Value Iteration with Value of Information Networks

Samantha N. Johnson
University of Chicago, Chicago, IL 60637
snjohnso@uchicago.edu

Michael A. Buice
Allen Institute, Seattle, WA 98109
University of Washington, Seattle, WA 98195
michaelbu@alleninstitute.org

Koosha Khalvati
Allen Institute, Seattle, WA 98109
koosha.khalvati@alleninstitute.org

Abstract

Despite great success in recent years, deep reinforcement learning architectures still face a tremendous challenge in dealing with uncertainty and perceptual ambiguity. Similarly, networks that learn to build the world model from the input and perform model-based decision making in novel environments (e.g., value iteration networks) are mostly limited to fully observable tasks. In this paper, we propose a new planning module architecture, the VI^{2}N (Value Iteration with Value of Information Network), that learns to act in novel environments with a high amount of perceptual ambiguity. This architecture over-emphasizes reducing the uncertainty before exploiting the reward. Our network outperforms other deep architecture in challenging partially observable environments. Moreover, it generates interpretable cognitive maps highlighting both rewarding and informative locations. The similarity of principles and computations of our network with observed cognitive processes and neural activity in the Hippocampus draw a strong connection between VI^{2}N and principles of computations in the biological networks.

1 Introduction

Deep neural networks have provided powerful end-to-end solutions to Reinforcement Learning (RL) problems that map perception to action [7]. One can approach this end-to-end learning in a classic supervised fashion especially when provided an expert policy to imitate. However, several studies have shown that incorporating cognitive/classic RL mechanisms such as simulation of future events and experience replay improve the learning process significantly [29, 14]. For example, Value Iteration Networks (VINs) incorporate long-term planning (the simulation of future events) by implementing the value iteration algorithm (i.e. a sequence of Bellman updates) via convolutional layers [3, 29, 21, 31, 12]. Trained either by reward or through imitation of an expert's actions, VINs can learn to navigate in fully observable novel environments significantly better than fully connected and untied convolutional networks [29]. Furthermore, its generated model of the environment correctly identifies the rewarding areas (e.g., the goal state).

While VINs and deep reinforcement learning architectures in general have been very successful in many applications, they face a tremendous challenge in many real-world scenarios due to perceptual ambiguity. Perceptual ambiguity, often called *partial observability*, introduces uncertainty about the current state of the environment. This uncertainty must be accounted for to make decisions that produce high rewards. In other words, the agent must form a probability distribution, or "belief", over its current state and choose its action based on this belief. Even simple networks with a probabilistic belief representation outperform networks with more sophisticated encoding and RL modules that perform well in challenging fully-observable environments [20, 15]. However, the main challenge in

NeurIPS 2023 Workshop on Generalization in Planning (GenPlan 2023)

uncertain environments is the action selection based on the current belief, not the belief representation itself. Therefore, more advanced policy/planning modules are required to perform well in more complex uncertain environments.

Formally, defined within the Partially Observable Markov Decision Process (POMDP) framework [30], optimal decision making under uncertainty is not achievable in polynomial time [28]. Additionally, powerful sub-optimal approximations involve sampling and tree search techniques with no differentiable implementation, hindering our ability to use them in neural network implementations. Because of this limitation, the current state-of-the-art value iteration network for decision making under partial observability is founded upon a very simple POMDP-solver, QMDP, which assumes that all uncertainty disappears after the first step [14]. While this allows for a differential heuristic, this assumption causes the solver to fail in highly uncertain environments. This paper proposes a new network architecture, the VI^{2}N (Value Iteration with Value of Information Network), that can learn to plan in unseen environments with high uncertainty. VI^{2}N is based on the *Pairwise Heuristic* [16], which calculates the solution of sub-problems where states are considered pairwise. Since the Pairwise Heuristic can be calculated by the Bellman equation, it can be implemented with a neural network similar to the VIN. We demonstrate the power of our approach by testing it on navigation problems in the presence of uncertainty in different environments. VI^{2}N outperforms other networks, especially in challenging environments with high ambiguity.

The computational principle, or Marr's algorithmic level of computation from the cognitive neuroscience perspective, of the Pairwise Heuristic plays a more important role in our work. The Pairwise Heuristic emphasizes information gathering and resolving uncertainty before maximizing the expected reward [16], hence *value of information* in the name of our architecture. While information gathering is a fundamental part of decision making under uncertainty, it is not always necessary for obtaining the optimal solution. In other words, the uncertainty about the hidden state of the environment does not have to be fully resolved to reach optimality. Nonetheless, resolving uncertainty beyond what is necessary may still provide better solutions. Notably, such over-emphasis on information gain has been extensively observed in humans and other animals [24]. Specifically, in many cognitive tasks such as visual search, subjects gather different pieces of evidence, even those irrelevant to the outcome of their choice [11]. Previous life experience, the complexity of learning the exact relevant pieces of information to the final outcome, and flexibility in changing environments are among the causes of such biases [6, 11, 24].

The incorporation of cognitive processes such as simulation of future events and information gathering make VI^{2}N and other value iteration networks a useful tool to study natural intelligence. Specifically, in our case, VI^{2}N represents informative areas in addition to rewarding ones. Besides interpretability, such representation resembles the activity of hippocampal neurons, such as border cells, during decision making of biological agents [27]. The emergence of these representations in our principled model-based network provides a theoretical/computational explanation for computing units of natural intelligence.

2 Background

Markov Decision Process (MDP): Sequential decision making is usually expressed as a Markov Decision Process (MDP) [30]. Formally, an MDP is (S, A, T, R, γ) where S is the set of states of the environment, A is the set of all available actions to the agent, the transition function $T : |S| \times |A| \times |S| \rightarrow [0, 1]$ defines $T(s, a, s') = P(s'|s, a)$, the probability of ending up in state s' by performing action a in state s, $R : |S| \times |A| \rightarrow \mathbb{R}$ is a bounded function determining the reward gained in state s, shown as $R(s)$, and $\gamma \in (0, 1]$ is the discount factor for the reward [30].

Starting from an initial state, s_0, the goal of the agent is to come up with a recipe for action selection, called a policy π, that maximizes the total discounted reward. Since the system is Markovian, the policy can be expressed as a mapping from states to actions, i.e. $\pi : |S| \times |A| \rightarrow [0, 1]$. The optimal policy π^* is $\pi^* = \arg\max_\pi \sum_{t=0}^{H} \gamma^t \mathbb{E}[R(s_t)|\pi, s_0]$ where the horizon H defines the length of this sequence. In deep reinforcement learning, this optimal policy/mapping is learned with a network with the state S (or a representation of it, $\phi(s)$) as the input and the action as the output of the network [7].

Value Iteration Network (VIN): Algorithms for finding the optimal policy of an MDP are generally divided into two categories: "model-free" and "model-based". Model-based approaches use the structure of the environment, i.e., transition and reward function, to determine the optimal policy. In

contrast, model-free approaches try to learn the optimal policy directly from the accumulated obtained reward. Consequently, model-based approaches adapt faster upon changes in the environment as they only need to update their model, i.e., T and/or R. The value iteration algorithm is a model-based approach where the optimal value of each state, which is the expected gained reward in the future given the optimal policy, is computed through a series of Bellman updates [3]:

$$V_t(s) = \max_a \left[R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V_{t-1}(s') \right] \quad (t \leq H). \tag{1}$$

When the transition function is spatially invariant, a neural network can learn T and R by implementing the Bellman equation with convolutional layers [29]). More specifically, given the map of the environment and the current state of the agent (e.g., its position on the map) as the inputs and an expert's action or reward as the output, VIN learns convolutional kernels of f_R and f_P representing reward and transition functions. Such a network with the integration of value iteration as an explicit planning module, generally known as a Value Iteration Network (VINs), significantly outperforms networks with similar computational power (e.g., layers) in learning to plan in unseen environments [29]. Originally built for simple lattice worlds with spatially invariant transition functions, value iteration networks have been significantly improved in terms of applicability to domains with more complex structures over the past years [21, 31, 12]. All of these improvements, however, are still mainly limited to fully observable environments.

Partially Observable Markov Decision Process (POMDP): Existence of uncertainty in the real world, especially in the form of perceptual ambiguity, has made MDPs impractical in many situations [30]. Similarly, state-of-the-art networks reaching extraordinary performance in very complicated yet fully observable tasks often fail to handle seemingly small amounts of ambiguity in the environment [20]. Partially Observable MDPs (POMDPs) represent the closest approach to MDPs that deals with the uncertainty by adding an observation set and observation function to its framework. Formally, a POMDP is a tuple $(S, A, Z, T, O, R, \gamma)$ where S, A, T, R, and γ are defined very similar to their definition in MDP. Z is the set of observations and $O : |S| \times |Z| \rightarrow [0, 1]$ is the observation function determining probability of observation z in state s, i.e. $O(s, z) = P(z|s)$. In a POMDP, the agent is not fully aware of its current state. Therefore, it has to maintain a probability distribution over states, often called its *belief* $b(s)$. Starting from a prior probability distribution over states of the environment, called the initial belief (b_0), the goal is to maximize the expected discounted reward [23]. For a POMDP, the optimal decision policy π^* can be expressed as a mapping from belief states (probability distributions over states) to distribution of actions that maximizes the total expected reward [28], i.e. $\pi^* = \arg\max_\pi \sum_{t=0}^{H} \gamma^t E[R(s_t, a_t, z_{t+1})|b_t, \pi]$.

The uncertainty about the state makes the agent navigate in the belief state space instead of the state space. At time step t, the belief state b_t is updated based on the previous belief state b_{t-1} after action a_{t-1} and observation z_t as follows: $b_t(s) \propto P(z_t|s, a_{t-1}) \sum_{s' \in S} P(s|s', a_{t-1}) b_{t-1}(s')$.

Partial observability also makes the problem of finding the optimal policy exponentially more complex than the MDP. While the optimal policy of an MDP can be found in polynomial time, finding the optimal policy of a POMDP is NP-hard [30]. As a result, the optimal policy can only be approximated by methods such as heuristics, sampling, and search trees [23, 16].

Deep networks for solving POMDPs: Since POMDPs have an additional observation function compared to MDPs, deep architectures for decision making under partial observability represent observation function as a convolutional kernel f_Z in addition to transition function kernel f_P and reward kernel f_R. More specifically, given the map of the environment and an observation instead of the current state, e.g. in the form of a small 3×3 window showing the surroundings of the agent in the map, as inputs and an expert's action or reward as the output, a value iteration network for POMDPs learns f_R, f_Z, and f_P to be able to act in novel environments.

Similar to classic POMDP solvers, a POMDP solver network consists of two modules of belief update and policy (action selection), forming a recurrent architecture together. The belief update can be easily implemented in a network, as exemplified in the QMDP-Net architecture [14]. However, designing a powerful policy module is very challenging due to the differentiability requirement. As a result, current networks for solving POMDPs have a very simple policy module, e.g., a model-free RL module [20] and QMDP [14].

Artificial Networks as a tool to study the brain: Since the success of artificial networks in achieving close-to-human-level performance in many tasks, such as object recognition, many researchers have used them as a tool to study natural intelligence [22]. In the field of sequential decision-making and reinforcement learning, this research is mainly limited to vanilla Recurrent Neural Networks (RNNs) or model-free reinforcement learning [4]. For example, it has been shown that training a recurrent network for navigation tasks results in a grid-like representation of the environment, similar to the "grid cells" in the hippocampus [2]. Another example is the success of recurrent networks in solving simultaneous mapping and localization and the emergence of "head-direction" selective cells [13]. Several works have studied the behavior of deep model-free RL, such as DQN, in experimental setups and animals' natural environments to study the brain (e.g., [1, 5, 26]). To the best of our knowledge, no work is using model-based deep networks in uncertain environments to study natural intelligence. Notably, a recent study has shown that the emergence of cell-like representations in unconstrained artificial networks is usually an artifact of posthoc implementation choices, highlighting the need for networks that are designed based on computational principles in studying the biological brain [25].

3 Model

Our main goal is to provide a better policy module for the value iteration networks in partially observable environments. Our architecture is founded upon a "Pairwise Heuristic". Originating from Bayesian active learning in which the heuristic is used to find the correct hypothesis with a set of noisy tests [9], the Pairwise Heuristic has also been used in robot localization [17] and a general-purpose POMDP-solver when the environment model is fully known [16]. Here we present the POMDP-solver version, slightly modified for our framework.

3.1 The Pairwise Heuristic for solving POMDPs

The main idea of the pairwise heuristic is to use solutions of the smallest sub-problems that still consider the uncertainty about the true hypothesis/state, which would be pairs (sets of 2) of hypotheses/states [9]. In a POMDP, this would be the set of $n(n-1)/2$ optimal policies in each of which the belief is .5 for two states. The expected total reward of each of these policies is the *value of the pair*, shown by $V(s, s')$, for $s, s' \in S$. Calculating pairwise optimal policies is still computationally very expensive. Therefore, the pairwise heuristic for POMDPs applies an additional heuristic to calculate $V(s, s')$ [16]. For each pair, it tries to resolve the uncertainty first and then exploits the reward. As mentioned before, resolving uncertainty is not always necessary to gain the optimal reward. However, it produces a "good enough" solution.

Given the observation function, the uncertainty is already resolved for some pairs of states. To be more precise, it is highly unlikely to have a notable probability/belief for states with different observations. These pairs are "distinguishable". For other pairs of states, i.e. indistinguishable ones, the Pairwise Heuristic resolves the uncertainty by going to distinguishable pairs. Two states are distinguishable if there is a high probability that different observations are recorded in the two states. Formally, s and s' are distinguishable if and only if:

$$\sum_{o} \sum_{s',s'} p(o|s)(1 - p(o|s')) + p(o|s')(1 - p(o|s)) \geq 2\lambda \tag{2}$$

λ is a constant that is specified by a domain expert. If there is no noise in observations, this value is 1. Otherwise, this threshold is set to a value close to but less than 1.

The pairwise value ($V(s, s')$) of distinguishable pairs is simply the average of the value function of each of the states in the underlying MDP model of the environment (assuming full observability in the environment), i.e., $.5(V(s) + V(s'))$. To find the value function of the indistinguishable pairs, we use a value iteration algorithm in an MDP where the states are pairs of states of our original problem. The transition function of this MDP is determined by the joint transition probability distribution of the original environment:

$$T((s, s'), a, (s'', s''')) = p((s'', s''')|(s, s'), a) = p(s''|s, a)p(s'''|s', a) \tag{3}$$

The reward of each pair is simply the average reward of the two states in the original problem:

$$R(s, s') = 0.5(R(s) + R(s')) \tag{4}$$

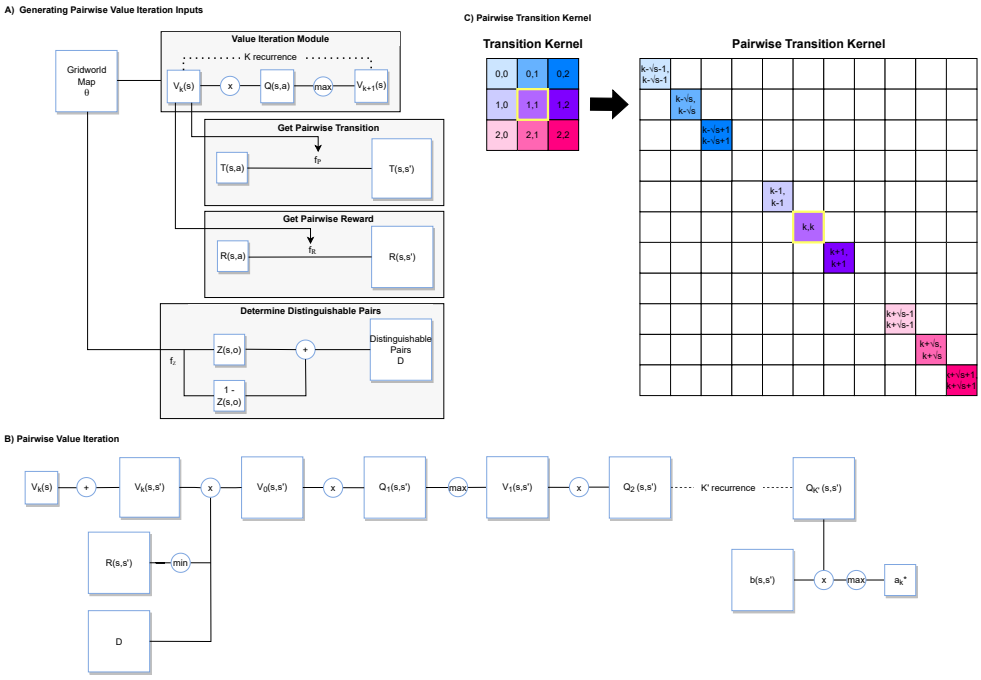

Figure 1: VI^{2}N Overview. a) Required preprocessing to prepare for Pairwise Value Iteration b) Pairwise Value Iteration, which uses outputs from part (a) to determine the selected action. c) Example conversion of one channel of the transition kernel f_P to the corresponding channel of pairwise transition kernel. Example assumes a 4x4 environment of 16 states, where S = number of states, kernel size = $2(\sqrt{s}+1)+1$, and k = kernel size$//2 = \sqrt{s}+1$

Therefore, the Bellman equation of our pairwise value iteration algorithm is as follows:

$$V_k(s,s') = max_a[R(s,s') + \gamma \sum_{s'',s'''} T((s,s'),a,(s'',s'''))V_{k-1}(s'',s''')] \tag{5}$$

Initial pairwise values, i.e., $V_0(s,s')$, in the above equation is $.5(V(s)+V(s'))$ for distinguished pairs and the minimum possible reward for indistinguished ones.

To select an action, the Pairwise Heuristic POMDP-solver maximizes the expected value of pairs using the joint belief state, i.e., $b(s,s') = b(s)b(s')$:

$$a_k^* = \arg\max_a \sum_{(s,s')} b(s,s')Q((s,s'),a) \tag{6}$$

If the probabilities of all states, except the most likely one, become negligible, the selected action would be the optimal action of the underlying MDP for that most likely state.

3.2 VI2N Architecture

All of the pairwise heuristic POMDP solver processes have a straightforward differentiable implementation. The central part of this solver is the pairwise value iteration (Eq. 5), which uses the pairwise transition (Eq. 3), and the pairwise reward functions (Eq. 4). Moreover, the initial pairwise values are determined by the value of states ($V(s)$) in the underlying MDP (Eq. 1) and the distinguishability of each pair of states (Eq. 2). The network implementation of these components is demonstrated in figure 3.2, part A.

Starting from the value iteration algorithm implemented by a VI module, the network learns f_P and f_R, determining $T(s,a,s')$ and $R(s)$ of the environment, in addition to the value of single states,

$V(s)$. From this point, the objective becomes converting elements of the environment to a pair-space representation to allow for the VI^2 module implementation. Specifically, we must convert $T(s, a)$ and $R(s)$ into $T((s, s'), a)$ and $R(s, s')$ for all $s, s' \in S \times S$.

We can convert $R(s)$ using an averaging layer across all s. Transition T is used as the kernel in the VI Module (f_P) and must be transformed into a transition kernel for the pairwise state space, which involves increasing the size of the kernel from $(3, 3)$ to $(2(\sqrt{S}+1)+1, 2(\sqrt{S}+1)+1)$ to allow for row and column transitions between pairs (assuming the grid world is a $\sqrt{S} \times \sqrt{S}$ square). This kernel is constructed using the learned transition probabilities from the VI Module and has a number of channels equal to the number of actions available in the environment similar to f_P. All nine values of each channel of f_P would be mapped to the main diagonal of the pairwise transition kernel in the corresponding channel as demonstrated in figure 3.2, part C.

We must also determine which set of pairs (s, s') are distinguishable. We implement this by applying the convolutional kernel for the observation function, f_Z, to all states (the grid world map) to get matrix Z. Then we use the outer product of Z and $1 - Z$ and compare it with a threshold to implement Eq. 2. We express the distinguishability by a binary $|S| \times |S|$ matrix, D.

The pairwise value initialization ($V_0(s, s')$) is done using matrix multiplication of D and $.5(V(s) + V(s'))$ (for distinguished pairs) in addition to multiplication of $(1 - D)$ and $min_{R(S)}$ in the shape of an $|S| \times |S|$ matrix (for indistinguishable pairs). With the pairwise reward and transition function calculated, the pairwise value iteration (Eq. 5) is just another VI module, which we call VI^2 module since it is in the pairwise space. Finally, the action selection (Eq. 6) is done by multiplying the pairwise belief state (outer product of belief by itself) with pairwise Q values and applying the max pooling layer (Figure 3.2, part b).

4 Results

We compared VI^{2}N with the QMDP-Net on several 20×20 binary grid-world navigation environments, each of which with various amounts of perceptual ambiguity controlled by the sparsity/density percentage of obstacles. Since QMDP-Net has been shown to perform significantly better than unconstrained networks [14], and there is no mapping between a classic RL expert and the network in the networks, we did not include them in our analysis.

Our environments were designed in a way to resemble biological and artificial agents' real environments. We kept the observation and action function constant among the environments to be able to have a systematic comparison in terms of uncertainty and complexity of the decision-making. The actions were 'right', 'up', 'down', and 'left', moving the agent one cell in the direction specified by the name and also action 'stay'. Moreover, the agent was able to observe the cell it was on and also the neighboring cells in each of the cardinal directions. Since our focus is on perceptual ambiguity, not handling noise in sensors and actuators, both of our sensors and actuators were noise-free. We also used the same belief update mechanism for both agents to have a fair comparison between the two policy modules, i.e., QMDP-Net and VI^{2}N.

Moreover, we set the number of recurrences in the VI module of QMDP-net equal to the total number of recurrences in VI (40) and VI^2 modules of our network (20), which is 60. Importantly, the transition kernel did not get updated in the pairwise module (VI^2) (we did not pass gradients in this module). Therefore, Q-MDP net did actually have a computational advantage over VI^{2}N in terms of a number of free parameters.

For each type of environment, labels of "correct" policies were generated using two types of expert solvers: the QMDP and Pairwise solvers. This allowed us to explore reinforcement learning, which necessitates the training expert to be the same as the planner embedded in the networks. For the QMDP-net, QMDP was used as the expert solver, and the VI^{2}N, which uses the Pairwise Heuristic planner, uses the Pairwise Heuristic as the expert. Models were trained on each environment type separately, using policies of expert solvers as labels. It is valuable to note that not every expert policy label is a success, as some environments are too difficult for either the QMDP or Pairwise Heuristic to solve. Networks were trained only on successful trials (less than 50 steps needed to reach the goal) to resemble positive reinforcement. The training set contained 20000 to 30000 action labels. Training performance was evaluated through $95\% - 5\%$ train-validation process. The test success rate was then calculated by running the generated model on 1,000 novel environments of the same type, each

of which with 20 different start states for each density. The initial belief state for the tests was always uniform among all possible states (starting from an obstacle or goal was not possible).

We started with a "random" environment. In the random environment, obstacles are randomly placed within the arena at both 5% and 10% density/sparsity levels (Figure 2, top left). With an average of 20 or 40 obstacles in this environment, uncertainty would be resolved in a few steps. As a result, both networks had a very high success rate (table 4, top row).

We increased the ambiguity by adding the constraint of minimal continuity in each axis to the random environment, which produces very few blocks with a side size of 4. This type of environment called "blocks", was generated to model environments where obstacles are randomly placed as independent clusters, such as desert landscapes ((Figure 2, bottom left). With the increase in perceptual ambiguity compared to the "random" environment, both networks' performance (success rate) dropped. However, the drop was lower for VI^{2}N (table 4, the second row from the top).

Our third environment called "walls", contained long walls parallel to the border in an empty arena, resembling long hallways for robots with sonar sensors or rodents' navigation using their whiskers ((Figure 2, top right). The superiority of VI^{2}N became appreciable in this challenging environment, where the middle walls and borders are not easily distinguishable (table 4, the second row from the bottom).

Our most challenging environment was called "symmetric". In the symmetric environment, four copies of a smaller random environment are placed in each corner of a larger grid-world (Figure 2, bottom right). The density of each of the four "rooms" (small environment block) was $5\%, 10\%,$ or 15%. This environment requires more long-term planning and information gathering, as it has more indistinguishable states that could lead to incorrect assumptions about belief in simpler updates. The symmetric grid-world model's environments where obstacles are closer together in a maze-like orientation with lots of repetition, such as trails through the forest or identical floors in a building. In this environment, the VI^{2}N drastically outperformed the QMDP-Net, which demonstrates its ability to use long-term planning to generate effective policies. We can also observe that the VI^{2}N was more robust to changes in sparsity in complex environments, whereas the QMDP-Net performance dropped at a higher rate as each type of environment became more challenging with the change in density/sparsity.

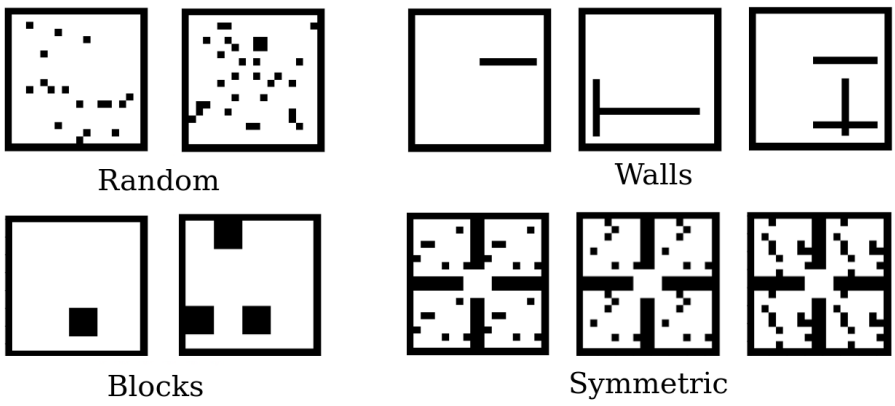

Figure 2: Example testing environments. There are four types of random, blocks, walls, and symmetric environments, each of which with various density rates. The goal is not shown in these maps.

5 Interpretability and the emergence of code for informative locations

Besides the superiority of VI^{2}N, measured by an objective measure of success rate, our method produces representations important both in terms of interpretability and understanding the computational foundations of natural intelligence. Specifically, in addition to producing value maps representing the space in terms of the reward values via the value function of single states ($V(s)$), VI^{2}N specifies informative areas via marginal pairwise values, i.e., $\sum_s V(s, s')$, as demonstrated in

Table 1: Success Rate of network solvers over various environments

Model	Environment				
	Random(5%)	Random(10%)	Walls(1)	Walls(2)	Walls(3)
VI^{2}N	93%	95%	77%	83%	82%
QMDP-Net	93%	96%	69%	78%	80%
	Blocks(5%)	Blocks(10%)	Symm(5%)	Symm(10%)	Symm(15%)
VI^{2}N	91%	91%	76%	74%	65%
QMDP-Net	88%	89%	61%	51%	41%

figure 3. This map significantly contributes to the interpretability of our method, explaining why specific actions were performed, especially when they were not directly related to the source of reward (goal). Informative states are not represented in the QMDP-net value function, even in the environments where the QMDP-Net performs well, such as "random" and "block". This is, in fact, expected as QMDP, the algorithm behind policy generation of QMDP-Net, does not take resolving uncertainty into account.

Notably, both rewarding and informative areas are represented in the hippocampal cells, broadly called "place cells", during navigation [32, 18]. The representation of informative areas is more visible when the environment is ambiguous, and a "landmark" is needed to resolve the uncertainty [10, 8]. These observations from neural recordings, along with extensive behavioral studies such as [11, 24], point to the importance, and even overemphasis, of informative states and information gathering in the decision-making of biological agents. The success of our network in different environments proves the usefulness of information-seeking behavior in decision making under uncertainty. Notably, in real world scenarios and for biological agents, almost all decision making situations are under uncertainty/partial observability.

Our representation results in simulated environments (figure 3) highlight something even more interesting related to the hippocampal cells. In all of the environments, especially non-dense ones (first three), borders are boldly represented in the pairwise values. Importantly, "border cells" (or "boundry cells") are one the most widely recognized place cells in the hippocampus [27, 19]. Our results suggest a theoretical/computational foundation for these cells: Borders provide a strong source of information about the state of the animals in the environment, helping them to navigate the world and eventually gain higher utility.

Our results are especially noteworthy since VI^{2}N structure is strictly shaped by known cognitive process and theories of decision making such as simulation of future events (planning) and information gathering. Therefore, it is more immune to common biases of general/unconstrained networks such as over-fitting and arbitrary implementation choices. This make its results more trustworthy in terms of connection to natural intelligence [25].

6 Discussion

We have introduced the VI^{2}N as a deep learning architecture for decision making under uncertainty, modeled after the fully differentiable Pairwise Heuristic. The VI^{2}N architecture demonstrates the ability for long-term planning for resolving the uncertainty which exceeds the capacity of previously proposed network architectures seen in the VIN and the QMDP-Net, especially in challenging environments with high perceptual ambiguity. Moreover, in addition to *reward value* maps, it generates *information value* maps, highlighting the informative areas in respect to the reward (goal). Besides interpretability, this representation resembles hippocampal place cells in the biological brain.

Since the main focus of our work is on the planning/policy module of the network, our environments were simple 2D binary grid worlds similar to VIN and QMDP-Net [29, 14]. We expect improvements of classic VIN over the past years [21, 31, 12] to be easily applicable to our network as the main component of our network is still a VI module. In fact, applying these improvements is an exciting future research direction to extend the applicability of our network.

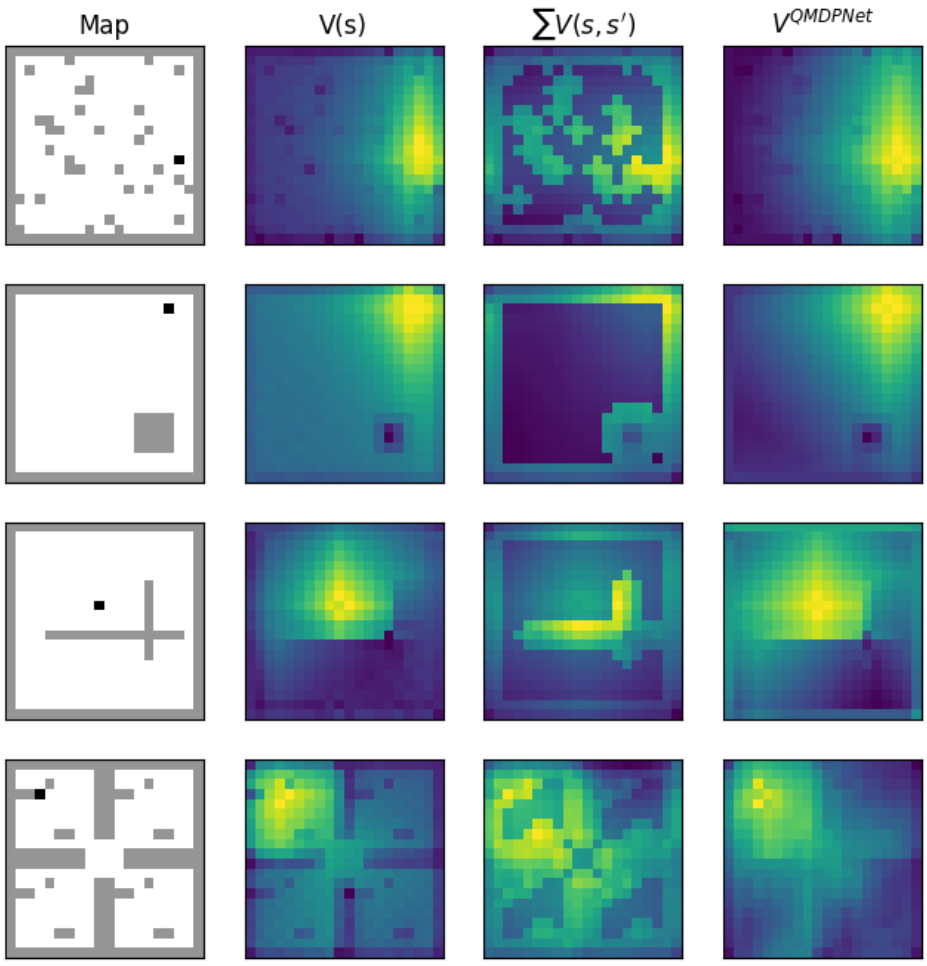

Figure 3: VI^{2}N represents rewarding areas through the value function of single states and informative areas through the value function of pairs. QMDP-Net (rightmost maps) focuses only on the reward. The leftmost maps represent the environment where the gray areas are obstacles and the black cell represents the goal.

All of our tasks were navigation to a known goal, while not knowing the agent's own position. This made us able to have environments with different levels of complexity for planning easily. Testing other tasks, such as grasping, would definitely contribute to the reliability of our results. However, designing scalable, challenging, and intuitive setups for other tasks is unfortunately complicated. For example, as shown in the QMDP-Net paper, the available grasping environment is not even challenging for the classic QMDP algorithm with more than 98 percent success rate [14].

Finally, our results are limited to learning from an expert (and not from reward reinforcement). Since in our setup, the expert uses the same type of algorithm, but on the perfect model, this learning resembles imitation learning from peers who know the environment or previous self-experience in familiar environments. Therefore, besides the plausibility of such learning in biological agents, we expect that our network can also learn from reward signals, only within a higher number of epochs in training.

References

[1] Misha B. Ahrens. Zebrafish Neuroscience: Using Artificial Neural Networks to Help Understand Brains. *Current Biology*, 29(21):R1138–R1140, November 2019.

[2] Andrea Banino, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, Martin J. Chadwick, Thomas Degris, Joseph Modayil, Greg Wayne, Hubert Soyer, Fabio Viola, Brian Zhang, Ross Goroshin, Neil Rabinowitz, Razvan Pascanu, Charlie Beattie, Stig Petersen, Amir Sadik, Stephen Gaffney, Helen King, Koray Kavukcuoglu, Demis Hassabis, Raia Hadsell, and Dharshan Kumaran. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705):429–433, May 2018. Number: 7705 Publisher: Nature Publishing Group.

[3] Richard Bellman. A Markovian Decision Process. *Journal of Mathematics and Mechanics*, 6(5):679–684, 1957. Publisher: Indiana University Mathematics Department.

[4] Matthew Botvinick, Jane X. Wang, Will Dabney, Kevin J. Miller, and Zeb Kurth-Nelson. Deep Reinforcement Learning and Its Neuroscientific Implications. *Neuron*, 107(4):603–616, August 2020.

[5] Logan Cross, Jeff Cockburn, Yisong Yue, and John P. O'Doherty. Using deep reinforcement learning to reveal how the brain encodes abstract state-space representations in high-dimensional environments. *Neuron*, 109(4):724–738.e7, February 2021.

[6] Chiara Della Libera and Leonardo Chelazzi. Learning to attend and to ignore is a matter of gains and losses. *Psychological Science*, 20(6):778–784, June 2009.

[7] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G. Bellemare, and Joelle Pineau. An Introduction to Deep Reinforcement Learning. *Foundations and Trends® in Machine Learning*, 11(3-4):219–354, December 2018. Publisher: Now Publishers, Inc.

[8] Tristan Geiller, Mohammad Fattahi, June-Seek Choi, and Sébastien Royer. Place cells are more strongly tied to landmarks in deep than in superficial CA1. *Nature Communications*, 8(1):14531, February 2017. Number: 1 Publisher: Nature Publishing Group.

[9] Daniel Golovin, Andreas Krause, and Debajyoti Ray. Near-optimal Bayesian active learning with noisy observations. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, NIPS'10, pages 766–774, Red Hook, NY, USA, December 2010. Curran Associates Inc.

[10] K. M. Gothard, W. E. Skaggs, K. M. Moore, and B. L. McNaughton. Binding of hippocampal CA1 neural activity to multiple reference frames in a landmark-based navigation task. *Journal of Neuroscience*, 16(2):823–835, January 1996. Publisher: Society for Neuroscience Section: Articles.

[11] Jacqueline Gottlieb, Mary Hayhoe, Okihide Hikosaka, and Antonio Rangel. Attention, Reward, and Information Seeking. *The Journal of Neuroscience*, 34(46):15497–15504, November 2014.

[12] Shu Ishida and João F. Henriques. Towards real-world navigation with deep differentiable planners. In *2022 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2022.

[13] Ingmar Kanitscheider and Ila Fiete. Training recurrent networks to generate hypotheses about how the brain solves hard navigation problems. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[14] Peter Karkus, David Hsu, and Wee Sun Lee. QMDP-Net: Deep Learning for Planning under Partial Observability. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[15] Peter Karkus, David Hsu, and Wee Sun Lee. QMDP-Net: Deep Learning for Planning under Partial Observability. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[16] Koosha Khalvati and Alan Mackworth. A Fast Pairwise Heuristic for Planning under Uncertainty. *Proceedings of the AAAI Conference on Artificial Intelligence*, 27(1):503–509, June 2013.

[17] Koosha Khalvati and Alan K. Mackworth. Active robot localization with macro actions. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 187–193, October 2012. ISSN: 2153-0866.

[18] Jae Sung Lee, John J. Briguglio, Jeremy D. Cohen, Sandro Romani, and Albert K. Lee. The Statistical Structure of the Hippocampal Code for Space as a Function of Time, Context, and Value. *Cell*, 183(3):620–635.e22, October 2020.

[19] Colin Lever, Stephen Burton, Ali Jeewajee, John O'Keefe, and Neil Burgess. Boundary vector cells in the subiculum of the hippocampal formation. *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience*, 29(31):9771–9777, August 2009.

[20] Tianwei Ni, Benjamin Eysenbach, and Ruslan Salakhutdinov. Recurrent Model-Free RL Can Be a Strong Baseline for Many POMDPs, June 2022. Number: arXiv:2110.05038 arXiv:2110.05038 [cs].

[21] Sufeng Niu, Siheng Chen, Hanyu Guo, Colin Targonski, Melissa Smith, and Jelena Kovačević. Generalized Value Iteration Networks:Life Beyond Lattices. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), April 2018. Number: 1.

[22] Blake A. Richards, Timothy P. Lillicrap, Philippe Beaudoin, Yoshua Bengio, Rafal Bogacz, Amelia Christensen, Claudia Clopath, Rui Ponte Costa, Archy de Berker, Surya Ganguli, Colleen J. Gillon, Danijar Hafner, Adam Kepecs, Nikolaus Kriegeskorte, Peter Latham, Grace W. Lindsay, Kenneth D. Miller, Richard Naud, Christopher C. Pack, Panayiota Poirazi, Pieter Roelfsema, João Sacramento, Andrew Saxe, Benjamin Scellier, Anna C. Schapiro, Walter Senn, Greg Wayne, Daniel Yamins, Friedemann Zenke, Joel Zylberberg, Denis Therien, and Konrad P. Kording. A deep learning framework for neuroscience. *Nature Neuroscience*, 22(11):1761–1770, November 2019. Number: 11 Publisher: Nature Publishing Group.

[23] S. Ross, J. Pineau, S. Paquet, and B. Chaib-draa. Online Planning Algorithms for POMDPs. *Journal of Artificial Intelligence Research*, 32:663–704, July 2008.

[24] Anthony W. Sali, Brian A. Anderson, and Susan M. Courtney. Information processing biases in the brain: Implications for decision-making and self-governance. *Neuroethics*, 11(3):259–271, October 2018.

[25] Rylan Schaeffer, Mikail Khona, and Ila R. Fiete. No Free Lunch from Deep Learning in Neuroscience: A Case Study through Models of the Entorhinal-Hippocampal Circuit. July 2022.

[26] Satpreet H. Singh, Floris van Breugel, Rajesh P. N. Rao, and Bingni W. Brunton. Emergent behaviour and neural dynamics in artificial agents tracking odour plumes. *Nature Machine Intelligence*, 5(1):58–70, January 2023. Number: 1 Publisher: Nature Publishing Group.

[27] Trygve Solstad, Charlotte N. Boccara, Emilio Kropff, May-Britt Moser, and Edvard I. Moser. Representation of Geometric Borders in the Entorhinal Cortex. *Science*, 322(5909):1865–1868, December 2008. Publisher: American Association for the Advancement of Science.

[28] Edward J. Sondik. The Optimal Control of Partially Observable Markov Processes over the Infinite Horizon: Discounted Costs. *Operations Research*, 26(2):282–304, 1978. Publisher: INFORMS.

[29] Aviv Tamar, YI WU, Garrett Thomas, Sergey Levine, and Pieter Abbeel. Value Iteration Networks. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[30] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents), 2005.

[31] Li Zhang, Xin Li, Sen Chen, Hongyu Zang, Jie Huang, and Mingzhong Wang. Universal Value Iteration Networks: When Spatially-Invariant Is Not Universal. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):6778–6785, April 2020. Number: 04.

[32] H Freyja Ólafsdóttir, Caswell Barry, Aman B Saleem, Demis Hassabis, and Hugo J Spiers. Hippocampal place cells construct reward related sequences through unexplored space. *eLife*, 4:e06063, June 2015. Publisher: eLife Sciences Publications, Ltd.