Factorized Value Iteration Network for Decision Making under Mixed Observability

Cynthia Chen, Michael Buice, Koosha Khalvati

Allen Institute
cynthia.chen@alleninstitute.org,
michaelbu@alleninstitute.org,
kooshakhalvati@alleninstitute.org

Abstract

Deep neural networks that incorporate classic reinforcement learning methods, such as value iteration, into their structure significantly outperform randomly structured networks in learning and generalization. These networks however, are mostly limited to environments with no or very low amounts of uncertainty. Value iteration with value of information network (VI^{2}N) shows a decent performance in highly uncertain environment. However, the network size grows quadratically in size of the state space, making it struggle in large environments. Here we show how taking advantage of mixed observability in many environments can be incorporated into VI^{2}N to make it significantly smaller and thus trainable in large domains. Mixed observability refers to situations where the state space could be divided into fully and partially observable components. We tested our network on a goal-based navigation task where the location of the goal is unknown to the agent. Our network significantly outperformed Q-MDP net, the only existing competitor for decision making under uncertainty among these types of networks.

Introduction

Deep neural networks have had tremendous success in Reinforcement Learning (RL) by providing an end-to-end solution from perception to action (François-Lavet et al. 2018). While networks with random structures could be trained based on expert policy, similar to supervised learning, incorporation of classic RL methods into them boosts their performance significantly (Tamar et al. 2016; Karkus, Hsu, and Lee 2017). For example, Value Iteration Networks (VINs) use long-term planning by implementing the value iteration algorithm (i.e. a sequence of Bellman updates) via convolutional layers (Bellman 1957; Tamar et al. 2016; Niu et al. 2018; Zhang et al. 2020; Ishida and Henriques 2022). 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 (Tamar et al. 2016).

Fully observable environments modeled by Markov Decision Processes (MDPs) are a well-explored domain in both classic and deep reinforcement learning. This is not true for Partially Observable Markov Decision Processes (POMDPs) with substantial complexity due to the inherent partial observability of the environment. Lack of full knowledge about the current state forces the agent to work with a belief space rather than a state space. This belief space consists of all probability distributions over possible states, and therefore has as many dimensions as there are states. The curse of dimensionality results in computationally intractable solutions for this exponentially growing belief space. As a result, there are significantly fewer methods and networks that work well in partially observable environments. For example, there are only two networks in the VIN family that are able to perform in partially observable environments, QMDP-Net and VI^{2}N (Karkus, Hsu, and Lee 2017). Being based on a very crude heuristic, i.e. QMDP, QMDP-Net can not perform well in environments with high degree of uncertainty. The VI^2N neural network utilizes a CNN to implement a POMDP solver, named the pairwise heuristic, in a differentiable manner (Khalvati and Mackworth 2013; Johnson, Buice, and Khalvati 2022). This method surpasses the performance of the QMDP-net, but takes significantly longer to learn the policy of a given environment. This is because although the dimensionality of the belief space is vastly reduced due to the heuristic that only considers pairs of beliefs rather than every possible combination of beliefs, the belief space could still be extremely large as it is quadratic in state space size.

Mixed Observable Markov Decision Processes (MOMDPs) are an alternative to the POMDP problem, where certain features about the state are fully observable while others are not (Araya-López et al. 2010). For example, in goal-directed navigation, the agent might know its own position but not where the goal is. Factorization of state space into fully and partially observable components can reduce the computational complexity of some solver's significance. Notably, not all POMDP solvers can benefit from this factorization. Here, first, we show how the pairwise heuristic can be modified for MOMDPs and take advantage of factorization. Based on this modification, we further show how VI^{2}N can be adapted to solve MOMDPs in an efficient and generalizable fashion (Johnson, Buice, and Khalvati 2022). We show the success of our approach in a goal-based navigation problem where the agent does not know where the goal is and needs to reach a certain landmark to gain that knowledge.

Background

Partially Observable Markov Decision Processes

The way the environment our agent operates in is represented as a Partially Observable Markov decision Process (POMDP). This is an extension of the Markov Decision Process (MDP) representation of an environment, which contains a set of states $s \in S$, and a set of actions $a \in A$, that connect states to each other based on a defined set of transition probabilities $P(s'|s,a)$. Each state has an associated reward given by the reward function $R(s,a)$, and the agent's goal in this environment is to maximize the total gained reward. In the partially observable case, we also have an observation function, which maps the likelihood of encountering an observation $z \in Z$ to each state $P(z|s)$. Finally, a POMDP agent operates on a belief state rather than a given state, which is a probability distribution over states, $b(s)$. The goal of the agent in a POMDP environment is to maximize the total gained reward.

Mixed Observable Markov Decision Process

A Mixed Observable Markov Decision Process (MOMDP) represents a middle ground between POMDPs and MDPs. In a MOMDP, certain dimensions of the state are visible, S_v while others are hidden, S_h (Araya-López et al. 2010). When a portion of the state is known, a factorized representation of the belief space can be used to reduce the dimensionality and complexity of the problem. As a result $B_v = S_v$, and any belief state can be represented as a combination of (S_v, B_h) (Ong et al. 2010). The dimensionality of B is therefore reduced from $|S|$ to $|S_h|$, yielding a large improvement in computational efficiency of some POMDP solvers.

Value Iteration Networks

The value iteration algorithm is an MDP solver where the optimal value of each state, which equals the expected total reward in the future, is computed through a series of Bellman updates (Bellman 1957):

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

When the transition function is spatially invariant, a neural network can learn the transition (T) and reward (R) functions by implementing the above equation with a series of convolutional layers (Tamar et al. 2016)). More specifically, given the map of the environment presented as an image, and the current state of the agent as the inputs and an expert's action or reward as the output, this network, called Value Iteration Network (VIN) learns convolutional kernels of f_R and f_P representing reward and transition functions. Value Iteration Networks (VINs), significantly outperform networks with similar computational power (e.g., layers) in learning to plan in novel environments (Tamar et al. 2016). These networks have been significantly improved in terms of applicability to domains with more complex structures over the past years (Niu et al. 2018; Zhang et al. 2020; Ishida and Henriques 2022). All of these improvements, however, are still mainly limited to fully observable environments.

POMDP has an additional observation function. As long as this function could be implemented by a network, for example with a convolutional kernel mimicking observing the surroundings, the network can learn it through training. The main challenge in solving POMDPs with neural networks is the implementation of the POMDP solver, which needs to be differentiable. Almost none of the POMDP solvers meet this criteria. Therefore, there are only two POMDP network solvers in this domain. The first one is QMDP-Net, which implements the QMDP heuristic (Karkus, Hsu, and Lee 2017). Founded upon a very simple heuristic, QMDP-Net fails in environments with high degree of uncertainty. VI^{2}N is the second network in this domain, which implements a POMDP solver named Pairwise Heuristic (Johnson, Buice, and Khalvati 2022). Implementing a more powerful heuristic, VI^{2}N performs well even in challenging environments. However, as the pairwise in the name suggests, the network size grows quadratically with state space size. This complexity makes VI^{2}N impractical in large domains.

Model

Our goal is to take advantage of the factorizability of MOMDPs to design an efficient network solver for an environment with Mixed Observability. To do this, we derive the MOMDP version of the Pairwise Heuristic and VI^{2}N.

Factorized Pairwise Heuristic

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 (Golovin, Krause, and Ray 2010; Khalvati and Mackworth 2012, 2013). In a POMDP, this would be the set of $|S|(|S|-1)/2$ optimal policies in each of which the belief is .5 for two states (Khalvati and Mackworth 2013). By factorizing out the portion of the belief space that is fully observable, this can be decreased to $|S_v||S_b|(|S_b|-1)/2$, which combines the visible state subspace with the pairwise combination of the hidden state subspace. The pairwise heuristic is easily factorizable because the state space of the pairwise function can be easily reduced, unlike in other algorithms like QMDP, where there is no reducible state.

When looking at the paired states, the expected total reward of each of these policies is the *value of the pair*, shown by $V(s,s')$, for $s, s' \in S$. When factorizing S into S_v and S_h, only the S_h component contains possible ambiguity, so the pairwise value can be expressed as $V(s_v, s_h, s'_h)$, decreasing the total possible values that need to be calculated for the model. Finding the optimal solution when the uncertainty is about two states is still computationally expensive. As a result, the Pairwise Heuristic further simplifies its policy by prioritizing resolving uncertainty when calculating $V(s_v, s_h, s'_h)$, before exploiting the reward. Resolving uncertainty is not always necessary to gain the optimal reward. However, it produces a good enough solution.

The Pairwise Heuristic defines a pair of states as distinguishable if there is a high probability that different obser-

vations are recorded in the two states. Since our factorized MOMDP implies that the a portion of the state is already known, the distinguishabilty of that dimension of the states does not need to be calculated. Therefore, we only have to determine the pairwise distinguishabilty of hidden dimension, keeping the visible dimension constant across pairs. Formally, for each visible factor s_v, (s_v, s_h) and (s_v, s'_h) are distinguishable if and only if:

$$\sum_{o} \sum_{s_h, s'_h} p(o|(s_v, s_h))(1-p(o|(s_v, s'_h)))+p(o|(s_v, s'_h))(1-p(o|(s_v, s_h))) > 2\lambda \tag{2}$$

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 ($V(s_v, s_h, s'_h)$) 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., $.5(V(s_v, s_h) + V(s_v, s'_h))$. 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. This requires a pairwise transition function, which is determined by the joint transition probability of the states in the pair. The reward of each pair is the average reward of the two states in the original problem:

$$\forall s_v : R(s_v, s_h, s'_h) = 0.5(R(s_v, s_h) + R(s_v, s'_h)) \tag{3}$$

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

$$V_k(s_v, s_h, s'_h) = \max_a [R(s_v, s_h, s'_h)+\gamma \sum_{s'_v, s''_h, s'''_h} T(s_v, s_h, s'_h, a, s'_v, s''_h, s'''_h) V_{k-1}(s'_v, s''_h, s'''_h)] \tag{4}$$

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

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

$$\forall s_v : a^*_k = \arg\max_a \sum_{(s_h, s'_h)} b(s_v, s_h, s'_h) Q((s_v, s_h, s'_h), a) \tag{5}$$

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.

Environment

To demonstrate the factorized VI^{2}N architecture, we develop an environment with mixed observability that consists of the visible state variables (the agent x and y coordinate locations), and a hidden variable that indicates the true distribution of rewards out of the possible goal locations indicated on the map. We can manipulate the number of possible goal locations displayed on the map, G, increasing the uncertainty of the environment by adding more possible goal locations. The maps are all squares, with a side length of n and states S of size $|S| = n * n * G$. Our environment contains one informational landmark that reveals the actual location of the goal, although a more complex observation function could be used and still yield a factorizable environment. Since the agent's x and y coordinate location is known, this landmark is the only relevant observation. The size of our observation space is $|G + 1|$, as the observation can reveal the true state that designates one rewarding location out of G, or nothing. The landmark observation is akin to an oracle revealing the actual state of the world. The agent could rely on the oracle for information or act purely on its own priors. In our environment, the observational cues are few but dense.

The agent is able to take the actions right, up, down, and left, moving itself one cell in the direction specified by the name and also action stay. Taking more actions leads to a reduced reward, as each a discount is applied to each reward based on how long each sequence of actions took to get there. Additionally, there is a positive reward when the agent finds the true goal, while there is a large negative reward when the agent reaches any of the false goals. This reward set up encourages the agent to quickly and correctly identify the true goal.

Factorized VI^{2}N Architecture

In general, the fundamentals of (factorized) VI^{2}N architecture remain the same across different types of problems. Only minor implementations, such as transition kernel, are different.

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. 4), which uses the pairwise transition, and the pairwise reward functions (Eq. 3). 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 1.

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 $R(s)$ into $R(s_v, s_h, s'_h)$ for all $s_h, s'_h \in S_h \times S_h$. Because of mixed observability, we only have to calculate the reward for the pairs of s_h for a given s_v, instead of calculating every combination of (s_v, s'_v) and (s_h, s'_h), as shown in Figure 1. Since the underlying model of our environment structures the goal states as giving a small reward or large penalty, the pairwise reward is negative, encouraging the agent to determine the true value of the goal state before reaching it.

In our environment, the hidden states (s_h) are fully disconnected. Therefore, the transition function T is used as the kernel in the VI Module (f_P) and is only applied on S_v. This allows us to simplify our transition function to, $T(s_v, a, s'_v)$,

applying this uniform transition function on all pairs of s_h instead of creating a separate pairwise transition function.

Initial pairwise values, $V(s_v, s_h, s'_h)$ or V_{pair}, are set as R_{min} for the indistinguishable states. This assumes the worst case scenario and uses it as the default value for each pair of states. As the pairwise heuristic dictates, for all states in s_v where aspects of the hidden variable can be distinguished through observation, the initial value is set to $0.5[R(s_v, s_h)+R(s_v, s'_h)+V(s_v, s_h)+V(s_v, s'_h)]$, for all of the pairs that can be distinguished in that s_v, in an attempt to approximate the true value of that state. The transition kernel is then recursively applied to $V_{pair} + R_{pair}$ in order to perform value iteration and propagate the values of the distinguishable pairs to the rest of the state space.

We use the observation function to determine which states are able to be distinguished from each other. Since we already know that s_v is fully observable, the distinguishabilty matrix dictates which states in s_v also have information about s_h, reducing the representation from $|S|$ to $|(s_v, s_b)|$. Our observation function is derived from the landmark layer of the map, since the landmark block, (x_l, y_l), is the only informative position. Because of that property of the environment, the only pairs that are distinguishable are pairs where $s_v = (x_l, y_l)$

The pairwise value initialization ($V_0(s_v, s_h, s'_h)$) is done using matrix multiplication of D, the distinguishabilty matrix, and $.5(V(s_v, s_h) + V(s_v, s'_h))$ (for distinguished pairs) in addition to multiplication of $(1 - D)$ and $min_{R(S)}$ in the shape of an $|S_v| \times |S_v|$ matrix (for indistinguishable pairs). With the pairwise reward and transition function calculated, the pairwise value iteration (Eq. 4) is just another VI module, which we call VI^2 module since it is in the pairwise space. Finally, the action selection (Eq. 5) is done by multiplying the pairwise belief state (outer product of belief by itself) with pairwise Q values and taking the sum of the weighted Q values.

Results

We compared our factorized VI^{2}N model with the other POMDP network solver, QMDP-net. We trained and tested these two models on various grid worlds that have the environmental attributes described earlier. The size and number of possible goals were manipulated in these grid worlds to investigate the generalizability of our architecture and the effect of increasing uncertainty on the results. We kept the observation and action function constant among the environments to have a systematic comparison in terms of uncertainty and complexity of the decision-making. Finally, we used the same belief update mechanism for both methods (QMDP-Net and factorized VI^{2}N) to have a fair comparison between the two policy modules.

The two neural networks were trained on the same datasets with a set of Pairwise Heuristic expert solutions. Networks were trained only on successful trials (less than 50 steps needed to reach the goal) to resemble positive reinforcement, although the failure rate was very low, under 1%. The training set contained 1000 independently generated environments, each containing 10 solution paths to

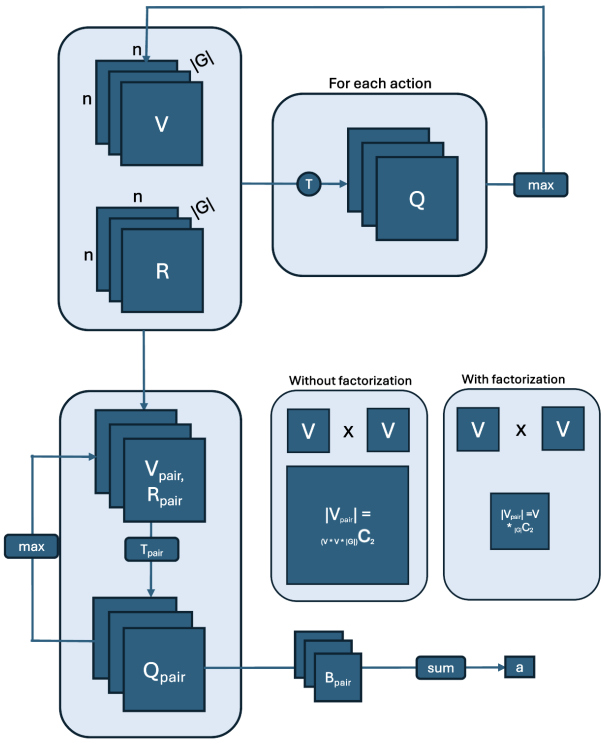

Figure 1: Factorized VI^{2}N architecture and comparison of network size with or without factorization.

yield 9000-10000 action labels when accounting for the unsuccessful solutions. Training performance was evaluated through $90\% - 10\%$ train-validation process, and the result reported is the average success rate of 5 training sequences for each environment. The test success rate was then calculated by running the generated model on 1,000 novel environments of the same type, defining a success as a trial where the agent reaches the goal by solely following the policy in less than 50 steps. The initial belief state for the tests was always uniform among all aspects of S_h.

Our results are displayed in Table 1. We first tested the models on our benchmark environment, a 10×10 grid world with two possible goals. In this case, the pairwise solver reached the goal far more often than the QMDP-net, proving its superiority in the most basic environment. We then moved on to more complex environments to show how the pairwise net maintains its advantage over the QMDP-net despite increasing complexity.

The first variable that we manipulated was the amount of possible goal states in each environment. When increasing the amount of goals from 2 to 3, the performance of QMDP-net increases, going from a 28% success rate to a 32% success rate. This is odd but is explained by a switch in strategy as the number of goals increases to a strategy that chooses one goal to navigate to regardless of the actual underlying

Table 1: Success Rate of network solvers over various environments.

Side Length	No. Goal Places	VI^{2}N	QMDP-net
10	2	0.98 ± 0.01	0.28 ± 0.01
10	3	0.97 ± 0.01	0.32 ± 0.02
10	4	0.96 ± 0.01	0.22 ± 0.02
15	2	0.96 ± 0.01	0.14 ± 0.02

state, leading to a success rate of about $1/|G|$. When the number of goals is further increased to 4 goals, that trend is followed, with a success rate of 22%. For the pairwise solver, an increased number of goals slightly degraded the performance of the model, but the VI2N solver still consistently performed far better than QMDP-Net.

After testing the network's performance on the environment with more goals, we also tested the two networks on a larger environment. We expected the performance of both networks to decrease on a larger network, but hypothesized that the use of factorization in the VI2N would allow it to handle an increase in the observable state space better than the QMDP-net model. We trained both models on a new set of 15x15 grid world environments, with two possible goals for each environment. As predicted, the VI2N-net model performed slightly less well on the larger environment, but still solved most cases, whereas the QMDP-net's performance decreased dramatically.

Interpretability and the emergence of code for informative locations

Overall, we found that the factorized VI^{2}N model was able to successfully learn the proper representations for the environment dynamics and reward function. Furthermore, looking at the value maps that our model generated for each environment, we found mappings that followed our intuitions about information seeking and value exploitation (Figure 2). In uncertain environments, where pairwise values are considered, the agent prioritizes information acquisition and localization, assigning the landmark state greater value. After the agent has cleared the ambiguity and determined the true goal state, the agent assigns increasing value to the state the true goal is in, avoiding the red herring states.

When looking at the navigation strategy of VI^{2}N, the model consistently reaches the landmark to determine where the goal is and subsequently navigates to the goal. Irrespective of the number of potential goals or size of the environment, the VI^{2}N navigated to the landmark first in about 95% of the total trials. On the other hand, QMDP-net was shown to go to the landmark in about 30% - 50% of the test trials in all cases and generally does not usefully incorporate this information into its navigation strategy.

These observed differences in performance and behavior imply that the VI^{2}N model is better at solving MOMDP problems. The structure of the VI^{2}N encourages the agent to decrease ambiguity before exploiting high-value actions. This yields good results in the MOMDP environment because the hidden variable is resolved, and then value is exploited according to the optimal action mapping based on the known state. Especially because the hidden state is factorized, resolving uncertainty for the MOMDP is not too computationally expensive, making it reasonable to prioritize information gain. In contrast, QMDP-net does not have a structure that inherently prioritizes information seeking, which makes finding the reward less reliable because the state information frequently remains ambiguous throughout the navigation process.

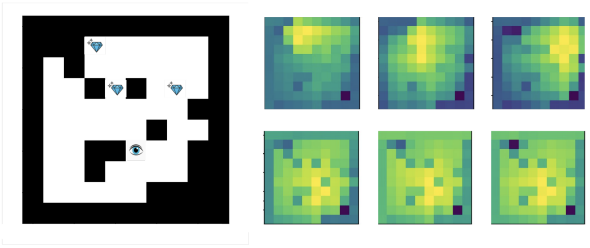

Figure 2: $S(left), V(s)(top), V(s, s')(bottom)$ for three possible places of the goal.

Discussion

We have introduced the factorized VI^{2}N as a deep learning architecture for decision making under mixed observability, modeled after the fully differentiable Pairwise Heuristic. The VI^{2}N architecture demonstrates the ability for long-term planning for resolving uncertainty which exceeds the capacity of previously proposed network architectures seen in the VIN and the QMDP-Net. As shown in Figure 2, in addition to *reward value* maps, it generates *information value* maps, highlighting the informative areas in respect to the reward (goal). Taking advantage of the factorizability of the heuristic, which was the main focus of our paper, allows us to compute value functions for complex environments far more efficiently.

In all of our tasks, the agent had full observability of their own location while being presented with a set of potential locations for the goal, where only one was the actual goal. We were able to manipulate this task to account for varying levels of uncertainty but kept the basic structure of the task the same. We have yet to explore varying the size and scope of the landmark structures or changing the dynamics that dictate where the reward is located. Additionally, formulating other environments with mixed observability, such as grasping, where the cartesian location of the arm is known, but maybe its height is obscured, would contribute to our results' reliability. 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 relatively easy for the classic QMDP algorithm with more than 98 percent success rate (Karkus, Hsu, and Lee 2017).

References

Araya-López, M.; Thomas, V.; Buffet, O.; and Charpillet, F. 2010. A Closer Look at MOMDPs. In *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, volume 2, 197–204. ISSN: 2375-0197.

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

François-Lavet, V.; Henderson, P.; Islam, R.; Bellemare, M. G.; and Pineau, J. 2018. An Introduction to Deep Reinforcement Learning. *Foundations and Trends® in Machine Learning*, 11(3-4): 219–354. Publisher: Now Publishers, Inc.

Golovin, D.; Krause, A.; and Ray, D. 2010. Near-optimal Bayesian active learning with noisy observations. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, NIPS'10, 766–774. Red Hook, NY, USA: Curran Associates Inc.

Ishida, S.; and Henriques, J. F. 2022. Towards real-world navigation with deep differentiable planners. In *2022 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Johnson, S.; Buice, M. A.; and Khalvati, K. 2022. VI $ˆ 2$ N: A Network for Planning Under Uncertainty based on Value of Information. In *Deep Reinforcement Learning Workshop NeurIPS 2022*.

Karkus, P.; Hsu, D.; and Lee, W. S. 2017. QMDP-Net: Deep Learning for Planning under Partial Observability. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Khalvati, K.; and Mackworth, A. 2013. A Fast Pairwise Heuristic for Planning under Uncertainty. *Proceedings of the AAAI Conference on Artificial Intelligence*, 27(1): 503–509.

Khalvati, K.; and Mackworth, A. K. 2012. Active robot localization with macro actions. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 187–193. ISSN: 2153-0866.

Niu, S.; Chen, S.; Guo, H.; Targonski, C.; Smith, M.; and Kovačević, J. 2018. Generalized Value Iteration Networks:Life Beyond Lattices. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1). Number: 1.

Ong, S. C. W.; Png, S. W.; Hsu, D.; and Lee, W. S. 2010. Planning under Uncertainty for Robotic Tasks with Mixed Observability. *The International Journal of Robotics Research*, 29(8): 1053–1068. Publisher: SAGE Publications Ltd STM.

Tamar, A.; WU, Y.; Thomas, G.; Levine, S.; and Abbeel, P. 2016. Value Iteration Networks. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Zhang, L.; Li, X.; Chen, S.; Zang, H.; Huang, J.; and Wang, M. 2020. Universal Value Iteration Networks: When Spatially-Invariant Is Not Universal. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04): 6778–6785. Number: 04.