Perception-Based Beliefs for POMDPs with Visual Observations

Miriam Schäfers
Ruhr-University Bochum
Bochum, Germany
m.schaefers@rub.de

Merlijn Krale
Radboud University
Nijmegen, The Netherlands
merlijn.krale@ru.nl

Thiago D. Simão
Eindhoven University of Technology
Eindhoven, The Netherlands
t.simao@tue.nl

Nils Jansen
Ruhr University Bochum
Bochum, Germany
Radboud University
Nijmegen, The Netherlands
n.jansen@rub.de

Maximilian Weininger
Ruhr University Bochum
Bochum, Germany
maximilian.weininger@rub.de

ABSTRACT

Partially observable Markov decision processes (POMDPs) are a principled planning model for sequential decision-making under uncertainty. Yet, real-world problems with high-dimensional observations—such as camera images—remain intractable for traditional belief- and filtering-based solvers. To tackle this problem, we introduce the **P**erception-based **B**eliefs for **P**OMDPs framework (PBP), which complements such solvers with a perception model. This model takes the form of an image classifier which maps visual observations to probability distributions over states. PBP incorporates these distributions directly into belief updates, so the underlying solver does not need to reason explicitly over high-dimensional observation spaces. We show that the belief update of PBP coincides with the standard belief update if the image classifier is exact. Moreover, to handle classifier imprecision, we incorporate uncertainty quantification and introduce two methods to adjust the belief update accordingly. We implement PBP using two traditional POMDP solvers and empirically show that (1) it outperforms existing end-to-end deep RL methods and (2) uncertainty quantification improves robustness of PBP against visual corruption.

KEYWORDS

POMDP; Partial Observability; Planning; Perception; Belief

ACM Reference Format:
Miriam Schäfers, Merlijn Krale, Thiago D. Simão, Nils Jansen, and Maximilian Weininger. 2026. Perception-Based Beliefs for POMDPs with Visual Observations. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/XUFR9329

1 INTRODUCTION

Partially observable Markov decision processes [POMDPs; 20] formalise sequential decision-making under uncertainty in partially observable settings [35, Chapter 16.4]. Many applications of POMDPs, such as robotics and autonomous driving [2, 23], require reasoning about observation spaces based on high-dimensional images. We call such problems *vision POMDPs* [VPOMDPs; 5].

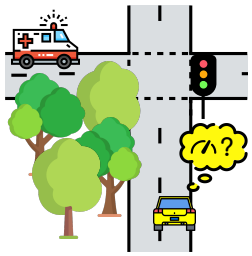

Figure 1: A vision POMDP example: A car must determine the presence of an ambulance using an audio sensor and the traffic light's color by interpreting camera images.

Example 1. Consider an autonomous vehicle that needs to determine how to approach an intersection (Figure 1). Its goal is to cross only if (1) the traffic light is green, and (2) no ambulance approaches the junction. The vehicle is equipped with a camera to detect the traffic light and an audio sensor to detect an ambulance. Both sensors may provide imperfect information, e.g. due to weather conditions, occlusion, or background noise. This problem can be viewed as a POMDP, where states describe the location of the car, the color of the traffic light, and the presence of an ambulance, and observations correspond to the sounds and camera images. Modeling the observation function would require assigning probabilities to every possible observation of the traffic light; in particular, the image can be subject to various sources of *visual corruption*, caused by weather conditions, occlusion, etc.

The main challenge in solving VPOMDPs are the massive spaces of visual observations. *End-to-end deep reinforcement learning (DRL)* methods, described in the related work below, may be able to deal with such observations. However, DRL agents are often hard to explain, do not exhibit any guarantees on their behavior, and their performance is unpredictable. These drawbacks render them unsuitable for safety-critical settings. In contrast, *planning* methods, like SARSOP [24], POMCP [38], DESPOT [46], and AdaOPS [45], often provide results that are easier to interpret, have guarantees on optimality, and find better policies than end-to-end learning approaches. However, these *belief-based* methods are restricted to

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/XUFR9329

smaller state- and action spaces and, in particular, currently cannot deal with massive observation spaces. We address this research gap.

Our Contribution

Our approach. This paper presents the **P**erception-based **B**eliefs for **P**OMDPs framework (PBP) for solving VPOMDPs. The key idea of our approach is separating interpretation of visual observations from belief-based planning. More precisely, PBP uses a perception model to map visual observations to probability distributions over states. We use these probability distributions to perform an efficient belief update without having to reason over the entire observation space. This new belief update rule relies on structural assumptions that naturally arise in many VPOMDP settings, see Section 3. PBP can integrate this belief update into any belief-based POMDP solver, allowing the resulting solver to scale to massive observation spaces.

Uncertainty quantification. Perception models are susceptible to erroneous predictions, especially in the presence of visual corruption. To increase the robustness of PBP, we use *uncertainty quantification* [10] to estimate the uncertainty of the perception model about its prediction and introduce two methods to incorporate the uncertainty into the decision-making process: The *threshold-based* method discards predictions with high uncertainty during the belief update. The *weighting-based* method adjusts the impact of the perception on the belief update according to its uncertainty.

Empirical evaluation. To demonstrate the feasibility of our approach, we implement two instantiations of PBP using the classical POMDP algorithms HSVI [39] and POMCP [38]. We evaluate both variants on novel benchmarks that feature small state and action spaces but complex visual observations. Our experiments confirm that PBP operates as intended and can be effectively combined with existing POMDP solvers. In particular, PBP with HSVI achieves competitive performance compared to state-of-the-art VPOMDP solvers, especially under visually corrupted image observations.

In summary, our main contributions are:

- We introduce a novel method of performing a **belief update** in POMDPs using predictions from a perception model.
- We use **uncertainty quantification** to make this update more robust against visual corruption.
- We introduce the **PBP** framework, which allows integrating our belief update into existing POMDP solvers, and empirically show this approach is competitive with prior work.

Related Work

Discretization. Many methods combine standard algorithms with discretizing the observation space, such as POMCPOW [43] and DESPOT-α [11] (which both use progressive widening [3]), or the dynamic methods of Hoey and Poupart [17]. However, finding a discretization with small information loss is challenging for images.

Monte Carlo methods. Methods based on particle filters, such as AdaOPS [45], LABECOP [18], SPARSE-PFT [27], and BetaZero [30] use Monte-Carlo sampling to avoid reasoning over the full observation space. However, such methods still require full knowledge of the observation function to update the particle filter, which is unrealistic for VPOMDPs.

End-to-end deep RL. Deep reinforcement learning methods often transform visual observations into latent representations via, e.g., convolutional layers (such as in [16]), variational auto encoders (such as in [15]), or transformers (as discussed in Ni et al. [31]). Other deep reinforcement learning methods aim to learn state estimation by learning components of a particle filter through neural networks [29], by learning components of the belief update through variational autoencoders [19]. These methods are trained end-to-end, which means the network needs to synchronously learn a policy and corresponding latent representation or state estimation. This is often computationally- and data-inefficient, and makes methods hard to explain or verify.

Perception models. Some methods use perception models to predict state features from images. However, it remains an open question how to best integrate such predictions into decision-making, especially in partially observable settings. Recent work has explored ways to incorporate uncertainty quantification [26, 28], though these methods do not incorporate history. In contrast, our framework uses beliefs to unify perception and decision-making and takes both history and uncertainty into account.

Compositional learning. Visual Tree Search (VTS) [5] extends DualSMC [44] by learning generative models offline to define particle filter components, and employs a particle belief-based Monte Carlo Tree Search planner [43] to select actions. In contrast, our framework generalizes to any belief-based method (of which we consider particle filters a subset). Moreover, our framework directly uses a perception model, which allows the take advantage of off-the-shelf *Computer Vision* architectures and pretrained models to explicitly interpret the visual observations in belief construction.

Partially supervised RL. PSRL [25] uses image classifier predictions as inputs to an otherwise end-to-end trained decision maker. However, these predictions are only used at the current timestep, whereas our method provides a principled way to incorporate all past predictions into a single belief.

2 PRELIMINARIES

$\Delta(X)$ denotes the set of all probability distributions over a set X.

2.1 Planning with State Uncertainty

POMDPs. A *partially observable Markov decision process* (POMDP) is a tuple $\mathcal{M} = \langle \mathsf{S}, \mathsf{A}, \mathsf{T}, \mathsf{R}, \gamma, b_0, \mathsf{Z}, \mathsf{O} \rangle$, with finite sets of states S, actions A, and observation Z; a probabilistic transition function $\mathsf{T}\colon \mathsf{S}\times\mathsf{A} \to \Delta(\mathsf{S})$; a reward function $\mathsf{R}\colon \mathsf{S}\times\mathsf{A} \to \mathbb{R}$; a discount factor $\gamma \in (0,1)$; an initial state distribution $b_0 \in \Delta(\mathsf{S})$; and an observation function $\mathsf{O}\colon \mathsf{S} \to \Delta(\mathsf{Z})$.[1] We employ a factored state and observation representation with a combination of different *variables*: $\mathsf{S} = \times_{i\in[1,\dots,n]} S_i$ and $\mathsf{Z} = \times_{i\in[1,\dots,m]} Z_i$, with states and observations given as tuples $\mathbf{s} = (s_1, \dots, s_n) \in \mathsf{S}$ and $\mathbf{z} = (z_1, \dots, z_m) \in \mathsf{Z}$. Single state-variables need not exactly correspond to single observation variables, hence $n \neq m$ is valid. In our setting, some observation variables z_i for $1 \leq i \leq m$ are images, see Section 3.

Semantics of POMDPs. A POMDP evolves as follows: At each time step $t \in \mathbb{N}_0$, the system is in a state $\mathbf{s}_t \in \mathsf{S}$, initially sampled according to b_0. The agent selects an action $a_t \in \mathsf{A}$, obtains reward

[1] Our method can be generalized to observation functions that also depend on actions and next states and reward functions that also depend on next states.

$\mathsf{R}(\mathbf{s}_t, a_t)$, and the system transitions to the next state $\mathbf{s}_{t+1}$ that is sampled according to $\mathsf{T}(\mathbf{s}_t, a_t)$. Repeating this interaction of agent and POMDP yields a *path* $\rho \in (S \times A) \times (S \times A) \times \ldots$; however, in POMDPs, the agent cannot observe $\mathbf{s}_t$, but instead obtains observations $\mathbf{z}_t \sim \mathsf{O}(\mathbf{s}_t)$. Thus, instead of ρ, the agent only knows the observable *history* $h \in (\mathsf{Z} \times \mathsf{A}) \times (\mathsf{Z} \times \mathsf{A}) \times \ldots$; such a history can be summarized using a *belief* $b_t \in \mathsf{B} \coloneqq \Delta(\mathsf{S})$, i.e. a probability distribution for the current state.

A *policy* $\pi \colon \mathsf{B} \to \Delta(\mathsf{A})$ describes the agent's behaviour by mapping beliefs to actions. The objective of the agent is to maximize its expected cumulative discounted reward, or *value*, defined as:

$$\mathsf{V}_\pi \coloneqq \mathbb{E}_\pi\Big[\sum_{t=0}^{\infty} \gamma^t \mathsf{R}(\mathbf{s}_t, a_t)\Big],$$

where $\mathbb{E}_\pi$ is the expectation over paths under the probability measure induced by executing a POMDP with fixed policy π.

2.2 Perception Using Deep Neural Networks

The main goal in perception is to learn a model that can reliably extract meaningful information from images. Crucially, the model should generalize to unseen examples from the same distribution, rather than overfitting to the training data. For this task, (convolutional) deep neural networks are the standard, outperforming traditional methods when trained on large datasets [35].

DNN classifiers. A *deep neural network (DNN) classifier* is a function $f \colon X \to \Delta(L)$, mapping an input image $x \in X$ to a probability distribution over possible classes $l \in L$, where L is the set of possible classes [13]. The probability distribution is obtained by normalizing raw confidence scores through the *softmax* function [35, Chapter 22.2.2]. Typically, a single class is predicted by choosing the one with the highest probability. DNNs are trained using supervised learning based on a dataset $D \subset X \times L$ consisting of pairs of images and classes. In practice, the predicted probabilities reflect the model's confidence based on its training experience, rather than true posterior likelihoods. However, these confidence scores often do not align with the actual probability of correctness, which we mitigate using *temperature scaling* [14], a post-hoc calibration method that adjusts the sharpness of the softmax distribution without changing the order of the probabilities.

Uncertainty functions. Uncertainty quantification [12] evaluates the uncertainty in a DNN's predictions, allowing to ignore predictions that are likely incorrect. Given a classifier $f : X \to \Delta(L)$, a function $u_f : X \to [0, 1]$ mapping an image $x \in X$ to a score $u_f(x)$ is an uncertainty function of f if a high value of $u_f(x)$ indicates greater uncertainty, i.e. a higher likelihood of an incorrect prediction. Many existing uncertainty functions are surveyed in [12]. We use (i) the prediction confidence $u_f(x) \coloneqq 1 - \max_{i=1,\ldots,|L|} f(l_i|x)$ and (ii) the entropy of the softmax output $u_f(x) \coloneqq -\sum_{i=1,\ldots,|L|} f(l_i|x) \cdot \log_2 f(l_i|x)$. Further, (iii) we employ the advanced Monte Carlo Dropout method, introduced in [10].

3 THE VISION POMDP PROBLEM

We define *vision POMDPs* (VPOMDPs) in close alignment with the informal definition of vision POMDPs by Deglurkar et al. [5].

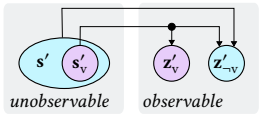

Figure 2: An illustration (as Bayesian network representation) of the observation function of a VPOMDP under Assumption 1. Notably, $\mathbf{z}'_{\mathrm{v}}$ may only depend on the visual variables of $\mathbf{s}'$, i.e. $\mathbf{s}'_{\mathrm{v}}$, while $\mathbf{z}_{\neg\mathrm{v}}$ can depend on all variables of $\mathbf{s}'$.

VPOMDPs are POMDPs in which some of the observation variables correspond to high-dimensional images.

Definition 1 (Vision POMDP)**.** Let $\mathcal{M}$ be a POMDP with a factored state and observation representation $\mathsf{S} = \times_{i\in[1,\ldots,n]} S_i$ and $\mathsf{Z} = \times_{i\in[1,\ldots,m]} Z_i$. Let $\mathsf{I}_\mathsf{Z} \subseteq [1, \ldots, m]$ be the indices of the observation variables that correspond to images. Then $\mathsf{Z}_\mathrm{v} = \times_{i\in\mathsf{I}_\mathsf{Z}} Z_i$ are the possible *vision observation components* and $\mathsf{Z}_{\neg\mathrm{v}} = \times_{i\in[1,\ldots,m]\setminus\mathsf{I}_\mathsf{Z}} Z_i$ are the possible non-vision observation components. We call $\mathcal{M}$ a *vision POMDP* if $\mathsf{Z}_\mathrm{v} \neq \emptyset$, and a *pure vision POMDP* if $\mathsf{Z}_{\neg\mathrm{v}} = \emptyset$.

The key challenge for VPOMDPs lies in the massive size of the observation space, which is effectively unbounded (see Example 1). Technically, this challenge renders it intractable to specify the probability of each possible observation, which is necessary to formally model the observation function O. We therefore assume that the observation function is not fully known, but can only be approximated using data. To make our problem tractable, we assume we have some knowledge about which state variables can have an effect on which observation variables.

We first introduce some additional notation. For a vision POMDP, let I_S denote the set of indices that factors the state space S into vision state components $\mathsf{S}_\mathrm{v} = \times_{i\in\mathsf{I}_\mathsf{S}} S_i$ and non-vision state components $\mathsf{S}_{\neg\mathrm{v}} = \times_{i\in[1,\ldots,n]\setminus\mathsf{I}_\mathsf{S}} S_i$, such that the vision observation components Z_v are independent of the non-vision state components $\mathsf{S}_{\neg\mathrm{v}}$, see Figure 2 for an illustration. We write $\mathbf{s}_\mathrm{v}$ and $\mathbf{s}_{\neg\mathrm{v}}$ for vision and non-vision components of a state $\mathbf{s}$, and analogously $\mathbf{z}_\mathrm{v}$ and $\mathbf{z}_{\neg\mathrm{v}}$ for an observation $\mathbf{z}$. Such a factorization is always possible, since independence between vision observation components and non-vision state components trivially holds for $\mathsf{I}_\mathsf{S} = [1 \ldots n]$. Assumption 1 ensures the factorization is meaningful.

Assumption 1 (Vision-factorizable.)**.** A VPOMDP $\mathcal{M}$ is *vision-factorizable* if the observation function O factorizes into a vision observation function $\mathsf{O}_\mathrm{v} \colon \mathsf{S}_\mathrm{v} \to \Delta(\mathsf{Z}_\mathrm{v})$ and a non-vision observation function $\mathsf{O}_{\neg\mathrm{v}} \colon \mathsf{S} \to \Delta(\mathsf{Z}_{\neg\mathrm{v}})$, such that:

$$\mathsf{O}(\mathbf{z} \mid \mathbf{s}) = \mathsf{O}_\mathrm{v}(\mathbf{z}_\mathrm{v} \mid \mathbf{s}_\mathrm{v}) \cdot \mathsf{O}_{\neg\mathrm{v}}(\mathbf{z}_{\neg\mathrm{v}} \mid \mathbf{s}).$$

Interpretation. Recall our motivating Example 1. The vision component of the observation, $\mathbf{z}_\mathrm{v}$, consists of camera images of the traffic light, while the non-vision component, $\mathbf{z}_{\neg\mathrm{v}}$, consist of audio input to detect an ambulance. $\mathbf{s}_\mathrm{v}$ describes all state components that affect $\mathbf{z}_\mathrm{v}$, which is only the color of the traffic light. The observations of the traffic light are independent of hearing the ambulance siren. Thus, the probability of receiving a particular observation, $\mathsf{O}(\mathbf{z} \mid \mathbf{s})$, can simply be written as the product of the probabilities of receiving the corresponding image and audio input, i.e. $\mathsf{O}_\mathrm{v}(\mathbf{z}_\mathrm{v} \mid \mathbf{s}_\mathrm{v}) \cdot \mathsf{O}_{\neg\mathrm{v}}(\mathbf{z}_{\neg\mathrm{v}} \mid \mathbf{s})$. Thus, Assumption 1 holds.

Vision datasets. Assumption 1 allows us to decompose the observation function into two parts. The non-vision observation function has a discrete and small domain, which means it can be learned or approximated using standard methods. In this work, we assume the non-vision observation function is given, and instead focus on the more challenging task of dealing with the vision observation function, which cannot be fully learned due to its large, high-dimensional domain. We assume to have access to a *vision dataset*, which we define as follows:

Definition 2 (Vision dataset)**.** A *vision dataset* is a finite set of pairs $(\mathbf{z}_v, \mathbf{s}_v) \in \mathsf{Z}_v \times \mathsf{S}_v$, where each vision component of an observation $\mathbf{z}_v$ is sampled from the conditional distribution defined by the vision observation function O_v, given the vision component of the corresponding state $\mathbf{s}_v$:

$$D \subset \left\{ (\mathbf{z}_v, \mathbf{s}_v) \;\middle|\; \mathbf{s}_v \in \mathsf{S}_v,\ \mathbf{z}_v \sim \mathsf{O}_v(\cdot \mid \mathbf{s}_v) \right\}.$$

We highlight that D is a strict subset of the full relation between vision components of states and their corresponding observations, since the complete set of all possible vision components of observations for all vision components of states is infeasibly large.

For a given VPOMDP, a vision dataset can be constructed by sampling from the model or a simulator, if these are available. Additionally, we can leverage existing vision datasets: In the field of computer vision, vision datasets are made publicly available on a wide range of vision tasks, e.g. [6, 32]. If public dataset images closely resemble a concrete problem (e.g., traffic light images for Example 1), we can use the images for that problem.

With such a dataset, we use the methods of Section 2.2 to train a perception model that predicts $\Pr(\mathbf{s}_v \mid \mathbf{z}_v)$, the likelihood of a state's vision component $\mathbf{s}_v \in \mathsf{S}_v$ given an observation's vision component $\mathbf{z}_v \in \mathsf{Z}_v$. In Section 4, we explain how these probabilities can be used to solve the following problem:

> **Problem Statement** (Perception POMDP problem)**.** *Given a vision POMDP $\mathcal{M}$ satisfying Assumption 1, with known model dynamics but unknown vision part of the observation function* O*, and given a vision dataset D, compute a policy π that maximizes the expected cumulative discounted reward* V_π.

Discussion of Assumption 1. Verifying Assumption 1 is part of the modeling process, as it depends on the semantic relationships between observation variables. Nevertheless, Assumption 1 can also be interpreted as an additional restriction on the policy rather than as an assumption about the environment: We disallow the agent to use $\mathbf{z}_v$ to infer information about $\mathbf{s}_{\neg v}$. Thus, the policies we find are valid (though possibly suboptimal) even if Assumption 1 is violated. For instance, in Example 1, we may consider the location of the car to be a non-vision variable and the color of a traffic light as a vision variable. In that case, our assumption does not hold: The images of the traffic light could be used to infer information about our location. However, ignoring this information will still yield a valid policy that could perform reasonably well.

We emphasize that, unlike other works using factored representations [1, 21], we do not require the existence of a factored representation of the transition function. Moreover, in contrast to approaches that aim to predict states directly from images, we do not rely on the *block assumption* [8, 41, 48], which requires that each observation corresponds to just one state. Lastly, our assumption is not related to those used for PORL (e.g., [42]).

4 THE PERCEPTION-BASED BELIEFS FOR POMDPS FRAMEWORK (PBP)

This section formalizes the *Perception-based Beliefs for POMDPs* framework (PBP). Section 4.1 provides the core ingredient: the perception-based belief update that is able to handle massive observation spaces. Section 4.2 explains how we use uncertainty quantification to address an imprecise perception model. Section 4.3 describes the overall framework, summarized in Figure 3, and embeds the perception-based belief update into various POMDP solvers.

4.1 Belief Update for Vision POMDPs

We recall the standard belief update, see e.g. [35, Chapter 16.4]. We denote $\Pr(\mathbf{s}' \mid b, a) \coloneqq \sum_{\mathbf{s} \in \mathsf{S}} b(\mathbf{s}) T(\mathbf{s}' \mid \mathbf{s}, a)$. For a current belief b_t, the next belief b_{t+1} upon selecting action a and observing $\mathbf{z}$ is:

$$b_{t+1}(\mathbf{s}') \coloneqq \frac{\mathsf{O}(\mathbf{z} \mid \mathbf{s}') \cdot \Pr(\mathbf{s}' \mid b_t, a)}{\sum_{\mathbf{s}'' \in \mathsf{S}} \mathsf{O}(\mathbf{z} \mid \mathbf{s}'') \cdot \Pr(\mathbf{s}'' \mid b_t, a)}. \tag{1}$$

This update depends on the probabilities $\mathsf{O}(\mathbf{z} \mid \mathbf{s})$; however, due to the massive observation space, the vision component $\mathsf{O}_v(\mathbf{z}_v \mid \mathbf{s}_v)$ is unknown. We rewrite O_v using *Bayes' rule*:

$$\mathsf{O}_v(\mathbf{z}_v \mid \mathbf{s}_v) = \frac{\Pr(\mathbf{s}_v \mid \mathbf{z}_v) \cdot \Pr(\mathbf{z}_v)}{\Pr(\mathbf{s}_v)}. \tag{2}$$

$\Pr(\mathbf{s}_v \mid \mathbf{z}_v)$ can be approximated using a *perception model* $f \colon \mathsf{Z}_v \to \Delta(\mathsf{S}_v)$, such that $\Pr(\mathbf{s}_v \mid \mathbf{z}_v) \approx f(\mathbf{s}_v \mid \mathbf{z}_v)$. We can obtain such a perception model by learning an image classifier using the vision dataset D, defined in Definition 2. For the prior $\Pr(\mathbf{z}_v)$, the specific choice has no impact, because we prove below that the term cancels out in the belief update. For $\Pr(\mathbf{s}_v)$, we choose the uninformative prior [47, pp. 41–43], i.e. a uniform distribution over S_v. Intuitively, this means we do not introduce any bias into the approximation of the observation function, but solely rely on the perception model. To show that this choice is appropriate, we show in [36, Appendix A] that the belief update can be viewed as a form of multiplicative *opinion pooling* [7]. In this view, we find that the uniform prior is the only choice that reproduces the standard belief update.

Perception-based belief update. We insert Equation (2) into Equation (1) to define a new belief update for vision POMDPs. For simplicity, we first define the next belief b^{pv}_{t+1} for ***p**ure **v**ision POMDPs* where $\mathsf{Z}_{\neg v} = \emptyset$ and thus $\mathbf{s} = \mathbf{s}_v$.

$$b^{pv}_{t+1}(\mathbf{s}') = \frac{\frac{\Pr(\mathbf{z}_v)}{\Pr(\mathbf{s}'_v)} f(\mathbf{s}'_v \mid \mathbf{z}_v) \cdot \Pr(\mathbf{s}' \mid b_t, a)}{\sum_{\mathbf{s}'' \in \mathsf{S}} \frac{\Pr(\mathbf{z}_v)}{\Pr(\mathbf{s}''_v)} f(\mathbf{s}''_v \mid \mathbf{z}_v) \Pr(\mathbf{s}'' \mid b_t, a)}. \tag{3}$$

As stated above, $\Pr(\mathbf{z}_v)$ cancels out. Moreover, since we assume a uniform prior over states, we have $\Pr(\mathbf{s}'_v) = \Pr(\mathbf{s}''_v)$ for all pairs of states. Thus, these also cancel out, leading to the simplified form:

$$b^{pv}_{t+1}(\mathbf{s}') = \frac{f(\mathbf{s}'_v \mid \mathbf{z}_v) \cdot \Pr(\mathbf{s}' \mid b_t, a)}{\sum_{\mathbf{s}'' \in \mathsf{S}} f(\mathbf{s}''_v \mid \mathbf{z}_v) \Pr(\mathbf{s}'' \mid b_t, a)}. \tag{4}$$

In general, i.e. when $\mathsf{Z}_{\neg v} \neq \emptyset$, we utilize the same idea and the fact that by Assumption 1, we can rewrite the belief update in terms

of O_v and $O_{\neg v}$ to obtain Equation (5):

$$b^v_{t+1}(\mathbf{s}'_v) \coloneqq \frac{f(\mathbf{s}'_v \mid \mathbf{z}_v)\cdot O_{\neg v}(\mathbf{z}_{\neg v} \mid \mathbf{s}')\cdot \Pr(\mathbf{s}' \mid b_t, a)}{\sum_{\mathbf{s}''\in S} f(\mathbf{s}''_v|\mathbf{z}_v)\cdot O_{\neg v}(\mathbf{z}_{\neg v}|\mathbf{s}'')\cdot \Pr(\mathbf{s}''|b_t, a)}. \tag{5}$$

We state in Theorem 1 (proven in [36, Appendix B]) that if the image classifier was exact, Equation (5) allows to recover the standard belief update, thereby overcoming the problem that the massive observation space makes the standard belief update infeasible.

Theorem 1 (Soundness of Equation (5)). *Consider a vision POMDP satisfying Assumption 1, and let f be a perfect perception model, i.e. $f(\mathbf{s}_v|\mathbf{z}_v) = \Pr(\mathbf{s}_v|\mathbf{z}_v)$. Then, for every belief $b_t \in \Delta(S)$, the next belief upon playing action $a \in A$ and observing $\mathbf{z} \in Z$ computed using Equations (1) and (5) coincide, i.e. $b^v_{t+1}(\mathbf{s}') = b_{t+1}(\mathbf{s}')$.*

4.2 Uncertainty Awareness

In practice, we do not know the true probabilities $\Pr(\mathbf{s}_v \mid \mathbf{z}_v)$, but can only learn an approximation by training a perception model f on a vision dataset D. Errors arise from two sources: Firstly, the vision dataset is necessarily incomplete, and thus the perception model has to generalize to unseen images. Secondly, ideally every image would correspond to a unique $\mathbf{s}_v \in S_v$.[2] Realistically, however, images may be corrupted (e.g. the traffic light being covered in snow), such that they cannot be assigned to a unique class. Overall, the predicted probabilities are not perfect. Therefore, it is important to know when these probabilities are likely to be incorrect, i.e., to quantify the uncertainty of the perception model.

To address these imperfections, we employ an *uncertainty-aware perception model*. It utilizes an uncertainty function u_f to measure the uncertainty of the perception model, and, intuitively, ignores the output of the perception model in case of high uncertainty. Ignoring the perception model means assuming that for each vision state variable $\mathbf{s}_v \in S_v$, we assign uniform probability to every value in its domain: $U(\mathbf{s}_v) = \frac{1}{|S_v|}$.

We define two *uncertainty-aware* perception models: Firstly, ***t**hreshold-based **u**ncertainty **q**uantification* (TUQ) simply uses a *threshold value* ε to determine which probabilities to use, closely matching the use of uncertainty quantification in existing literature [4].

$$f_{\text{TUQ}}(\mathbf{s}_v \mid \mathbf{z}_v) = \begin{cases} f(\mathbf{s}_v \mid \mathbf{z}_v) & \text{if } u_f(\mathbf{z}_v) \le \varepsilon \\ U(\mathbf{s}_v) & \text{otherwise.} \end{cases} \tag{6}$$

However, TUQ is sensitive to the choice of ε. Thus, secondly, we define ***w**eighted **u**ncertainty **q**uantification* (WUQ) to eliminate this dependency. WUQ provides a smooth interpolation between the perception model and the uniform distribution as a function of uncertainty. When the uncertainty is below 0.5, the perception model is considered trustworthy and influences WUQ; if the uncertainty exceeds this threshold, WUQ disregards the perception output and defaults to the uniform distribution:

$$f_{\text{WUQ}}(\mathbf{s}_v|\mathbf{z}_v) = \begin{cases} u_f(\mathbf{z}_v)U(\mathbf{s}_v) + \big(1-u_f(\mathbf{z}_v)\big)f(\mathbf{s}_v|\mathbf{z}_v) & \text{if } u_f(\mathbf{z}_v)<0.5 \\ \Pr(\mathbf{s}_v) & \text{otherwise.} \end{cases} \tag{7}$$

[2] This assumption is related to, but weaker than, the one used in *Block MDPs* [8, 41, 48], which requires uniqueness of the entire state $\mathbf{s}$ rather than only its vision component $\mathbf{s}_v$.

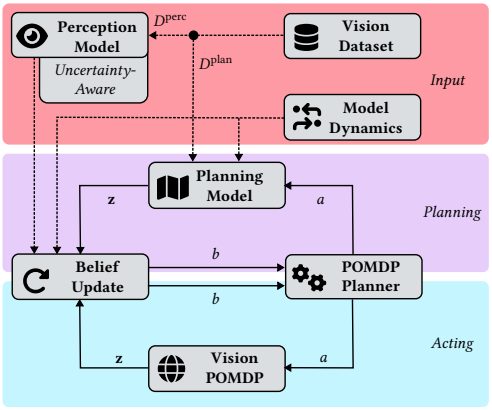

Figure 3: Overview of the *Perception-based Beliefs for POMDPs* Framework (PBP). See Section 4.3 for a detailed explanation.

Both functions degenerate to f if $u_f = 0$; thus, Theorem 1 also applies when using uncertainty-aware perception models. In cases where the output of the perception model is ignored, the belief update relies solely on the transition probabilities and the non-vision observation function, avoiding the inclusion of potentially unreliable information from the perception model.

In practice, the perception model may yield incorrect probabilities, and its associated uncertainty may not fully reflect this, potentially leading to the following corner case: Let the support of a distribution d be defined as $\text{supp}(d) = \{x \mid d(x) > 0\}$. If the support of the perception model and of the belief after propagation through the transition dynamics do not overlap, the belief update produces an empty belief. In this case, we adopt a uniform distribution over all states as an uninformative *fallback belief*.

4.3 Overall Algorithm

We outline in Figure 3 how the perception-based belief update is incorporated into the overall solution process of VPOMDPs. We now provide a detailed description, split into (1) the planning cycle (middle, purple box), (2) the acting cycle (bottom, blue box), and (3) details on how different planners use the new belief update.

The planning cycle. A **POMDP Planner** maintains value estimates for different beliefs. Improving these estimates requires querying the next belief b' upon playing action a and receiving observation $\mathbf{z}$ (see below and [36, Appendix C]). However, when dealing with VPOMDPs, it is unclear how the vision component of an observation should be sampled during planning, as the distribution O_v is unknown. PBP addresses this by using an approximation of O_v based on a subset D^{plan} of the **Vision Dataset**:

$$\hat{O}_v(\mathbf{z}_v|\mathbf{s}_v) = \frac{|\{(\mathbf{z}'_v, \mathbf{s}'_v) \in D^{\text{plan}} \mid \mathbf{z}'_v = \mathbf{z}_v, \mathbf{s}'_v = \mathbf{s}_v\}|}{|\{(\mathbf{z}'_v, \mathbf{s}'_v) \in D^{\text{plan}} \mid \mathbf{s}'_v = \mathbf{s}_v\}|}. \tag{8}$$

Together with the known **Model Dynamics**, this approximation yields the **Planning Model** $\hat{\mathcal{M}}$, a fully specified POMDP approximating the true VPOMDP. Using this, the **POMDP Planner** can sample observations $\mathbf{z}$.

In principle, the **Belief Update** could work on the **Planning Model**, using the known approximation of the observation function

$\hat{O}_v$. However, as every vision component of an observation in the vision dataset is labelled with a unique $\mathbf{s}_v$, the planner would never encounter uncertainty about the vision component of the state. Consequently, we instead apply the perception-based belief update (Equation (5)) already during planning, allowing the planner to reason about both the state information in the images and the uncertainty in their interpretation. Naturally, this belief update uses the **Perception Model (with Uncertainty Quantification)**, trained[3] based on the second subset D^{perc} of the **Vision Dataset**.

The acting cycle. The **POMDP Planner** computes a policy π. Starting from the initial belief b_0, the action a recommended by π is executed in the **VPOMDP**, yielding the next observation $\mathbf{z}$. This observation (and knowledge of the chosen action a and previous belief b) is used by the perception-based **Belief Update** to compute the next belief b'. The cycle repeats with the next action $\pi(b')$.

Planning with the new belief update. The perception-based belief update can be incorporated into many planning algorithms. To demonstrate this versatility, we describe how to do so for different approaches in [36, Appendix C]. We provide a brief overview here.

For *Point-based methods*, such as PBVI [33], HSVI [39, 40], and SARSOP [24] the belief update affects two components of the algorithm: belief sampling and the backup operation. We first introduce a non-standard way to represent the backup operator, which makes its use of the belief update explicit. Then, for both belief sampling and the backup operator, we use $\hat{\mathcal{M}}$ to approximate the possible observations and use Equation (5) to perform belief updates.

Particle filter based methods, such as POMCP [38], DESPOT [46] and AdaOPS [45] use a set of particles as an approximation of the belief. Most particle filters use the probabilities $\mathrm{O}(\mathbf{z} \mid \mathbf{s})$ in the update step. As with beliefs, we can use the perception model to approximate these probabilities. We show that rejection sampling, such as used in POMCP, can be viewed as rejecting a particle x with probability $1 - \mathrm{O}(\mathbf{z} \mid x)$, which can be approximated as above. As in the belief update, $\Pr(\mathbf{z})$ does not affect these updates.

Deep RL methods can utilize our belief update in a similar fashion as PSRL [25]. In that case, the belief represents the latent representation that is used by the agent.

We emphasize that, via Theorem 1, PBP 's replacement of the belief update preserves the solver's properties as long as the perception model is good enough. The runtime is dominated by the chosen solver, as the (new) belief update itself is relatively inexpensive.

5 EMPIRICAL ANALYSIS

In this section, we empirically evaluate our approach to answer the following questions:

- **Q1. Performance:** How does PBP compare to existing baseline methods for vision POMDP tasks?
- **Q2. Robustness:** Is PBP robust against visual corruption?
- **Q3. Uncertainty awareness:** How do different uncertainty awareness approaches affect the performance of PBP?

[3] PBP can utilize different computer vision methods to learn the perception model. In particular, the training of the perception model could be bootstrapped with pretrained models if these are available for the problem setting.

Table 1: Average cumulative discounted rewards (V) and policy computation times (t) of different method on a number of benchmarks. For the methods using PBP or PSRL, training times of the perception model are not included.

	Intersection		*FlowerGrid*		*FrozenLake* (4)		*FrozenLake* (8)	
Algorithms	V	t (s)	V	t (s)	V	t (s)	V	t (s)
PBP-HSVI	-3.57	405	**56.6**	412	0.64	50.5	**0.31**	445
tPBP-HSVI	-4.47	404	52.3	392	0.64	52.4	**0.31**	461
wPBP-HSVI	-4.11	406	56.5	447	**0.65**	53.2	**0.31**	454
tPBP-POMCP	-15.7	53K	30.8	105K	0.54	54K	0.26	108K
PSRL-HSVI	**-3.41**	303	45.1	308	0.64	13.2	**0.31**	318
DQN	-6.35	1.2K	34.5	3.9K	0.64	1.4K	0.01	2.1K
PSRL-DQN	-4.03	1.7K	40.1	1.8K	0.64	1.7K	**0.31**	1.8K
Oracle	-3.32	302	57.8	13.4	0.64	3.03	0.31	18.6
NoPerc	-15.4	319	24.3	325	0.45	336	0.23	334

We first give an overview of our experimental setup, then answer these questions. Our code is available online [37], and further technical specifications are included in [36, Appendix F].

5.1 Experimental Setup

Benchmarks. We consider three environments. These have medium-sized state spaces but massive visual observation spaces that are sufficiently non-trivial to investigate the necessity and applicability of our framework. All environments use a discount factor of 0.95. For further environment details, see [36, Appendix D].

- *Intersection* is a custom environment that models Example 1. We use realistic traffic light images from [22].
- *FlowerGrid* is a custom 5×5 gridworld where every state corresponds to a flower type from the 102 Category Flower Dataset [32]. The goal of the agent is to pick a target flower and reach a goal while avoiding poisonous flowers. A visualization of the environment is given in Figure 4.
- *FrozenLake* is a variant of Gymnasium's *FrozenLake* [9], but with an added latent variable that determines surface slipperiness. The agent observes this slipperiness and can infer its location using a top-down image of the environment. We consider the default 4×4 and 8×8 maps.

Vision datasets. For each environment, the corresponding vision dataset is split into three distinct parts: D^{perc} and D^{plan} as described in Section 4.3,[4] , and D^{act}, which is used to evaluate the computed policy. Thus, the agent is evaluated with images that have not been used in its training, which simulates a situation where the observation space is unknown or intractably large.

Algorithms. We instantiate PBP with two existing POMDP solvers: HSVI [40] and POMCP [38], see [36, Appendix C] for implementation details. tPBP-HSVI and tPBP-POMCP utilize threshold-based uncertainty quantification (Equation (6)), with $\epsilon = 0.1$ unless noted otherwise. wPBP-HSVI uses weighted uncertainty quantification (Equation (7)). Unless noted otherwise, we use Monte Carlo Dropout (MCDO) as uncertainty function. [36, Appendix E] details the DNN architectures of the perception model for each environment. All DNNs achieve an accuracy of over 0.8 on their test sets (see Figure

[4] For the end-to-end algorithm DQN, we use $D^{\text{perc}} \cup D^{\text{plan}}$ for planning.

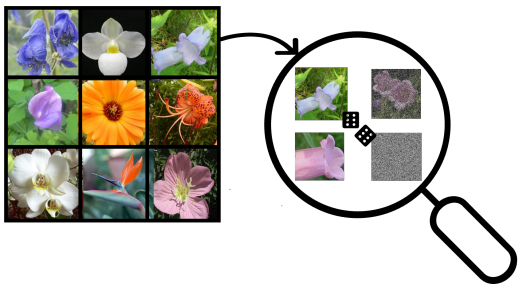

Figure 4: Visualization of *FlowerGrid*. For each cell, the set of possible observations corresponds to images of a particular class in the 102 Category Flower Dataset [32]. The magnifying glass shows two normal images, as well as an *additive-noise* (top right) and *pure-noise image* (bottom left) image.

11 in [36, Appendix E]) and were trained in under 10 minutes. We compare to the following baselines:

- DQN [16], an end-to-end deep RL method. We use the *Stable-Baselines3* [34] and add memory via framestacking.
- PSRL-DQN and PSRL-HSVI, two implementation of the PSRL framework [25]. Like PBP, PSRL utilizes a perception model to compute probability distributions over states. These probabilities are interpreted as latent variables for PSRL-DQN and as beliefs for PSRL-HSVI.
- NoPerc, a naive variant of HSVI that does not incorporate the vision component of observations in the belief update.
- Oracle, an implementation of HSVI that has full observability of $\mathbf{s}_\mathrm{v}$, mimicking a perfect perception model.

For all PBP- and PSRL-based methods, we first use the perception function to precompute probabilities for all images in our dataset. Then, HSVI-algorithms are restricted to a computational time of 300s for policy computations, while tPBP-POMCP has a time budget of 600s per step. DQN and PSRL-DQN may take arbitrary time, only being restricted to 300,000 environment interactions for training. For the specific runtimes, see Table 1. Our results represent the average after 1,000 episodes for all methods except tPBP-POMCP, where we take the average over 10 episodes.

5.2 Performance and Computational Cost

To address question Q1, we run all methods on our set of benchmarks and record both the average cumulative discounted reward V and the policy computation time t. Recall that runtime depends more on the underlying solver than our framework (see end of Section 4.3). Results are shown in Table 1.

PBP-HSVI outperforms existing methods. Across all benchmarks, PBP-HSVI performs on-par with or higher than all other algorithms, and only slightly lower than that of the Oracle baseline. Both tPBP-HSVI and wPBP-HSVI perform slightly worse than PBP-HSVI, but comparably on all benchmarks except *Intersection*.

The performances of the existing baselines (PSRL-HSVI, DQN, and PSRL-DQN) is more inconsistent. PSRL-HSVI performs roughly on-par with PBP-HSVI on all benchmarks except *FlowerGrid*. The accuracy of our perception model is lower for *FlowerGrid* than for the others, which causes more errors in the perception model. Such errors have a larger impact on PSRL-HSVI than on PBP-based approaches, since these can instead use knowledge of the dynamics and previous belief to compute an accurate belief. Despite their significantly larger time budget, DQN and PSRL-DQN never achieve higher returns than PSRL-HSVI. Moreover, DQN perform significantly worse on both *Intersection* and *FrozenLake* (8). These results suggest that combining our PBP with belief-based solvers is competitive with deep learning methods for VPOMDPs.

tPBP-POMCP requires optimization to be competitive. Lastly, tPBP-POMCP outperforms NoPerc on all environments, but performs significantly worse than all other algorithms. This comes from scalability limitations of the current implementation rather than any inherent weakness of the approach. An ablation study in [36, Appendix G] supports this claim: The particle filter of tPBP-POMCP accurately tracks the true belief across all environments, even under visual corruption (see Section 5.3). Since PBP only modifies the particle filter update, this indicates that the reduced performance stems from implementation inefficiencies rather than conceptual shortcomings. A more optimized implementation that leverages efficient data structures would yield substantially better results. Therefore, we exclude tPBP-POMCP from the remaining experiments and provide additional results in [36, Appendix G].

5.3 Robustness Against Visual Corruption

Next, we investigate question Q2 by adding *corrupted images* to our environments: For each dataset D^{plan} and D^{act}, we create two additional variants: (1) a set of *additive-noise images*, where we add salt-and-pepper noise to each image such that the perception model has an accuracy of roughly 0.4, and (2) a set of *pure-noise images*, where images are fully replaced by salt-and-pepper noise. For an example of corrupted images, see Figure 4. Intuitively, additive-noise images are hard to classify, but the probabilities of the perception model could still contain valuable information; for pure-noise images, the probabilities from the perception model cannot be useful. Further details on corrupted images are in [36, Appendix D.4]. For our experiments, we select a portion of the images in D^{plan} and D^{act} according to a *noise probability*, and replace these with corrupted variants. The performance of our methods at different noise probabilities, for both types of corruption, is shown in Figure 5.

PBP is robust against visual corruption. PBP-HSVI, tPBP-HSVI and wPBP-HSVI perform well for *FlowerGrid*, *FrozenLake* (4), and *FrozenLake* (8) at all noise probabilities. The performance of all PBP-based methods degrades more slowly than that of the other methods, and all three outperform the NoPerc baseline until almost all images are affected by noise. Most notably, tPBP-HSVI performs better than NoPerc in all our experiments. Still, for *Intersection*, the performance of all three methods degrades more than in the other environments. This is because there, the perception model is commonly overconfident for noisy images, i.e., it yields inaccurate but highly confident predictions (see Figure 11 in [36, Appendix E.2]). Such predictions cannot be detected by uncertainty quantification and lead to larger errors in the belief update.

In comparison, the existing methods are less robust against noise: While DQN is competitive for additive noise on *Intersection* and *FrozenLake* (4), it performs worse in all other cases. Similarly, PSRL-DQN is comparable to PBP on *Intersection* and *FrozenLake*, but

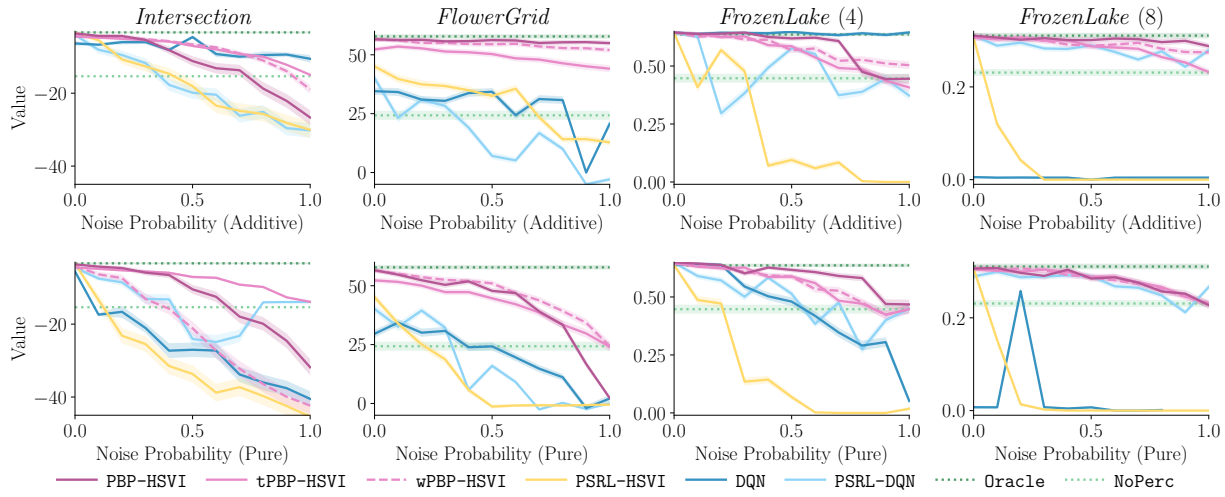

Figure 5: Average discounted returns (*Value*) for different algorithms at different probabilities of receiving noisy observations. Additive noise refers to images that are correctly classified with 0.4 probability, while full noise refers to pure salt-and-pepper images. Shaded areas show 95% confidence in the value of the tested policy.

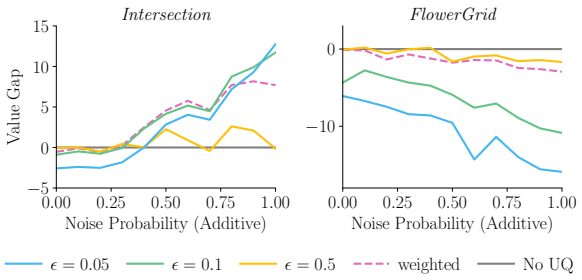

Figure 6: Value gap between PBP-HSVI and variants of wPBP-HSVI and tPBP-HSVI with different uncertainty thresholds, at different (additive) noise probabilities.

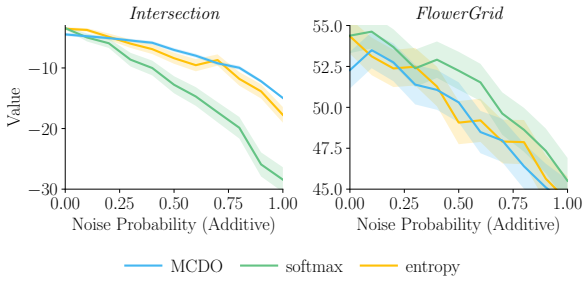

Figure 7: Value for tPBP-HSVI using different uncertainty functions, at different (additive) noise probabilities.

is worse on *FlowerGrid*. Both existing methods can perform significantly worse than just ignoring perceptions using NoPerc.

5.4 Uncertainty Quantification

To investigate question Q3, we compare PBP-HSVI with different variants of our belief update and different uncertainty functions.

Performance of UQ methods is environment-dependent. Figure 6 shows the value gap between the uncertainty-agnostic PBP-HSVI and with wPBP-HSVI, as well as several variants of tPBP-HSVI with different thresholds ϵ. Results are mixed: For *Intersection*, we find that lower thresholds improve performance at high noise probabilities, as expected. However, for *FlowerGrid*, lower thresholds lead to worse performance; in particular, PBP-HSVI without uncertainty quantification (i.e. $\epsilon = 1$) performs best. We conjecture that the perception model for *FlowerGrid* still provides a more accurate prediction than a uniform distribution, only being uncertain between a few of the many possible classes. In such cases, even if the perception model has limited accuracy, it can be advantageous to use it. wPBP-HSVI performs well on both environments and does not require finetuning, which makes it an attractive choice.

Next, Figure 7 shows the value of tPBP-HSVI (with $\epsilon = 0.1$) for the different uncertainty functions introduced in Section 2.2. All functions achieve roughly equal performance for *FlowerGrid*, with prediction confidence performing slightly better than the others. However, prediction confidence performs significantly worse on *Intersection*. We thus conclude that the optimal choice of uncertainty function depends on the specific environment, though MCDO and entropy seem to yield more consistent results.

6 CONCLUSION

PBP is a principled and modular approach for solving vision POMDPs using belief-based POMDP solvers. By integrating the perception model and its associated uncertainty with the belief update, PBP avoids the need to reason over massive observation spaces. Our results demonstrate that using PBP with the POMDP solver HSVI outperforms existing baselines.

ACKNOWLEDGEMENTS

This work has been partially funded by the ERC Starting Grant DEUCE (101077178).

REFERENCES

[1] Craig Boutilier, Richard Dearden, and Moisés Goldszmidt. 2000. Stochastic dynamic programming with factored representations. *Artificial Intelligence* 121, 1 (2000), 49–107. https://doi.org/10.1016/S0004-3702(00)00033-3

[2] Anthony R Cassandra. 1998. A survey of POMDP applications. In *Working notes of AAAI 1998 fall symposium on planning with partially observable Markov decision processes*, Vol. 1724.

[3] Adrien Couëtoux, Jean-Baptiste Hoock, Nataliya Sokolovska, Olivier Teytaud, and Nicolas Bonnard. 2011. Continuous Upper Confidence Trees. In *LION (Lecture Notes in Computer Science, Vol. 6683)*. Springer, 433–445.

[4] Sina Däubener, Kira Maag, David Krueger, and Asja Fischer. 2024. Integrating uncertainty quantification into randomized smoothing based robustness guarantees. *CoRR* abs/2410.20432 (2024). https://doi.org/10.48550/ARXIV.2410.20432

[5] Sampada Deglurkar, Michael H. Lim, Johnathan Tucker, Zachary N. Sunberg, Aleksandra Faust, and Claire J. Tomlin. 2023. Compositional Learning-based Planning for Vision POMDPs. In *L4DC (Proceedings of Machine Learning Research, Vol. 211)*. PMLR, 469–482. https://proceedings.mlr.press/v211/deglurkar23a.html

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*. IEEE Computer Society, 248–255. https://doi.org/10.1109/CVPR.2009.5206848

[7] Franz Dietrich and Christian List. 2016. 519Probabilistic Opinion Pooling. In *The Oxford Handbook of Probability and Philosophy*. Oxford University Press. https://doi.org/10.1093/oxfordhb/9780199607617.013.37 arXiv:https://academic.oup.com/book/0/chapter/365891921/chapter-ag-pdf/45096062/book_43657_section_365891921.ag.pdf

[8] Simon S. Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudík, and John Langford. 2019. Provably efficient RL with Rich Observations via Latent State Decoding. In *ICML (Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 1665–1674. http://proceedings.mlr.press/v97/du19b.html

[9] Farama Foundation. 2023. Gymnasium: A modern, maintained fork of OpenAI Gym. https://github.com/Farama-Foundation/Gymnasium.

[10] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *ICML (JMLR Workshop and Conference Proceedings, Vol. 48)*. JMLR.org, 1050–1059. http://proceedings.mlr.press/v48/gal16.html

[11] Neha Priyadarshini Garg, David Hsu, and Wee Sun Lee. 2019. DESPOT-Alpha: Online POMDP Planning with Large State and Observation Spaces. In *Robotics: Science and Systems*. https://doi.org/10.15607/RSS.2019.XV.006

[12] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna M. Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, Muhammad Shahzad, Wen Yang, Richard Bamler, and Xiaoxiang Zhu. 2023. A survey of uncertainty in deep neural networks. *Artif. Intell. Rev.* 56, S1 (2023), 1513–1589. https://doi.org/10.1007/S10462-023-10562-9

[13] Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org/

[14] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On Calibration of Modern Neural Networks. In *ICML (Proceedings of Machine Learning Research, Vol. 70)*. PMLR, 1321–1330. http://proceedings.mlr.press/v70/guo17a.html

[15] David Ha and Jürgen Schmidhuber. 2018. Recurrent World Models Facilitate Policy Evolution. In *NeurIPS*. 2455–2467.

[16] Matthew J. Hausknecht and Peter Stone. 2015. Deep Recurrent Q-Learning for Partially Observable MDPs. *CoRR* abs/1507.06527 (2015).

[17] Jesse Hoey and Pascal Poupart. 2005. Solving POMDPs with Continuous or Large Discrete Observation Spaces. In *IJCAI*. 1332–1338.

[18] Marcus Hörger and Hanna Kurniawati. 2021. An On-Line POMDP Solver for Continuous Observation Spaces. In *ICRA*. IEEE, 7643–7649. https://doi.org/10.1109/ICRA48506.2021.9560943

[19] Maximilian Igl, Luisa M. Zintgraf, Tuan Anh Le, Frank Wood, and Shimon Whiteson. 2018. Deep Variational Reinforcement Learning for POMDPs. In *ICML (Proceedings of Machine Learning Research, Vol. 80)*. PMLR, 2122–2131. http://proceedings.mlr.press/v80/igl18a.html

[20] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. 1998. Planning and Acting in Partially Observable Stochastic Domains. *Artif. Intell.* 101, 1-2 (1998), 99–134.

[21] Sammie Katt, Frans A. Oliehoek, and Christopher Amato. 2019. Bayesian Reinforcement Learning in Factored POMDPs. In *AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems, 7–15. http://dl.acm.org/citation.cfm?id=3331668

[22] Shashank Kumbhare. 2018. Traffic Light Classifier. https://github.com/ShashankKumbhare/traffic-light-classifier. Accessed: 2025-07-29.

[23] Hanna Kurniawati. 2022. Partially Observable Markov Decision Processes and Robotics. *Annual Review of Control, Robotics, and Autonomous Systems* 5, 1 (2022), 253–277.

[24] Hanna Kurniawati, David Hsu, and Wee Sun Lee. 2008. SARSOP: Efficient Point-Based POMDP Planning by Approximating Optimally Reachable Belief Spaces. In *Robotics: Science and Systems*. The MIT Press. https://doi.org/10.15607/RSS.2008.IV.009

[25] Michael Lanier, Ying Xu, Nathan Jacobs, Chongjie Zhang, and Yevgeniy Vorobeychik. 2024. Learning Interpretable Policies in Hindsight-Observable POMDPs Through Partially Supervised Reinforcement Learning. In *ICMLA*. IEEE, 312–319. https://doi.org/10.1109/ICMLA61862.2024.00048

[26] Dachuan Li, Bowen Liu, Zijian Huang, Qi Hao, Dezong Zhao, and Bin Tian. 2024. Safe Motion Planning for Autonomous Vehicles by Quantifying Uncertainties of Deep Learning-Enabled Environment Perception. *IEEE Trans. Intell. Veh.* 9, 1 (2024), 2318–2332. https://doi.org/10.1109/TIV.2023.3297735

[27] Michael H Lim, Tyler J Becker, Mykel J Kochenderfer, Claire J Tomlin, and Zachary N Sunberg. 2023. Optimality guarantees for particle belief approximation of POMDPs. *Journal of Artificial Intelligence Research* 77 (2023), 1591–1636.

[28] Jiaxin Liu, Hong Wang, Liang Peng, Zhong Cao, Diange Yang, and Jun Li. 2022. PNNUAD: Perception Neural Networks Uncertainty Aware Decision-Making for Autonomous Vehicle. *IEEE Trans. Intell. Transp. Syst.* 23, 12 (2022), 24355–24368. https://doi.org/10.1109/TITS.2022.3197602

[29] Xiao Ma, Péter Karkus, David Hsu, Wee Sun Lee, and Nan Ye. 2020. Discriminative Particle Filter Reinforcement Learning for Complex Partial observations. In *ICLR*. OpenReview.net. https://openreview.net/forum?id=HJl8_eHYvS

[30] Robert J. Moss, Anthony Corso, Jef Caers, and Mykel J. Kochenderfer. 2024. BetaZero: Belief-State Planning for Long-Horizon POMDPs using Learned Approximations. *RLJ* 1 (2024), 158–181.

[31] Tianwei Ni, Michel Ma, Benjamin Eysenbach, and Pierre-Luc Bacon. 2023. When Do Transformers Shine in RL? Decoupling Memory from Credit Assignment. In *NeurIPS*.

[32] Maria-Elena Nilsback and Andrew Zisserman. 2008. Automated Flower Classification over a Large Number of Classes. In *ICVGIP*. IEEE Computer Society, 722–729. https://doi.org/10.1109/ICVGIP.2008.47

[33] Joelle Pineau, Geoffrey J. Gordon, and Sebastian Thrun. 2003. Point-based value iteration: An anytime algorithm for POMDPs. In *IJCAI*. Morgan Kaufmann, 1025–1032. http://ijcai.org/Proceedings/03/Papers/147.pdf

[34] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. 2021. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research* 22, 268 (2021), 1–8. http://jmlr.org/papers/v22/20-1364.html

[35] Stuart J. Russell and Peter Norvig. 2022. *Artificial Intelligence: a modern approach. Fourth Edition. Global Edition.* Pearson Education Limited.

[36] Miriam Schäfers, Merlijn Krale, Thiago D. Simão, Nils Jansen, and Maximilian Weininger. 2026. Perception-Based Beliefs for POMDPs with Visual Observations. *CoRR* abs/2602.05679 (2026).

[37] Miriam Schäfers, Merlijn Krale, Thiago D. Simão, Nils Jansen, and Maximilian Weininger. 2026. *Perception-Based Beliefs for POMDPs with Visual Observations - Code*. https://doi.org/10.5281/zenodo.18266744

[38] David Silver and Joel Veness. 2010. Monte-Carlo Planning in Large POMDPs. In *NIPS*. Curran Associates, Inc., 2164–2172. https://proceedings.neurips.cc/paper/2010/hash/edfbe1afcf9246bb0d40eb4d8027d90f-Abstract.html

[39] Trey Smith and Reid G. Simmons. 2004. Heuristic Search Value Iteration for POMDPs. In *UAI*. 520–527.

[40] Trey Smith and Reid G. Simmons. 2005. Point-Based POMDP Algorithms: Improved Analysis and Implementation. In *UAI*. 542–547.

[41] Shagun Sodhani, Franziska Meier, Joelle Pineau, and Amy Zhang. 2022. Block Contextual MDPs for Continual Learning. In *L4DC (Proceedings of Machine Learning Research, Vol. 168)*. PMLR, 608–623. https://proceedings.mlr.press/v168/sodhani22a.html

[42] Jayakumar Subramanian, Amit Sinha, Raihan Seraj, and Aditya Mahajan. 2022. Approximate Information State for Approximate Planning and Reinforcement Learning in Partially Observed Systems. *J. Mach. Learn. Res.* 23 (2022), 12:1–12:83.

[43] Zachary N. Sunberg and Mykel J. Kochenderfer. 2018. Online Algorithms for POMDPs with Continuous State, Action, and Observation Spaces. In *ICAPS*. 259–263.

[44] Yunbo Wang, Bo Liu, Jiajun Wu, Yuke Zhu, Simon S. Du, Li Fei-Fei, and Joshua B. Tenenbaum. 2020. DualSMC: Tunneling Differentiable Filtering and Planning under Continuous POMDPs. In *IJCAI*. ijcai.org, 4190–4198.

[45] Chenyang Wu, Guoyu Yang, Zongzhang Zhang, Yang Yu, Dong Li, Wulong Liu, and Jianye Hao. 2021. Adaptive Online Packing-guided Search for POMDPs. In *NeurIPS*. 28419–28430.

[46] Nan Ye, Adhiraj Somani, David Hsu, and Wee Sun Lee. 2017. DESPOT: Online POMDP Planning with Regularization. *J. Artif. Intell. Res.* 58 (2017), 231–266. https://doi.org/10.1613/JAIR.5328

[47] Arnold Zellner. 1971. *An introduction to Bayesian inference in econometrics*. Wiley.

[48] Xuezhou Zhang, Yuda Song, Masatoshi Uehara, Mengdi Wang, Alekh Agarwal, and Wen Sun. 2022. Efficient Reinforcement Learning in Block MDPs: A Model-free Representation Learning approach. In *ICML (Proceedings of Machine Learning Research, Vol. 162)*. PMLR, 26517–26547. https://proceedings.mlr.press/v162/zhang22aa.html