplete or noisy observations rather than full state knowledge. Formally, the POMDP is defined as a tuple:

$$\mathcal{M} = (S, A, T, R, \Omega, O, \gamma) \tag{2}$$

where each component is defined as follows:

State and Observation Spaces (S, Ω)

At time step t, the environment is in an unobservable state $s_t \in S$. Before choosing the control action a_t, the agent possesses the composite observation

$$o_t = (I_t, L_t) \in \Omega,$$

where $I_t \in \mathbb{R}^{H\times W\times D}$ is the RGB image captured by the UAV's forward-facing camera, and $L_t = \{(x_t^k, y_t^k)\}_{k=1}^{K}$ is the set of K glimpse centers predicted by the Emission Network at the end of the previous step $t-1$.

Action Space (A)

The action space comprises a set of discrete actions defined as:

$$A = \{a_{\text{left}},\ a_{\text{right}},\ a_{\text{forward}}\}$$

corresponding respectively to UAV movements to the left, right, and forward by a fixed distance. The altitude and yaw orientations are stabilized autonomously by the UAV's onboard flight controller.

Since AG-DQN is fundamentally a value-based algorithm derived from Deep Q-Learning (DQN), it inherently requires a discrete action space to function. Continuous action spaces would necessitate entirely different RL frameworks, such as policy-gradient or actor-critic methods, that are fundamentally incompatible with AG-DQN's design and underlying Q-learning mechanism. Therefore, adopting a discrete action space ensures compatibility and leverages the strengths of value-based methods. Similar discrete-action formulations have been consistently employed in prior studies on vision-based UAV navigation [4,25,33,37].

Transition Function (T)

The transition probability between states given an action is defined as

$$T(s_{t+1} \mid s_t, a_t) = P\big(S_{t+1} = s_{t+1} \mid S_t = s_t, A_t = a_t\big).$$

In this research, the transitions are deterministic and fully defined by the current state-action pair.

Observation Probability (O).

Given the hidden state s_t and the previous action a_{t-1}, the likelihood of observing $o_t = (I_t, L_t)$ can be decomposed as

$$O\big(o_t \mid s_t, a_{t-1}\big) = P_{\text{cam}}\big(I_t \mid s_t\big) \cdot P_{\text{glimpse}}\big(L_t \mid s_{t-1}, a_{t-1}, \theta\big),$$

where P_{cam} models camera noise and illumination variation, and P_{glimpse} is the glimpse-location distribution produced by the Emission Network with parameters θ.

Reward Function (R)

The reward function $R(s_t, a_t)$ assigns immediate scalar feedback based on the current state-action pair:

$$R(s_t, a_t) = \begin{cases} R_{\text{goal}}, & \text{goal reached} \\ R_{\text{collision}}, & \text{collision detected} \\ R_{\text{forward}}, & a_t = a_{\text{forward}} \\ R_{\text{sideways}}, & a_t \in \{a_{\text{left}}, a_{\text{right}}\} \end{cases}$$

Colliding with obstacles results in a significant penalty, discouraging unsafe actions. Forward movements yield a positive reward, encouraging goal-oriented behavior, whereas sideways maneuvers attract a minor penalty to discourage non-goal-oriented behavior. Successfully reaching the destination yields a substantial positive reward, incentivizing efficient navigation. Numerical values and further justification for these rewards are detailed in "Evaluation".

Observation Probability (O)

Given the hidden state s_t and the previous control action a_{t-1}, the likelihood of observing $o_t = (I_t, L_t)$ can be decomposed into

$$O\big(o_t \mid s_t, a_{t-1}\big) = P_{\text{cam}}\big(I_t \mid s_t\big) \cdot P_{\text{glimpse}}\big(L_t \mid s_{t-1}, a_{t-1}, \theta\big),$$

where P_{cam} models camera noise and illumination variation, and P_{glimpse} is the glimpse-location distribution induced by the Emission Network with parameter θ.

Discount Factor (γ)

The discount factor $\gamma \in [0, 1]$ quantifies the importance of future rewards relative to immediate rewards. AG-DQN aims to maximize the expected cumulative reward, defined by the Bellman Optimality Equation:

$$Q_*(s, a) = E\left[R_{t+1} + \gamma \max_{a'} Q_*(s', a')\right]$$

AG-DQN iteratively refines Q-values according to this equation, approximating the highest expected returns and guiding optimal action selection. The numerical choice for γ is provided in "Evaluation".

The AG-DQN operates within this POMDP framework using a dynamic multi-glimpse strategy, essential for selecting relevant state features for navigation. The Glimpse Network processes observations o_t to extract salient features, while the Emission Network predicts future glimpse locations L_{t+1}. The Recurrent Network captures temporal dependencies, and the Q-Network, as the decision-making component, computes Q-values for action selection.

Figure 1 provides an overview of the AG-DQN architecture, illustrating how the agent interacts with the UAV simulation environment and the execution order of different components. This visual representation illustrates the AG-DQN's execution within the Markov Decision Process (MDP), beginning with Step 1: processing the state $s_t(I_t, L_t)$, which includes the RGB image I_t from the UAV's front-facing camera and the predicted glimpse locations L_t. In Step 2, the state is passed to the Glimpse Network, which extracts a feature vector G_t from the identified regions of the image using the glimpses. The process then advances to Step 3, where the extracted features G_t are forwarded to the Recurrent Network for temporal pattern analysis. Following this, in Step 4, the

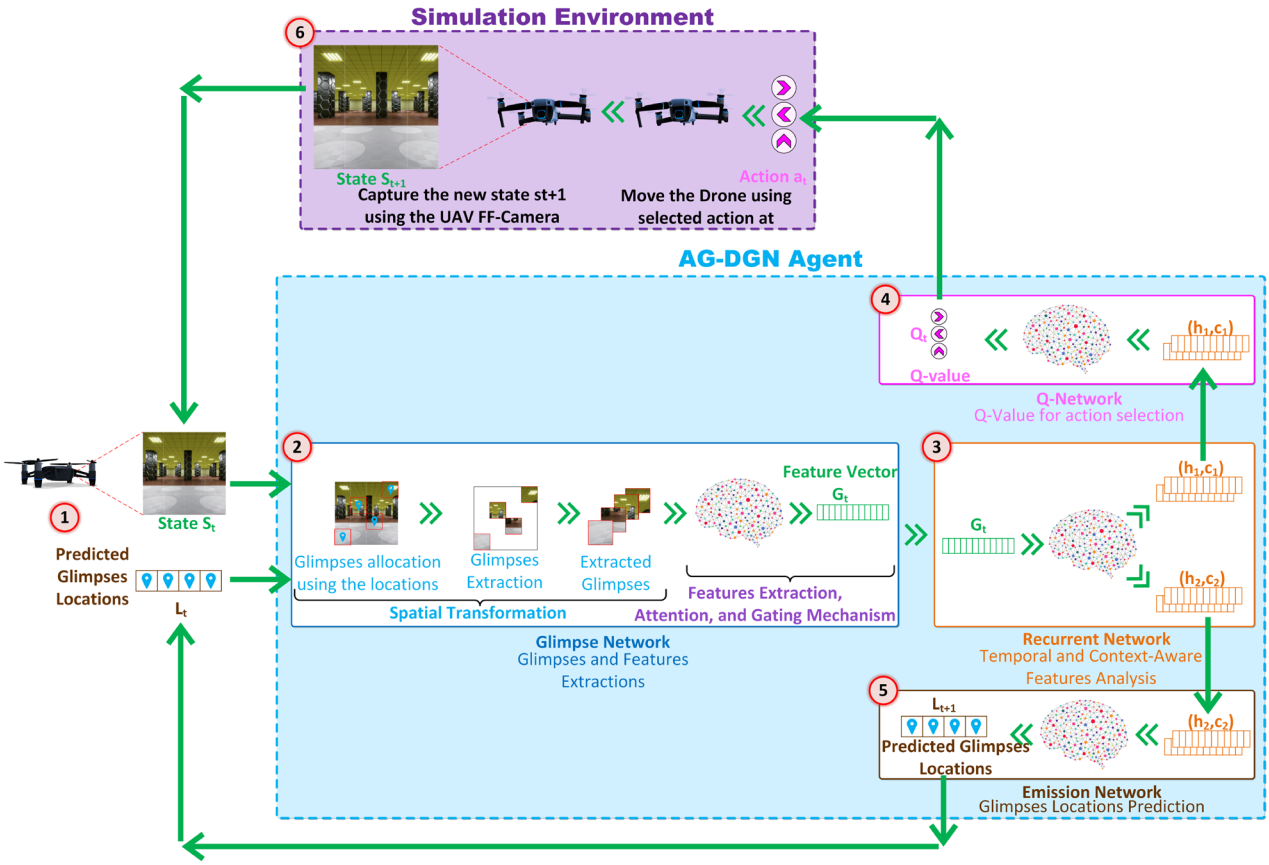


Figure 1. AG-DQN overview showing the agent's interaction with the UAV simulation environment and the execution order.

output from the Recurrent Network is channeled to the Q-Network, leading to the computation of the Q-value for the subsequent action. Concurrently, in Step 5, the Emission Network predicts the next set of glimpses' locations. While, Step 6 involves action execution, where the UAV moves to a new position based on the Q-value, and captures a new image. This step completes the cycle and sets the stage for next AG-DQN's operational loop using the new state $s_{t+1}(I_{t+1}, L_{t+1})$.

Agile DQN

The proposed Agile Deep Q-Network (AG-DQN) is a DRL framework designed for autonomous UAV obstacle avoidance in complex 3D environments. By augmenting the foundational principles of DARQN[9] and incorporating the Enriched Deep Recurrent Attention Model (EDRAM)[10], AG-DQN enhances the agent's adaptability and performance by enabling the agent to selectively focus on pertinent state features instead of processing the entire input state through its dynamic multi-glimpse strategy. This adaptive and selective strategy improves learning and task performance by enabling dynamic focus adjustment towards areas of interest based on learned representations.

EDRAM uses a single-glimpse strategy for object recognition, which has demonstrated its effectiveness in managing high-dimensional data and outperforming state-of-the-art algorithms. However, this strategy identifies and processes key input segments sequentially rather than simultaneously, limiting its potential in complex decision-making scenarios, such as those found in autonomous UAV obstacle avoidance.

AG-DQN builds further on EDRAM's single-glimpse strategy by adopting a multi-glimpse strategy and leveraging experience replay, which is a commonly employed technique in DQN. AG-DQN benefits from integrating this approach to accelerate the learning process convergence. Utilizing ER during training plays a crucial role in breaking the correlation among sequential experiences, which promotes enhanced stability, reduces the risks of oscillations or divergence[38], and ultimately enhances the robustness of AG-DQN's learning capability.

Moreover, the integration of experience replay within AG-DQN aids in refining the prediction of glimpse locations. By replaying past observations, AG-DQN improves its ability to estimate the optimal locations in the state where significant features are likely to be present. This refinement of glimpse locations enables improved feature extraction, as the attention mechanism can selectively focus on informative regions within the state space. Consequently, AG-DQN empowers more accurate and informed decision-making processes, leading to improved performance in complex environments.

Comprised of four integral components, the AG-DQN architecture encompasses: (1) Glimpse Network, (2) Recurrent Network, (4) Emission Network, and (4) Q-Network. Figure 2 depicts this structure, exhibiting the temporal dependency and the information flow among these components over different time steps.

Through integrating these network components, AG-DQN provides significant improvements over traditional DQN approaches, addressing the challenges associated with partial observability and the lack of adaptability. The architecture's selective processing, incorporation of temporal information, adaptive attention allocation, and optimized action selection contribute to AG-DQN's enhanced performance and applicability in complex environments.

The following subsections describe the specifics of each network component within the AG-DQN illustrated in Fig. 3, providing a comprehensive understanding of AG-DQN architecture's functionality.

Glimpse network

The Glimpse Network is tasked with the selective extraction of salient features from specific locations within the given state. By processing only restricted segments of the state and adaptively fusing the local and global features derived, AG-DQN evades the processing of irrelevant information. This focused attention to critical state aspects enhances the agent's perception and understanding of the environment. The Glimpse Network operations comprise a sequence of steps, each facilitated by a designated sub-component: *Spatial Transformer Network*, *Extraction Network*, *Attention Mechanism*, and *Gating Mechanism*.

Spatial transformer network

This network[39], operating as a part of the Glimpse Network, utilizes the *Affine Grid Generator* to extract localized glimpses from the input image. The process commences with the *Glimpse Coordinate Scaling*, which normalizes glimpses locations received from the *Emission Network* to [-1,1]. These coordinates representing the center of glimpses are transformed to align with the actual coordinates of the image, ensuring that the glimpses accurately represent a specific region within the image bounds.

Upon scaling the glimpse locations, they are forwarded to the *Affine Grid Generator* along with the input image. This component leverages an affine transformation, a geometric transformation that combines linear transformations (rotation, scaling, and shearing) and translation (shifting). Affine transformation, while altering the image shape, size, and position, preserves collinearity and ratios of distances, which aids in maintaining the integrity of the image's inherent features. In essence, the Affine Grid Generator applies this transformation to the input image resulting in a transformed output grid representing a spatial reference framework with respect to the input image.

Then *Grid Sampler* receives the output from *Affine Grid Generator* to extract pixel values from the input image at the transformed grid points. Given that the transformed grid points might not perfectly align with the input image's pixel grid, a bi-linear interpolation technique[40] is employed to estimate pixel values at these coordinates, thereby generating localized glimpses.

The coordinated sequence of the *Spatial Transformer Network* operations ensures that the glimpses retain their visual quality and capture the most relevant features of the input image. This dynamic transformation reduces the *Glimpse Network's* sensitivity to the effects of distortions and variations, allowing for a more accurate and informative representation of the input.

Extraction network

Once the glimpses are generated through the *Spatial Transformer Network*, the *Extraction Network* component, consisting of a Convolutional Neural Network (CNN), is employed to process the extracted glimpses and obtain feature maps. The CNN consists of repeated pairs of convolution and max-pooling layers, enabling it to capture local patterns such as edges, corners, and textures. As the information progresses through the network and

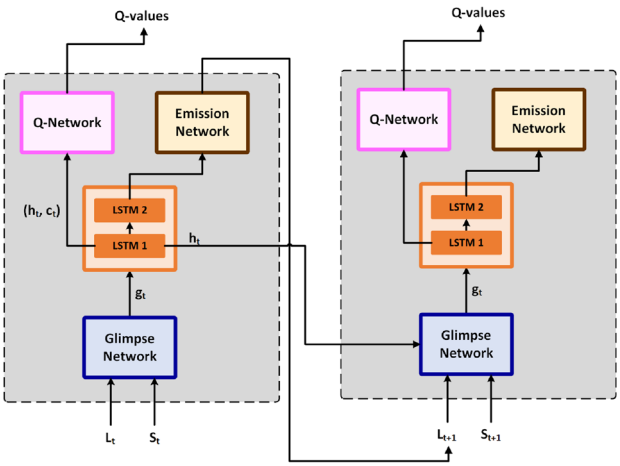


Figure 2. High-level AG-DQN architecture showing the temporal dependencies between time steps.

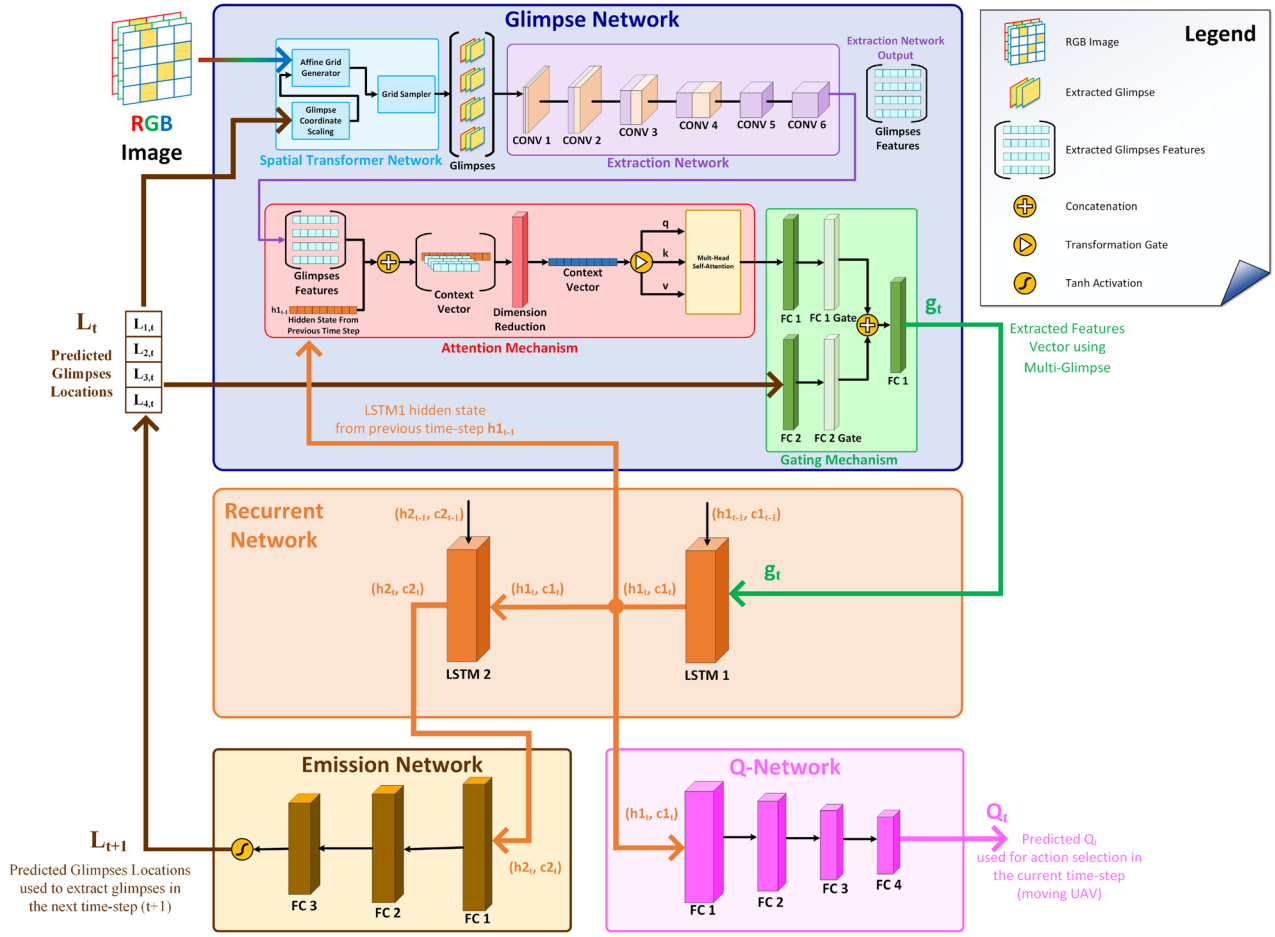


Figure 3. AG-DQN architecture showing the four main neural network components and their interactions.

passes through multiple layers, the CNN begins to capture more complex patterns and higher-level features within the context of the glimpses[41].

The affine transformation applied in the previous step enables the CNN to focus on the most relevant regions of the environment and extract robust and reliable features, regardless of the object's position, orientation, or scale within the input image[39]. By identifying and emphasizing salient features within the localized glimpses, The feature extraction process furnishes the agent with a detailed and comprehensive understanding of the current state of the environment. Additionally, the feature extraction process enhances the ability to recognize and concentrate on the state's most impactful patterns and structures.

Attention mechanism

This process begins with concatenating the glimpses extracted by the *Extraction Network* with the hidden cell information from the previous time step (h_{t-1}). The combined data is processed through a fully connected layer, reducing its dimensionality. As a result, a context vector is formed, encapsulating both the hidden cell information and the extracted glimpses. The context vector is then subjected to the multi-head attention mechanism, which directs attention to different features extracted from various glimpses.

The adapted *Attention Mechanism*[42] captures global information from glimpses and integrates hidden cell data from previous time steps – a design choice inspired by the DARQN algorithm. Using a multi-head attention mechanism, the model creates a context vector that processes and weighs multiple relationships between glimpses and the hidden cell temporal information from the LSTM concurrently. This mechanism allows the model to leverage the memory of the agent's past observations alongside the features extracted from the glimpses. Consequently, the model can concentrate on the most relevant state regions, considering both the agent's past experiences and current state.

Gating mechanism

The *Gating Mechanism* in our model orchestrates the fusion of information derived from the glimpses' features, represented by the context vector, and the glimpses' locations. Both streams undergo a similar process: they are passed through a fully connected (FC) layer that applies the Relu activation function and subsequently through a second FC layer that applies a Sigmoid activation function. The second FC layer producing gating values between (0, 1) is crucial in emphasizing or suppressing different features or locations.

Rather than employing an element-wise multiplication between the extracted features and their glimpse location – similar to the EDRAM, AG-DQN employs the gating mechanism to the extracted features and the glimpses' locations, then concatenates these two data streams using a final FC layer. This action creates a unified data representation that merges the context vector and the glimpses' locations, each modulated by its respective gating value.

Gating mechanisms have demonstrated their versatility and effectiveness across various tasks and domains. For instance, they have outperformed traditional models in the field of natural language processing[43] and facilitated the training of deep neural networks by modulating the information flow within the network[44]. These instances highlight the efficiency of gating mechanisms in learning complex hierarchical features via the adaptive combination of information from diverse sources and layers.

Our approach retains the adaptivity inherent in conventional gating mechanisms and offers an increased capacity for capturing more complex relationships. The gating mechanism's capacity to adaptively weigh the significance of each source of information based on the context enhances decision-making and promotes improved learning.

Recurrent network

The *Recurrent Network* in the AG-DQN architecture comprises two LSTM layers. This layered processing enables the network to capture various levels of temporal patterns, with the first LSTM layer receiving the context vector g_t as the input and splitting the output into two distinct streams. The first stream, containing the hidden cell state (h_t) and the cell output (c_t), is forwarded to the *Q-Network* and the second LSTM layer for processing. The second stream, solely consisting of the hidden cell state information (h_t), is directed to the attention mechanism within the Glimpse Network in the next time step. This design choice, inspired by the DARQN algorithm[9], ensures that the agent retains the memory of vital lower-level temporal information across time steps.

The second LSTM layer utilizes the abstracted information from the first layer. Its output is directed towards the *Emission Network*, which predicts the glimpse locations for the next time step. By incorporating the refined information from the second LSTM layer, the *Emission Network* can generate more accurate glimpse locations, improving the overall performance of the AG-DQN agent in the reinforcement learning environments.

Emission network

Composed of fully connected layers, the *Emission Network* takes input from the second LSTM layer of the *Recurrent Network*, which encapsulates a rich blend of the agent's spatial and temporal experiences. The *Emission Network* primary role involves predicting the subsequent glimpses' locations, compensating for the lack of adaptability in traditional DQN approaches.

The *Emission Network* processes the input from the *Recurrent Network*, distilling spatial and temporal patterns to generate coordinates for the next glimpses' locations normalized to [-1, 1] by employing a *Tanh* activation function in the final layer, which yields excepted range for all potential glimpses' locations for improved convergence.

Therefore, the *Emission Network* guides the agent's attention towards salient areas within the environment and reduces the necessity of processing the entire scene while refining the glimpses' locations predictions as the agent collects experience from its interactions with the environment. This mechanism results in more accurate and context-aware glimpse selections in future steps.

Q-network

The *Q-Network*, composed of several fully connected layers, functions as the decision-making hub for the agent within the reinforcement learning environment. It receives the input from the first LSTM layer of the *Recurrent Network* analogous to the EDRAM classification network component, thereby obtaining insights into the agent's current state and its history of interactions. These fully connected layers integrate the *Recurrent Network's* temporal information and facilitate the Q-values computation for each potential action. These Q-values, representing the estimated cumulative rewards the agent could accrue by executing specific actions in the current state, inform the agent's selection of actions.

The *Q-Network* refines its Q-values based on the agent's interactions with the environment, embodying the dynamism of AG-DQN. As the agent accumulates experiences and deepens its understanding of the environment, the *Q-Network* iteratively fine-tunes its reward expectations. This iterative learning and refinement process encourages enhanced decision-making capabilities, improving agent performance in complex reinforcement learning tasks.

Theoretical analysis and justification

AG-DQN theoretically advances reinforcement learning-based UAV navigation by addressing critical challenges associated with high-dimensional visual inputs and partially observable states. Conventional DRL algorithms typically process entire images uniformly, inadvertently diminishing critical features with irrelevant information, thereby complicating the state representation and decision-making process. Unlike conventional attention models such as the Enriched Deep Recurrent Attention Model (EDRAM) [10], which iteratively applies a single-glimpse strategy to sequentially explore different parts of the same image, AG-DQN uniquely employs a dynamic multi-glimpse attention strategy. Specifically, AG-DQN simultaneously predicts multiple glimpse locations dynamically, enabling the extraction of salient information from diverse spatial regions of each image at once. This theoretically provides significant advantages over single-glimpse iterative approaches by improving spatial coverage, capturing richer contextual information simultaneously, and enhancing the quality and diversity of the extracted features.

From an information-theoretic viewpoint, dynamically predicting multiple glimpse locations simultaneously allows AG-DQN to maximize mutual information between selected regions and the agent's internal state representation, effectively reducing uncertainty and entropy in learned embeddings [45]. Moreover, utilizing partial image processing through multiple smaller glimpses instead of the full image achieves computational efficiency and better state abstraction by prioritizing informative regions over redundant areas. This selective processing aligns with cognitive neuroscience theories, demonstrating how selective attention mechanisms optimize cognitive resources and enhance decision-making efficiency and accuracy by focusing on task-relevant salient information [46,47].

Furthermore, AG-DQN's recurrent LSTM architecture theoretically addresses the limitations inherent in partially observable environments, explicitly modeled as a POMDP. Theoretical studies confirm that recurrent architectures like LSTMs facilitate temporal credit assignment by integrating information across multiple timesteps, thereby providing stable and accurate estimates of state values and actions in partially observable scenarios [8,48].

Additionally, the AG-DQN's Glimpse Network integrates several theoretically significant components that enhance its effectiveness. First, using the Spatial Transformer Network [39] theoretically improves robustness to spatial variations by dynamically adapting glimpses to relevant image regions regardless of object scale, rotation, or translation. This capability theoretically enhances generalization across diverse environmental conditions by explicitly learning spatial invariances.

AG-DQN also incorporates an advanced attention mechanism that leverages both the glimpses extracted from the current observation and the hidden state from the previous timestep. This form of temporal-spatial attention enables the network to explicitly integrate current spatial observations with historical context, thereby effectively addressing the challenges of temporal dependencies and partial observability. This combined attention approach has been shown to enhance the consistency and stability of state representations over time [9,42].

Lastly, the gating mechanism employed in AG-DQN represents a theoretical improvement in feature integration efficiency. By adaptively combining the attention mechanism outputs and glimpse locations, the gating mechanism explicitly modulates information flow and balances the relevance of spatial locations and extracted features dynamically. Theoretical studies indicate that such adaptive gating facilitates better hierarchical representation learning, improves convergence stability, and enhances the model's capacity to learn complex mappings from partial visual observations to decision-making policies [43,44].

Thus, AG-DQN's novel approach of dynamically predicting multiple glimpse locations simultaneously, combined with adaptive temporal attention, offers superior theoretical capabilities compared to conventional DRL methods and attention models, leading to improved efficiency, effectiveness, and accuracy in complex visual navigation tasks.

Evaluation

This section first outlines the experimental setup for the AG-DQN algorithm assessment. Next, a detailed presentation of the derived results is provided, with a thorough analysis delving into the algorithm's performance compared to other algorithms with similar traits. Experiments further investigate the implications of the algorithm's glimpse settings and the influence of various DQN extensions on the AG-DQN reward. Finally, the section concludes with a discussion of the overall findings to understand the combined impact of these experiments.

Experimental setup

The proposed AG-DQN algorithm is evaluated using autonomous UAV visual navigation in complex 3D indoor environments, although its applicability extends to various visual RL tasks where image interpretation is crucial for state representation. The experimental implementation utilizes the modular off-policy DRL framework VizNav [5], specifically adapted to support discrete action spaces mandated by Agile-DQN for consistent and reproducible evaluation in this study.

The virtual 3D simulation environment used for UAV navigation experiments is constructed using the Unreal Engine [49] integrated with the Microsoft AirSim flight simulator [50], as depicted in Fig. 4.

The primary objective across all experimental scenarios is successful UAV navigation to a predetermined finish line, with minimum steps and no collisions. Key simulation and training parameters, including all relevant experimental values, are summarized in Table 1.

In our experimental setup for the evaluation of AG-DQN and comparative RL algorithms, two key metrics have been selected for a comprehensive assessment of performance in autonomous UAV navigation within complex 3D indoor environments:

- Average Discounted Reward per Episode is a primary metric widely recognized and employed across numerous studies, recording the discounted reward of each episode[6–9,11,12,14–17,51]. This metric captures the average cumulative reward the UAV obtains throughout an episode, effectively reflecting the agent's performance in achieving its objectives. By applying a discount factor over time, the metric allows the agent to prioritize immediate rewards while still accounting for the value of future rewards. This balance emphasizes the algorithm's capability to navigate toward short-term objectives while considering long-term outcomes.
- Average Flight Distance per Episode is a secondary metric for measuring navigational efficiency under various operational settings. The flight distance per episode is employed to assess the impact of different glimpse configurations. This metric quantifies the total distance the UAV covers in each episode before reaching a terminal state.

Figure 4. 3D indoor environment viewed from different angles, constructed using Unreal Engine for AG-DQN evaluation.

Parameter group	Parameter	Value
Training	Number of episodes	10,000
	Maximum steps per episode	20
	Discount factor (γ)	0.999
Exploration (ϵ-greedy)	Initial ϵ	0.99
	Final ϵ	0.01
	ϵ decay rate	0.0005
Replay Buffer	Replay buffer size	32,768
	Minimal priority (ϵ)	0.001
	Priority exponent (α)	0.6
	Importance sampling exponent (β)	0.4
	β increment per sampling	0.0001
	Minibatch size	32
Reward	Collision penalty	-20
	Destination reward	+10
	Forward movement reward	+2
	Sideways movement penalty	-1
Optimizer	Optimizer	Adam (AMSGrad)
	Learning rate	1×10^{-4}
AirSim Simulator	Action displacement	1 m/step
	Altitude and yaw control	Autonomous (fixed)
	Actions	Left, Right, Forward
Image Specifications	Image height (H)	227
	Image Width (W)	227
	Color channels (D)	3 (RGB)

Table 1. Detailed simulation and training parameters grouped by category.

Performance analysis

This section embarks on an in-depth examination of AG-DQN's performance in a virtual 3D indoor environment, utilizing the average reward as the evaluation metric. The evaluation follows a comparative approach, benchmarking the performance of AG-DQN against established algorithms that have common characteristics (i.e. temporal dependencies and attention mechanism) with AG-DQN, namely DRQN[8] and DARQN[9]. The intent of this strategy is not to augment the overall performance of UAV navigation but to analyze the AG-DQN performance as a specialized reinforcement learning strategy for navigation tasks, compared to analogous algorithms with shared traits.

This study further explores the impact of varying numbers and sizes of glimpses, a critical step towards unveiling AG-DQN's unique competencies and advantages, widening its applicability across various use cases. The research also examines the potential enhancements conferred by integrating various DQN extensions into

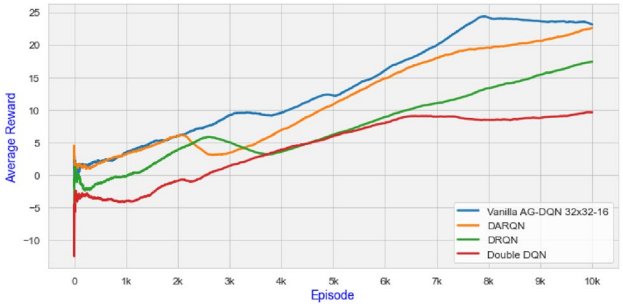


Figure 5. Vanilla AG-DQN using 16 glimpses of size 32×32 compared to other DQN algorithms.

Settings #	Glimpse size	# of Glimpses	Total pixels	Coverage %
1	16×16	16	4,096	8%
2	16×16	64	16,384	32%
3	16×16	96	24,576	48%
4	32×32	4	4,096	8%
5	32×32	16	16,384	32%
6	32×32	24	24,576	48%

Table 2. Various glimpses settings used by AG-DQN.

the AG-DQN algorithm. Specifically, it assesses the impact of Dueling Networks[15], Prioritized Experience Replay (PER)[14], Noisy Nets[17], and Distributional DQN (C51)[16].

Comparative analysis of AG-DQN and other DQN variants

This subsection emphasizes a comparison between Vanilla AG-DQN, utilizing 16 glimpses of size 32×32, and DRQN[8], and DARQN[9]. Additionally, Double DQN[7] is included in the evaluation to provide a baseline for these experiments. While DRQN, DARQN, and Double DQN algorithms process the entire state/image, AG-DQN uniquely employs selective and dynamic attention to 32% of the total image size 227×227(51529 pixels). AG-DQN accomplishes this by extracting salient features from 16384 pixels, which are covered by 16 glimpses of size 32×32. This ability to focus on distinct regions of the image sets AG-DQN apart and underscores its value in environments where significant details may be distributed across the state space.

As illustrated in Fig. 5, AG-DQN outperforms other techniques by securing higher rewards throughout the course of the experiment. In this comparative evaluation, a performance hierarchy emerges, with AG-DQN yielding the highest rewards, followed by DARQN and DRQN. As anticipated, Double DQN, serving as the baseline algorithm, ranks at the lower end of the performance spectrum.

Despite utilizing less than one-third of the entire image, AG-DQN yields better results due to the algorithm's selective attention mechanism that facilitates a concentrated analysis of the state's most significant characteristics. This mechanism gives AG-DQN an advantage in complex environments where processing the entire state/image may not be practical or efficient.

It is essential to highlight that AG-DQN demonstrates clear and stable convergence around episode 8000, achieving a consistent average reward of approximately 22, significantly outperforming DARQN, DRQN, and Double DQN throughout training. In contrast, DARQN and DRQN exhibit gradual improvements; their slower convergence rates and consistently lower performance reinforce AG-DQN's superior sample efficiency and effectiveness. Consequently, additional training episodes for these methods would likely not alter the established performance hierarchy, especially given AG-DQN's clear advantage in both convergence speed and reward magnitude. Moreover, the training duration aligns with common practice in DRL research, ensuring fair and practical comparative analysis [6,8,9].

Examining glimpse size impact on AG-DQN performance using average reward

This subsection investigates the influence of the number and size of glimpses on AG-DQN's performance. Given the essential role of the number and size of glimpses in the AG-DQN algorithm, these parameters essentially define the proportion of the image leveraged to extract pertinent features. Thus, the glimpses shape the state representation that the agent leverages for decision-making, ultimately influencing the reward outcome.

Two distinct sets of experimental configurations were considered to elucidate the influence of the number and size of glimpses on AG-DQN's performance, as depicted in Table 2. The first set, defined by settings (1–3), employs 16, 64, and 96 glimpses, each with a consistent size of (16×16). In contrast, the second set, represented by settings (4–6), employs 4, 16, and 24 glimpses, each with a uniform size of (32×32). The total number of pixels attended to by AG-DQN, referred to as *Total Pixels*, is derived by multiplying the *Glimpse Size* by the *# of Glimpses*. Meanwhile, the proportion of the entire image that these glimpses cover, or the *Coverage %*, is

calculated by dividing the *Total Pixels* by the total *Image Size* ($227 \times 227 = 51{,}529$ pixels) and then multiplied by 100 to express it as a percentage.

In order to further interpret these results, it is significant to note that the pairs of settings, namely (1,4), (2,5), and (3,6), are identical in terms of coverage percentage, i.e., the proportion of the image from which these glimpses derive features. This enables a direct comparison of the performance of the AG-DQN algorithm using different glimpse sizes and numbers while keeping the total pixel count constant.

The outcomes from the first set of experiments, corresponding to settings (1–3), are depicted in Fig. 6a. It can be observed that AG-DQN yields higher rewards with 64 glimpses (setting #2), followed by configurations using 16 and subsequently 96 glimpses. Results from the second set, corresponding to settings (4–6), are shown in Fig. 6b. AG-DQN achieves the highest rewards with 16 glimpses (setting #5). Notably, the performances achieved with 4 and 24 glimpses exhibited near parity, with the configuration using 4 glimpses slightly improving the reward compared to 24 glimpses.

Comparing AG-DQN using the same coverage percentage but with different glimpse settings, Fig. 7a illustrates the performance of glimpse settings (1,4), covering 8% of the state image. Next, Fig. 7b compares glimpse settings (2,5) that cover 32% of the state image. Lastly, Fig. 7c offers a performance comparison for glimpse settings (3,6) covering 48% of the state image.

In analyzing the performance of AG-DQN with glimpses that have equivalent coverage percentages, a consistent advantage of larger glimpse sizes (32×32) is apparent, yielding higher rewards. These observations carry several implications. For one, the dimensions and count of glimpses directly dictate the portion of the image used for feature extraction, inherently affecting the algorithm's performance. While a larger number of glimpses allows AG-DQN to sample from a wider array of image regions, this does not automatically guarantee improved performance. Interestingly, the benefit of larger glimpse sizes, despite maintaining the same total pixel count, could be attributed to their capacity to capture more contextual data within each region. This comprehensive information capture enables a more accurate state evaluation, refining the policy selection.

These experiments show that an optimal equilibrium between the number and size of glimpses is crucial in AG-DQN's performance. This balance might fluctuate based on the specific task or environment; however, the findings demonstrated through the results offer invaluable guidance for configuring AG-DQN in similar visual navigation tasks. As shown in Fig. 8, the best performance was achieved using 16 glimpses of size (32×32) with 32% state coverage.

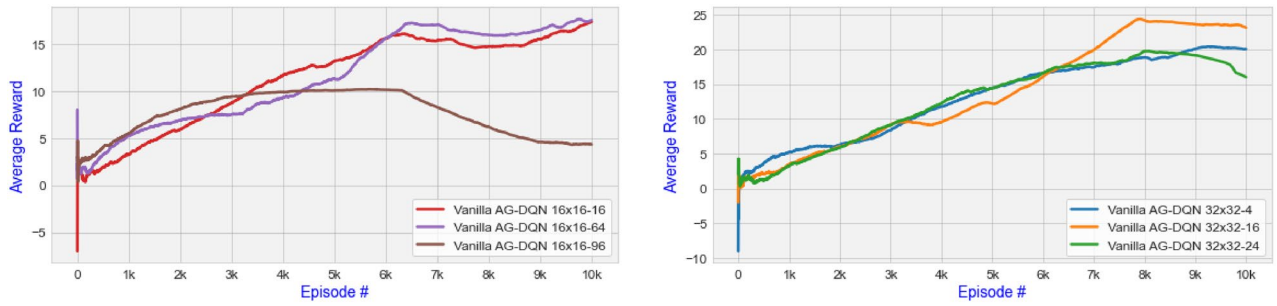
However, it is worth noting that while larger glimpse sizes were more effective within the UAV navigation context, the optimal glimpse size selection could be significantly influenced by the task at hand-particularly by the nature and distribution of the features extracted from the state image. As such, the choice of glimpse size ought to be adapted based on the distinctive demands of the task and environment.

Examining various DQN extensions impact on AG-DQN performance

Although the DQN extensions are well-known, this subsection briefly evaluates their direct impact on AG-DQN to identify whether incorporating them enhances or impairs AG-DQN performance.

- Dueling DQN: The AG-DQN incorporated the Dueling DQN approach by implementing separate paths within the Q-Network for the state-value functions $\boldsymbol{V(s)}$ and advantage-value functions $\boldsymbol{A(s,a)}$. The Q-values were then generated by combining these two streams in the final layer.
- Noisy Net: The Noisy Net technique was applied to the FC layers of the AG-DQN's Q-Network, introducing noise to stimulate exploration.
- Distributional DQN: The Categorical DQN (C51) approach was utilized with 51 discrete atoms, enabling the Q-Network to output a distribution of probabilities over these atoms for each action, as opposed to a vector of Q-values in vanilla AG-DQN.
- PER: The AG-DQN model was also modified take advantage of Prioritized Experience Replay (PER), prioritizing the replay of the most significant experiences.

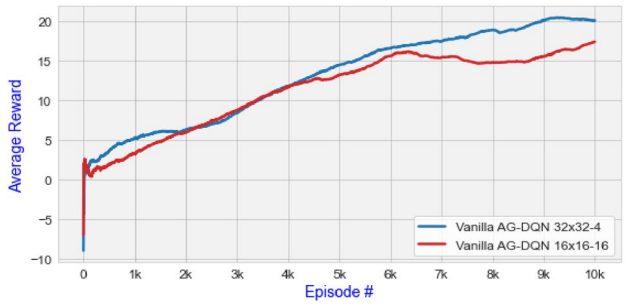
As demonstrated in Fig. 9, among the extensions tested, only Dueling AG-DQN delivered improved rewards over the base AG-DQN due to the model's ability to independently assess state values and action advantages, a



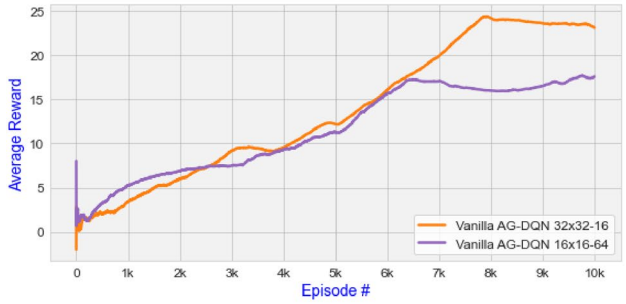
(a) Different number of glimpses of 16x16 size

(b) Different number of glimpses of 32x32 size

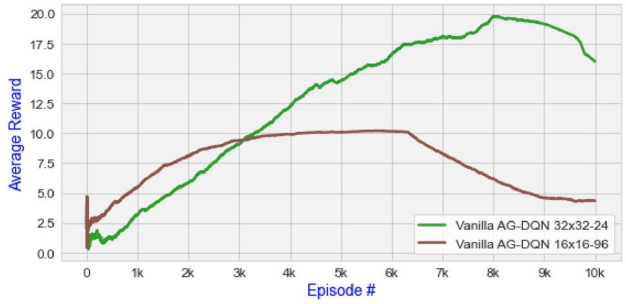
Figure 6. Comparing Vanilla AG-DQN of the same size using different numbers of glimpses.



(a) Different number of glimpses of 4096 pixels



(b) Different number of glimpses of 16384 pixels



(c) Different number of glimpses of 24576 pixels

Figure 7. Comparing Vanilla AG-DQN of the different number of glimpses and different sizes that have the same number of pixels count.

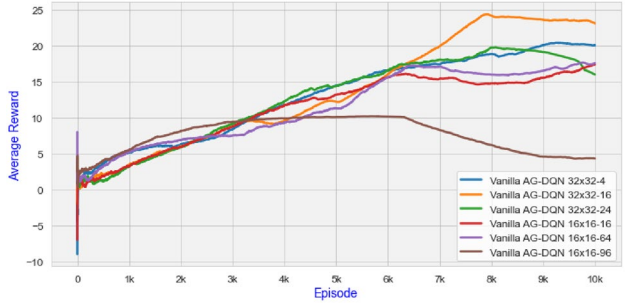


Figure 8. Comparing all Vanilla AG-DQN of different glimpse sizes and number of glimpses.

capability particularly effective for tasks where state value largely outweighs the significance of individual action choices. This configuration can enhance performance by highlighting key state representation elements.

The other DQN extensions (i.e. Noisy Net, Distributional DQN, and PER) did not enhance the reward. In the context of the navigation task, the Noisy Nets technique's encouraging extensive exploration may have been

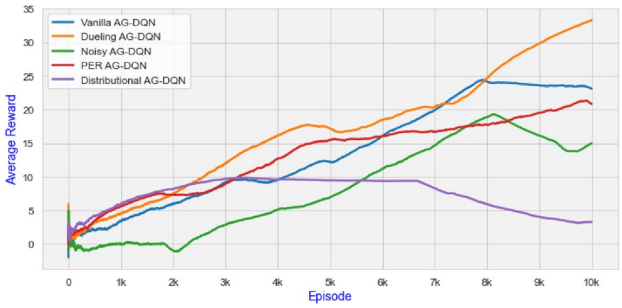


Figure 9. Comparing variations of AG-DQN using different DQN extensions.

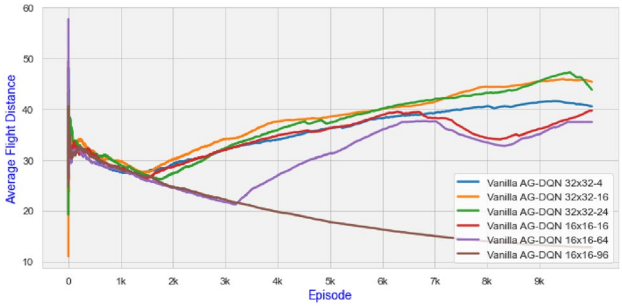


Figure 10. Examining the impact of various glimpses settings on the average flight distance.

counterproductive, leading to increased instances of the drone colliding with obstacles or walls. It is possible that the random perturbations introduced by Noisy Nets proved disadvantageous for this specific task.

In the AG-DQN framework, the Categorical DQN (C51) did not improve the reward, which can be attributed to the task environment and reward structure. Specifically, AG-DQN operates within a sparse reward context, providing intermittent feedback. This situation can challenge C51's ability to model the complete distribution of future rewards, leading to increased variance in the learned distributions. This complexity is compounded in UAV navigation, characterized by high-magnitude rewards. Furthermore, the recurrent LSTM architecture of AG-DQN, which incorporates temporal dependencies, may not naturally align with C51's assumption of discrete action-value distributions, thereby reducing the performance of C51 within the AG-DQN framework, despite its proven effectiveness in denser reward environments like Atari games[16].

These experimental findings lead to further investigation into another DQN extension, Prioritized Experience Replay (PER), which prioritizes experiences based on Temporal Difference (TD) error to optimize the learning of the Q-function. However, within the AG-DQN architecture, LSTM layers inherently maintain a memory capturing temporal dependencies within state sequences. This functionality could potentially minimize the advantages of PER. Furthermore, within an LSTM framework, the current Q-value is determined by combined factors: not only the current state and action but also past states due to the recurrent nature of LSTMs. This interplay suggests that the TD-error may not fully reflect the learning potential of an experience, rendering the prioritization process less effective.

Examining glimpse size impact on AG-DQN performance using average flight distance

This subsection presents an analysis of glimpse size on AG-DQN's performance using the flight distance, as illustrated in Fig. 10, which offers a distinct perspective on navigation efficiency using various glimpse settings. Unlike the reward-based evaluation, flight distance here measures the total distance traveled by the UAV before an episode ends, whether due to reaching the target, a crash, or reaching the maximum limit of 21 episodes.

In this context, the 32×32 pixel glimpse configuration with 16 glimpses (32×32-16) emerges as a leading setup as elucidated in Fig. 10. This configuration not only demonstrated effective performance in terms of rewards but also achieved the longest average flight distance.

In contrast, the 32×32 pixel glimpse configuration with 24 glimpses (32×32-24), while showing close performance to the 32×32-16 setting in certain training phases, did not consistently yield a better reward in terms of the overall traveled distance. This suggests that while the number of glimpses plays a crucial role, an excessive count may not always translate to more efficient navigation.

The 16×16-96 glimpse configuration, aligning with prior observations using the average reward, was found to be less effective. It generally resulted in shorter average flight distance due to crashes, indicating that smaller glimpses, despite their higher count, might not effectively capture the necessary environmental context for optimal navigation.

These insights gathered from flight distance measurements complement the reward-based analysis, reinforcing the importance of optimizing the glimpse configuration in AG-DQN. The larger glimpses (32×32) with a moderate count (16 or 24) provide a balanced approach, ensuring comprehensive environment understanding while avoiding collisions.

Discussion

This study has extensively evaluated the AG-DQN against several state-of-the-art algorithms with similar traits, namely DRQN and DARQN. The results confirm that AG-DQN outperforms these models on a UAV navigation task. The recurrent architecture and adaptive glimpsing mechanism of AG-DQN are key contributors to its improved performance. The recurrent architecture empowers AG-DQN to integrate historical information, enhancing its understanding of temporal dependencies within the environment. On the other hand, the adaptive glimpsing mechanism optimizes the information extracted from the environment, enabling the model to focus on the most salient aspects of the state for decision-making.

Further analysis of AG-DQN's glimpsing mechanism revealed its relationship with performance. Our experiments identified that while both the number and size of glimpses directly impact the proportion of the image used for feature extraction, they also contribute differently to the algorithm's performance. It was found that larger glimpses of size 32×32 consistently yielded higher rewards than smaller glimpses of size 16×16, even when they covered the same percentage of the state image. This is attributed to the capacity of larger glimpses to capture more contextual information about each region, thus facilitating a more accurate state assessment and an improved policy. However, the optimal glimpse size might depend on the specifics of the task, and distribution of the features in the state image.

Our investigation also explored the impact of various DQN extensions on AG-DQN performance various DQN extensions. Only Dueling AG-DQN showed an improvement over vanilla AG-DQN. While other DQN extensions did not enhance AG-DQN's performance.

Overall, these findings underscore the robustness of AG-DQN in visual navigation tasks. They highlight the critical role of an optimal balance between the number and size of glimpses and the thoughtful selection of DQN extensions in maximizing AG-DQN's performance. While the optimal settings may vary depending on the specific task and environment, the insights from this study provide a valuable starting point for configuring AG-DQN for similar tasks.

Conclusion

This paper proposed Agile Deep Q-Network (AG-DQN) for autonomous drone navigation tasks, an area of growing importance for various applications. AG-DQN represents a significant contribution to the field of Deep Reinforcement Learning (DRL), particularly for high-dimensional input domains like UAV navigation. This algorithm's innovation lies in its dynamic multi-glimpse strategy that focuses on critical aspects of the state for decision-making, reducing the need for processing the entire state. Thus, AG-DQN efficiently manages high-dimensional input representation, enhancing model adaptability and performance. Moreover, AG-DQN introduces a dynamic temporal attention strategy, which effectively handles temporal dependencies among observations. This strategy offers a balance between focusing on recent observations and maintaining a historical perspective, thereby enabling the agent to adapt fluidly within changing environments.

AG-DQN outperformed other state-of-the-art methods in complex UAV navigation tasks in the empirical evaluation, like DRQN and DARQN. Furthermore, performance variations in AG-DQN have been tied to the number and size of glimpses, with larger glimpses leading to enhanced results, underscoring the importance of the adaptive glimpsing mechanism. Additionally, while incorporating the Dueling DQN extension improved performance, other extensions did not elevate AG-DQN's performance within the studied context.

While the results are promising, further research is needed to generalize these insights across different tasks and environments. Moreover, due to computational constraints, the experiments were conducted using a single random seed, limiting the analysis to single-run results without mean and variance calculations. Future research should incorporate multiple random seeds to provide a comprehensive assessment of robustness and generalizability. Additionally, while our current experiments establish AG-DQN's overall superior performance, conducting detailed component-wise ablation studies remains a valuable direction for future research, offering deeper insights into the contributions of individual architectural components. Future work will also focus on developing more sophisticated adaptive glimpsing mechanisms and exploring other reinforcement learning algorithms that can work in synergy with AG-DQN. Specifically, addressing the integration of AG-DQN with safety mechanisms for real-world UAV applications will ensure the transition from simulation-based validation to practical, safe deployment. Through this ongoing research, we move closer to creating fully autonomous UAVs capable of operating effectively and safely in complex, real-world environments. AG-DQN's potential extends beyond UAV navigation, promising to enhance decision-making in diverse domains, such as robotics and autonomous vehicles, wherever complex decision-making in high-dimensional spaces is required.

Data availability

Data are contained within the article.

Received: 6 August 2024; Accepted: 20 May 2025
Published online: 23 May 2025

References

1. AlMahamid, F. & Grolinger, K. Autonomous unmanned aerial vehicle navigation using reinforcement learning: A systematic review. *Elsevier Eng. Appl. Artif. Intell.* **115**, 105321. https://doi.org/10.1016/j.engappai.2022.105321 (2022).
2. Wang, H., Song, S., Guo, Q., Xu, D., Zhang, X. & Wang, P. Cooperative motion planning for persistent 3d visual coverage with multiple quadrotor uavs. *IEEE Trans. Autom. Sci. Eng.* (2023).
3. O'Flaherty, R. & Egerstedt, M. Low-dimensional learning for complex robots. *IEEE Trans. Autom. Sci. Eng.* **12**(1), 19–27 (2014).
4. AlMahamid, F. & Grolinger, K. Reinforcement learning algorithms: An overview and classification. In *IEEE Canadian Conference on Electrical and Computer Engineering*, pp. 1–7 (2021). https://doi.org/10.1109/CCECE53047.2021.9569056
5. AlMahamid, F. & Grolinger, K. Viznav: A modular off-policy deep reinforcement learning framework for vision-based autonomous uav navigation in 3d dynamic environments. *MDPI Drones* **8**(5). https://doi.org/10.3390/drones8050173 (2024).
6. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. & Riedmiller, M.: Playing atari with deep reinforcement learning. arXiv:1312.5602 (2013).
7. Hasselt, H. Double q-learning. *Adv. Neural. Inf. Process. Syst.* **23**, 2613–2621 (2010).
8. Hausknecht, M. & Stone, P. Deep recurrent q-learning for partially observable mdps. AAAI Sequential Decision Making for Intelligent Agents (2015).
9. Sorokin, I., Seleznev, A., Pavlov, M., Fedorov, A. & Ignateva, A. Deep attention recurrent q-network. arXiv e-prints, 1512 (2015).
10. Ablavatski, A., Lu, S. & Cai, J. Enriched deep recurrent visual attention model for multiple object recognition. In *IEEE Winter Conference on Applications of Computer Vision*, pp. 971–978 (2017).
11. Ba, J., Mnih, V. & Kavukcuoglu, K. Multiple object recognition with visual attention. In *International Conference on Learning Representations* (2015).
12. Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M. & Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In *AAAI Conference on Artificial Intelligence* (2018).
13. Lin, L.-J. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Springer Mach. Learn.* **8**(3–4), 293–321 (1992).
14. Schaul, T., Quan, J., Antonoglou, I. & Silver, D. Prioritized experience replay. arXiv:1511.05952 (2015).
15. Wang, Z. et al. Dueling network architectures for deep reinforcement learning. *Int. Conf. Mach. Learn.* **48**, 1995–2003 (2016).
16. Bellemare, M.G., Dabney, W. & Munos, R. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pp. 449–458 (2017)
17. Fortunato, M., Azar, M.G., Piot, B., Menick, J., Hessel, M., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., Blundell, C. & Legg, S. Noisy networks for exploration. In *International Conference on Learning Representations* (2018).
18. Placed, J.A., Strader, J., Carrillo, H., Atanasov, N., Indelman, V., Carlone, L. & Castellanos, J.A. A survey on active simultaneous localization and mapping: State of the art and new frontiers. *IEEE Trans. Robot.* (2023).
19. Shen, G., Lei, L., Zhang, X., Li, Z., Cai, S. & Zhang, L. Multi-uav cooperative search based on reinforcement learning with a digital twin driven training framework. *IEEE Trans. Vehic. Technol.* (2023).
20. Liu, Z. et al. Visuomotor reinforcement learning for multirobot cooperative navigation. *IEEE Trans. Autom. Sci. Eng.* **19**(4), 3234–3245 (2021).
21. Yan, Z., Kreidieh, A. R., Vinitsky, E., Bayen, A. M. & Wu, C. Unified automatic control of vehicular systems with reinforcement learning. *IEEE Trans. Autom. Sci. Eng.* **20**(2), 789–804 (2022).
22. Wang, X. et al. Deep reinforcement learning-based air combat maneuver decision-making: Literature review, implementation tutorial and future direction. *Springer Artif. Intell. Rev.* **57**(1), 1 (2024).
23. 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. Chen, Y., Dong, C., Palanisamy, P., Mudalige, P., Muelling, K. & Dolan, J.M. Attention-based hierarchical deep reinforcement learning for lane change behaviors in autonomous driving. In *CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1326–1334 (2019).
25. Huang, X., Chen, W., Zhang, W., Song, R., Cheng, J. & Li, Y. Autonomous multi-view navigation via deep reinforcement learning. In *IEEE International Conference on Robotics and Automation*, pp. 13798–13804 (2021).
26. Wei, K. et al. High-performance uav crowdsensing: A deep reinforcement learning approach. *IEEE Internet Things J.* **9**(19), 18487–18499 (2022).
27. Chen, C., Liu, Y., Kreiss, S. & Alahi, A. Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In *IEEE International Conference on Robotics and Automation*, pp. 6015–6022 (2019).
28. Josef, S. & Degani, A. Deep reinforcement learning for safe local planning of a ground vehicle in unknown rough terrain. *IEEE Robot. Autom. Lett.* **5**(4), 6748–6755 (2020).
29. Chen, Y., Liu, C., Shi, B. E. & Liu, M. Robot navigation in crowds by graph convolutional networks with attention learned from human gaze. *IEEE Robot. Autom. Lett.* **5**(2), 2754–2761 (2020).
30. Mousavi, S., Schukat, M., Howley, E., Borji, A. & Mozayani, N. Learning to predict where to look in interactive environments using deep recurrent q-learning. arXiv:1612.05753 (2016).
31. Mezghan, L., Sukhbaatar, S., Lavril, T., Maksymets, O., Batra, D., Bojanowski, P. & Alahari, K. Memory-augmented reinforcement learning for image-goal navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3316–3323 (2022).
32. Mayo, B., Hazan, T. & Tal, A. Visual navigation with spatial attention. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16898–16907 (2021).
33. Chipka, J., Zeng, S., Elvitigala, T. & Mudalige, P. A computer vision-based attention generator using dqn. In *IEEE/CVF International Conference on Computer Vision*, pp. 2942–2950 (2021). https://doi.org/10.1109/ICCVW54120.2021.00329
34. Shi, H. et al. Path planning of randomly scattering waypoints for wafer probing based on deep attention mechanism. *IEEE Trans. Syst. Man Cybern. Syst.* **53**(1), 529–541 (2022).
35. Huang, W., Zhang, C., Wu, J., He, X., Zhang, J. & Lv, C. Sampling efficient deep reinforcement learning through preference-guided stochastic exploration. *IEEE Trans. Neural Netw. Learn. Syst.* (2023).
36. Bonyadi, M. R., Wang, R. & Ziaei, M. Self-punishment and reward backfill for deep q-learning. *IEEE Trans. Neural Netw. Learn. Syst.* **34**(10), 8086–8093 (2022).
37. Singla, A., Padakandla, S. & Bhatnagar, S. Memory-based deep reinforcement learning for obstacle avoidance in uav with limited environment knowledge. *IEEE Trans. Intell. Transp. Syst.* **22**(1), 107–118 (2019).
38. Sutton, R.S. & Barto, A.G. Reinforcement Learning: An Introduction (MIT Press, 2018)
39. Jaderberg, M., Simonyan, K., Zisserman, A., et al. Spatial transformer networks. *Adv. Neural Inf. Process. Syst.* **28** (2015).
40. Keys, R. Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoust. Speech Signal Process.* **29**(6), 1153–1160 (1981).
41. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **60**(6), 84–90 (2017).
42. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. & Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30** (2017).
43. Dauphin, Y.N., Fan, A., Auli, M. & Grangier, D. Language modeling with gated convolutional networks. In *International Conference on Machine Learning*, pp. 933–941 (2017). PMLR
44. Srivastava, R.K., Greff, K. & Schmidhuber, J. Highway networks. arXiv:1505.00387 (2015).

45. Itti, L. & Baldi, P. Bayesian surprise attracts human attention. *Elsevier Vis. Res.* **49**(10), 1295–1306 (2009).
46. Itti, L., Koch, C. & Niebur, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(11), 1254–1259 (1998).
47. Desimone, R. & Duncan, J. Neural mechanisms of selective visual attention. *Annu. Rev. Neurosci.* **18**(1), 193–222 (1995).
48. Zhu, P., Li, X., Poupart, P. & Miao, G. On improving deep reinforcement learning for pomdps. arXiv:1704.07978 (2017).
49. Games, E. Epic Games Unreal Engine Home Page. https://www.unrealengine.com. (Accessed: 07.05.2024) (2024).
50. Microsoft: Microsoft AirSim Home Page. https://microsoft.github.io/AirSim/. (Accessed: 07.05.2024) (2024)
51. Sutton, R. S. Learning to predict by the methods of temporal differences. *Springer Mach. Learn.* **3**, 9–44. https://doi.org/10.1007/BF00115009 (1988).

Author contributions

Fadi AlMahamid (First Author): Conceptualization, Methodology, Formal Analysis, Software, Validation, Writing—Original Draft, Writing—Review and Editing; Katarina Grolinger (Corresponding Author): Supervision, Funding Acquisition, Writing—Review and Editing.

Funding

This research has been supported by NSERC under grant RGPIN-2018-06222.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to K.G.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.