Joint Belief Tracking and Reward Optimization through Approximate Inference

Pavel Shvechikov[1,2] shvechikov.p@gmail.com
Alexander Grishin[2] grishin.alexgri@yandex.ru
Arsenii Kuznetsov[1] arseny.k@samsung.com
Alexander Fritzler[2] afritzler449@gmail.com
Dmitry Vetrov[1,2] vetrovd@yandex.ru

Abstract

We propose to unify the filtering and control in Partially Observable Markov Decision Processes (POMDPs) via maximizing the marginal likelihood of observations, rewards, *and* surrogate optimality variables. Our approach allows for trust region optimization of policy parameters and filtering to occur simultaneously, increasing the learning speed.

1 Introduction

Igl et al. (2018) introduced the framework of variational sequential Monte Carlo (SMC) (Le et al., 2017; Maddison et al., 2017; Naesseth et al., 2017) for policy optimization in POMDPs. Reported improvement in practical performance may be attributed to an inductive bias imposed by learning of a generative model of the environment. The joint optimization of policy and transition dynamics were formulated as a weighted sum of four losses: Advantage Actor Critic loss (A2C), evidence lower bound on observation marginal log-likelihood, entropy regularization and value prediction loss.

In this work, we unify policy optimization and belief tracking via a single probabilistic model and, consequently, single loss function – evidence lower bound on marginal likelihood of observations, rewards, and optimality of policy. This unification is possible due to a dual role of rewards in POMDP: they are both the target for policy optimization and an observation for belief tracking.

The proposed method subsumes several forms, which are closely related to Max-Entropy learning (Ziebart, 2010) and Trust Region Policy Optimization (Schulman et al., 2015); improves filtering, as it accounts for additional information about hidden states, revealed by emitted rewards; and regularizes the learning process by penalizing the discrepancy between the policy and its prior.

2 Method

To reformulate the control problem as approximate inference, we introduce a latent binary variable $\mathcal{O}_t$, which signifies if the t-th step was played optimally – the greater the reward r_t is, the more probable it orginated from the optimal action at t-th step:

$$p(r_t, \mathcal{O}_t = 1 | s_t, a_t) \propto \exp(r_t) p(r_t | s_t, a_t), \tag{1}$$

where s_t, a_t, r_t are state, action and reward at time step t. The approximate inference in MDPs with such a likelihood and a uniform prior over actions yields the policy that maximizes entropy-

[1] Samsung AI Center, Moscow, Russia
[2] Higher School of Economics, Moscow, Russia

Preprint. Work in progress.

regularized total reward objective (see Levine (2018) for further details and Appendix A for a minimal exposition).

Acting in POMDPs inevitable relies on inference of a state s_t given previous observations $o_{1:t}$. One approach to quantifying the uncertainty in s_t (the so-called belief state) is particle filtering (PF). PF methods approximate the distribution $p(s_t|o_{1:t})$ at each time step with a mixture of K weighted particles.

We propose to unify belief state tracking and reward maximization by performing inference on states *and* optimal actions in a joint probabilistic model (see Appendix B). We approximate the posterior over optimal actions and states with particle-weighted mixture of delta functions:

$$p(a_t, s_{t+1}|\mathcal{O}_{1:t}, r_{1:t}, o_{1:t+1}) \approx \sum_{k=1}^{K} W_{t+1}^k \delta(a_t - a_t^1)\delta(s_{t+1} - s_{t+1}^k), \tag{2}$$

where $W_{t+1}^k = \frac{w_{t+1}^k}{\sum_1^K w_{t+1}^k}$ is the normalized version of unnormalized importance sampling weight:

$$w_{t+1}^k = \frac{p(\mathcal{O}_t|r_t)\mu(a_t|s_t^k)}{\sum_{i=1}^K W_t^i \pi(a_t|s_t^i)} \cdot \frac{p_\theta(r_t|s_t^k, a_t)p_\theta(s_{t+1}|s_t^k, a_t)p_\theta(o_{t+1}|s_{t+1}^k, a_t)}{q_\phi(s_{t+1}^k|s_t^k, a_t, r_t, o_{t+1})}. \tag{3}$$

An action is sampled from the policy averaged over the state uncertainty $a_t \sim \sum_k W_t^k \pi(a_t|s_t^k)$, while the next state representation $s_{t+1} \sim q(s_{t+1}|s_t, a_t, r_t, o_{t+1})$ (see algorithm in Appendix C).

The parameters ϕ, θ are optimized by maximizing the lower bound (Appendix B) on the marginal likelihood:

$$\log p(r_{1:\tau}, \mathcal{O}_{1:T}, o_{1:\tau+1}) \geq \mathbb{E}\Big[\sum_{t=1}^{\tau} \log \frac{1}{K} \sum_{k=1}^{K} w_t^k + \log p(\mathcal{O}_{\tau+1:T}|s_{\tau+1})\Big],$$

where τ is the length of the observed part of the episode. As we show in Appendix B, one part of this lower bound is equivalent to the entropy-regularized advantage actor critic loss, while another part – to the belief tracking in state-space models.

The concept of optimal variable $\mathcal{O}_t$ and explicit probabilistic model of rewards allows for the joint optimization of policy and system dynamics. In Appendix B we derive the proposed lower bound, show the correctness of the iterative update of the posterior, and discuss the connections with some of the Reinforcement Learning methods.

3 Experiments & discussion

In preliminary experiments on flickering Atari environments (see Appendix E for learning curves and Appendix D for the experiment setup) our method performed comparably to the state of the art DVRL algorithm (Igl et al., 2018).

As Table 1 suggests, in some games – i.e. Pong (P), Chopper Command (CC) – our method scored much higher total rewards given the same amount of experience, while not losing significantly in any one of the environments. This is notable, since this testbench is not the one, in which our method is designed to shine – it is characterized by sparse rewards, that are intentionally clipped to few distinctive values, $\{-1, 0, 1\}$. The detailed empirical evaluation of the algorithm, as well as the quality of the learned transition dynamics is the work in progress.

Table 1: Resulting scores. We report here both models trained on 10M frames. Game names are indicated with first letters: A: Asteroids, BR: Beam Rider, B: Bowling, C: Centipede, CC: Chopper Command, DD: Double Dunk, F: Frostbite, IH: Ice Hockey, MP: Ms. Pacman, P: Pong.

	A	BR	B	C	CC	DD	F	IH	MP	P
DVRL	1456	773	29	3825	2703	-16.3	254	-6.7	**1752**	-10
Ours	**1603**	**928**	28	3865	**3329**	-16.4	262	-6.8	1592	**-4**

Acknowledgments

The study has been supported by Russian Science Foundation (grant 17-71-20072) and Samsung Research, Samsung Electronics.

We thank Ekaterina Lobacheva for the help with the paper.

References

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*.

Igl, M., Zintgraf, L., Le, T. A., Wood, F., and Whiteson, S. (2018). Deep variational reinforcement learning for pomdps. *arXiv preprint arXiv:1806.02426*.

Kolchinsky, A. and Tracey, B. D. (2017). Estimating mixture entropy with pairwise distances. *Entropy*, 19(7):361.

Laubenfels, R. (2005). Feynman–kac formulae: Genealogical and interacting particle systems with applications.

Le, T. A., Igl, M., Rainforth, T., Jin, T., and Wood, F. (2017). Auto-encoding sequential monte carlo. *arXiv preprint arXiv:1705.10306*.

Levine, S. (2018). Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*.

Maddison, C. J., Lawson, J., Tucker, G., Heess, N., Norouzi, M., Mnih, A., Doucet, A., and Teh, Y. (2017). Filtering variational objectives. In *Advances in Neural Information Processing Systems*, pages 6573–6583.

Naesseth, C. A., Linderman, S. W., Ranganath, R., and Blei, D. M. (2017). Variational sequential monte carlo. *arXiv preprint arXiv:1705.11140*.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897.

Ziebart, B. D. (2010). Modeling purposeful adaptive behavior with the principle of maximum causal entropy. *PhD thesis, Carnegie Mellon University.*

A Control as inference in MDPs

As usual in variational inference, we specify the prior $p(\cdot)$ and the posterior $q(\cdot)$. In RL context these are specified over trajectories $\tau = \langle s_1, a_1, ..., s_T, a_T \rangle$, i.e.

$$p(\tau) = p(s_1) \prod_{t=1}^{T} p(s_{t+1}|s_t, a_t)\mu(a_t|s_t), \qquad q(\tau) = p(s_1) \prod_{t=1}^{T} p(s_{t+1}|s_t, a_t)\pi(a_t|s_t),$$

where μ and π are prior and posterior policies respectively.

To reformulate the total reward optimization as approximate inference, we reweight the prior likelihood of the trajectory proportionally to the exponentiated reward obtained along it:

$$p(\mathcal{O}_{1:T} = 1|\tau) \;=\; \prod_{t=1}^{T} p(\mathcal{O}_t = 1|s_t, a_t) \;\propto\; \prod_{t=1}^{T} \exp r(s_t, a_t)$$

Under this model, approximate inference is equivalent to maximization of the reward, penalized for the high discrepancy between prior and posterior policies via KL divergence:

$$\log p(\mathcal{O}_{1:T}) \geq \mathbb{E}_{q(\tau)} \log \frac{p(\mathcal{O}_{1:T} = 1|\tau)p(\tau)}{q(\tau)} = \mathbb{E}_{q(\tau)} \sum_{t=1}^{T} \left[r(s_t, a_t) - \mathrm{KL}\left(\pi(\cdot|s_t)||\mu(\cdot|s_t)\right) \right]$$

Uniform prior, μ, yield a Max Entropy framework (Haarnoja et al., 2018; Ziebart, 2010). For the thorough discussion of relations with the usual RL objective, see Levine (2018).

B Joint filtering and control for POMDPs

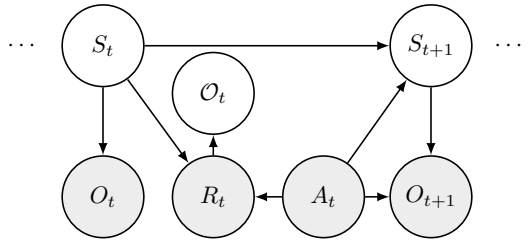

Figure 1: Graphical model of POMDP with optimality variables $\mathcal{O}_t$. Grey nodes are observed, white nodes are hidden.

B.1 Lower bound

Assume an agent has collected experience up to the time step $t = \tau$. We aim to maximize the marginal likelihood of the observed data and optimality indicators:

$$\begin{aligned} p(r_{1:\tau}, \mathcal{O}_{1:T}, o_{1:\tau+1}) &= p(\mathcal{O}_{\tau+1:T}|r_{1:\tau}, \mathcal{O}_{1:\tau}, o_{1:\tau+1}) \cdot \prod_{t=0}^{\tau} p(r_t, \mathcal{O}_t, o_{t+1}|r_{1:t-1}, \mathcal{O}_{1:t-1}, o_{1:t}) \\ &= p(\mathcal{O}_{\tau+1:T}|r_{1:\tau}, \mathcal{O}_{1:\tau}, o_{1:\tau+1}) \cdot \prod_{t=0}^{\tau} p(\mathcal{O}_t|r_t) p(r_t, o_{t+1}|r_{1:t-1}, o_{1:t}). \end{aligned},$$

where we assume that $p(r_t, \mathcal{O}_t, o_{t+1}|r_{1:t-1}, \mathcal{O}_{1:t-1}, o_{1:t})$ equals to $p(o_1)$ if $t = 0$.
For this we introduce the joint likelihood defined with respect to some prior policy $\mu(a_t|s_t)$ and system dynamics $p(s_{t+1}|s_t, a_t)$:

$$\begin{aligned} &p(a_{1:\tau}, r_{1:\tau}, \mathcal{O}_{1:T}, s_{1:\tau+1}, o_{1:\tau+1}) \\ &= p(\mathcal{O}_{\tau+1:T}|s_{\tau+1}) \prod_{t=0}^{\tau} p(\mathcal{O}_t|r_t) \mu(a_t|s_t) p(r_t|s_t, a_t) p(s_{t+1}|s_t, a_t) p(o_{t+1}|s_{t+1}, a_t), \end{aligned}$$

where we employed conditional independencies (Figure 1) to simplify

$$p(\mathcal{O}_{\tau+1:T}|a_{1:\tau}, r_{1:\tau}, \mathcal{O}_{1:T}, s_{1:\tau+1}, o_{1:\tau+1}) = p(\mathcal{O}_{\tau+1:T}|s_{\tau+1})$$

and defined $p(\mathcal{O}_t|r_t)\mu(a_t|s_t)p(r_t|s_t, a_t)p(s_{t+1}|s_t, a_t)p(o_{t+1}|s_{t+1}, a_t)$ at $t = 0$ as $p(o_1|s_1)p(s_1)$.

Thus, the lower bound on the marginal likelihood:

$$\begin{aligned} \log p(r_{1:\tau}, \mathcal{O}_{1:T}, o_{1:\tau+1}) &\geq \int q(s_{1:\tau+1})\pi(a_{1:\tau}) \log \frac{p(a_{1:\tau}, r_{1:\tau}, \mathcal{O}_{1:T}, s_{1:\tau+1}, o_{1:\tau+1})}{q(s_{1:\tau+1})\pi(a_{1:\tau})} ds_{1:\tau+1} da_{1:\tau} \\ &= \mathbb{E}\left[\sum_{t=0}^{\tau} \log w_{t+1}(s_t, a_t, s_{t+1}) + \log p(\mathcal{O}_{\tau+1:T}|s_{\tau+1})\right], \end{aligned} \tag{4}$$

where

$$w_{t+1}(s_t, a_t, s_{t+1}) = \frac{p(\mathcal{O}_t|r_t)\mu(a_t|s_t)}{\pi(a_t|h_t)} \cdot \frac{p(r_t|s_t, a_t)p(s_{t+1}|s_t, a_t)p(o_{t+1}|s_{t+1}, a_t)}{q(s_{t+1}|s_t, a_t, r_t, o_{t+1})}.$$

B.2 Improving the lower bound

To make the bound tighter and enable sequential sampling we use Variational Sequential Monte Carlo (Le et al., 2017; Maddison et al., 2017; Naesseth et al., 2017) and approximate with particles the posterior over states and optimal actions:

$$p(a_t, s_{t+1}|r_{1:t}, \mathcal{O}_{1:t}, o_{1:t+1}) = \frac{p(a_t, r_t, \mathcal{O}_t, s_{t+1}, o_{t+1}|r_{1:t-1}, \mathcal{O}_{1:t-1}, o_{1:t})}{p(r_t, \mathcal{O}_t, o_{t+1}|r_{1:t-1}, \mathcal{O}_{1:t-1}, o_{1:t})}. \tag{5}$$

This allows us to refine the distribution over the hidden state, accounting for the new observations.

Assume at time step t we approximate a distribution $p(s_t|\mathcal{O}_{1:t-1}, r_{1:t-1}, o_{1:t})$ with a weighted mixture of delta functions $\sum_{k=1}^{K} W_t^k \delta(s_t - s_t^k)$, s.t. $\sum_k W_t^k = 1$.

First, for each k we sample an ancestor indexes $u_t^k \sim \text{Discrete}\left(\{W_t^k\}_{k=1}^K\right)$, which makes the approximation equally weighted $\sum_{k=1}^{K} W_t^k \delta(s_t - s_t^k) \approx \frac{1}{K}\sum_{k=1}^{K} \delta(s_t - s_t^{u_t^k})$.

For clarity we derive approximations for the numerator and denominator of (5) separately. To reduce the clutter, we also shorthand $\pi(a_t|r_{1:t-1}, o_{1:t}, \mathcal{O}_{1:T}) = \pi(a_t|h_t)$. The numerator

$$
\begin{aligned}
&p(a_t, r_t, \mathcal{O}_t, s_{t+1}, o_{t+1}|r_{1:t-1}, \mathcal{O}_{1:t-1}, o_{1:t}) \\
&= \int p(s_t|r_{1:t-1}, \mathcal{O}_{1:t-1}, o_{1:t}) p(a_t, r_t, \mathcal{O}_t, s_{t+1}, o_{t+1}|s_t) ds_t \\
&= \int p(s_t|r_{1:t-1}, \mathcal{O}_{1:t-1}, o_{1:t}) w_{t+1}(s_t, a_t, s_{t+1}) \pi(a_t|h_t) q(s_{t+1}|s_t, a_t, r_t, o_{t+1}) ds_t \\
&\approx \frac{1}{K}\sum_{k=1}^{K} w_{t+1}(s_t^{u_t^k}, a_t, s_{t+1}) \pi(a_t|h_t) q(s_{t+1}|s_t^{u_t^k}, a_t, r_t, o_{t+1}) \\
&\approx \frac{1}{K}\sum_{k=1}^{K} w_{t+1}(s_t^{u_t^k}, a_t^1, s_{t+1}^k) \delta(a_t - a_t^1) \delta(s_{t+1} - s_{t+1}^k).
\end{aligned}
$$

Here we approximate the measures $q(\cdot|s_t^{u_t^k}, a_t, r_t, o_{t+1})$ and $\pi(a_t|h_t)$ by sampling from the respective distributions (once for the distribution over actions and once for each $s_t^{u_t^k}$ for the distributions over the next state).

From now on we write w_{t+1}^k for $w(s_t^{u_t^k}, a_t^1, s_{t+1}^k)$ The denominator is just an integral of the numerator over a_t, s_{t+1}, that is

$$
p(r_t, \mathcal{O}_t, o_{t+1}|r_{1:t-1}, \mathcal{O}_{1:t-1}, o_{1:t}) \approx \frac{1}{K}\sum_{j=1}^{K} w_{t+1}^j \tag{6}
$$

This allows us to arrive at the expression of self-normalized importance sampling:

$$
\begin{aligned}
p(a_t, s_{t+1}|r_{1:t}, \mathcal{O}_{1:t}, o_{1:t+1}) &\approx \frac{\frac{1}{K}\sum_{k=1}^{K} w_{t+1}^k \delta(a_t - a_t^1)\delta(s_{t+1} - s_{t+1}^k)}{\frac{1}{K}\sum_{j=1}^{K} w_{t+1}^j} \\
&= \sum_{k=1}^{K} W_{t+1}^k \delta(a_t - a_t^1)\delta(s_{t+1} - s_{t+1}^k)
\end{aligned}
$$

We can integrate out the a_t in the the last equation to obtain approximation for

$$
p(s_{t+1}|r_{1:t}, \mathcal{O}_{1:t}, o_{1:t+1}) = \sum_{k=1}^{K} W_{t+1}^k \delta(s_{t+1} - s_{t+1}^k)
$$

Since the product of (6) over time gives an unbiased estimate of the marginal likelihood (Laubenfels, 2005):

$$
p(r_{1:\tau}, \mathcal{O}_{1:\tau}, o_{1:\tau+1}) = \prod_{t=1}^{\tau} \frac{1}{K}\sum_{k=1}^{K} w_{t+1}^k,
$$

the improved sequential Monte Carlo (SMC) lower bound (4) can be written as

$$
\log p(r_{1:\tau}, \mathcal{O}_{1:T}, o_{1:\tau+1}) \geq \mathbb{E}\Big[\sum_{t=1}^{\tau} \log \frac{1}{K}\sum_{k=1}^{K} w_t^k + \log p(\mathcal{O}_{\tau+1:T}|s_{\tau+1})\Big]
$$

The full algorithm is given in Appendix C.

B.3 Analysis of the lower bound

For $k = 1$ the lower bound admits an illuminating expression

$$\mathbb{E}\Bigg[\log p(\mathcal{O}_{\tau+1:T}|s^1_{\tau+1}) + \sum_{t=1}^{\tau} \log p(\mathcal{O}_t|r_t) - \mathrm{KL}(\pi(\cdot|h), \mu(\cdot|s^1_t)) \\ + \log \frac{p(r_t|s^1_t, a^1_t)p(s^1_{t+1}|s^1_t, a^1_t)p(o_{t+1}|s^1_{t+1}, a^1_t)}{q(s^1_{t+1}|s^1_t, a^1_t, r_t, o_{t+1})} \Bigg]$$

In particular, the first three terms inside the expectation are equivalent to REINFORCE algorithm applied to rewards, adjusted for the discrepancy between π and μ (assuming $p(\mathcal{O}_t|r_t) = \exp(r_t)$). The last term is responsible for learning dynamics. From this expression, we can see that the choice of μ may yield different algorithms: for the uniform μ we obtain the analog of Maximum Entropy framework (Ziebart, 2010), for the μ equal to the previous version of π (i.e. before the last gradient update) makes our algorithms similar to TRPO (Schulman et al., 2015) in a POMDP setting.

The policy $\pi(a_t|h_t)$ does not depend on the summation index k. So, we can move it out of the averaging over particles and arrive at the explicit entropy maximization:

$$\mathbb{E}\Bigg[\log p(\mathcal{O}_{\tau+1:T}|s^k_{\tau+1}) + \sum_{t=1}^{\tau} \log p(\mathcal{O}_t|r_t) + \mathrm{H}(\pi(\cdot|h_t)) \\ + \log \frac{1}{K} \sum_{k=1}^{K} \frac{\mu(a_t|s_t^{u_t^k})p(r_t|s_t^{u_t^k}, a^1_t)p(s^k_{t+1}|s_t^{u_t^k}, a^1_t)p(o_{t+1}|s^k_{t+1}, a^1_t)}{q(s^1_{t+1}|s_t^{u_t^k}, a^1_t, r_t, o_{t+1})} \Bigg]$$

The term $\log p(\mathcal{O}_{\tau+1:T}|s^k_{\tau+1})$ is essentially the approximation of the $V^\pi(s^k_{\tau+1})$ (state conditioned expectation over all future observed and hidden variables). Thus, the expression $\sum_{t=1}^{\tau} \log p(\mathcal{O}_t|r_t) + \log p(\mathcal{O}_{\tau+1:T}|s^k_{\tau+1})$ is equivalent to the τ-step return $r_1 + ... + r_\tau + V(s^k_{\tau+1})$. Moreover, we can recover the analogue of Advantage Actor Critic entropy-regularized update, if we additionally subtract from the objective the baseline $V(s^k_1)$ and average objectives with different lengths of collected experience, e.g. $\tau = 1, 2, 3$.

C Algorithm

C.1 Latent state

To represent the belief state with particles we follow Igl et al. (2018) and use triplets $\{z^k_t, h^k_t, w^k_t\}$. Latent variable h^k_t is the latent state of an RNN, which aggregates all previous history. Stochastic latent variable z^k_t represents stochastic transition via proposal distribution q. Weights w^k_t represent importance weights.

C.2 Latent state update

To update latent state we proceed as follows:

$$u^k_t \sim \text{Discrete}(\frac{w^k_t}{\sum_{k=1}^K w^k_t})$$

$$z^k_{t+1} \sim q_\phi(z^k_{t+1}|h_t^{u_t^k}, a_t, r_t, o_{t+1})$$

$$h^k_{t+1} = \psi^{\text{RNN}}_\theta(z^k_{t+1}, h_t^{u_t^k}, a_t, r_t, o_{t+1})$$

$$w^k_{t+1} = \frac{p(\mathcal{O}_t|r_t)\mu(a_t|z_t^{u_t^k}, h_t^{u_t^k})}{\sum_{i=1}^K W^i_t \pi_\phi(a_t|z^i_t, h^i_t)} \frac{p_\theta(r_t|h_t^{u_t^k}, a_t)p_\theta(z^k_{t+1}|h_t^{u_t^k}, a_t)p_\theta(o_{t+1}|h_t^{u_t^k}, a_t, z^k_{t+1})}{q_\phi(z^k_{t+1}|h_t^{u_t^k}, a_t, r_t, o_{t+1})}$$

C.3 Pseudocode

We introduce some additional notations to clarify algorithm's pseudocode. Slighly abusing the notation let $\bar{\pi}_t = \sum_{k=1}^K W^k_t \pi_\phi(a_t|z^k_t, h^k_t)$ and $\bar{V}_t = \sum_{k=1}^K W^k_t V(z^k_t, h^k_t)$. We divide the variational

lower bound into three summands:

$$l = \underbrace{\mathbb{E}\left[p(\mathcal{O}_{\tau+1:T}|s^k_{\tau+1}) + \sum_{t=t_s}^{t_e} \log p(\mathcal{O}_t|r_t)\right]}_{l_1} + \underbrace{\mathbb{E}\left[-\sum_{t=t_s}^{t_e} \log(\bar{\pi}_{t-1})\right]}_{l_2} + \underbrace{\mathbb{E}\left[\log \frac{1}{K}\sum_{k=1}^{K} \tilde{w}^k_t\right]}_{l_3}$$

where

$$\tilde{w}^k_t = \frac{\mu(a_{t-1}|z^{u^k_{t-1}}_{t-1}, h^{u^k_{t-1}}_{t-1}) p_\theta(r_{t-1}|h^{u^k_{t-1}}_{t-1}, a_{t-1}) p_\theta(z^k_t|h^{u^k_{t-1}}_{t-1}, a_{t-1}) p_\theta(o_t|h^{u^k_{t-1}}_{t-1}, a_{t-1}, z^k_t)}{q_\phi(z^k_t|h^{u^k_{t-1}}_{t-1}, a_{t-1}, r_{t-1}, o_t)}$$

The first summand (l_1) manifests the REINFORCE algorithm which can be transformed into Advantage Actor Critic algorithm by approximating the $\log p(\mathcal{O}_{\tau+1:T}|s_{\tau+1})$ and introducing the baseline.

The second summand (l_2) is the policy entropy, which could be computed analytically for categorical distribution and lower-bounded in case of continuous control for mixture of Gaussians policy (e.g. following Kolchinsky and Tracey (2017)).

The last summand (l_3) estimates the quality of environment modelling. Note, that if we omit the gradients with respect to categorical resampling of u^k_t (following Le et al. (2017); Maddison et al. (2017); Naesseth et al. (2017)) the last addend becomes fully differentiable.

Algorithm 1 Training algorithm

Initialize parameters θ for environment model and ϕ for variational approximation (see (3))
Initialize prior update rate λ
$\tau \leftarrow 1$
repeat
  Initialize play interval $\tau_\text{s} \leftarrow \tau; \tau_\text{e} \leftarrow \tau + n - 1$ and log-likelihoods $l_2, l_3 \leftarrow 0$
  for $\tau = \tau_\text{s}$ to $\tau = \tau_\text{e}$ **do**
    if $\tau = 1$ **then**
      $\{z^k_1, h^k_1, w^k_1\}_{k=1}^K, \{o_1\} \leftarrow$ Initial step()
    else
      $\{z^k_\tau, h^k_\tau, w^k_\tau\}_{k=1}^K, \{a_{\tau-1}, r_{\tau-1}, o_\tau, \text{done}\} \leftarrow$ Forward step$(\{z^k_{\tau-1}, h^k_{\tau-1}, w^k_{\tau-1}\}_{k=1}^K)$
      Update entropy with analytical estimate $l_2 \leftarrow l_2 + \mathrm{H}(\bar{\pi}_{\tau-1})$
      Update model likelihood with Monte Carlo estimate $l_3 \leftarrow l_3 + \log \frac{1}{K}\sum_{k=1}^K \tilde{w}^k_\tau$
    end if
    if done **then**
      $\tau_\text{e} \leftarrow \tau$
      break
    end if
  end for
  Gradient of RL part of loss is computed akin to Advantage Actor Critic algorithm:
  $\nabla l_1 = \sum_{t=\tau_s}^{\tau_e} \left(\nabla \log\left(\bar{\pi}_t\right)\right)\left[\sum_{i=t}^{\tau_e}\left[\sum_{j=t}^{i} r_j + \bar{V}^-_{i+1} - \bar{V}^-_t\right]\right]$, where $(\cdot)^-$ is stop gradient
  Gradient of policy entropy and environment model parts of loss is computed via usual autograd:
  $\nabla l_{2,3} = \nabla \sum_{t=\tau_s}^{\tau_e}\left[\mathrm{H}(\bar{\pi}_{t-1}) + \log \frac{1}{K}\sum_{k=1}^K \tilde{w}_t\right]$, where $\mathrm{H}(\cdot)$ is entropy
  Make baseline more precise (in fact, TD loss):
  $\nabla l_\text{TD} = \nabla \sum_{t=\tau_s}^{\tau_e}\left[\sum_{i=t}^{\tau_e}\left[\sum_{j=t}^{i} r_j + \bar{V}^-_{i+1} - \bar{V}_t\right]^2\right]$
  Gradient update
  if prior policy update **then**
    Soft update for prior policy parameters:
    $\phi_\mu \leftarrow (1-\lambda)\phi_\mu + \lambda\phi_\pi$
  end if
  if done **then**
    Set $\tau = 1$
  end if
until Convergence

Algorithm 2 Forward step

Input: $\{z_t^k, h_t^k, w_t^k\}_{k=1}^K$
Sample action according to belief state $a_t \sim \sum_{k=1}^K \pi_\phi(a_t|z_t^k, h_t^k) w_t^k$
Take action, get reward and observation r_t, o_{t+1}, done
for $k = 1$ **do** to K
  Sample ancestor index $u_t^k \sim \text{Discrete}\left(w_t^k / \sum_{j=1}^K w_t^j\right)$
  Sample first part of latent state $z_{t+1}^k \sim q_\phi(z_{t+1}^k | h_t^{u_t^k}, a_t, r_t, o_{t+1})$
  Update second part of latent state $h_{t+1}^k = \psi_\theta^{\text{RNN}}(z_{t+1}^k, h_t^{u_t^k}, a_t, o_{t+1})$
  Compute weights w_{t+1}^k following equation (3)
end for
return $\{z_{t+1}^k, h_{t+1}^k, w_{t+1}^k\}_{k=1}^K, \{a_t, r_t, o_{t+1}, \text{done}\}$

Algorithm 3 Initial step

Reset environment, observe initial state o_1
Let $h_0^k \leftarrow h_{\text{init}}$, $z_0^k \leftarrow z_{\text{init}}$, $a_0 \leftarrow a_{\text{init}}$, $r_0 \leftarrow r_{\text{init}}$
for $k = 1$ **do** to K
  Sample first part of particle $z_1^k \sim q_\phi(z_1^k | h_0^k, a_0, r_0, o_1)$
  Update second part of particle $h_1^k = \psi_\theta^{\text{RNN}}(z_1^k, h_0^k, a_0, o_1)$
  Compute initial weights w_1^k following equation (3)
end for
return $\{z_1^k, h_1^k, w_1^k\}_{k=1}^K, \{o_1\}$

D Experimental setup

We have parametrized distributions in the same way as in DVRL Igl et al. (2018).

Transition distribution $p_\theta(z_{t+1}|h_t, a_t)$, emission model $p_\phi(o_{t+1}|z_{t+1}, h_t, a_t)$ were exactly the same as in Igl et al. (2018). The proposal distribution $q_\phi(z_{t+1}|h_t, a_t, r_t, o_{t+1})$ was conditioned on an additional argument, r_t, and thus is the function of $[h_t, a_t, r_t, o_{t+1}]$.

We have modelled the reward $p_\theta(r_t|h_t, a_t)$ as a discrete distribution (rewards are clipped to $\{-1, 0, -1\}$ as usual) whose parameters are determined by a neural network with the same architecture as of the emission network).

The prior $\mu(a_t|s_t)$ was exactly of the same architecture as the policy $\pi(a_t|s_t)$ (see Igl et al. (2018)), albeit with different parameters.

E Learning curves on the Flickering Atari

We compare our method with the state of the art DVRL algorithm (Igl et al., 2018) on flickering Atari games. In these environments, each game screen and reward are blacked out with probability 0.5, hiding the state from the agent. Additionally, as in Igl et al. (2018), our model receives only a single frame per time step, making it more difficult for an agent to account for velocities and accelerations of various objects.

Our preliminary experiments are 5 times shorter than experiments in original DVRL paper (i.e. 10 million frames vs 50 million frames). However, they are illustrative, since our algorithm performs comparably to the state of the art and marginally better in some cases.

Empirical evaluation of the proposed approach is in progress, and we conjecture that the very same algorithm will shine on the problems with more variability in rewards (i.e. on Flickering Atari games *without* reward clipping).

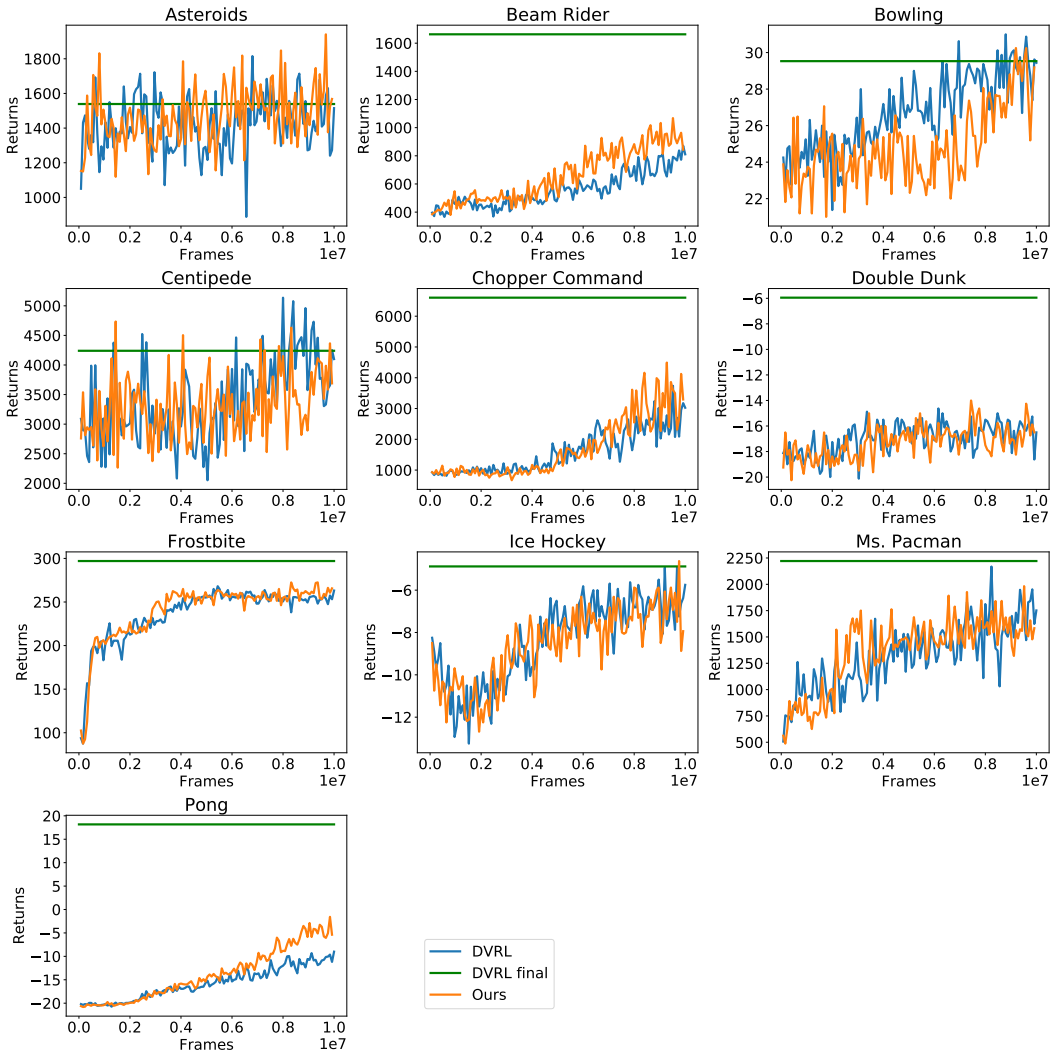

Figure 2: Results for DVRL (Igl et al., 2018) and our model on Flickering Atari environments. The green horizontal line represents the DVRL performance after five times as many environment interactions, as is reported by Igl et al. (2018).