Recursive Bayesian state estimation from raw pixels using variational inference

Simon Steffens - 402-0738-10L Bayesian Statistical Methods and Data Analysis SS2024

***Abstract*—This work presents a Bayesian graphical model to estimate an agent's position in a two-dimensional virtual arena from raw visual input. A variational convolutional autoencoder approximates the observation distribution $p(M|X)$, which is then used by a Kalman filter to recursively infer the agent's location. The variational method effectively reconstructs images and accurately approximates $p(X)$. The Kalman filter demonstrated accurate trajectory estimation and effective handling of discontinuities, indicated by Bayesian surprise metrics. Future work may explore non-linear recursive Bayesian models for improved state estimation in dynamic environments.**

***Index Terms*—Bayesian filtering, recursive Bayesian estimation, Kalman filters, variational inference, continual learning**

I. Introduction

In dynamic real-world environments, organisms and artificial agents build models of their environment to behave effectively. Many of such models can be seen as inferring unobserved relevant quantities from observed variables. For example, the odor of a preferred food m (observable) may be transformed into a hidden state variable s (reward) using a model that was formed using previous exposure to the odor and the outcome following its consumption. In robotics, simultaneous localization and mapping (SLAM) can be formulated as an inference problem where various sensors (observables m) are integrated to estimate the robot's position s (hidden) within a mapped space.

$$p(s|m) = \frac{p(m|s) \cdot p(s)}{p(m)} \tag{1}$$

The broad spectrum of Bayesian methods presents a great fit for this highly general formulation of how agents act within an environment. In the Bayesian framework, the prior distribution $p(s)$ can be elegantly represented as the integrated observation history of the time series data encountered by biological or artificial agents. In this context, the prior distribution can be interpreted as the accumulated knowledge of an agent through time. Given new evidence under the current model represented by the likelihood distribution $p(m|s)$, the posterior is an update to the agent's knowledge weighting the certainty of previous beliefs and new evidence. The explicit formulation of uncertainty within the probabilistic model is advantageous for modulating adaptation in a dynamic environment. Although the real environment is highly non-stationary, conventional problem formulation in machine learning assumes training and inference distributionss to be stationary and sampled independently (IID). With their inherent property of capturing the uncertainty in their prediction, Bayesian approaches present a powerful alternative framework for continual adaption. Utilizing the uncertainty estimation of generative models as a signal to increase the learning rate has been explored before [1] [2]. Also, in theoretical neuroscience, probabilistic quantities such as novelty and surprise have been suggested to explain a broad range of empirical observations [3]

A common critique of Bayesian methods is that Bayes rule involves the computation of integrals that are intractable for high-dimensional problems. While the integral over the evidence distribution $p(s)$ can be treated as a constant when estimating the posterior, it becomes unavoidable for model comparison. Sampling methods such as rejection sampling, Metropolis-Hastings, slice sampling, or Gibbs sampling offer a solution for estimating the evidence integral, but although capable in up to hundreds of dimensions, real data processed by agents in the environment can be orders of magnitude larger, vision being the most prominent example. In such cases, variational methods for approximating a complex posterior have proven highly effective.

In this work, we implement a Bayesian graphical model to recursively estimate the x y position of an agent in a two-dimensional arena from raw visual input. We utilize a generative model of visual observations to draw out a path for continual learning in non-stationary environments. More specifically, we present two interacting models. A variational convolutional autoencoder estimating the observation distribution, and a Kalman filter that recursively infers the agent's position from the variational distribution. We first show that variational inference is well suited for approximating the high dimensional visual input distribution with a 24-dimensional multivariate Gaussian. The Kalman filter utilizes this distribution as the measurement or observable state to correct the future position predicted by a kinematics model. Finally, we show how the Kullback-Leibler (KL) divergence between the prior and posterior position estimate (also termed Bayesian surprise [3]) behaves when the agent is suddenly teleported.

II. Theoretical Background & Implementation

The following section describes the mathematical background of variational inference and recursive Bayesian estimation specifically introducing variational autoencoders (VAE) and Kalman filters.

Given a dataset $X = \{(I_1, s_1), (I_2, p_2), \ldots, (I_N, s_N)\}$, where each data point consists of an image I_i and a position (or state) s_i, the goal is to infer the posterior distribution

$p(M|X)$. Direct computation of $p(M|X)$ is intractable, so variational inference approximates this posterior with a simpler distribution $q(M|\lambda)$, parameterized by λ. The latent variable is denoted as M instead of Z since it will be interpreted as a measurement by the Kalman filter.

$$p(M|X) \approx q(M|\lambda). \tag{2}$$

The dissimilarity between $p(M|X)$ and $q(M|\lambda)$ is measured using the KL divergence:

$$D_{\mathrm{KL}}(q(M|\lambda)\|p(M|X)) = \int q(M|\lambda) \log \frac{q(M|\lambda)}{p(M|X)}\, dM. \tag{3}$$

Since the marginal likelihood $p(X)$ is intractable due to the integration over all possible M:

$$p(X) = \int p(X, M)\, dM, \tag{4}$$

we use variational inference to maximize the Evidence Lower Bound (ELBO), which is derived from the KL divergence. The ELBO is defined as:

$$\log p(X) \geq \mathbb{E}_{q(M|\lambda)}[\log p(X|M)] - D_{\mathrm{KL}}(q(M|\lambda)\|p(M)). \tag{5}$$

Maximizing the ELBO involves balancing the reconstruction accuracy $\mathbb{E}_{q(M|\lambda)}[\log p(X|M)]$ and the KL divergence $D_{\mathrm{KL}}(q(M|\lambda)\|p(M))$. In this project, we adopt the ELBO as an objective function to approximate the visual observation distribution over $I \in \mathbb{R}^{224\times224}$ with a 24-dimensional multivariate Gaussian $\lambda = \{\mu_1, ..., \mu_{24}, \sigma_1^2, ..., \sigma_{24}^2\}$, or embedding vector $m \in \mathbb{R}^{24}$. In the context of Variational Autoencoders (VAEs) [4], this framework is applied to generative models. A VAE consists of 1, the encoder $q_\phi(M|X)$ approximating the posterior distribution over the latent variables m, and the decoder $p_\theta(X|M)$ reconstructing the data (I, s) from the latent representation m. The parameters of the encoder $h_\phi(I)$ and decoder $f_\theta(m)$ were learned via convolutional neural networks (CNNs) and gradient descent.

To allow subsequent use in the Kalman filter, we want to learn a latent distribution $p(M|X)$ that not only approximates the image space $X_I = \{I_1, ..., I_N\}$ but also the agents state or position $X_s = \{s_1, ..., s_N\}$. To satisfy the linearity assumption for Kalman filters, this joint embedding should allow the position to be linearly reconstructed from $p(M|X)$. In the context of the neural network, we define a linear layer without bias that transforms the true position into embedding space using a matrix $sH = \hat{m}$. Analogous to the image decoder $f_\theta(m)$, a second matrix W reconstruct the point $\hat{s} = \hat{m}^T W$. The complete network architecture is shown in Figure 1.

Derived from the ELBO equation 5, the final loss function $\mathcal{L}(\theta, \phi, H, W)$ of our joint variational autoencoder then consists of 4 terms:

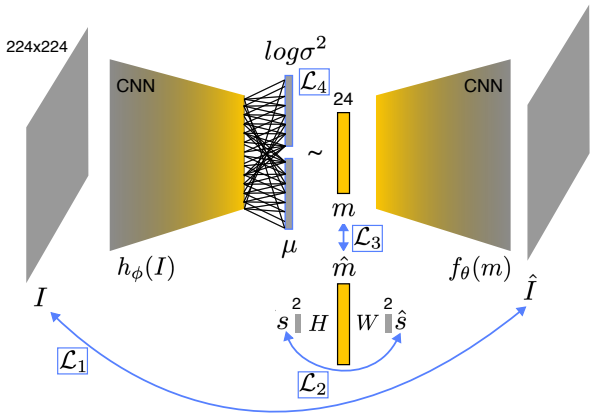

Fig. 1. Illustration of variational autoencoder (vAE) architecture. The embedding m is depicted in yellow, and the loss terms $\mathcal{L}$ are in blue. Note that $m \sim \mathcal{N}_{1...24}(\mu_i, \sigma_i^2)$

$$\mathcal{L}_1(I, \hat{I}; \phi, \theta) = \mathbb{E}_{q_\phi(M|X)}\left[\log p_\theta(I|M)\right] \tag{6}$$

$$\mathcal{L}_2(s, \hat{s}; H, W) = \mathbb{E}_{q_H(M|X)}\left[\log p_M(s|M)\right] \tag{7}$$

$$\mathcal{L}_3(m, \hat{m}; \phi, \theta, H, W) = \mathbb{E}_{q_\phi(M|X)}\left[\|m - \hat{m}\|^2\right] \tag{8}$$

$$\mathcal{L}_4 = -D_{\mathrm{KL}}\left(q_\phi(M|X)\|p(M)\right) \tag{9}$$

$$\mathcal{L} = \alpha\mathcal{L}_1 + \beta\mathcal{L}_2 + \gamma\mathcal{L}_3 + \mathcal{L}_4 \tag{10}$$

$$1 - (\alpha + \beta + \gamma) = 0 \tag{11}$$

The goal of the variational autoencoder described above is to approximate the input distribution over visual observations and matching positions $p(M|X)$. In the context of the Kalman filter, this distribution is used to evaluate measurements. Next, the mathematical background of Kalman filters is briefly introduced.

Kalman filters are a specific case of Bayesian graphical models or more specifically Bayesian recursive estimation models. The state estimation problem is approached by constructing a probabilistic model that captures the dependencies between variables. The hidden state s_{t-1} evolves over time according to a state transition model, while observations m_{t-1} provide noisy information about the current state (Figure 2).

The most general way to formulate recursive Bayesian estimation is defined below.

$$p(s_t|m_t) = \frac{p(m_t|s_t) \cdot p(s_t|m_{t-1})}{N} \tag{12}$$

Using the state estimate without considering new measurements $p(s_t|m_{t-1})$ (also called the predict step) and the likelihood of the new measurement $p(m_t|s_t)$, the state estimate is updated to $p(s_t|m_t)$. The distribution $p(s_t|m_{t-1})$ (predict

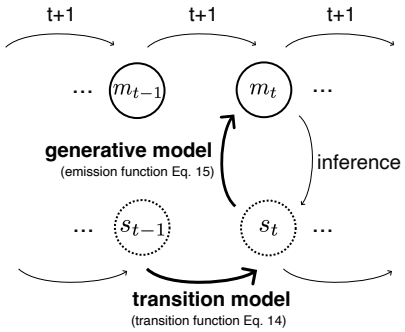

Fig. 2. Bayesian network model over observed measurements m_t, and hidden state variables s_t. Bayesian inference is performed to estimate s_t using a transition model and generative model.

step) is obtained by marginalizing over the previous state s_{t-1}. This is also called the Chapman-Kolmogorov equation:

$$p(s_t|m_{t-1}) = \int p(s_t|s_{t-1}) \cdot p(s_{t-1}|m_{t-1}) ds_{t-1} \tag{13}$$

In the specific case of the Kalman filter, equations 11, and 12 become computable by assuming a linear Gaussian state space model. This constrains all probability distributions to be Gaussian and the transition and emission functions (generative model) to be linear. For position estimation using visual observations, we define the transition model in the following way:

$$s_t = I s_{t-1} + \tau\nu + q \quad q \sim \mathcal{N}(0, Q) \tag{14}$$

The new state s_t is the sum of the previous state estimate s_{t-1}, a velocity input signal ν, and Q which describes the covariance matrix or uncertainty in the transition model. Once the prediction is made using equation 13, the state estimate is updated using the emission model matrix H:

$$m_t = H s_t + r \quad r \sim \mathcal{N}(0, R) \tag{15}$$

The implementation of both the VAE and Kalman filter are written in Python using, among others, PyTorch and is freely available at github.com/loaloaf/vaeKalmanPositionEstimation. The experiments were conducted in a custom-built Unity simulation environment which is part of a physical setup where rats participate in a virtual reality experiment.

The experiments were conducted in a custom-built Unity simulation environment which is part of a physical setup where rats participate in a virtual reality experiment. For training and cross-validating the VAE, we acquired a static dataset. The Kalman filter is used in a closed-loop where velocity inputs and rendered outputs are communicated with Unity via a shared memory architecture.

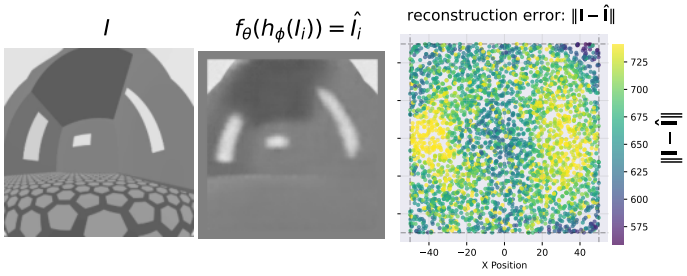

Fig. 3. Image reconstruction performance of the VAE model. The image shown is in the lower right corner. $\alpha = 0.9$, $\beta = \gamma = 0$

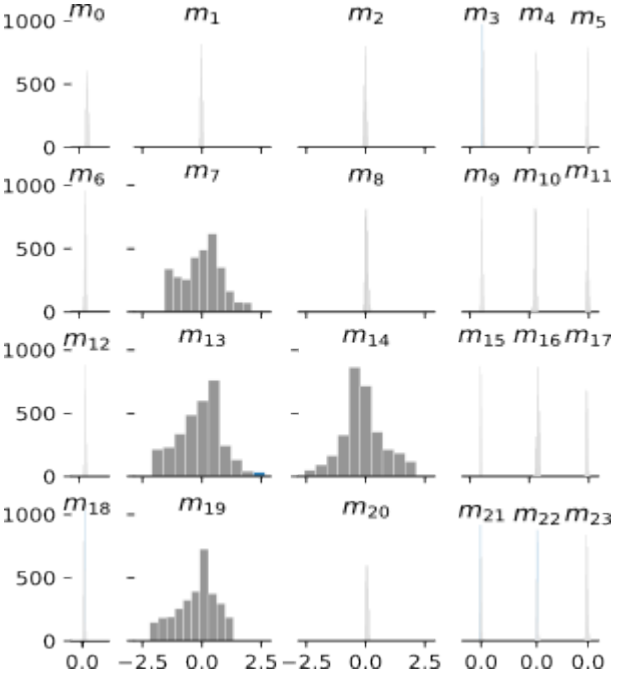

Fig. 4. Distribution $p(M = m_{1,\ldots,24}|X)$. With a low weight on the KL loss term, 20 of 24 dimensions converge towards $\mathcal{N}(0, 0)$

III. Results

We first trained the variational autoencoder to approximate the distribution over visual observations. To confirm the network's ability to reconstruct the image I, we first trained without $\mathcal{L}_2$ and $\mathcal{L}_3$ loss terms. This resulted in a fast convergence on $p(M|X_I)$ as shown in Figure 3. The approximation successfully identifies the most salient features while ignoring challenging patterns such as the floor texture. The reconstruction accuracy is measured as the Euclidean distance between input and output image I, $\hat{I}$. The reconstruction accuracy is highest at the center of the arena.

We next investigated the specific form underlying the distribution $p(M|X_I)$. We find that the VAE training objective successfully reduces the dimensionality of the original input distribution while retaining image quality, given the KL loss component is kept small (0.1 in Figure 4 and 5). With this low weight on regularizing the distribution, 20 of the 24 dimensions are ignored (Figure 4). The 4

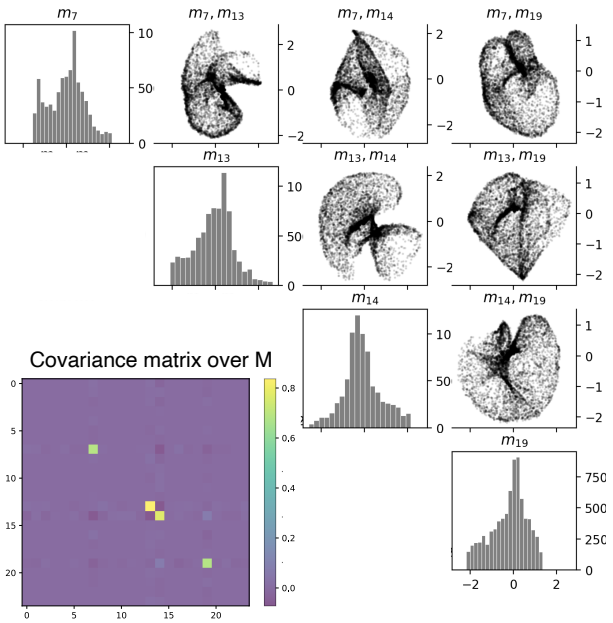

Fig. 5. Joint distribution over components $p(m_{7,13,14,19}|X_I)$. The histograms along the diagonal show the marginal distribution. Joint distributions are plotted with the x, y coordinate from the corresponding row and column histograms, respectively.

non-zero-variance marginal distributions $p(m_{7,13,14,19}|X_I)$ generally follow a Gaussian shape (Figure 4). The joint distribution as shown in Figure 5 over the marginals appear disentangled, and not linearly correlated. This is also reflected in the covariance matrix of $p(m_{7,13,14,19}|X_I)$. With increasing the KL component beyond 0.1, we find that the reconstruction performance drops significantly (not shown).

To enable subsequent use in with the Kalman filter, we incorporated the loss terms $\mathcal{L}_2$ and $\mathcal{L}_3$ into the training objective. These loss terms add effectively a second encoder-decoder architecture for the current state/location s (see Figure 1). To ensure that both the image encoder $h_\theta(I_i)$ and linear state transformation $s_i H$ encode the same embedding m_i, we had to train with a high γ value. Only when we reduced the KL loss component to an absolute minimum, we could obtain $p(M|X)$ such that both s and I were mapped to a similar m. Figure 6 shows how both $h_\theta(I)$ and sH yield an accurate and similar embedding m from which the the state can be linearly reconstructed using the matrix W. The mean euclidean distance between $s, \hat{s}$ was between 2 and 4. Given that both $h_\theta(I)$ and sH reconstruct s using the same linear map W, we know that $h_\theta(I_i)$ and $s_i H$ encode a similar m_i.

With the VAE model over the observation space $p(M|X)$ we implemented the Kalman filter. To simplify the problem, the agents actions space was limited to (x_{vel}, y_{vel}), excluding angular velocity input. We uniformly sample a point in the arena that the agent approached in discrete time steps. Each

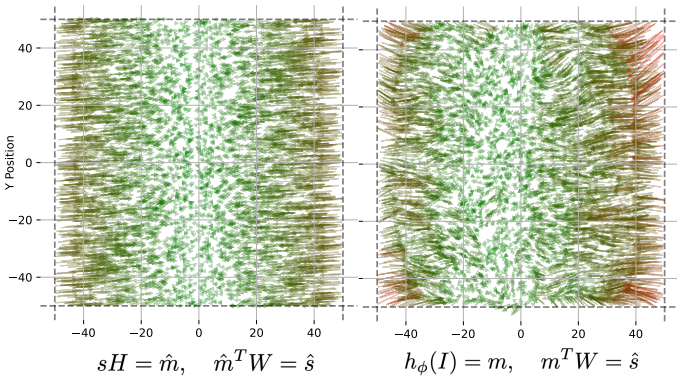

Fig. 6. Linear mapping of the agents state (2D position) from $p(M|X)$. Arrows point from the ground truth s to the reconstructed position $\hat{s}$. Arrows with a Euclidean distance great than 10 are drawn increasingly red.

step was subjected to noise Q defined in equation 14. At each step, the motion model predicted $p(s|m_{t-1})$ integrating the current motion input ν as defined in equation 14. We then encoded the current visual observation $h_\theta(I_t) = m_t$ using the VAE encoder model. To subsequently evaluate the update-step (equation 12), we calculate the Kalman gain which integrates the covariances R and Q with the uncertainty of the previous state estimate. The update then involves scaling the residual between the measured m_t and predicted $\hat{m} = Hs_t$ by the Kalman gain to update the state estimate.

In Figure 7 we show two example trajectories of the agent in the environment with the motion and emission model covariances $Q = R = I \cdot 3$. The left example shows how the Kalman filter accurately approximates the ground truth trajectory with the motion model biasing the estimates towards the goal point (matching the velocity input ν), and the measurements correcting these estimates. This trajectory was obtained in the center of the arena where the position mapping W showed better results than at the edges (compare Figure 6). This also reflected in the second example shown on the right of Figure 7. Here, the initial measurement estimate erroneously updated the position towards the center. This matches the directional bias of the position estimates shown in Figure 6.

Finally, we investigated how the probabilistic model reacts to discontinuous ground truth position updates. At some point in the trajectory, we teleported the agent to a random location. For an unexpected event like this, we would expect the measurement model to significantly differ from what the motion model predicts. In other words, the posterior state estimate $p(s|m)$ would highly differ from the prior $p(s|m_{t-1})$ and instead be mostly influenced by the likelihood $p(m|s)$ (see equation 12). This can be formalized by the KL divergence between the prior and posterior, a quantity that has been termed *Bayesian surprise*. In Figure 8 we show the results of this experiment. Again, estimated trajectory is slightly off from the ground truth in the beginning. At the point of teleportation ($t = 7$), the Bayesian surprise peaks. This is

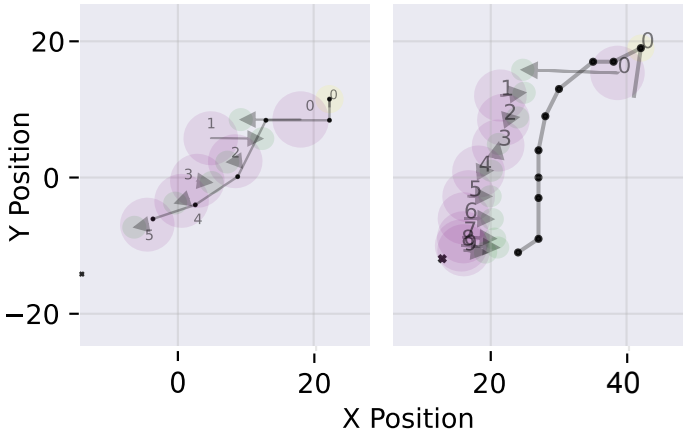

Fig. 7. Two example trajectories of the agent in the environment. The ground truth path is shown as the black-dotted line. Predict-estimates $p(s|m_{t-1})$ are shown in purple with the circle reflecting one standard deviation from the mean (center). The green circle shows the updated estimate $p(s|m)$ with an arrow connecting the two steps at one specific t.

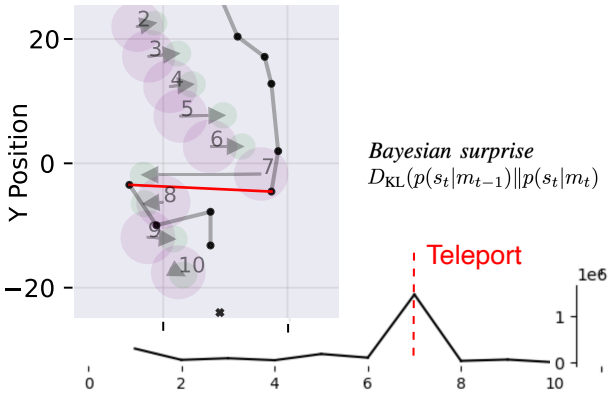

Fig. 8. Example trajectory with teleportation at $t = 7$ depicted in red. The plot at the bottom shows the Bayesian surprise over time as defined by the KL divergence between the prior and posterior. The color code matches with Figure 7.

directly proportional to the length of the arrow between the purple and green circles at $t = 7$. This shows how the visual observation successfully corrects the state estimate while being explicit about the *surprise* of the discontinuity.

IV. Discussion

In this work, we developed two interacting probabilistic models for estimating the position of an agent from raw visual input. We utilized variational inference to approximate the high-dimensional distribution over visual observations $p(M|X)$ for subsequent use as measurements in a Kalman filter. Utilizing these measurements, the Kalman filter was able to recursively estimate the true agent location over time.

To satisfy the Kalman filter's assumptions of linear transition and emission functions, we introduced a joint VAE architecture and learning objective where the distribution $p(M|X)$ not only reconstructs the input but also allows for linear mapping to the agent's state s. Combining these learning objectives proved challenging. While we successfully approximated $p(X_I)$ with a decorrelated multivariate Gaussian (Figure 5), the introduction of $\mathcal{L}_2$ and $\mathcal{L}_3$ to enforce the state to be linearly mapped from m did not converge to a valid multivariate Gaussian that could also reconstruct s (Figure 6). Only when we minimized the KL loss component $\mathcal{L}_4$ could we construct a functional embedding space $p(M|X)$ for use with the Kalman filter. In future work, we may consider non-linear recursive Bayesian estimation models, such as particle filters.

The initial goal of this project was to utilize uncertainty estimates from the probabilistic model to guide model adaptation. Initial steps were made in this direction by demonstrating that the KL divergence between prior and posterior state estimates peaks when the agent is teleported. As a second step, we aimed to use the distribution $p(M|X)$ to estimate uncertainty in a non-stationary environment (e.g., changes in wall textures). However, due to the difficulties in combining the learning objectives as described above, we were unable to reliably estimate the likelihood of incoming measurements given $p(M|X)$.

Nonetheless, the initial steps of approximating the visual observation space $p(M|X)$ provide a solid foundation for building on with non-linear state estimation models. The exploration of probabilistic Bayesian models for continual learning is a promising direction, as evidenced by recent publications [5], [6].

References

[1] J. Gordon and J. M. Hernández-Lobato, "Combining deep generative and discriminative models for bayesian semi-supervised learning," *Pattern Recognition*, vol. 100, p. 107156, 4 2020.

[2] T. Lesort, H. Caselles-Dupre, M. Garcia-Ortiz, A. Stoian, and D. Filliat, "Generative models from the perspective of continual learning," *IEEE International Joint Conference on Neural Network*, vol. 2019-July, 7 2018.

[3] A. Modirshanechi, S. Becker, J. Brea, and W. Gerstner, "Surprise and novelty in the brain," *Current Opinion in Neurobiology*, vol. 82, p. 102758, 10 2023.

[4] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 12 2013. [Online]. Available: https://arxiv.org/abs/1312.6114v11

[5] V. Liakoni, A. Modirshanechi, W. Gerstner, J. Brea, V. Liakoni, A. Modirshanechi, W. Gerstner, and J. Brea, "Learning in volatile environments with the bayes factor surprise," *Neural Computation*, vol. 33, pp. 269–340, 2 2021. [Online]. Available: https://dx.doi.org/10.1162/neco_a_01352

[6] M. L. Barry and W. Gerstner, "Fast adaptation to rule switching using neuronal surprise," *PLOS Computational Biology*, vol. 20, p. e1011839, 2 2024. [Online]. Available: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1011839