Navigation and planning in latent maps

Baris Kayalibay [1] **Atanas Mirchev** [1] **Maximilian Soelch** [1] **Patrick van der Smagt** [1] **Justin Bayer** [1]

Abstract

We investigate the problem of agents planning paths to efficiently traverse environments only observed through the agent's limited and local sensors such as laser range finders. We use a probabilistic sequential latent-variable model with an explicit global map. The entire model is trained only from local sensor readings, leveraging recent advances in approximate Bayesian inference. The generative nature of the model allows us to plan traversals of the real environment solely based on simulations within the learnt model. Further, we can exploit the Euclidean geometry imposed on our map to make use of well-known path planning algorithms. We showcase this ability by reliably and efficiently finding optimal paths in random mazes.

1. Introduction

Sequential decision making is a framework to represent the interaction of an agent with its environment: an observation of the world is presented to the agent, upon which the agent picks an action, which in turn alters the world's state. The agent then observes the new state of the world, and the process repeats.

In the case of mobile agents, the state of the world includes the location of the agent. Knowing that location, or at least having an accurate estimate of it, is crucial for devising and successfully executing plans. Several works were proposed recently that overcome the intractabilities of Bayes filters via approximate inference (Gu et al., 2015; Krishnan et al., 2015; Karl et al., 2016; Fraccaro et al., 2016; Maddison et al., 2017). Higher flexibility comes at the cost of only obtaining an approximate solution; yet these approaches often achieve outstanding results. Typically, the optimisation of finding an approximate posterior is amortised into a fixed computation implemented as a neural network; this approach is hence often referred to as amortised inference (Rezende et al., 2014; Kingma & Welling, 2014). These techniques have recently been extended with structured map memory (Mirchev et al., 2018; Fraccaro et al., 2018) to tackle the SLAM problem.

In this work, we make use of the generative nature of these models. We show that we can reliably plan paths in the latent space using the learnt map. In the process, we show that it is straightforward to implement collision avoidance in the probabilistic model. The obtained optimal latent trajectories are successfully executed in the real environment, on a par with planning algorithms that have access to the true map.

2. Methods

We revise the sequential graphical model with a global map, including learning procedure and efficient inference.[1] Then, we discuss how we use the learnt generative model to do planning in latent space.

2.1. The graphical model

We extend the vanilla latent-variable graphical model for sequential data, $p(\bar{\mathbf{x}}, \bar{\mathbf{z}} \mid \bar{\mathbf{u}}) = p(\bar{\mathbf{x}} \mid \bar{\mathbf{z}})\, p(\bar{\mathbf{z}} \mid \bar{\mathbf{u}})$, where

$\bar{\mathbf{x}} = \mathbf{x}_{1:T} \in \mathbb{R}^{T \times D_x}$ is a sequence of observations,
$\bar{\mathbf{z}} = \mathbf{z}_{1:T} \in \mathbb{R}^{T \times D_z}$ is a sequence of poses, and
$\bar{\mathbf{u}} = \mathbf{u}_{1:T-1} \in \mathbb{R}^{T-1 \times D_u}$ is a sequence of control inputs.

Specifically, we incorporate a global map $\mathcal{M} \sim p(\mathcal{M})$. To use it in the local transitions, we introduce a latent middle layer of local charts $\mathbf{m}_t \sim p(\mathbf{m}_t \mid \mathbf{z}_t, \mathcal{M})$. Intuitively, the chart $\mathbf{m}_t$ represents the currently relevant region of the map. It shapes the transition of poses over time, $p(\mathbf{z}_{t+1} \mid \mathbf{z}_t, \mathbf{m}_t, \mathbf{u}_t)$. The observation emission model operates solely on these charts, $\mathbf{x}_t \sim p(\mathbf{x}_t \mid \mathbf{m}_t)$. In total, this yields the graphical model (cf. fig. 1.1)

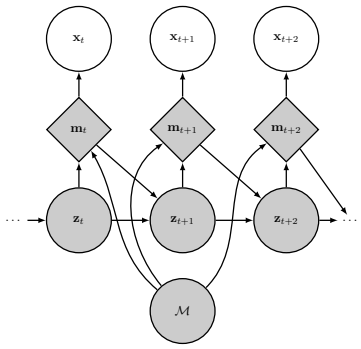
$$p(\bar{\mathbf{x}}, \bar{\mathbf{z}}, \bar{\mathbf{m}}, \mathcal{M} \mid \bar{\mathbf{u}}) = p(\mathcal{M}) \cdot \rho(\mathbf{z}_1) \cdot \cdot \prod_{t=1}^{T} p(\mathbf{x}_t \mid \mathbf{m}_t)\, p(\mathbf{m}_t \mid \mathbf{z}_t, \mathcal{M})\, p(\mathbf{z}_{t+1} \mid \mathbf{z}_t, \mathbf{m}_t, \mathbf{u}_t)\,.$$

This model is commonly used in robotic navigation tasks (e.g. (Montemerlo et al., 2002)). We assume that all compo-

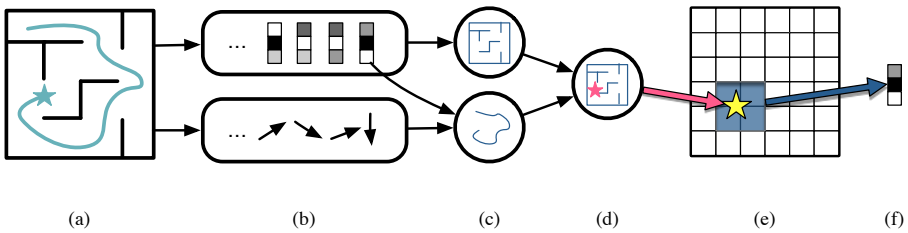
[1]Data:Lab, Volkswagen Group. Correspondence to: Atanas Mirchev <atanas.mirchev@volkswagen.de>.

Accepted at the FAIM workshop "Prediction and Generative Modeling in Reinforcement Learning", Stockholm, Sweden, 2018.

[1]Details can be found in the recent (Mirchev et al., 2018).



(1) Sequential graphical model with global map $\mathcal{M}$ and local charts $\mathbf{m}_t$.



(2) *Illustration of pose inference.* (a) An agent (teal star) traverses a maze, (b) collecting sensor readings (top) and control signals (bottom). (c) A belief of the map is formed from observations (top), a belief of the trajectory is formed from observations and controls (bottom). (d) The two beliefs are fused. (e) The map is indexed with a pose-based attention mechanism. (f) The attended region of the map is used to reconstruct the observation.

Figure 1.

nents are parameterised, e.g. by neural networks mapping to parameters. We collect all learnable parameters as a set $\boldsymbol{\theta}$.

Following (Murphy, 1999), our map $\mathcal{M} \in \mathbb{R}^{w \times h \times D_m}$ is a finite grid of real-valued vectors of dimension D_m, with a factorised Gaussian prior, $\mathcal{M}_{ij} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$. A *local chart* $\mathbf{m}_t = f_{\mathbf{m}}(\mathcal{M}, \mathbf{z}_t)$ is a *convex* combination of map cells,[2]

$$\mathbf{m}_t = \sum_{i,j} \alpha(\mathbf{z}_t)_{ij} \mathcal{M}_{ij}, \qquad \text{s.t.} \sum_{i,j} \alpha(\mathbf{z}_t)_{ij} = 1,$$

as found in attention models, e.g. (Bahdanau et al., 2014). In our implementation, $\mathbf{m}_t$ is a bilinear interpolation of the four closest map cells.

The *emission likelihood model* as well as *transition model* are conditional Gaussian distributions. The respective means are given by neural networks $\boldsymbol{\mu}_E$ and $\boldsymbol{\mu}_T$, i.e.

$$p\left(\mathbf{x}_t \mid \mathbf{m}_t\right) = \mathcal{N}\left(\boldsymbol{\mu}_E(\mathbf{m}_t), \operatorname{diag}(\boldsymbol{\sigma}_E^2)\right),$$
$$p\left(\mathbf{z}_{t+1} \mid \mathbf{z}_t, \mathbf{u}_t, \mathbf{m}_t\right) = \mathcal{N}\left(\boldsymbol{\mu}_T(\mathbf{z}_t, \mathbf{u}_t, \mathbf{m}_t), \sigma_T^2 \mathbf{1}\right).$$

Variances $\boldsymbol{\sigma}_E^2$ and σ_T^2 are learnt, but independent of inputs.

2.2. Amortised inference and SGVB training

Exact inference in such models is typically intractable. We obtain amortised variational approximations $q\left(\mathcal{M}\right)$ and $q\left(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T}, \mathbf{u}_{1:T-1}\right)$, with all learnable paramaters collected in $\boldsymbol{\phi}$, through stochastic gradient on the negative evidence lower bound,

$$\begin{aligned}
&\mathcal{L}_{\text{elbo}}\left(\mathbf{x}_{1:T}, \mathbf{u}_{1:T-1}; \boldsymbol{\phi}, \boldsymbol{\theta}\right) \\
&= \mathbb{E}_{q(\mathcal{M}) q(\bar{\mathbf{z}} \mid \bar{\mathbf{x}}, \bar{\mathbf{u}}, \mathcal{M})}\left[-\log p\left(\bar{\mathbf{x}} \mid \bar{\mathbf{z}}, \mathcal{M}, \bar{\mathbf{u}}\right)\right] \\
&\quad + \mathbb{E}_{q(M)}\left[\mathrm{KL}\left[q\left(\bar{\mathbf{z}} \mid \bar{\mathbf{x}}, \bar{\mathbf{u}}, \mathcal{M}\right) \mid\mid p\left(\bar{\mathbf{z}} \mid \bar{\mathbf{u}}, \mathcal{M}\right)\right]\right] \\
&\quad + \mathrm{KL}\left[q\left(\mathcal{M}\right) \mid\mid p\left(\mathcal{M}\right)\right]. \qquad (1)
\end{aligned}$$

This amounts to Bayes by backprop (Blundell et al., 2015) on the map, and SGVB (Kingma & Welling, 2014) on poses.

[2]The resulting distribution $p\left(\mathbf{m}_t \mid \mathbf{z}_t, \mathcal{M}, \boldsymbol{\theta}_M\right)$ is a point mass.

The variational approximation of the posterior map $q\left(\mathcal{M}\right)$ was chosen to follow a mean-field approach with a factorised Gaussian $q\left(\mathcal{M}\right) = \prod_i \prod_j \mathcal{N}(\boldsymbol{\mu}_{\mathcal{M}_{ij}}, \boldsymbol{\sigma}^2_{\mathcal{M}_{ij}})$, with variational parameters $\boldsymbol{\mu}_{\mathcal{M}_{ij}}, \boldsymbol{\sigma}^2_{\mathcal{M}_{ij}} \in \boldsymbol{\phi}$.

Pose inference is done via particle filtering. That is, $q\left(\mathbf{z}_t \mid \mathbf{x}_{1:t}, \mathcal{M}\right)$ is represented by a set of particles $\left\{\mathbf{z}_t^{(k)}\right\}$, which are resampled according to

$$\omega_k = \frac{p\left(\mathbf{x}_t \mid \mathbf{z}_t^{(k)}, \mathcal{M}\right) p\left(\mathbf{z}_t^{(k)}\right)}{\hat{q}\left(\mathbf{z}_t^{(k)}\right)},$$

where $\hat{q}\left(\mathbf{z}_t^{(k)}\right)$ is a proposal distribution, typically the transition, in which case the weight reduces to the emission likelihood. This means that the map implicitly, through the weight, selects fitting particles.

The interested reader is referred to (Karl et al., 2016) for an elaborate discussion on the design of the amortised inference model $q\left(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T}, \mathbf{u}_{1:T-1}\right)$, and to (Mirchev et al., 2018) in the case of a map-enhanced model.

2.3. Generative path planning

Latent spaces in unsupervised generative models do not necessarily exhibit Euclidean geometrical structures, (Chen et al., 2017; Arvanitidis et al., 2017). Specifically, close latents need not correspond to related data. However, the map $\mathcal{M}$ is learnt such that it roughly conserves the Euclidean geometry of the true environment. This allows us to use classical path planning algorithms. We emphasize though that planning is performed in latent space only, i.e., the agent has no access to a true map of the environment at no point in the process. In particular, collision avoidance needs to be done solely based on the learnt map.

2.3.1. LATENT HYBRID-A^*

The goal of the algorithm is to find a path from a *continuous* starting pose to a *continuous* target pose. We rely on *hybrid-A^** (Dolgov et al., 2010): we discretise the search space into a grid of N cells $\{\mathbf{c}_n\}_{n\in[N]}$, which the conventional A^*-search (Hart et al., 1968) can operate on. To obtain smooth navigation trajectories, every discrete state $\mathbf{c}_n$ is associated with a continuous state $\mathbf{z}_n$—the agent's state when that cell was explored for the first time. New cells $\mathbf{c}_{n+1}$ are explored by picking random sequences of controls $\mathbf{u}^n_{1:K}$ and predicting a following continuous state by applying the controls to the current $\mathbf{z}_n$. Upon reaching an unexplored cell, we assign

$$\mathbf{z}_{n+1} = \arg\max_{\mathbf{z}_{n+1}} p\left(\mathbf{z}_{n+1} \mid \mathbf{z}_n, \mathbf{u}^n_{1:K}, \mathcal{M}\right).$$

When the target state is found, we can backtrack to obtain a consistent sequence of controls $\mathbf{u}_{1:T}$, which can be executed by the agent to reach the target.

2.3.2. PATH PLANNING VIA GRADIENT DESCENT

Instead of using established navigation heuristics like hybrid-A^*, we can also directly optimise control trajectories via gradient descent. The navigation problem becomes an optimisation problem of the form

$$\arg\max_{\mathbf{u}_{1:T}} \log p\left(\mathbf{z}_T = \mathbf{z}_{\text{target}} \mid \mathbf{u}_{1:T}\right).$$

In practice, we jointly optimise multiple control trajectory proposals and pick the best. To encourage exploration, we add a kernel-based inter-trajectory entropy term $\mathcal{L}_{\mathbb{H}} = \frac{1}{NT}\sum_i\sum_j\sum_m\sum_n k(\mathbf{z}^i_m, \mathbf{z}^j_n)$ as a regulariser.

To obtain safe trajectories, we also add a wall-avoidance term: $\mathcal{L}_{\text{wall}} = \sum_n\sum_t\sum_i s(l^n_{ti})$ Here, $1 - s(\cdot)$ is a sigmoidal function. Its parameters are selected so the agent receives a high penalty as soon as it faces a wall closer than a certain distance d. The l^n_{ti} are predicted proximity sensor readings (e.g. lidars) of the agent at time t in trajectory n. Using a sigmoidal function can be interpreted as thresholding a cumulative distribution function. That is, we punish trajectories if the probability of falling below a certain distance to an obstacle is high.

This gradient-based approach serves as a proof of concept. While it is less efficient than, e.g., the proposed hybrid-A^*, it showcases the potential of using a fully differentiable model in combination with probabilistically motivated goal statements like collision avoidance.

3. Related Work

The problem of concurrent estimation of an agent's pose and its surrounding has seen considerable attention in the last decades. We refer the interested reader to the survey of (Cadena et al., 2016), which contains a thorough review of the relevant publications and methods.

Mapping and localisation has been adopted in the machine-learning community mostly to solve reinforcement-learning or visual-navigation problems where the model is often tightly integrated with the inference method. (Bhatti et al., 2016) integrated an engineered pipeline into an agent for learning to play the computer game Doom. (Oh et al., 2016) took a different approach by equipping an agent with an external memory which is not informed about the spatial structure of its environment. This prior knowledge was then added to the architecture of the agent's policy by (Parisotto & Salakhutdinov, 2017). To the same end (Fraccaro et al., 2018) integrated this prior into the model for model-based reinforcement learning. While their approach is similar to ours their focus was primarily on simulator performance over long time spans. Further, an external memory is used which does not directly represent a random variable as part of a graphical model. (Gupta et al., 2017) integrated their planning within a mapping framework. Interestingly, (Savinov et al., 2018) reduced visual navigation to a few supervised learning components; they explicitly side-stepped the task of metric localisation.

4. Experiments

In our experiments, we verify that models as described in section 2 can be applied to navigation. To that end, we first learn maps $\mathcal{M}$ for a set of unknown maze environments. The agent is then given a starting pose and a target it must reach based exclusively on the generative model's predictions and in the absence of further ground-truth information (cf. fig. 2.1). We consider two scenarios: a latent pose target—*pose-to-pose* navigation—or an observation target—*pose-to-observation* navigation. In the latter case, a pose corresponding to the observation is inferred based on the learned map. The success rate of the navigation task is then quantitatively evaluated.

Generative model learning For simulation, we used a precisely controlled environment, and hence implemented our own simulator using pybox2d [3]. For a detailed description, we refer the reader to (Mirchev et al., 2018). In this case, the observations are the readings of 20 laser range finders covering a full circle surrounding the agent. We start off by collecting data from the simulator for six randomised distinct mazes. The data consists of the collected sensory readings (observations) and applied controls during traversal of a maze. A generative model is then learned for each envi-

[3] https://github.com/pybox2d/pybox2d

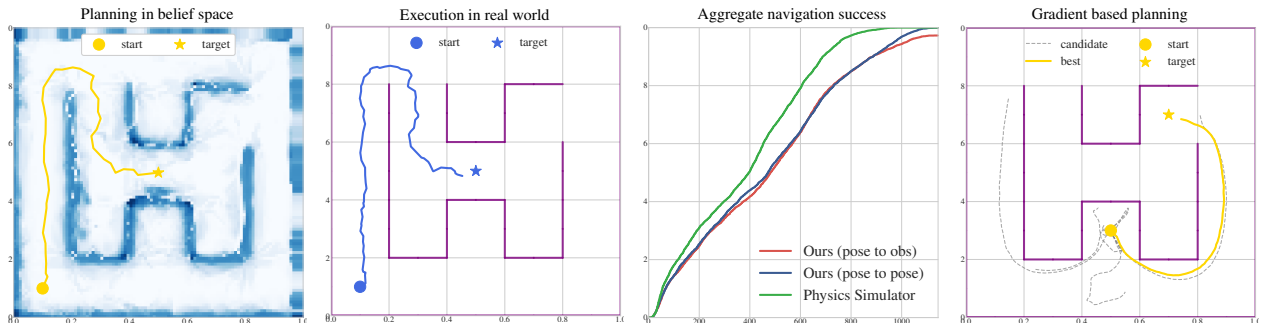


Figure 2. (1) An example navigation plan constructed by our method using the belief space of the learned environment. (2) The corresponding executed trajectory in the actual simulator. (3) Navigation success across different mazes and exhaustively sampled pairs of start and target state. Ratio of successful navigation traversals over the number of simulation steps in A^*. In green the upper bound, A^*on the true environment with access to the ground truth map and transition. (4) Gradient-based optimisation for navigation. The yellow curve is the chosen proposal, dashed grey shows candidates.

ronment. This comes down to approximating the posterior of the poses and the map $p\left(\mathbf{z}_{1:t}, \mathcal{M} \mid \mathbf{x}_{1:t}, \mathbf{u}_{1:t-1}\right)$ through the optimisation of eq. (1) with respect to the variational posteriors $q\left(\mathbf{z}_{1:T}\right)$ and $q\left(\mathcal{M}\right)$.

Pose-to-pose navigation with hybrid-A^* For each maze, we exhaustively pick pairs of starting and target pose from a grid over the map. For each pair of poses, we apply the hybrid-A^*as described in section 2.3.1 to plan a navigation trajectory based on the generative model of the environment. The obtained controls are then executed in the simulator. The navigation task is considered successful if, after executing the planned trajectory, the agent lands in a proximity of 0.05 or less from the target pose. As a baseline, we consider the navigation performance when the planning algorithm is executed directly in the simulator, with access to all ground truth data. Figure 2.3 shows the results of the evaluation. Planning based on our generative model comes very close in terms of navigation efficiency to planning in the simulator, affirming the usability of the learned environment maps for navigation tasks. Furthermore, all planned trajectories are successful in reaching the target.

Pose-to-observation navigation with hybrid-A^* In this scenario, the observation targets are picked by obtaining sensor readings from the environment simulator. The corresponding starting poses are the same as in the previous case. Since the navigation algorithm from 2.3.1 cannot operate on observation targets, we choose to first map them to corresponding poses. To that end, we train a separate variational auto-encoder (VAE, (Kingma & Welling, 2014)) on the same data used for learning a map of the environment. We set the emission model of the VAE to the emission model from section 2.1 and freeze its parameters. Thus, we obtain an approximation $q\left(\mathbf{z} \mid \mathbf{x}\right)$ of the posterior over poses $p\left(\mathbf{z} \mid \mathbf{x}\right)$ that conforms with the learned spatial map. This is done once for each of the six mazes. The obtained approximation can be reused for multiple navigation tasks in the given environment. The rest of the evaluation proceeds analogously to the *pose-to-pose* case, using the mode of the approximate posterior, $\mathbf{z}^* = \arg\max_{\mathbf{z}} q\left(\mathbf{z} \mid \mathbf{x}\right)$, as a target. Figure 2.3 illustrates the results of the evaluation. Performance is very similar to the *pose-to-pose* case, with less than 2% of all trajectories failing to reach their target. The slight drop in performance can be attributed to the *perceptual aliasing*—ambiguity in the pose given an observation—of the sensory reading that is typical for spatial environments.

Navigation via gradient descent Finally, we shortly explore the feasibility of solving the navigation task through the gradient descent method described in section 2.3.2. Figure 2.4 illustrates an example application of the algorithm for the *pose-to-pose* case. The proposed entropy term indeed leads to multiple alternative paths to the target being explored, which side steps the issue with local minima otherwise present in the objective function. Since the approach is fully differentiable w.r.t. the model parameters, we hope it can facilitate future research.

5. Discussions and Conclusion

We examined a latent variable model with a global map variable, which can be trained only from local sensor readings in an unsupervised fashion. We applied the fully-trained generative model and specifically the map to perform path planning. In experiments on random mazes, we showcased the potential of our approach, highlighting the value of structured generative models. Only because we learnt a structured map were we able to exploit efficient planning algorithms. The resulting representation enables, e.g., collision avoidance in a straightforward way.

References

Arvanitidis, G., Hansen, L. K., and Hauberg, S. Latent space oddity: On the curvature of deep generative models. 2017. URL http://arxiv.org/abs/1710.11379.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. URL http://arxiv.org/abs/1409.0473.

Bhatti, S., Desmaison, A., Miksik, O., Nardelli, N., Siddharth, N., and Torr, P. H. S. Playing doom with slam-augmented deep reinforcement learning. *CoRR*, abs/1612.00380, 2016. URL http://arxiv.org/abs/1612.00380.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks. *CoRR*, abs/1505.05424, 2015. URL http://arxiv.org/abs/1505.05424.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I. D., and Leonard, J. J. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robotics*, 32(6):1309–1332, 2016. doi: 10.1109/TRO.2016.2624754. URL https://doi.org/10.1109/TRO.2016.2624754.

Chen, N., Klushyn, A., Kurle, R., Jiang, X., Bayer, J., and van der Smagt, P. Metrics for deep generative models. 2017. URL http://arxiv.org/abs/1711.01204.

Dolgov, D., Thrun, S., Montemerlo, M., and Diebel, J. Path planning for autonomous vehicles in unknown semi-structured environments. *The International Journal of Robotics Research*, 29(5):485–501, 2010. doi: 10.1177/0278364909359210. URL https://doi.org/10.1177/0278364909359210.

Fraccaro, M., Sønderby, S. K., Paquet, U., and Winther, O. Sequential neural models with stochastic layers. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 2199–2207, 2016. URL http://papers.nips.cc/paper/6039-sequential-neural-models-with-stochastic-layers.

Fraccaro, M., Rezende, D. J., Zwols, Y., Pritzel, A., Eslami, S. M. A., and Viola, F. Generative temporal models with spatial memory for partially observed environments. *CoRR*, abs/1804.09401, 2018. URL http://arxiv.org/abs/1804.09401.

Gu, S., Ghahramani, Z., and Turner, R. E. Neural adaptive sequential monte carlo. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 2629–2637, 2015. URL http://papers.nips.cc/paper/5961-neural-adaptive-sequential-monte-carlo.

Gupta, S., Davidson, J., Levine, S., Sukthankar, R., and Malik, J. Cognitive mapping and planning for visual navigation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 7272–7281, 2017. doi: 10.1109/CVPR.2017.769. URL https://doi.org/10.1109/CVPR.2017.769.

Hart, P. E., Nilsson, N. J., and Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, July 1968. ISSN 0536-1567. doi: 10.1109/TSSC.1968.300136.

Karl, M., Sölch, M., Bayer, J., and van der Smagt, P. Deep variational bayes filters: Unsupervised learning of state space models from raw data. *CoRR*, abs/1605.06432, 2016. URL http://arxiv.org/abs/1605.06432.

Kingma, D. and Welling, M. Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014.

Krishnan, R. G., Shalit, U., and Sontag, D. Deep kalman filters. *CoRR*, abs/1511.05121, 2015. URL http://arxiv.org/abs/1511.05121.

Maddison, C. J., Lawson, J., Tucker, G., Heess, N., Norouzi, M., Mnih, A., Doucet, A., and Teh, Y. W. Filtering variational objectives. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 6576–6586, 2017. URL http://papers.nips.cc/paper/7235-filtering-variational-objectives.

Mirchev, A., Kayalibay, B., van der Smagt, P., and Bayer, J. Approximate Bayesian inference in spatial environments. 2018. URL http://arxiv.org/abs/1805.07206.

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications*

of Artificial Intelligence, July 28 - August 1, 2002, Edmonton, Alberta, Canada., pp. 593–598, 2002. URL http://www.aaai.org/Library/AAAI/2002/aaai02-089.php.

Murphy, K. P. Bayesian map learning in dynamic environments. In *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, pp. 1015–1021, 1999. URL http://papers.nips.cc/paper/1716-bayesian-map-learning-in-dynamic-environments.

Oh, J., Chockalingam, V., Singh, S. P., and Lee, H. Control of memory, active perception, and action in minecraft. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pp. 2790–2799, 2016. URL http://jmlr.org/proceedings/papers/v48/oh16.html.

Parisotto, E. and Salakhutdinov, R. Neural map: Structured memory for deep reinforcement learning. *CoRR*, abs/1702.08360, 2017. URL http://arxiv.org/abs/1702.08360.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 1278–1286, 2014. URL http://jmlr.org/proceedings/papers/v32/rezende14.html.

Savinov, N., Dosovitskiy, A., and Koltun, V. Semi-parametric topological memory for navigation. *CoRR*, abs/1803.00653, 2018. URL http://arxiv.org/abs/1803.00653.