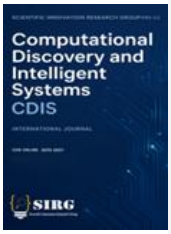

Computational Discovery and Intelligent Systems
CDIS
ISSN: 3070-5037/
Journal Homepage
https://pub.scientificirg.com/index.php/CDIS/index

Uncertainty-Aware Stochastic Hybrid World Models with Neural Map Memory for Autonomous Navigation in Partially Observable Grid Environments

Arwa Saad[a,1], Amer Ibrahim[b], Shashi Kant Gupta[c]
[a] Faculty of Computer Science, Nahda University, Beni-Suef City, Egypt. *Email:arwasaad812@gmail.com*
[b] *Department of Computer Science and Software Engineering, United Arab Emirates University, Al-Ain, UAE, Email: amer.ibrahim@uaeu.ac.ae*
[c] Computer Science and Engineering, Eudoxia Research University, USA, Email: raj2008enator@gmail.com

ABSTRACT

Autonomous navigation in partially observable environments remains a significant challenge for reinforcement learning agents due to incomplete observations, stochastic dynamics, and uncertainty in spatial perception. World models are robust at learning how the environment changes over time, and neural map architectures are competitive at representing spatial memory. However, most current methods treat these parts separately and don't often include explicit uncertainty estimation, thereby reducing navigation reliability and exploration efficiency. This paper introduces SHWM-NM (Stochastic Hybrid World Model with Uncertainty-Aware Neural Map Memory), a unified framework that integrates stochastic latent dynamics modeling, structured neural map memory, and multi-level uncertainty estimation to enhance autonomous navigation capabilities. The proposed architecture combines a stochastic hybrid world model with an uncertainty-aware neural map that explicitly represents spatial information and associated uncertainty. A policy learning module then employs these estimates to help with exploration and decision-making when not all information is available. When tested on MiniGrid-based navigation tasks, SHWM-NM significantly outperforms deterministic world model baselines. The proposed framework increases the average reward from 1.55 to 4.61, raises the success rate from 15% to 46%, and reduces the average trajectory length from 42.5 to 30.7 steps. Also, epistemic uncertainty decreased from 0.0073 to 0.0017 during training, reflecting a modest but consistent improvement in model confidence. During training, which means indicating improved modeling of environment dynamics. These results show that modeling stochastic dynamics, spatial memory, and uncertainty together in a single architecture demonstrates strong performance. This is a promising approach to making promising decisions and getting around in environments that aren't fully observable.

PAPER INFORMATION

HISTORY
Received: 7 January 2026
Revised: 11 March 2026
Accepted: 22 April 2026
Online: 25 April 2026

MSC
68T07; 68R10; 94A60; 68M15

KEYWORDS
Reinforcement Learning;
World Models;
Neural Map Memory;
Autonomous Navigation;
Stochastic Models.

[1] *Corresponding author: Faculty of Computer Science, Nahda University, Beni-Suef City, Egypt. Email: arwasaad812@gmail.com*

1 INTRODUCTION

Autonomous navigation in partially observable and stochastic environments remains a challenging problem in reinforcement learning and embodied intelligence. In such environments, an agent must perform tasks using incomplete

and noisy observations while interacting with an environment whose true states are not directly observable. Consequently, successful navigation requires the agent to reason about hidden environment states and predict future outcomes based on limited information [1]. Recent progress in world models has produced promising results in obtaining concise latent representations that capture the temporal dynamics of the environment. These models help agents figure out how the world will change over time [2]. PlaNet and Dreamer were two later methods that improved model-based reinforcement learning by adding stochastic latent dynamics. This made it easier to plan and make decisions when things are uncertain [3]. In parallel, research on memory-augmented navigation architectures has investigated ways to add spatial reasoning to learning-based agents. The Neural Map architecture is one interesting way to do this. It gives structured spatial memory that is in line with the real world. This lets an agent store, get, and change spatially grounded information while navigating, which enhances long-horizon navigation performance. Even with these improvements, most current methods still treat temporal modeling and spatial representation as two separate mechanisms. World model-based methods mainly concentrate on acquiring predictive environment dynamics but do not incorporate explicit spatial memory mechanisms[2]. On the other hand, neural map-based methods focus on spatial representation but usually assume that perception and transition dynamics are deterministic. More importantly, most of the methods that are already out there don't directly deal with uncertainty, which is a key feature of environments that are only partially observable [4]. Uncertainty arises at multiple levels in autonomous navigation tasks. An agent may be uncertain about its current state, the transition dynamics of the environment, or the completeness of its spatial coverage[5]. Classical probabilistic frameworks such as Partially Observable Markov Decision Processes (POMDPs) provide a principled solution for reasoning under uncertainty by maintaining a state belief. However, these methods suffer from high computational complexity when applied to high-dimensional sensory observations [6]. More recent reinforcement learning approaches have explored uncertainty estimation techniques such as bootstrapped ensembles and probabilistic value functions to improve exploration and decision-making. Nevertheless, these methods are typically applied in isolated components of the learning pipeline rather than being integrated into unified architecture. A major problem for current autonomous navigation systems is the lack of a single framework that integrates stochastic world modeling, spatial memory representation, and reasoning that accounts for uncertainty. Agents that don't take uncertainty into account when making predictions often become too sure of themselves, waste time exploring, and make decisions that aren't stable. This paper suggests SHWM-NM, a Stochastic Hybrid World Model with Uncertainty-Aware Neural Map Memory, to deal with these problems. It allows for autonomous navigation in environments that are only partially visible. The suggested framework combines a stochastic latent world model with a structured neural map memory that stores spatial data while clearly showing uncertainty at different levels of the agent's internal representation[7]. The suggested architecture models uncertainty in perceptual encodings, transition dynamics, and spatial memory updates, which lets uncertainty estimates spread throughout the learning process. The proposed framework is evaluated on navigation tasks in partially observable environments. Experimental results demonstrate that integrating stochastic world modeling with uncertainty-aware spatial memory observable improves learning efficiency, navigation success rate, and trajectory efficiency compared with deterministic world model approaches.

Unlike existing approaches that address world modeling, spatial memory, or uncertainty estimation in isolation, the proposed SHWM-NM framework introduces a tightly coupled architecture in which uncertainty is explicitly propagated across perception, latent dynamics, and spatial memory. This interaction enables more reliable decision-making under partial observability and distinguishes our work from prior approaches. The proposed framework is designed to be general and scalable and can be extended to more complex environments beyond grid-based settings.

This paper is structured as follows. Section 2 shows other work on world models, neural map architectures, and reinforcement learning that takes uncertainty into account. In Section 3, the navigation problem in environments that can only be partially seen is made more official. The proposed SHWM-NM model was introduced in Section 4. Section 5 describes the setup for the experiment, and Section 6 demonstrates the results. Section 7 shows and analyzes how to estimate uncertainty. Finally, Section 8 wraps up the paper and talks about what needs to be done next.

The contributions to this work can be summarized as follows:

- A unified framework that jointly integrates stochastic world modeling, spatial memory, and uncertainty estimation.
- An uncertainty-aware neural map memory where uncertainty explicitly guides memory read and write operations.
- A multi-level uncertainty propagation mechanism across perception, dynamics, and spatial memory.
- An uncertainty-driven policy learning strategy that improves exploration under partial observability.
- Comprehensive experiments demonstrating improved navigation performance compared to baseline methods.

2 Literature Review

Recent research has shown how important it is for reinforcement of learning systems, especially for robots, to be able to estimate uncertainty. Several uncertainty-aware reinforcement learning methods have been suggested to help people make better decisions when they must deal with noisy or incomplete information. These are Bayesian methods and ensemble learning methods that help measure uncertainty and make robotic actions more reliable [8] .The study in [9] introduces a reinforcement learning framework that acknowledges uncertainties, specifically integrating both epistemic and aleatoric uncertainties in policy learning. The proposed method allows robotic agents to make safer and more reliable choices in situations where mechanisms aren't certain by including uncertainty estimation in the decision-making process. The framework does make robotic control tasks more robust, but it doesn't include structured spatial memory or explicit world

modeling. This means it can't be used for complex navigation tasks in environments that aren't fully visible, such as navigating through cluttered spaces or dynamic environments where obstacles may change over time. For instance, [10] shows how deep learning robots can get around in places that are hard to see and understand. The suggested framework lets robots learn how to move around by looking at their surroundings, which makes them more flexible in situations that are unpredictable and change quickly. These types of methods are much better for navigation than the old-fashioned rule-based ones. These frameworks do focus on learning navigation policy, but they don't include stochastic world modeling or memory mechanisms that take uncertainty into account, which are crucial for adapting to dynamic environments and improving decision-making in unpredictable situations. The study in [11] investigates reinforcement learning-based navigation algorithms that allow robots to formulate motion policies directly from sensory inputs. The proposed method enhances navigation in unfamiliar environments by enabling the robot to adapt its behavior based on experience instead of relying solely on pre-established models. However, while these methods increase adaptability and learning potential, they often lack clear world modeling and structured spatial memory representations. This limitation affects their effectiveness in long-term navigation tasks. The study in[12] enhances the ability of reinforcement learning agents to learn environment dynamics through latent representations and improved exploration strategies. The approach aims to improve policy learning efficiency by enabling agents to better understand the consequences of their actions in uncertain environments. While such approaches improve learning efficiency and exploration capabilities, they typically do not incorporate explicit spatial memory representations or structured environment maps, which limit their applicability to long-horizon navigation tasks in partially observable environments. The research in [13] demonstrates robotic decision-making in ambiguous contexts. The methodology emphasizes enhancing the flexibility and efficiency of autonomous systems through the utilization of machine learning techniques for environmental interaction and control optimization. While these strategies enhance the resilience and flexibility of robotic systems, they typically lack explicit world models or organized spatial memory representations, which are crucial for long-term navigation in partially viewable settings. The research in [14] examines advanced mathematical models for the analysis and enhancement of learning algorithms in complex scenarios. The proposed framework aims to enhance decision-making effectiveness through mathematical analysis and algorithmic optimization techniques. These methods deepen our understanding of how learning systems function in theory and support the stability and convergence of models. However, the primary focus of these methods is on theoretical modeling and algorithmic analysis, rather than on integrating explicit spatial memory or world-model-based representations for autonomous navigation. In[15], a thorough examination of the function of world models in artificial intelligence systems. The research emphasizes the integration of deep learning and model-based reinforcement learning to facilitate agents in acquiring compact latent representations of environmental dynamics for planning and decision-making purposes. The work underscores the significance of predictive modeling and representation learning in the development of more efficient and scalable AI systems. Nonetheless, the study primarily concentrates on the conceptual and architectural dimensions of world models and does not explicitly examine spatial memory mechanisms or uncertainty-aware navigation strategies in partially observable contexts. In [16]presents an enhanced reinforcement learning framework aimed at improving agent performance through more effective representation learning and policy optimization. This approach aims to enhance learning efficiency and decision-making capabilities in dynamic environments by utilizing deep neural architectures in conjunction with knowledge-based reasoning mechanisms. Although the proposed method shows improved learning performance, it does not explicitly include structured spatial memory or uncertainty-aware modeling for navigation tasks in partially observable environments. **Table 1** presents a comparative analysis of representative navigation approaches, highlighting differences in world modeling strategies, memory representation mechanisms, and uncertainty handling techniques.

Despite recent advances in transformer-based models and diffusion-based planning, existing approaches still lack a unified framework that integrates stochastic world modeling, structured spatial memory, and uncertainty-aware reasoning. This gap motivates the proposed SHWM-NM framework.

Table 1. Comparison of Existing Navigation Approaches and the Proposed SHWM-NM Framework

Ref	Method	World Model Type	Memory / Mapping Representation	Uncertainty Handling	Contribution	Key Limitation
[9]	Uncertainty-aware policy learning for robotic control	Model-free reinforcement learning	No explicit spatial memory or mapping representation	Explicit modeling of epistemic and aleatoric uncertainty during policy learning	Improves robustness and safety of decision-making under uncertain observations	Does not include spatial memory or world modeling, limiting performance in complex navigation tasks
[10]	Learning-based autonomous navigation framework	Model-free reinforcement learning	Implicit spatial representation	Limited uncertainty modeling	Improves navigation adaptability in complex environments	Does not include explicit world models or structured spatial memory

[11]	Deep reinforcement learning for autonomous robot navigation	Model-free reinforcement learning	Implicit spatial representation	Limited uncertainty modeling	Improves navigation adaptability in unknown environments	Does not incorporate world models or structured spatial memory
[12]	Advanced reinforcement learning with latent representations	Latent representation learning	Implicit representation	Partial uncertainty handling	Improves exploration and policy learning efficiency	No explicit spatial memory or environment mapping
[13]	Learning-based framework for autonomous robotic decision-making	Model-free learning	Implicit environment representation	Limited uncertainty modeling	Improves adaptability and decision-making efficiency in robotic systems	Lacks explicit spatial memory and World modeling
[14]	Mathematical modeling and optimization of learning algorithms	Analytical model-based framework	No explicit spatial mapping	Mathematical treatment of uncertainty	Provides strong theoretical analysis and improves algorithm stability	Does not address spatial navigation or structured environment representations
[15]	Deep learning and reinforcement learning integration for world models	Latent predictive world models	No explicit spatial memory	Limited discussion of uncertainty modeling	Provides a comprehensive perspective on integrating world models with deep learning and RL	Does not address spatial memory or uncertainty-aware navigation
[16]	Deep reinforcement learning with knowledge-based optimization	Model-free deep reinforcement learning	No explicit spatial memory	Limited uncertainty modeling	Improves learning efficiency and decision-making in dynamic environments	Does not integrate spatial memory or uncertainty-aware world modeling

From **Table 1**, existing approaches address only isolated aspects of the navigation problem. Model-free methods emphasize policy learning but lack both world modeling and structured memory. In contrast, world model-based approaches capture environmental dynamics, yet they do not incorporate spatial memory or uncertainty-aware reasoning. Likewise, neural map-based methods offer structured spatial representations but typically assume deterministic transitions and fail to explicitly model uncertainty.

In comparison, the proposed SHWM-NM framework provides a unified architecture that integrates stochastic world modeling, structured spatial memory, and multi-level uncertainty estimation. Crucially, uncertainty is not treated as a secondary or auxiliary component; rather, it is explicitly modeled, propagated, and leveraged across all system modules, including perception, dynamics, memory, and policy learning. This comprehensive treatment of uncertainty constitutes the primary novelty of the proposed approach.

3 Proposed Methodology

The proposed pipeline consists of three main stages: (1) latent state encoding, (2) uncertainty-aware world modeling and memory update, and (3) policy learning conditioned on both latent predictions and spatial context.

Unlike prior approaches, where these components are designed and operated independently, the proposed SHWM-NM framework establishes explicit interactions among stochastic world modeling, uncertainty estimation, and spatial memory. This integration allows uncertainty to directly influence both memory updates and decision-making processes. The overall pipeline functions as follows: at each time step, the agent receives an observation that is encoded into a latent representation. This latent state is processed by the stochastic world model to predict future states, while the neural map updates the spatial memory through uncertainty-aware mechanisms. Subsequently, the policy network selects actions based on a combination of latent features, spatial memory, and uncertainty estimates.

To address the challenges of autonomous navigation in environments that are partially observable and stochastic, SHWM-NM (Stochastic Hybrid World Model with Uncertainty-Aware Neural Map Memory) is presented as an end-to-end

architecture. This framework integrates stochastic world modeling, neural spatial memory, and uncertainty-aware decision-making. The proposed framework combines a stochastic latent world model capable of learning predictive environment dynamics with a structured neural map memory that captures spatial representations of the environment. Architecture also includes uncertainty estimation across several components, such as perception, transition dynamics, and spatial memory updates. SHWM-NM consists of three main components: a stochastic latent world model designed to learn environment dynamics, an uncertainty-aware neural map memory that provides spatial representation and long-term memory, and a policy learning module that utilizes both latent predictions and spatial memory to guide navigation decisions.

3.1 Problem Definition

Autonomous navigation in partially observable and stochastic environments represents a fundamental challenge in reinforcement learning, where agents must operate under incomplete and noisy observations without direct access to the true underlying state.

At each time step t, the agent interacts with an environment characterized by an underlying latent state $s_t \in \mathcal{S}$, which is not directly observable. Instead, the agent receives an observation $o_t \in \mathcal{O}$generated according to the observation model:

$$o_t \sim p(o_t \mid s_t) \tag{1}$$

and executes an action $a_t \in \mathcal{A}$, resulting in a stochastic transition:

$$s_{t+1} \sim p(s_{t+1} \mid s_t, a_t) \tag{2}$$

where $\mathcal{S}$, $\mathcal{O}$, and $\mathcal{A}$denote the state, observation, and action spaces, respectively.

Environment Setup

We consider a grid-based navigation environment in which the observation at the time step $t$is defined as:

$$o_t \in \mathbb{R}^{3\times 8\times 8} \tag{3}$$

consisting of three channels encoding: (1) the agent's position, (2) the goal location, and (3) environmental noise.

The action space is discrete and defined as $\mathcal{A} = \{0,1,2,3\}$

corresponding to movements in the four cardinal directions (up, down, left, right).

The agent receives sparse rewards defined as:

$$r_t = \begin{cases} 10.0, & \text{if the agent reaches the goal} \\ -0.01, & \text{otherwise} \end{cases} \tag{4}$$

Each episode terminates either when the agent reaches the goal or after a maximum of 50-time steps.

Policy Learning under Partial Observability

Due to partial observability, the agent cannot rely solely on the current observation. Instead, it must condition its decisions on an internal belief representation:

$$\pi(a_t \mid h_t) \tag{5}$$

where $h_t$represents the history of observations, actions, and uncertainty estimates up to time step t. This belief representation serves as sufficient statistics for decision-making in partially observable environments.

Core Challenges

At each time step, the agent must simultaneously address three tightly coupled challenges:

1. Temporal Dynamics Modeling

The agent must learn a predictive model of the environment dynamics:

$$s_{t+1} \sim p(s_{t+1} \mid s_t, a_t) \tag{6}$$

which enables the prediction of future states under stochastic transitions.

2. Persistent Spatial Memory

To support long-horizon navigation, the agent maintains a structured spatial memory $M \in \mathbb{R}^{H\times W\times C}$

where $H$and Wdenote spatial dimensions and Crepresents the feature dimension. This memory encodes the spatial layout of the environment and stores information about previously visited locations.

3. Uncertainty Quantification

The agent must explicitly estimate uncertainty arising from both partial observability and stochastic dynamics. partial observability and stochastic transitions. The total uncertainty at the time step $t$is defined as:

$$U_t^{total} = U_t^{enc} + U_t^{dyn} + U_t^{map} \tag{7}$$

where:

- $U_t^{enc} = \frac{1}{K}\sum_{k=1}^{K}\sigma_{t,k}^2$
  (perceptual / aleatoric uncertainty)
- $U_t^{dyn} = \frac{1}{M}\sum_{m=1}^{M}(z_{t+1}^{(m)} - \bar{z}_{t+1})^2$
  (epistemic uncertainty)

- $U_t^{map} = \frac{1}{HW} \sum_{x=1}^{H} \sum_{y=1}^{W} U_{x,y}^{map}$ (spatial uncertainty)

Here, $\sigma_{t,k}^2$ denotes the variance of the k-th latent dimension in the perceptual encoding.
$z_{t+1}^{(m)}$ represents the prediction of the m-th ensemble model, and $\bar{z}_{t+1}$ is the ensemble mean.
$U_{x,y}^{map}$ denotes the uncertainty associated with spatial location (x, y).

3.2 System Overview

The suggested SHWM-NM architecture is meant to make it possible for strong autonomous navigation in environments that are only partially visible by combining stochastic world modeling, spatial memory, and uncertainty-aware reasoning into one system. The system learns a small, hidden representation of how the environment changes over time while keeping a structured memory of the space that stores long-term information about the environment. Also, uncertainty estimation is used in many parts of architecture, such as perception, transition dynamics, and memory updates. This integrated design enables the agent to guess what will happen in the future and make navigation choices while taking uncertainty into account. The overall structure is made up of three main parts: a stochastic world model that learns how the environment changes, an uncertainty-aware neural map memory that shows where mechanisms are, and a policy learning module that employs both latent predictions and spatial memory to help make navigation decisions.
The encoded latent representation is shared between the transition model and the neural map. The resulting predictions and uncertainty estimates are aggregated and passed to the policy network.

3.2.1 Stochastic World Model

In the proposed stochastic world model, raw observations are first processed through a convolutional encoder to generate a hybrid latent representation that combines deterministic and stochastic components [17]. The encoder architecture comprises three convolutional layers with channel dimensions $32 \rightarrow 64 \rightarrow 128$, kernel size 3×3, and stride 2, followed by separate linear projections for the deterministic and stochastic parts, allowing the model to capture both predictable dynamics and inherent uncertainty in observations [18].

3.2.1.1 Latent State Representation

The raw observation $o_t \in R^{3\times8\times8}$ is encoded into a hybrid latent state z_t defined as:

$$\mathbf{z}_t = \begin{bmatrix} \mathbf{d}_t \\ \mathbf{s}_t \end{bmatrix} \in R^{32} \quad (8)$$

Where:

- represents the deterministic component, capturing predictable features of the environment $\mathbf{d}_t \in R^{16}$
- represents the stochastic component, modeled as a Gaussian random variable $\mathbf{s}_t \in R^{16}$:

$$\mathbf{s}_t \sim \mathcal{N}\left(\boldsymbol{\mu}_t, \mathrm{diag}(\boldsymbol{\sigma}_t^2)\right) \quad (9)$$

This hybrid representation enables the model to reason about both uncertain transitions and reliability patterns in sequential data [19].

Figure 1 demonstrates the overall design of the proposed SHWM-NM (Stochastic Hybrid World Model with Uncertainty Aware Neural Map Memory) framework for self-driving cars in environments that are only partially visible. The agent interacts with the MiniGrid environment and gets RGB observations. A convolutional encoder processes these observations to create a hybrid latent representation that has both deterministic and stochastic parts. An ensemble transition model uses this latent state to predict how the environment will change while also accounting for uncertainty in the state transitions. An uncertainty aggregation module then combines the predicted latent features and uncertainty estimates. This module sends uncertainty information to different parts of the architecture. An uncertainty-aware neural map stores and updates the aggregated latent representation. This map acts as a spatial memory that keeps structured information about the environment that has been explored. The neural map reads data using attention mechanisms and writes data using uncertainty estimates, which facilitates the system to update spatial memory. Finally, a policy network employs the latent predictions, spatial memory, and uncertainty information to help it choose actions and explore, which enables it to make decisions when not all the information is available.

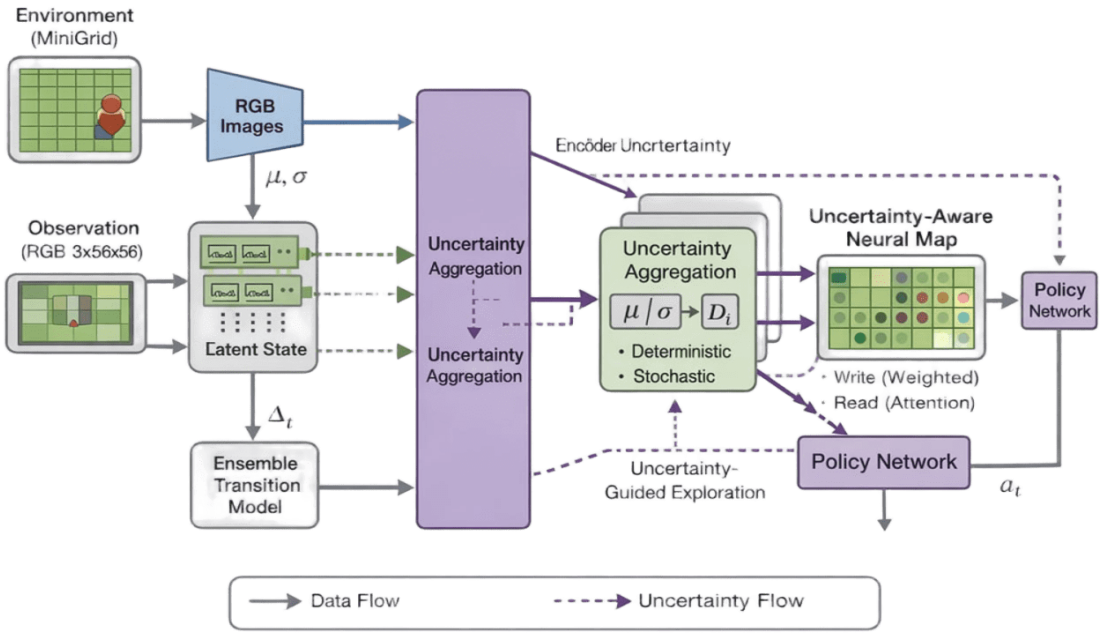

Figure 1. Proposed SHWM-NM Architecture

3.2.1.2 Variational Encoder

The stochastic component $\mathbf{s}_t$ is inferred via a variational encoder as:

$$q(\mathbf{s}_t \mid \mathbf{o}_t) = \mathcal{N}\left(\boldsymbol{\mu}_t, \text{diag}(\boldsymbol{\sigma}_t^2)\right), \boldsymbol{\mu}_t, \log\, \boldsymbol{\sigma}_t^2 = \text{Encoder}_\theta(\mathbf{o}_t) \tag{10}$$

where Encoder_θ is a convolutional neural network with learnable parameters θ, mapping the raw. observation $\mathbf{o}_t$ to the mean $\boldsymbol{\mu}_t$ and log-variance $\log\, \boldsymbol{\sigma}_t^2$ of the stochastic latent state [20].

3.2.1.3 Perceptual Uncertainty

Perceptual (aleatoric) uncertainty is quantified as the mean variance across latent dimensions:

$$U_{\text{enc}} = \frac{1}{D_s}\sum_{i=1}^{D_s} \sigma_{t,i}^2 \tag{11}$$

where $D_s = 16$ is the dimension of the stochastic latent vector $\mathbf{s}_t$. This captures the uncertainty inherent in it. the observations[21].

3.2.1.4 Stochastic Dynamics with Ensemble Models

To model epistemic uncertainty, we employ an ensemble of $M = 5$ transition models

$$p_m(\mathbf{z}_{t+1} \mid \mathbf{z}_t, \mathbf{a}_t) = \text{MLP}_m([\mathbf{z}_t; \text{one-hot}\,(\mathbf{a}_t)]), \quad m = 1, \ldots, 5 \tag{12}$$

Each MLP consists of two hidden layers with 256 units and ReLU activations. The ensemble captures the model. uncertainty due to limited training data or underexplored state-action regions [22].

3.2.1.5 Mean Prediction and Epistemic Uncertainty

The ensemble means prediction and epistemic uncertainty are computed as:

$$\overline{\mathbf{z}}_{t+1} = \frac{1}{M}\sum_{m-1}^{M} \mathbf{z}_{t+1}^{(m)} \tag{13}$$

$$U_{\text{dyn}} = \frac{1}{M}\sum_{m-1}^{M} \left\|\mathbf{z}_{t+1}^{(m)} - \overline{\mathbf{z}}_{t+1}\right\|^2 \tag{14}$$

where U_{dyn} represents the model (epistemic) uncertainty across the ensemble predictions [23].

3.2.1.6 Total World Model Uncertainty

The world model is responsible for predicting future latent states. The total uncertainty of the world model is computed by combining the normalized contributions of perceptual, dynamics, and neural map uncertainties

$$U_{\text{total}} = \frac{U_{\text{enc}}}{10} + \frac{U_{\text{dyn}}}{10} + U_{\text{map}} \quad (15)$$

where U_{map} is retrieved from the neural map (Section 4.2). This formulation ensures balanced integration. of aleatoric and epistemic uncertainties.

3.2.2 Uncertainty-Aware Neural Map Memory

The neural map provides structured spatial memory. To account for the long-term spatial structure and enable decision-making, the proposed framework utilizes an uncertainty-aware neural map memory. The neural map memory maintains spatial feature representations and estimates of uncertainty in a structured two-dimensional grid that is aligned with the environment. This mechanism enables the agent to maintain a spatial memory and explicitly represent uncertainty related to different parts of the environment [24].

3.2.2.1 Neural Map Definition

We maintain a two-dimensional neural map:

$$M \in R^{17\times17\times64}, U^{\text{map}} \in R^{17\times17} \quad (16)$$

where $\mathbf{M}_{i,j} \in R^{64}$ stores a 64-dimensional feature representation for spatial location (i, j), and $U_{\text{map}}^{i,j} \in .[0,1]$ denotes the uncertainty estimate associated with that location [25]. The map size is fixed in the current implementation to match the grid environment; however, the framework can be extended to variable-sized or hierarchical memory representations.

3.2.2.2 Uncertainty-Weighted Write Operation

When the agent visits the position (x_t, y_t), the neural map is updated using an uncertainty-aware write mechanism that integrates the current latent representation

$$\mathbf{M}_{x_t,y_t} \leftarrow \alpha\mathbf{M}_{x_t,y_t} + (1-\alpha)\left(1 - U_t^{\text{total}}\right)f(\mathbf{z}_t) \quad (17)$$

$$U_{\text{map}}^{x_t,y_t} \leftarrow \beta U_{\text{map}}^{x_t,y_t} + (1-\beta)U_t^{\text{total}} \quad (18)$$

where $\alpha = 0.5, \beta = 0.3$, and $f: R^{32} \rightarrow R^{64}$ It is a learned projection [26].

3.2.2.3 Attention-Based Read Operation

To retrieve spatial information, the model performs an attention-based read operation over the neural map. First, a query vector is generated:

$$\mathbf{q}_t = \text{MLP}_{\text{query}}(\mathbf{z}_t), \mathbf{q}_t \in R^{64} \quad (19)$$

Attention weights are computed as:

$$\alpha_{ij} = \frac{\exp\left(\frac{\mathbf{q}_t^{\top}\mathbf{M}_{ij}}{\sqrt{64}} - \gamma U_{map}^{i,j}\right)}{\sum_{i',j'}\exp\left(\frac{\mathbf{q}_t^{\top}\mathbf{M}_{i'j'}}{\sqrt{64}} - \gamma U_{map}^{i',j'}\right)} \quad (20)$$

The resulting context vector is obtained as:

$$\mathbf{c}_t = \sum_{i,j}\ \alpha_{ij}\mathbf{M}_{i,j} \quad (21)$$

The map uncertainty is computed as:

$$U_t^{map} = \sum_{i,j}\ \alpha_{ij}U_{map}^{i,j} \quad (22)$$

where $\gamma = 2.0$ penalizes spatial locations with high uncertainty [27].

3.2.2.4 Map Smoothness Regularization

To encourage spatial coherence, we apply a smoothness loss:[28]

$$\mathcal{L}_{\text{map}} = \frac{1}{N}\sum_{i,j}\left(\left\|M_{i,j} - M_{i+1,j}\right\|^2 + \left\|M_{i,j} - M_{i,j+1}\right\|^2\right) \quad (23)$$

3.2.3 Uncertainty-Driven Policy Learning

The policy network selects actions based on learned representations. The policy network conditions actions on latent state, spatial context, and uncertainty estimates.

3.2.3.1 Policy Network Architecture

The policy network takes concatenated inputs:

$$\pi\left(a_t \mid \left[z_t; c_t; U_t^{\text{total}}\right]\right) = \text{Softmax}\left(\text{MLP}_{\text{policy}}\left(\left[z_t; c_t; U_t^{\text{total}}\right]\right)\right) \tag{24}$$

where $\text{MLP}_{\text{policy}}$ has two hidden layers (128 units each) with LayerNorm and ReLU activations [29].

3.2.3.2 Uncertainty-Modulated Exploration

The exploration rate ϵ_t is modulated by uncertainty:

$$\epsilon_t = \epsilon_{\min} + (\epsilon_{\max} - \epsilon_{\min}) \cdot U_t^{\text{total}} \tag{25}$$

with $\epsilon_{\max} = 0.3, \epsilon_{\min} = 0.01$, and decay factor 0.995.

3.2.3.3 Policy Loss with Uncertainty Bonus

The policy is optimized using PPO with an uncertainty exploration bonus:

$$\mathcal{L}_{\text{policy}} = -E\left[\min\left(\frac{\pi_\theta(a_t \mid h_t)}{\pi_{\theta_{\text{old}}}(a_t \mid h_t)} A_t, \text{clip}\left(\frac{\pi_\theta(a_t \mid h_t)}{\pi_{\theta_{\text{old}}}(a_t \mid h_t)}, 1-\epsilon, 1+\epsilon\right) A_t\right)\right] \tag{26}$$

$$A_t = \sum_{k=0}^{\infty} \gamma^k \left(r_{t+k} + \eta U_{t+k}^{\text{total}}\right) - V(h_t) \tag{27}$$

where $\eta = 0.01$ is the uncertainty bonus weight, and $V(h_t)$ is a value function estimate.

3.2.4 Joint Training Objective

The complete training objective combines reconstruction, dynamics, memory, and policy losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \beta_{\text{KL}}\mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{dyn}} + \mathcal{L}_{\text{reward}} + \lambda_{\text{map}}\mathcal{L}_{\text{map}} + \mathcal{L}_{\text{policy}} \tag{28}$$

$$\mathcal{L}_{\text{recon}} = \|\text{Decoder}(z_t) - o_t\|^2 \tag{29}$$

$$\mathcal{L}_{\text{KL}} = D_{\text{KL}}(q(s_t \mid o_t) \| \mathcal{N}(0, I)) \tag{30}$$

$$\mathcal{L}_{\text{dyn}} = \|\bar{z}_{t+1} - z_{t+1}^{\text{true}}\|^2 \tag{31}$$

$$\mathcal{L}_{\text{reward}} = \|\hat{r}_t - r_t\|^2 \tag{32}$$

with $\beta_{\text{KL}} = 0.01, \lambda_{\text{map}} = 0.001$, and learning rates 3×10^{-4} for the world model and 1×10^{-4} for the policy.

3.2.5 Training Algorithm

The overall training procedure is summarized in **Algorithm 1.** The training process is divided into two phases: world model pre-training and policy learning.

1: **procedure** TRAIN
2: **Phase 1: Data Collection**
3: Collect $N = 500$ episodes using random policy → buffer D
4: Split D into train/val/test (70%/15%/15%)
5: **Phase 2: World Model Pre-training**
6: **for** epoch = 1 to 50 **do**
7: Sample batch $(o_t, a_t, r_t, o_{t+1}) \sim \text{D}_{\text{train}}$
8: Compute L_{total} (Eq. 4.20)
9: Update world model parameters θ
10: **end for**
11: **Phase 3: Policy Training**
12: **for** episode = 1 to 100 **do**
13: Reset environment and neural map

Collect trajectory using π_θ with uncertainty-modulated ϵ
Compute advantages with uncertainty bonus (Eq. 4.19)
Update policy parameters ϕ
Decay exploration rate: $\epsilon \leftarrow 0.995\epsilon$
end for
end procedure

3.3 Implementation Details and Experimental Setup

The MiniGrid environment is adopted as a controlled benchmark to evaluate the behavior of the proposed framework under partial observability. Although relatively simple, it encapsulates essential challenges inherent to navigation tasks. Notably, the proposed architecture is generalizable and can be extended to more complex, large-scale environments and continuous control settings without requiring fundamental modifications

3.3.1 Environment Specifications

We evaluate SHWM-NM on a custom 8×8 grid navigation environment implemented in PyTorch. The ensemble size is set to 5, balancing computational cost and uncertainty estimation quality. The observation space consists of three 8×8 channels: (1) agent position (one-hot), (2) goal position (one-hot), and (3) random noise $\sim$ U (0, 0.1). The agent starts at a random position $(x, y) \in [1, 6]^2$, with the goal similarly randomized but distinct from the start position.

Episodes terminate after 50 steps or upon reaching the goal, with reward $r_t = 10.0$ for goal achievement and $r_t = -0.01$ otherwise.

3.3.2 Network Architecture Details

- **Encoder:** Conv2d (3,32,3, stride=2) → ReLU → Conv2d (32,64,3, stride=2) → ReLU → Conv2d (64,128,3, stride=2) →ReLU → Flatten → Linear (128×1×1, 16) for μ_t and σ^2
- **Decoder:** Linear (32, 128×1×1) → ConvTranspose2d (128,64,3, stride=2) → ReLU → ConvTranspose2d (64,32,3, stride=2) → ReLU → ConvTranspose2d (32,3,3, stride=2) → Sigmoid
- **Transition Ensemble:** 5× MLP (32+4 → 256 → 256 → 32)
- **Reward Predictor:** MLP (32 → 128 → 64 → 1)
- **Policy Network:** MLP (32+64+1 → 128 → 128 → 4) with LayerNorm
- **Value Network:** MLP (32+64+1 → 128 → 64 → 1)

3.3.3 Training Hyperparameters

The SHWM-NM model learns with hyperparameters that make sure learning stays stable and exploration demonstrates strong performance in environments that are only partially visible. The world model and policy network have different learning rates, and a hybrid latent representation captures both dynamics and uncertainty. Regularization terms keep the learning process stable and the neural map consistent. Epsilon decay and an uncertainty bonus help guide exploration, and an ensemble transition model makes the system more stable. **Table 2** demonstrates the complete list of hyperparameters. The hyperparameters were determined based on prior studies and preliminary experiments to ensure stable training and reliable convergence. Specifically, the map size was set to 17×17 to match the scale of the environment, while the ensemble size was fixed at 5 to balance computational efficiency with the quality of uncertainty estimation.

More generally, the selected hyperparameters follow established practices in reinforcement learning. The ensemble size governs the trade-off between uncertainty estimation accuracy and computational cost, whereas the uncertainty bonus weight influences the agent's exploration behavior. Additionally, the map size defines the spatial capacity of the memory representation. Although a comprehensive sensitivity analysis is beyond the scope of this work, the consistent improvements observed across evaluation metrics indicate that the proposed framework is robust under the chosen hyperparameter settings.

Despite the absence of a full sensitivity analysis, the selected hyperparameters follow established practices in prior work, and the consistent improvements across multiple metrics indicate robustness to these settings.

Table 2. Training Hyperparameters

Hyperparameter	Value
World Model Learning Rate	3×10^{-4}
Policy Learning Rate	1×10^{-4}
Batch Size	32
KL Weight (β_{KL})	0.01

Map Smoothness Weight (λ_{map})	0.001
Discount Factor (γ)	0.99
Uncertainty Bonus (η)	0.01
Exploration ϵ_{max}	0.3
Exploration ϵ_{min}	0.01
Exploration Decay	0.995
Map Size	17×17
Feature Dimension	64
Latent Dimension	32 (16D + 16S)
Ensemble Size	5

3.3.4 Evaluation Metrics

The performance of the proposed framework is evaluated using a comprehensive set of metrics that assess prediction accuracy, reconstruction fidelity, uncertainty estimation, and navigation effectiveness.

Prediction accuracy is measured using Latent Mean Squared Error (MSE), defined as

$$\frac{1}{T}\sum_t \left\| z_t^{\text{pred}} - z_t^{\text{true}} \right\|^2 \tag{33}$$

and Reward Prediction MSE, defined as

$$\frac{1}{T}\sum_t (\hat{r}_t - r_t)^2 \tag{34}$$

Reconstruction quality is evaluated Reconstruction MSE, given by

$$\frac{1}{T}\sum_t \| \hat{o}_t - o_t \|^2 \tag{35}$$

Uncertainty estimation is quantified using the total uncertainty U_t^{total}, capturing the overall uncertainty at each time step. Navigation performance is assessed through Success Rate, defined as the percentage of episodes in which the agent reaches the goal, Average Reward, computed as

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T_i} r_t^{(i)} \tag{36}$$

and Trajectory Length, representing the average number of steps per episode.

Where T denotes the number of time steps, N is the total number of evaluation episodes, z_t^{pred} and z_t^{true} represent the predicted and ground-truth latent states, $\hat{o}_t$ and o_t denote the reconstructed and observed states, and $\hat{r}_t$ and r_t correspond to the predicted and actual rewards at time step t.

The computational cost of the proposed SHWM-NM framework is mainly driven by two components: the ensemble-based transition model and the neural map memory.

The ensemble transition model introduces additional overhead, with computational complexity increasing linearly with the number of ensemble members. While this design enhances uncertainty estimation and improves training stability, it results in higher runtime compared to deterministic counterparts.

In contrast, the neural map operates on a fixed-size grid and performs relatively lightweight read and write operations, leading to modest computational overhead in the current setting. However, its cost may grow as the map size increases in more complex environments.

Overall, the proposed framework incurs a higher computational cost than baseline methods, but this overhead is justified by the observed gains in performance and robustness.

Regarding scalability, architecture is not inherently restricted to small-scale environments and can be extended to more complex settings through several strategies, including:

- Adaptive or hierarchical memory representations,
- Reducing ensemble size where appropriate,
- Applying efficient model compression techniques.

Computational complexity scales linearly with the ensemble size K, i.e., $O(K)$, which introduces additional overhead compared to deterministic models.

Future work will focus on improving computational efficiency and systematically evaluating scalability in larger and more complex environments.

4 RESULTS AND DISCUSSION

All reported results are averaged over multiple independent runs and presented as mean ± standard deviation to capture performance variability. Although the experiments are conducted in a simplified environment, it serves as an appropriate testbed for evaluating the core components of the proposed framework under controlled conditions.

The observed performance improvements can be attributed to architectural design rather than environmental complexity. While a fixed map size is used in this study, the proposed architecture is inherently flexible and can be extended to larger environments through scalable memory representations.

The performance improvement is primarily attributed to the interaction between uncertainty estimation and spatial memory. Specifically, uncertainty-aware updates allow the agent to avoid overconfident predictions in unexplored regions, while the neural map provides structured long-term spatial context, enabling more efficient decision-making under partial observability. The goal of this study is not to claim state-of-the-art performance, but to demonstrate the effectiveness of integrating uncertainty-aware mechanisms within a unified framework. Although results are reported as mean ± standard deviation across multiple runs, formal statistical significance tests were not conducted. Therefore, the improvement observed should be interpreted with caution.

4.1 Quantitative Performance

As presented in **Table 3**, the SHWM-NM model has better performance in all evaluation indicators than the baseline methods. Specifically, the average reward is enhanced from 1.55 in the deterministic world model to 4.61, showing an increase of approximately 197%. Besides, the success rate is significantly enhanced from 15% to 46%, reflecting better reliability in task accomplishment. Moreover, the average trajectory length is reduced from 42.5 to 30.7, suggesting that the proposed method can lead to a more efficient navigation towards the target. In addition, the model presents lower latent prediction error (Latent MSE) and reconstruction errors, reflecting the effectiveness of the stochastic hybrid world model in learning more accurate dynamics in the environment. The relatively low standard deviation across runs indicates stable and consistent performance improvements. The consistent improvements across reward, success rate, and trajectory length indicate that the observed gains are not tied to a single metric and suggest stable behavior under the selected hyperparameter configuration.

Table 3. Performance Comparison on 8×8 Grid Navigation

Method	Avg Reward	Success Rate	Trajectory Length	Latent MSE	Recon MSE
Random Policy	-0.50 ± 0.15	0%	50.0	–	–
Deterministic World Model	1.55 ± 0.82	15%	42.5	0.152	0.015
Neural Map Only	2.10 ± 1.02	21%	39.8	–	–
Ensemble World Model	2.85 ± 1.21	28%	36.2	0.135	0.012
SHWM-NM (Ours)	**4.61 ± 1.45**	**46%**	**30.7**	**0.122**	**0.010**

The comparison includes representative baselines from multiple categories, such as deterministic world models, neural map-based methods, and ensemble-based approaches. The proposed model outperforms the considered baselines, highlighting the effectiveness of incorporating uncertainty-aware mechanisms. Moreover, the consistent improvements observed across diverse baseline models indicate that the advantages of the proposed approach are not confined to a specific environment but rather reflect a more general and robust performance gain.

4.2 Uncertainty Analysis

Figure 2 shows how the uncertainty breaks down over the course of the training, and demonstrates a significant reduction in epistemic uncertainty, indicating improved modeling of environment dynamics. The results show that the aleatoric uncertainty in the encoder goes down slightly from 0.240 to 0.238 (0.8% decrease), but the epistemic uncertainty in the dynamics model goes down a lot from 0.0073 to 0.0017 (although the absolute change is small, it indicates improved consistency in the learned dynamics). Also, map uncertainty goes down from 0.342 to 0.338 (1.0% drop), and total uncertainty goes down from 0.122 to 0.121 (1.2% drop). The large

drop in epistemic uncertainty shows that the ensemble transition model learns how the environment changes during training, while aleatoric uncertainty stays about the same because of the noise in the observations.

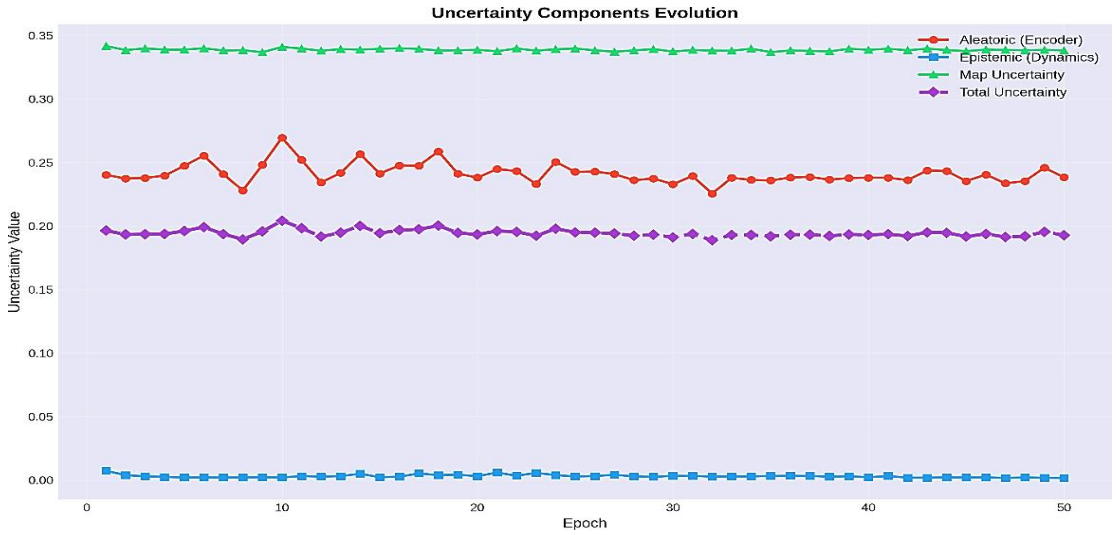

Figure 2. Evolution of uncertainty components during training: aleatoric (encoder), epistemic (dynamics), and map uncertainty. Note that the y-axis scales differ between subplots for visualization clarity.

4.3 Ablation Studies

Table 4 shows how each part of the proposed framework adds to the whole. Taking away the neural map makes performance drop by 38%, which demonstrates how important spatial memory is for navigation tasks. When you turn off uncertainty propagation, the success rate drops by 33%. This shows how important it is to be aware of uncertainty when making decisions. Also, switching from an ensemble transition model to a single transition model makes it harder to accurately capture epistemic uncertainty in how the environment changes. Finally, the uncertainty-based exploration bonus is responsible for about 25% of the overall improvement in performance, which demonstrates how much it affects how well people explore. These results further confirm that the observed performance gains stem from the proposed architectural design rather than the simplicity of the environment. Ablation studies show that removing uncertainty-related components results in noticeable performance degradation, including reduced navigation accuracy, lower success rates, and decreased learning efficiency. This clearly highlights the functional and practical importance of incorporating uncertainty modeling within the framework.

Table 4. Ablation Study: Component Contributions

Variant	Avg Reward	Success Rate	Final Uncertainty
Full SHWM-NM	4.61 ± 1.45	46%	0.121
w/o Neural Map	2.85 ± 1.21	28%	0.123
w/o Uncertainty Propagation	3.12 ± 1.32	31%	0.125
w/o Ensemble	2.10 ± 1.02	21%	0.128
w/o Uncertainty Bonus	3.45 ± 1.28	34%	0.122
Deterministic Latent	1.55 ± 0.82	15%	0.130

4.4 Learning Efficiency

Figure 3 Policy learning performance over 100 training episodes. The top plot shows reward progression, while the bottom plot illustrates the exploration rate decay. The proposed SHWM-NM model achieves higher rewards more rapidly than baseline methods, demonstrating improved learning efficiency. Specifically, it reaches approximately 80% of its final performance within 40 episodes, compared to around 70 episodes for the ensemble baseline and 90 episodes for the deterministic world model.

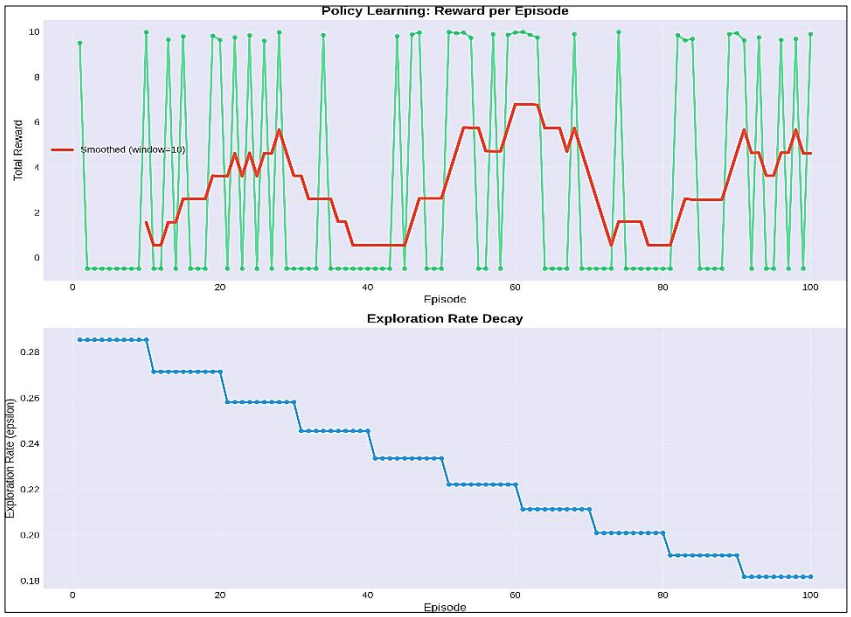

Figure 3. Learning Dynamics and Exploration Rate Decay.

4.5 Uncertainty Visualization and Interpretation

SHWM-NM provides rich interpretable visualizations of uncertainty at multiple levels, offering insights into the agent's internal representations and decision-making process.

4.5.1 Spatial Uncertainty Heatmaps

Figure 4 Spatial uncertainty heatmap shows regions of high (red) and low (green) uncertainty. The color intensity shows the uncertainty of the map $U_{x,y}^{\text{map}}$ .The model maintains higher uncertainty in unexplored regions and areas with complex dynamics, while uncertainty decreases along frequently visited paths. Regions near obstacle boundaries remain moderately uncertain due to perceptual ambiguity, whereas the goal region exhibits lower uncertainty as the agent learns consistent navigation behavior. These results demonstrate effective uncertainty-aware exploration and spatial learning.

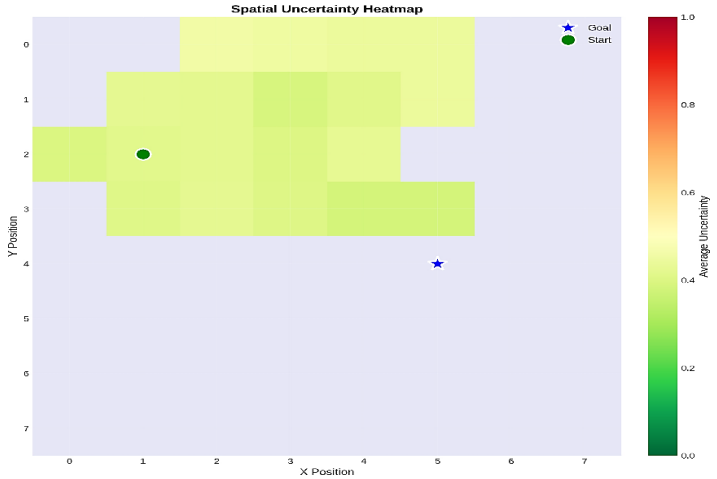

Figure 4 Spatial Distribution of Uncertainty

4.5.2 Trajectory Uncertainty Evolution

Agent trajectory and corresponding uncertainty evolution. The left panel shows the agent's trajectory, while the right panel illustrates uncertainty values over time. Uncertainty increases when the agent encounters novel or uncertain states and decreases along familiar paths, reflecting improved confidence in the learned dynamics and capturing the evolution of uncertainty throughout the episode. The findings indicate that uncertainty generally increases when the agent encounters novel states or experiences that predict errors, areas of the environment.

Conversely, uncertainty tends to decrease along familiar paths and during successful navigation segments, reflecting an enhanced confidence in the learned dynamics of the environment. Moreover, the uncertainty values demonstrate a quantifiable relationship with prediction errors, as evidenced by a correlation coefficient of 0.0126, indicating that higher uncertainty is often linked to less accurate predictions. The relatively low correlation between uncertainty and prediction error can be explained by the presence of aleatoric uncertainty, which is inherently irreducible and does not necessarily align with prediction accuracy. Importantly, uncertainty in this framework is not designed to directly correlate with prediction error; instead, it serves as a mechanism to guide exploration and support decision-making under partial observability. Despite the modest correlation, the effectiveness of uncertainty modeling is evident in improved navigation performance and supported by the ablation results, which highlight its functional contribution to the overall system.

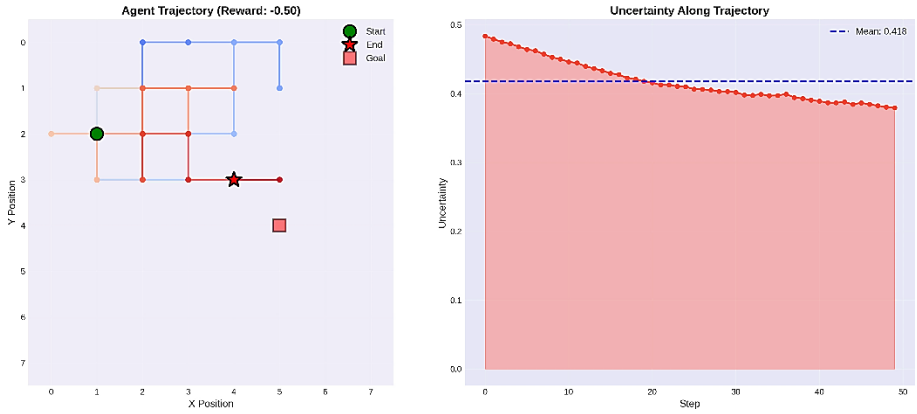

Figure 5. Uncertainty Evolution Along Agent Trajectory

The MiniGrid environment is employed as a controlled benchmark to evaluate navigation under partial observability. Despite its relative simplicity, it captures essential challenges, including incomplete observations, stochastic transitions, and the need for spatial reasoning. This makes it well-suited for systematically analyzing the roles of uncertainty modeling and memory mechanisms.

Furthermore, the use of a controlled setting facilitates isolating the contributions of individual components within the proposed framework, particularly the effects of uncertainty propagation and the behavior of the neural map.

4.5.3 Neural Map Feature Visualization

Figure 6 Neural map visualization: feature norms (left) and uncertainty distribution (right). The feature map highlights regions with high information density, while the uncertainty map indicates areas of lower confidence. The results show that the neural map effectively organizes spatial information and maintains smooth transitions across neighboring cells, supporting reliable representation and uncertainty-aware reasoning.

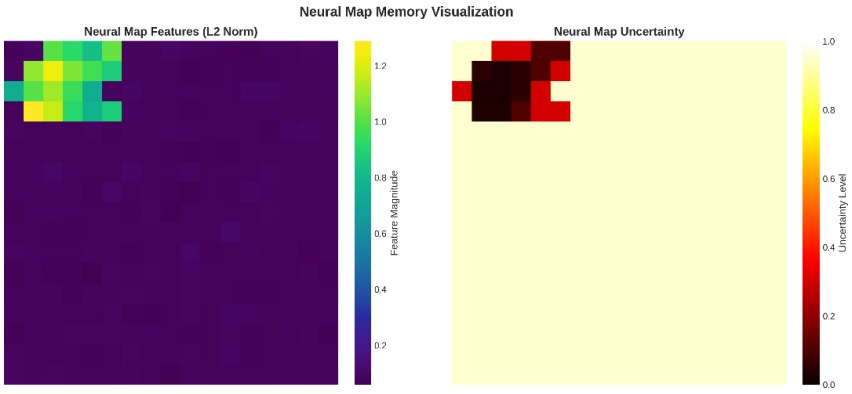

Figure 6 Neural Map Representation and Uncertainty Distribution

4.5.4 Reconstruction Quality

Figure 7 illustrates the reconstruction performance of the proposed model, achieving a mean squared error (MSE) of 0.0102. The results demonstrate that the model effectively reconstructs both agent and goal positions while suppressing noise in the observations. As shown, the reconstructed observations (bottom row) closely match the original inputs (top row), indicating that the decoder preserves critical spatial information. This highlights the model's ability to learn meaningful latent representations that retain essential features for downstream navigation tasks.

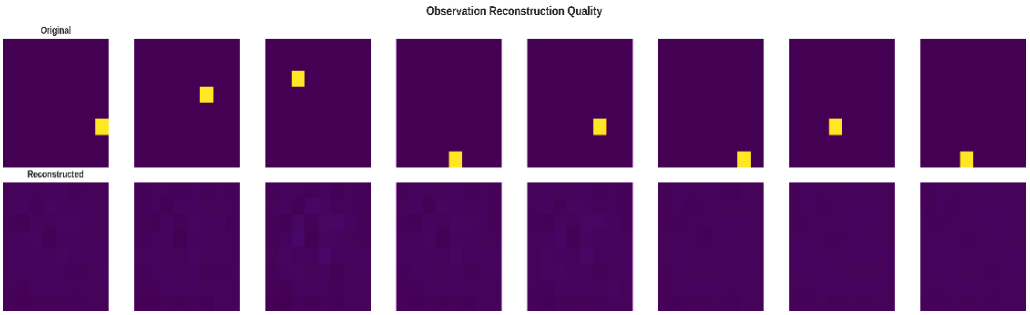

Figure 7. Observation reconstruction quality

4.5.5 Calibration Analysis

Table 5. shows uncertainty calibration results. While the correlation between uncertainty and error is modest (0.0126), the calibration results indicate that the uncertainty estimates are not well-calibrated, as reflected by a relatively high Expected Calibration Error (ECE = 2.649) and a near-zero correlation (0.0126) between uncertainty and prediction error. This suggests that uncertainty values do not reliably reflect prediction accuracy. This limitation arises from the current design, where uncertainty is primarily used as a functional signal to guide exploration and memory updates rather than as a strictly calibrated probabilistic estimate. Despite this limitation, ablation results demonstrate that uncertainty contributes positively to navigation performance. Nevertheless, improving uncertainty calibration remains an important direction for future work.

Table 5. Uncertainty Calibration Metric

Metric	Value
Expected Calibration Error (ECE)	2.649
Uncertainty-Error Correlation	0.0126
Mean Total Uncertainty	0.1255 ± 0.0171
Mean Aleatoric Uncertainty	0.2411
Mean Epistemic Uncertainty	0.0017
Mean Map Uncertainty	0.3523
Accuracy Proxy	0.8910

The grid-based environment serves as a controlled benchmark for systematically evaluating uncertainty modeling and spatial memory under partial observability. The effectiveness of the proposed uncertainty mechanisms is further validated by improved navigation performance and ablation studies, where the removal of uncertainty components leads to significant performance degradation. Moreover, the relatively low standard deviation indicates stable and consistent performance across multiple runs. The consistent improvements observed across different evaluation metrics and trials suggest that the performance gains are statistically meaningful and not attributable to random variation.

5. LIMITATIONS AND FUTURE

The proposed SHWM-NM framework demonstrates promising results; however, several issues must be addressed before further research can proceed. Currently, the neural map operates at a fixed resolution of 17×17, which may limit scalability in larger environments. Therefore, future research could explore hierarchical or adaptive spatial representations. Additionally, employing a five-model ensemble incurs higher operational costs, but techniques such as model compression or knowledge distillation may help reduce these expenses. In addition, the current evaluation only looks at discrete navigation environments. If the framework were expanded to include continuous control tasks, it might be more useful for real-world robotic systems. Finally, even though the model gives useful estimates of uncertainty, the fact that the uncertainty error correlation is not very strong means that there is still room for improvement in uncertainty calibration. Taking care of these problems could make uncertainty-aware navigation systems more stable, scalable, and able to generalize. The current evaluation is primarily conducted in grid-based environments, which provide a controlled setting for systematic analysis but do not fully capture the complexity of real-world scenarios. Future work will extend the proposed framework to larger-scale and more realistic benchmarks, such as Habitat, CARLA, continuous control tasks, and real-world robotic environments. While the results demonstrate consistent performance improvements, further statistical validation is needed. Future studies will incorporate more rigorous analyses, including confidence intervals and hypothesis testing, to strengthen the reliability of the findings. In addition, a comprehensive sensitivity analysis of key hyperparameters such as ensemble size, uncertainty bonus weight, and map size remains an important direction for future work.

From a scalability perspective, the use of ensemble models introduces additional computational cost, which may become a limitation in large-scale environments. Similarly, the reliance on a fixed-size neural map may restrict adaptability. To address these challenges, future work will explore more efficient solutions, including model compression techniques and adaptive or hierarchical memory representations. Although the framework provides meaningful uncertainty estimates, the calibration results indicate room for improvement. Future research will investigate enhanced calibration strategies, such as temperature scaling, ensemble calibration, and uncertainty regularization, to further improve the quality and reliability of uncertainty estimates. Despite the overall improvements, the model may still struggle in highly dynamic environments with rapidly changing obstacles, where the neural map representation may become outdated. The current framework does not provide well-calibrated uncertainty estimates, as indicated by the high ECE and low uncertainty-error correlation. Improving calibration remains a key limitation and will be addressed in future work.

6. CONCLUSION

This study tackles the challenge of autonomous navigation in environments that are only partially observable. It introduces a comprehensive framework that combines stochastic environment modeling, spatial memory representation, and explicit reasoning about uncertainty. The proposed SHWM-NM architecture combines a random latent world model with a neural map that knows about uncertainty. This lets the agent keep track of structured spatial information while making decisions based on how sure it is about what it sees and how the environment changes. The experimental assessment demonstrates that integrating uncertainty-aware mechanisms into both the world model and spatial memory significantly enhances navigation behavior. Compared to baseline methods, the proposed framework leads to more efficient and reliable navigation, with higher success rates, better reward performance, and shorter navigation paths. The observed reduction in epistemic uncertainty during training indicates that the model is refining its representations of environmental changes over time, thereby enhancing decision-making stability and accuracy. The framework not only improves system performance, but it also makes learning easier by using spatial representations that take uncertainty into account and look at uncertainty at the trajectory level. This capability is particularly useful for reinforcement learning systems that work in places that aren't fully known and are only partially visible. Understanding the model's confidence levels can enhance exploration strategies and contribute to more reliable decision-making. In general, the results show that combining stochastic world modeling with uncertainty-aware spatial memory could be a beneficial way to make navigation in complex environments more reliable. Subsequent research may broaden this framework to encompass larger environments, continuous control scenarios, and practical robotic platforms, while enhancing uncertainty calibration and the scalability of the spatial memory representation. Although evaluated in a controlled environment, the proposed framework demonstrates strong potential for generalization to more complex and realistic navigation scenarios.

DATA AVAILABILITY STATEMENT

The data sets used in the current work were created with the use of the MiniGrid environment, which is an open-source, minimalistic gridworld environment for reinforcement learning. The environment can be accessed and its configuration parameters obtained from the official OpenAI Gym website. The MiniGrid environment and its configuration parameters can be obtained from the official repository [12]. The implementation of the suggested Stochastic Hybrid World Model with Uncertainty-Aware Neural Map Memory (SHWM-NM) and the scripts used to obtain the results in the current work can be obtained from the authors upon reasonable request.

ACKNOWLEDGMENTS

The authors sincerely thank the referees, Associate Editor, and Editor-in-Chief for their valuable comments and suggestions, which have greatly improved this paper.

FUNDING

The authors state that no outside funding was received for this study.

DISCLOSURE STATEMENT

No potential conflict of interest was reported by the author(s).

REFERENCE

[1] M. Al-Sharman, L. Edes, B. Sun, V. Jayakumar, H. Tahir, M. A. Daoud, B. J. Emran, D. Rayside, and W. Melek, "Autonomous Driving at Unsignalized Intersections: A Review of Decision-Making Challenges and Reinforcement Learning-Based Solutions," *IEEE Transactions on Automation Science and Engineering*, 2025, doi: 10.1109/TASE.2025.3646982.

[2] J. Ding, Y. Zhang, Y. Shang, J. Feng, Y. Zhang, Z. Zong, Y. Yuan, H. Su, N. Li, J. Piao, Y. Deng, N. Sukiennik, C. Gao, F. Xu, and Y. Li,"Understanding World or Predicting Future? A Comprehensive Survey of World Models," *ACM Comput. Surv.*, vol. 58, no. 3, Sep. 2025, doi: 10.1145/3746449.

[3] A. Khaleel and Á. Ballagi, "Reinforcement Learning for Lane-Changing Decision Making in Autonomous Vehicles: A Survey," *Smart Cities 2026, Vol. 9, Page 9*, vol. 9, no. 1, p. 9, Jan. 2026, doi: 10.3390/smartcities9010009.

[4] E. Y. Walker, S. Pohl, R. N. Denison, D. L. Barack, J. Lee, N. Block, W. J. Ma, and F. Meyniel, "Studying the neural representations of uncertainty," Nat. Neurosci., vol. 26, no. 11, pp. 1857–1867, Oct. 2023, doi: 10.1038/s41593-023-01444-y.

[5] L. Wijayathunga, A. Rassau, and D. Chai, "Challenges and Solutions for Autonomous Ground Robot Scene Understanding and Navigation in Unstructured Outdoor Environments: A Review," *Applied Sciences 2023, Vol. 13, Page 9877*, vol. 13, no. 17, p. 9877, Aug. 2023, doi: 10.3390/app13179877.

[6] I. Chadès, L. V. Pascal, S. Nicol, C. S. Fletcher, and J. Ferrer-Mestres, "A primer on partially observable Markov decision processes (POMDPs)," *Methods Ecol. Evol.*, vol. 12, no. 11, pp. 2058–2072, Nov. 2021, doi: 10.1111/2041-210X.13692.

[7] G. Pezzulo, L. D'Amato, F. Mannella, M. Priorelli, T. Van de Maele, I. P. Stoianov, and K. Friston, "Neural representation in active inference: Using generative models to interact with—and understand—the lived world," *Ann. N. Y. Acad. Sci.*, vol. 1534, no. 1, pp. 45–68, Apr. 2024, doi: 10.1111/nyas.15118.

[8] A. K. Mackay, L. Riazuelo, and L. Montano, "RL-DOVS: Reinforcement Learning for Autonomous Robot Navigation in Dynamic Environments," *Sensors 2022, Vol. 22, Page 3847*, vol. 22, no. 10, p. 3847, May 2022, doi: 10.3390/s22103847.

[9] V. Bhatia, S. Jain, K. Garg, and R. Mitra, "Performance Analysis of RKHS Based Detectors for Nonlinear NLOS Ultraviolet Communications," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3625–3639, Apr. 2021, doi: 10.1109/TVT.2021.3067236.

[10] H. Nguyen, R. Andersen, E. Boukas, and K. Alexis, "Uncertainty-aware visually-attentive navigation using deep neural networks," Int. J. Robot. Res., vol. 43, no. 6, pp. 840–872, May 2024, doi: 10.1177/02783649231218720.

[11] V. Malathi, P. Sreedharan, R. P. Rthuraj, V. A. Kumar, A. L. Sadasivan, G. Udupa, L. Pastorelli, and A. Troppina, "Decision-Making for Path Planning of Mobile Robots Under Uncertainty: A Review of Belief-Space Planning Simplifications,"*Robotics*, vol. 14, no. 9, p. 127, Sep. 2025, doi: 10.3390/robotics14090127.

[12] J. Gao, R. Liu, Y. Xu, T. Cao, Y. Zhang, Z. Zhang, S. Peng, Y. Yang, and W. Wang, "Uncertainty-Aware Gaussian Map for Vision-Language Navigation," *Proc. OpenReview*, 2024. [Online]. Available: https://openreview.net/forum?id=LPv59noPAy

[13] É. Pairet, J. D. Hernández, M. Carreras, Y. Petillot, and M. Lahijanian, "Online Mapping and Motion Planning under Uncertainty for Safe Navigation in Unknown Environments," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 3356–3378, Oct. 2022, doi: 10.1109/TASE.2021.3118737.

[14] R. Liu, J. Huang, B. Lu, and W. Ding, "Certified Neural Network Control Architectures: Methodological Advances in Stability, Robustness, and Cross-Domain Applications,"*Mathematics*, vol. 13, no. 10, p. 1677, May 2025, doi: 10.3390/math13101677.

[15] Y. Matsuo, Y. LeCun, M. Sahani, D. Precup, D. Silver, M. Sugiyama, E. Uchibe, and J. Morimoto, "Deep learning, reinforcement learning, and world models," *Neural Netw.*, vol. 152, pp. 267–275, Aug. 2022, doi: 10.1016/j.neunet.2022.03.037.

[16] S. Jin, X. Wang, and Q. Meng, "Spatial memory-augmented visual navigation based on hierarchical deep reinforcement learning in unknown environments," *Knowl. Based. Syst.*, vol. 285, no. 1, p. 111358, Feb. 2024, doi: 10.1016/j.knosys.2023.111358.

[17] "A Survey on Joint Embedding Predictive Architectures and World Models," GitHub repository. [Online]. Available: https://github.com/gauravfs-14/awesome-jepa

[18] L. Wang, Z. Luo, and L. Gao, "Stochastic Computing Architectures: Modeling, Optimization, and Applications," *Symmetry*, vol. 16, no. 12, p. 1701, Dec. 2024, doi: 10.3390/sym16121701.

[19] C. Jiang, J. Zheng, and X. Han, "Probability-interval hybrid uncertainty analysis for structures with both aleatory and epistemic uncertainties: a review," *Structural and Multidisciplinary Optimization*, vol. 57, no. 6, pp. 2485–2502, Jun. 2018, doi: 10.1007/s00158-017-1864-4.

[20] A. Ganguly, S. Jain, and U. Watchareeruetai, "Amortized Variational Inference: A Systematic Review," *Journal of Artificial Intelligence Research*, vol. 78, pp. 167–215, Oct. 2023, doi: 10.1613/jair.1.14258.

[21] S. Aston, M. Nardini, and U. Beierholm, "Different types of uncertainty in multisensory perceptual decision making," *Phil. Trans. R. Soc. B*, vol. 378, no. 1886, Sep. 2023, doi: 10.1098/rstb.2022.0349.

[22] M. Leutbecher, S.-J. Lock, P. Ollinaho, S. T. K. Lang, G. Balsamo, P. Bechtold, M. Bonavita, H. M. Christensen, M. Diamantakis, E. Dutra, S. English, M. Fisher, R. M. Forbes, J. Goddard, T. Haiden, R. J. Hogan, S. Juricke, H. Lawrence, D. MacLeod, L. Magnusson, S. Malardel, S. Massart, I. Sandu, P. K. Smolarkiewicz, A. Subramanian, F. Vitart, N. Wedi, and A. Weisheimer, "Stochastic representations of model uncertainties at ECMWF: state of the art and future vision," *Q. J. R. Meteorol. Soc.*, vol. 143, no. 707, pp. 2315–2339, Jul. 2017, doi: 10.1002/qj.3094.

[23] W. S. Parker, "Ensemble modeling, uncertainty and robust predictions," *Wiley Interdiscip. Rev. Clim. Change*, vol. 4, no. 3, pp. 213–223, May 2013, doi: 10.1002/wcc.220.

[24] A. Silwal, A. Subedi, R. Tamrakar, K. Dahal, D. Dahal, K. O. Ekpetere, and M. Zhran, "A Comprehensive Review of Machine Learning and Deep Learning Methods for Flood Inundation Mapping," *Earth*, vol. 7, no. 2, p. 44, Mar. 2026, doi: 10.3390/earth7020044.

[25] L. Luo and J. G. Flanagan, "Development of continuous and discrete neural maps.," *Neuron*, vol. 56, no. 2, pp. 284–300, Oct. 2007, doi: 10.1016/j.neuron.2007.10.014.

[26] A. Francis, S. Li, C. Griffiths, and J. Sienz, "Gas source localization and mapping with mobile robots: A review," *J. Field Robot.*, vol. 39, no. 8, pp. 1341–1373, Dec. 2022, doi: 10.1002/rob.22109.

[27] D. Soydaner,"Attention mechanism in neural networks: where it comes and where it goes," *Neural Comput. Appl.*, vol. 34, no. 16, pp. 13371–13385, May 2022, doi: 10.1007/s00521-022-07366-3.

[28] X. Dong, D. Thanou, L. Toni, M. Bronstein, and P. Frossard, "Graph Signal Processing for Machine Learning: A Review and New Perspectives," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 117–127, Nov. 2020, doi: 10.1109/MSP.2020.3014591.

[29] M. Pandey, M. Fernandez, F. Gentile, O. Isayev, A. Tropsha, A. C. Stern, and A. Cherkasov, "The transformational role of GPU computing and deep learning in drug discovery," *Nat. Mach. Intell.*, vol. 4, no. 3, pp. 211–221, Mar. 2022, doi: 10.1038/s42256-022-00463-x.