An inference perspective on model-based reinforcement learning

Joseph Marino[1] **Yisong Yue**[1]

Abstract

Model-based reinforcement learning (RL) offers the prospect of improved sample complexity through model learning and model-based planning. However, the agent's model and policy are often considered separately, only to be combined heuristically in the objective function. Starting from the perspective of RL as probabilistic inference and learning, we derive the general objective for a model-based agent in a partially-observable environment. In comparison with conventional approaches to model-based RL, this objective includes additional terms that impact action selection and model learning. We interpret several recent approaches in light of this perspective, and by noting differences, we highlight directions for further investigation.

1. Introduction

Reinforcement learning (RL) has recently seen substantial progress, primarily as a result of model-free approaches (Mnih et al., 2015; Schulman et al., 2015; Lillicrap et al., 2015; Mnih et al., 2016; Schulman et al., 2017b). However, model-free RL tends to require many environment interactions to update action or value estimates, referred to as high *sample complexity*. Model-based approaches, in contrast, can more rapidly incorporate environment information through model learning and model-based planning, thereby improving sample complexity. Yet, model-based RL can be more difficult to train and typically suffers from worse asymptotic performance. Part of this difficulty may arise from considering the agent's model and policy separately, heuristically combining these components in the objective function. A more unified framing may illuminate the underlying interaction between these components.

We approach model-based RL from the perspective of probabilistic inference and learning (Levine, 2018). By framing an agent's interaction with a partially-observable environment as a probabilistic graphical model, we arrive at a single variational objective for model learning, state estimation, and action selection. This framing generalizes various special cases in both model-free and model-based RL. However, in comparison with conventional model-based approaches, this objective contains additional terms that impact action selection and model learning:

- In addition to model-based planning, an agent also contains a model-free *action prior*, which can help to initialize planning and be used as a roll-out policy.
- The objective contains the agent's *marginal log-likelihood of observations*, restricting the agent's model to learn task-relevant state information and biasing planning toward higher-confidence states.

In light of this perspective, we interpret several recent approaches and highlight directions for further investigation.

2. Related work

Recent improvements in generative models (Kingma & Welling, 2014; Rezende et al., 2014), particularly sequential latent variable models (Chung et al., 2015; Fraccaro et al., 2016; Gemici et al., 2017; Goyal et al., 2017), have facilitated renewed interest in model-based RL. Models can assist in learning state representations (Lange & Riedmiller, 2010; Jaderberg et al., 2016; Eslami et al., 2018; Igl et al., 2018), serve as environment emulators (Sutton, 1990; Ha & Schmidhuber, 2018; Kurutach et al., 2018; Kaiser et al., 2019), or enable model-based planning (Deisenroth et al., 2015; Henaff et al., 2017; Nagabandi et al., 2018; Chua et al., 2018; Hafner et al., 2019). In a typical set-up, a generative model is trained to capture the environment dynamics. The model can then be used in place of the environment for training or planning actions. However, it is unclear how to constructively combine these components. For instance, a policy could impact the model's internal state representation, and a model's uncertainty could impact training or planning.

To approach model-based RL from a more unified formulation, we start from the perspective of RL as probabilistic inference, recasting actions as latent variables and inferring actions that result in "observed" high reward. This

[1] California Institute of Technology, Pasadena, CA, USA. Correspondence to: Joseph Marino <jmarino@caltech.edu>.

ICML Workshop on Generative Modeling and Model-Based Reasoning for Robotics and AI, Long Beach, California, 2019. Copyright 2019 by the author(s).

is typically accomplished by introducing binary variables (Cooper, 1988) that map reward to a degree of "optimality" (Levine, 2018), then setting optimality to be observed as true. Previous works have investigated this probabilistic inference approach in the context of planning (Attias, 2003; Toussaint & Storkey, 2006; Toussaint, 2009; Hoffman et al., 2009; Botvinick & Toussaint, 2012; Piché et al., 2019) and control (Todorov, 2008; Rawlik et al., 2013; Haarnoja et al., 2017; Schulman et al., 2017a; Haarnoja et al., 2018b; Levine, 2018; Shvechikov et al., 2018). However, these approaches are generally considered outside of the context of model learning or only consider fully-observable environments. Our formulation, instead, centers on using learned models for state estimation and action selection in partially-observable environments.

Model-based RL is not without its limitations: planning is computationally expensive, and poor model quality can negatively impact asymptotic task performance. For these reasons, there has been considerable work in combining model-based and model-free RL (Heess et al., 2015; Gu et al., 2016; Chebotar et al., 2017; Bansal et al., 2017; Weber et al., 2017; Oh et al., 2017; Nagabandi et al., 2018; Pong et al., 2018; Co-Reyes et al., 2018). While many approaches exist, one line of work involves *consolidating* model-based planning into a model-free policy, e.g. (Buesing et al., 2018). As we describe in Sections 3 and 4, our formulation naturally frames this process as Bayesian estimation, converting model-based likelihoods into model-free priors. Likewise, our formulation points out inconsistencies between conventional model-based and model-free objectives.

3. Variational reinforcement learning

3.1. Set-up: agent, environment, and optimality

Consider an agent performing a sequence of actions within a partially-observable environment. At time step t, the environment is in state $\mathbf{s}_t$, but the agent is only able to observe sensory observations, $\mathbf{x}_t$, and reward, r_t. The agent performs action $\mathbf{a}_t$, at which point the environment transitions to state $\mathbf{s}_{t+1}$. Because the environment state is unobserved, we are free to assume it is Markov, potentially redefining it to capture the entire history. The distribution over the environment variables, denoted p_e, can thus be expressed as

$$p_\text{e}(\mathbf{x}_{1:T}, r_{1:T}, \mathbf{s}_{1:T} | \mathbf{a}_{1:T-1}) = \prod_{t=1}^{T} p_\text{e}(\mathbf{s}_t | \mathbf{s}_{t-1}, \mathbf{a}_{t-1}) \cdot p_\text{e}(\mathbf{x}_t | \mathbf{s}_t) p_\text{e}(r_t | \mathbf{s}_t). \quad (1)$$

Typically, we do not have analytical expressions for these distributions. Instead, we must resort to sampling $\mathbf{x}$ and r from the environment through interactions. We do, however, have access to the agent, which is subject to our design choices. The agent contains a distribution over actions, which, in accordance with the forward nature of the environment, may be conditioned on any past and present variables. The agent may also contain additional latent variables, which we denote by $\mathbf{z}$, allowing the agent to internally represent state information. A general form for the distribution over the agent's variables, denoted p_a, is

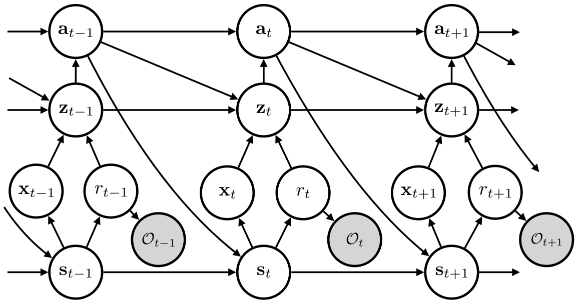

Figure 1. **Graphical Model of the Agent, Environment, and Optimality**. The graphical model depicts a simplified version of the joint distribution over all variables, which is comprised of the environment (Eq. 1), agent (Eq. 2), and optimality (Eq. 3) distributions. By constructing a graphical model of the agent-environment system, conditioning on optimality (maximal reward), and marginalizing over all other variables, RL is recast as a probabilistic inference and learning problem.

$$p_\text{a}(\mathbf{a}_{1:T}, \mathbf{z}_{1:T} | \mathbf{x}_{1:T}, r_{1:T}) = \prod_{t=1}^{T} p_\text{a}(\mathbf{a}_t | \mathbf{a}_{<t}, \mathbf{z}_{\leq t}, \mathbf{x}_{\leq t}, r_{\leq t}) \cdot p_\text{a}(\mathbf{z}_t | \mathbf{a}_{<t}, \mathbf{z}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}). \quad (2)$$

Note that marginalizing this distribution over $\mathbf{z}_{1:T}$ yields a distribution over $\mathbf{a}_{1:T}$. Although not explicitly written, the agent's distributions are parameterized by θ. Together, the conditional distributions for the agent (Eq. 2) and environment (Eq. 1) define the joint distribution of trajectories. However, to train the agent to perform tasks, we must formulate reward maximization in probabilistic terms, specifying preferences over states and actions. Following previous works, we introduce binary variables (Cooper, 1988), mapping reward to "optimality" (Levine, 2018). Denoting optimality as $\mathcal{O}$, the conditional likelihood of optimality at time t is a Bernoulli distribution, $\mathcal{B}$, with

$$p(\mathcal{O}_t | r_t) = \mathcal{B}(\exp(r_t)) \quad (3)$$

for $r_t \leq 0$. Conditioning on $\mathcal{O}_t = 1$, the conditional log-likelihood becomes $\log p(\mathcal{O}_t = 1 | r_t) = r_t$. Thus, maximizing the log-likelihood of optimality is equivalent to maximizing reward. Combining the agent, environment, and optimality distributions yields the joint distribution $p(\mathbf{x}_{1:T}, r_{1:T}, \mathbf{s}_{1:T}, \mathbf{a}_{1:T}, \mathbf{z}_{1:T}, \mathcal{O}_{1:T})$. Figure 1 illustrates a graphical model for a simplified form of this distribution.

In contrast to the standard RL set-up, in which an agent is trained and evaluated according to the expected sum of rewards, i.e. return, the objective is the marginal log-likelihood of optimality,

$$\begin{aligned}\theta^* &= \arg\max_{\theta} \mathbb{E}_{\mathcal{O}_{1:T}\sim\delta(\mathbf{1})}\left[\log p(\mathcal{O}_{1:T})\right] \\ &= \arg\max_{\theta} \log p(\mathcal{O}_{1:T} = \mathbf{1}).\end{aligned} \tag{4}$$

Thus, considering the distribution of all possible trajectories, the agent is trained such that maximal return is maximally likely. Through learning, we adjust the parameters, θ, underlying the agent's internal state and action distributions in an attempt to fit this desired outcome.

3.2. Variational inference and learning

Evaluating and optimizing $\log p(\mathcal{O}_{1:T} = \mathbf{1})$ directly is intractable, as it involves marginalizing over all variables in the agent and environment distributions. Instead, we lower bound the objective using techniques from variational inference (Jordan et al., 1998). As in (Levine, 2018), we introduce a *structured* approximate posterior distribution, q, over the variables for which sampling is under our control, i.e. $\mathbf{z}$ and $\mathbf{a}$. This approximate posterior is expressed in the following form:

$$\begin{aligned} q(\mathbf{z}_{1:T},&\mathbf{a}_{1:T}|\mathbf{x}_{1:T}, r_{1:T}, \mathcal{O}_{1:T}) \\ &= \prod_{t=1}^{T} q(\mathbf{z}_t|\mathbf{z}_{<t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T}) \\ &\qquad \cdot q(\mathbf{a}_t|\mathbf{z}_{\leq t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T}).\end{aligned} \tag{5}$$

Note that each distribution is conditioned on past and current variables, as well as future optimality observations. In Appendix A, we use this approximate posterior to derive a lower bound on the objective, $\mathcal{L} \leq \log p(\mathcal{O}_{1:T} = \mathbf{1})$, where $\mathcal{L}$ is defined as:

$$\begin{aligned}\mathcal{L} = \mathbb{E}\Bigg[\sum_{t=1}^{T} r_t &- \log \frac{q(\mathbf{z}_t|\mathbf{z}_{<t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T})}{p_{\mathrm{a}}(\mathbf{z}_t|\mathbf{a}_{<t}, \mathbf{z}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t})} \\ &- \log \frac{q(\mathbf{a}_t|\mathbf{z}_{\leq t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T})}{p_{\mathrm{a}}(\mathbf{a}_t|\mathbf{a}_{<t}, \mathbf{z}_{\leq t}, \mathbf{x}_{\leq t}, r_{\leq t})}\Bigg].\end{aligned} \tag{6}$$

The expectation involves sampling $\mathbf{s}$, $\mathbf{x}$, and r from the environment and sampling $\mathbf{z}$ and $\mathbf{a}$ from the agent. This is similar to the "max-entropy" bound derived in (Levine, 2018), but it includes the prior distribution on actions, $p_{\mathrm{a}}(\mathbf{a}_t|\mathbf{a}_{<t}, \mathbf{z}_{\leq t}, \mathbf{x}_{\leq t}, r_{\leq t})$, and additional corresponding terms for $\mathbf{z}$. From the perspective of the agent-environment distribution, $p_{\mathrm{a}}(\mathbf{z}_t|\mathbf{a}_{<t}, \mathbf{z}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t})$ is the prior distribution on $\mathbf{z}_t$. This distribution can be parameterized as a direct, discriminative mapping, which we refer to as a *discriminative agent*. Alternatively, we can use Bayes' rule to invert $p_{\mathrm{a}}(\mathbf{z}_t|\mathbf{a}_{<t}, \mathbf{z}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t})$ into a generative mapping:

$$\begin{aligned} p_{\mathrm{a}}(\mathbf{z}_t&|\mathbf{a}_{<t}, \mathbf{z}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}) \\ &= \frac{p_{\mathrm{a}}(\mathbf{x}_t, r_t|\mathbf{a}_{<t}, \mathbf{z}_{\leq t}, \mathbf{x}_{<t}, r_{<t}) p_{\mathrm{a}}(\mathbf{z}_t|\mathbf{a}_{<t}, \mathbf{z}_{<t}, \mathbf{x}_{<t}, r_{<t})}{p_{\mathrm{a}}(\mathbf{x}_t, r_t|\mathbf{a}_{<t}, \mathbf{z}_{<t}, \mathbf{x}_{<t}, r_{<t})}.\end{aligned} \tag{7}$$

This inversion results in terms modeling the likelihood of observations and rewards, providing an additional learning signal, a more complex internal state posterior, and an internal model that allows for model-based planning. We refer to an agent of this form as a *generative agent*. Substituting Eq. 7 into Eq. 6 and rearranging yields

$$\begin{aligned}\mathcal{L} = \mathbb{E}\Bigg[\sum_{t=1}^{T} r_t &+ \log \frac{p_{\mathrm{a}}(\mathbf{x}_t, r_t|\mathbf{a}_{<t}, \mathbf{z}_{\leq t}, \mathbf{x}_{<t}, r_{<t})}{p_{\mathrm{a}}(\mathbf{x}_t, r_t|\mathbf{a}_{<t}, \mathbf{z}_{<t}, \mathbf{x}_{<t}, r_{<t})} \\ &- \log \frac{q(\mathbf{z}_t|\mathbf{z}_{<t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T})}{p_{\mathrm{a}}(\mathbf{z}_t|\mathbf{a}_{<t}, \mathbf{z}_{<t}, \mathbf{x}_{<t}, r_{<t})} \\ &- \log \frac{q(\mathbf{a}_t|\mathbf{z}_{\leq t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T})}{p_{\mathrm{a}}(\mathbf{a}_t|\mathbf{a}_{<t}, \mathbf{z}_{\leq t}, \mathbf{x}_{\leq t}, r_{\leq t})}\Bigg].\end{aligned} \tag{8}$$

The additional term in Eq. 8 corresponds to the agent's *information gain* in $\mathbf{x}_t$ and r_t after estimating $\mathbf{z}_t$ from q. Maximizing this term encourages observations and rewards to be maximally informative to the agent's model, attempting to increase the information contained in $\mathbf{z}$. Simplified computation graphs for the discriminative and generative agents are shown in Figure 2.

We can optimize the bound using the variational EM algorithm (Dempster et al., 1977; Neal & Hinton, 1998), which alternates between optimizing the approximate posterior, q, during inference (E-step), and the parameters, θ, during learning (M-step). Both optimization procedures can be performed using Monte Carlo estimates of the bound (Ranganath et al., 2014; Hoffman et al., 2013). To simplify notation, let $\mathcal{L}_t$ denote the terms in the bound at time t:

$$\log p(\mathcal{O}_{1:T}) \geq \mathcal{L} = \mathbb{E}\left[\sum_{t=1}^{T} \mathcal{L}_t\right]. \tag{9}$$

For each sampled Monte Carlo trajectory, we optimize the bound sequentially w.r.t. the structured approximate posterior. Each step involves optimizing the approximate posterior distributions over $\mathbf{z}_t$ and $\mathbf{a}_t$, which appear directly in $\mathcal{L}_t$ and indirectly in the expectations around $\mathcal{L}_{t+1:T}$. In general, these optimization procedures involve unrolling the environment to evaluate these future terms. After optimizing the bound w.r.t. q for a trajectory or a mini-batch of trajectories, we can then optimize the bound w.r.t. the agent's parameters, θ. The efficiency of the inference procedures may be improved through amortization (Marino et al., 2018b;a). The overall procedure, which we refer to as Active Variational Expectation Maximization, is provided in Algorithm 1 in Appendix C.

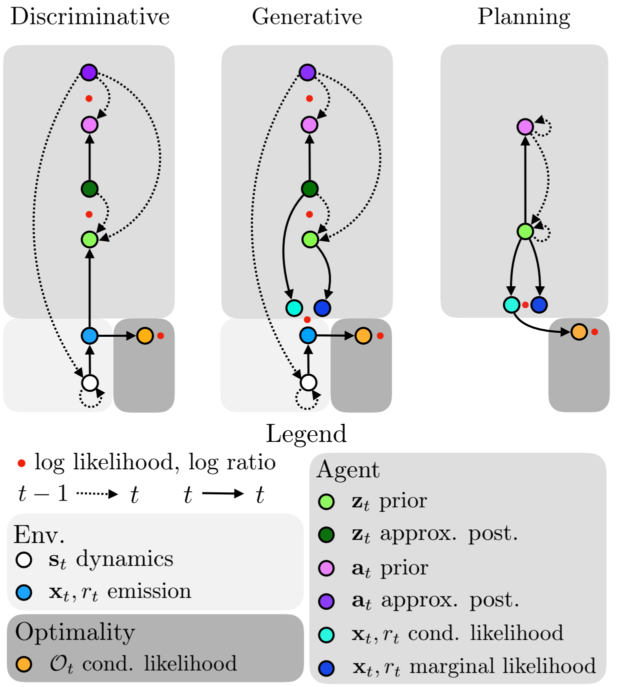
Discriminative
Generative
Planning
Legend
● log likelihood, log ratio
$t-1 \dashrightarrow t$
$t \longrightarrow t$
Env.
○ $\mathbf{s}_t$ dynamics
● $\mathbf{x}_t, r_t$ emission
Optimality
● $\mathcal{O}_t$ cond. likelihood
Agent
● $\mathbf{z}_t$ prior
● $\mathbf{z}_t$ approx. post.
● $\mathbf{a}_t$ prior
● $\mathbf{a}_t$ approx. post.
● $\mathbf{x}_t, r_t$ cond. likelihood
● $\mathbf{x}_t, r_t$ marginal likelihood

Figure 2. **Computation Graphs**. Shaded regions represent the environment, agent, and optimality. Circles with black outlines represent probability distributions. Arrows represent dependencies between time steps (dotted) and within (solid). Red dots represent terms in the objectives. Computation graphs are simplified, with a subset of dependencies. Inference computations are not shown.

3.3. Model-based planning and consolidation

We are often unable to roll out the environment during inference. For a generative agent, we can instead use the agent's distribution, p_a, to estimate and maximize future terms in the bound, thereby *planning*. At time t, we can replace p_e and q in $\mathcal{L}_{t+1:T}$ with p_a over the corresponding variables. The estimated lower bound over $\widehat{\mathcal{L}}_{t+1:T}$ is then:

$$\widehat{\mathcal{L}}_{t+1:T} = \mathbb{E}_{p_\mathrm{a}}\left[\sum_{\tau=t+1}^{T} r_\tau + \log \frac{p_\mathrm{a}(\mathbf{x}_\tau, r_\tau|\mathbf{a}_{<\tau}, \mathbf{z}_{\leq\tau}, \mathbf{x}_{<\tau}, r_{<\tau})}{p_\mathrm{a}(\mathbf{x}_\tau, r_\tau|\mathbf{a}_{<\tau}, \mathbf{z}_{<\tau}, \mathbf{x}_{<\tau}, r_{<\tau})}\right]. \tag{10}$$

During inference, we can maximize $\mathbb{E}[\mathcal{L}_t + \widehat{\mathcal{L}}_{t+1:T}]$ w.r.t. q at each time step, allowing the agent to internally evaluate the outcomes of its actions and internal state estimates under its own model. As with conventional approaches to planning, this estimated objective contains the predicted reward. However, the estimated objective also includes the *mutual information* between the internal state and the observations and reward. This term should bias planning toward states with lower epistemic uncertainty. An unrolled planning computation graph is shown in Figure 3 in Appendix C.

Through learning, the log ratio terms on $\mathbf{z}$ and $\mathbf{a}$ in the objective (Eq. 6 and 8) will bring the priors on these variables toward their approximate posteriors. Thus, model-based approximate posteriors will be consolidated into model-free priors. This provides a mechanism for converting goal-directed behavior into habits (Dickinson, 1985). After sufficient training of the priors, the agent can operate in a more efficient model-free manner. We also note that during planning (Eq. 10), actions are sampled according to the agent's prior, serving as a roll-out policy, cf. (Silver et al., 2016).

4. Discussion

We now contrast the inference perspective presented here with recent approaches to model-based RL. Multiple works have combined sequential latent variable models, using variational inference, with RL, e.g. (Buesing et al., 2018; Igl et al., 2018; Ha & Schmidhuber, 2018; Hafner et al., 2019; Zhang et al., 2019). In comparison with the objective in Eq. 8, these formulations do not contain the negative marginal log-likelihood of observations and reward, effectively maximizing reward *and* learning an environment model. This highlights an inconsistency between current model-based and model-free objectives. While reward maximization and model learning are often complementary, they may be at odds in complex environments. Including this term forces the agent's model to retain only task-relevant information. This presents a future research direction, investigating the task-specificity of learned models and their effect on asymptotic task performance. Optimizing this term may pose challenges, requiring approximations (Burda et al., 2016) and annealing schemes (Bowman et al., 2016).

Unlike previous formulations, the planning objective in Eq. 10 contains the mutual information between the internal state and the observations and reward, effectively downweighting uncertain outcomes. This presents another research direction, investigating the effect of this term on the quality and stability of planning optimization.

Our formulation relies on latent variables, $\mathbf{z}$, enabling the agent to maintain a stochastic internal state. While previous works have looked at adding latent variables to the policy (Gupta et al., 2018; Hausman et al., 2018; Haarnoja et al., 2018a), these approaches were considered in the context of meta-RL or hierarchical RL.

While a uniform action prior (Levine, 2018), as used in max-entropy RL, can have a stabilizing effect, a learned prior may provide further improvement (Shvechikov et al., 2018). We have described how this prior provides a principled Bayesian mechanism for consolidating model-based planning into a model-free policy (Weber et al., 2017; Nagabandi et al., 2018; Kurutach et al., 2018; Buesing et al., 2018). This presents yet another direction for further investigation.

References

Attias, H. Planning by probabilistic inference. In *AISTATS*. Citeseer, 2003.

Bansal, S., Calandra, R., Chua, K., Levine, S., and Tomlin, C. Mbmf: model-based priors for model-free reinforcement learning. *arXiv preprint arXiv:1709.03153*, 2017.

Botvinick, M. and Toussaint, M. Planning as inference. *Trends in cognitive sciences*, 16(10):485–488, 2012.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. Generating sentences from a continuous space. 2016.

Buesing, L., Weber, T., Racaniere, S., Eslami, S., Rezende, D., Reichert, D. P., Viola, F., Besse, F., Gregor, K., Hassabis, D., et al. Learning and querying fast generative models for reinforcement learning. *arXiv preprint arXiv:1802.03006*, 2018.

Burda, Y., Grosse, R., and Salakhutdinov, R. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2016.

Chebotar, Y., Hausman, K., Zhang, M., Sukhatme, G., Schaal, S., and Levine, S. Combining model-based and model-free updates for trajectory-centric reinforcement learning. In *International Conference on Machine Learning*, pp. 703–711, 2017.

Chua, K., Calandra, R., McAllister, R., and Levine, S. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, 2018.

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., and Bengio, Y. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pp. 2980–2988, 2015.

Co-Reyes, J. D., Liu, Y., Gupta, A., Eysenbach, B., Abbeel, P., and Levine, S. Self-consistent trajectory autoencoder: Hierarchical reinforcement learning with trajectory embeddings. *arXiv preprint arXiv:1806.02813*, 2018.

Cooper, G. F. A method for using belief networks as influence diagrams. In *Fourth Workshop on Uncertainty in Artificial Intelligence.*, 1988.

Deisenroth, M. P., Fox, D., and Rasmussen, C. E. Gaussian processes for data-efficient learning in robotics and control. *IEEE transactions on pattern analysis and machine intelligence*, 37(2):408–423, 2015.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

Dickinson, A. Actions and habits: the development of behavioural autonomy. *Phil. Trans. R. Soc. Lond. B*, 308 (1135):67–78, 1985.

Eslami, S. A., Rezende, D. J., Besse, F., Viola, F., Morcos, A. S., Garnelo, M., Ruderman, A., Rusu, A. A., Danihelka, I., Gregor, K., et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.

Fraccaro, M., Sønderby, S. K., Paquet, U., and Winther, O. Sequential neural models with stochastic layers. In *Advances in neural information processing systems*, pp. 2199–2207, 2016.

Gemici, M., Hung, C.-C., Santoro, A., Wayne, G., Mohamed, S., Rezende, D. J., Amos, D., and Lillicrap, T. Generative temporal models with memory. *arXiv preprint arXiv:1702.04649*, 2017.

Goyal, A. G. A. P., Sordoni, A., Côté, M.-A., Ke, N. R., and Bengio, Y. Z-forcing: Training stochastic recurrent networks. In *Advances in neural information processing systems*, pp. 6713–6723, 2017.

Gu, S., Lillicrap, T., Sutskever, I., and Levine, S. Continuous deep q-learning with model-based acceleration. In *International Conference on Machine Learning*, pp. 2829–2838, 2016.

Gupta, A., Mendonca, R., Liu, Y., Abbeel, P., and Levine, S. Meta-reinforcement learning of structured exploration strategies. In *Advances in Neural Information Processing Systems*, pp. 5307–5316, 2018.

Ha, D. and Schmidhuber, J. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, 2018.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, 2017.

Haarnoja, T., Hartikainen, K., Abbeel, P., and Levine, S. Latent space policies for hierarchical reinforcement learning. In *International Conference on Machine Learning*, pp. 1846–1855, 2018a.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, 2018b.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, 2019.

Hausman, K., Springenberg, J. T., Wang, Z., Heess, N., and Riedmiller, M. Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations*, 2018.

Heess, N., Wayne, G., Silver, D., Lillicrap, T., Erez, T., and Tassa, Y. Learning continuous control policies by stochastic value gradients. In *Advances in Neural Information Processing Systems*, pp. 2944–2952, 2015.

Henaff, M., Whitney, W. F., and LeCun, Y. Model-based planning with discrete and continuous actions. *arXiv preprint arXiv:1705.07177*, 2017.

Hoffman, M., Freitas, N., Doucet, A., and Peters, J. An expectation maximization algorithm for continuous markov decision processes with arbitrary reward. In *Artificial Intelligence and Statistics*, pp. 232–239, 2009.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Igl, M., Zintgraf, L., Le, T. A., Wood, F., and Whiteson, S. Deep variational reinforcement learning for pomdps. In *International Conference on Machine Learning (ICML)*, 2018.

Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., and Kavukcuoglu, K. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *NATO ASI SERIES D BEHAVIOURAL AND SOCIAL SCIENCES*, 89:105–162, 1998.

Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019.

Kingma, D. P. and Welling, M. Stochastic gradient vb and the variational auto-encoder. In *International Conference on Learning Representations*, 2014.

Kurutach, T., Clavera, I., Duan, Y., Tamar, A., and Abbeel, P. Model-ensemble trust-region policy optimization. In *International Conference on Learning Representations*, 2018.

Lange, S. and Riedmiller, M. Deep auto-encoder neural networks in reinforcement learning. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2010.

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

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Marino, J., Cvitkovic, M., and Yue, Y. A general method for amortizing variational filtering. In *Advances in Neural Information Processing Systems*, pp. 7867–7878, 2018a.

Marino, J., Yue, Y., and Mandt, S. Iterative amortized inference. In *International Conference on Machine Learning*, pp. 3400–3409, 2018b.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529, 2015.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937, 2016.

Nagabandi, A., Kahn, G., Fearing, R. S., and Levine, S. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7559–7566. IEEE, 2018.

Neal, R. M. and Hinton, G. E. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pp. 355–368. Springer, 1998.

Oh, J., Singh, S., and Lee, H. Value prediction network. In *Advances in Neural Information Processing Systems*, pp. 6118–6128, 2017.

Piché, A., Thomas, V., Ibrahim, C., Bengio, Y., and Pal, C. Probabilistic planning with sequential monte carlo methods. In *International Conference on Learning Representations*, 2019.

Pong, V., Gu, S., Dalal, M., and Levine, S. Temporal difference models: Model-free deep rl for model-based control. In *International Conference on Learning Representations*, 2018.

Ranganath, R., Gerrish, S., and Blei, D. Black box variational inference. In *Artificial Intelligence and Statistics*, 2014.

Rawlik, K., Toussaint, M., and Vijayakumar, S. On stochastic optimal control and reinforcement learning by approximate inference. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

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

Schulman, J., Chen, X., and Abbeel, P. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*, 2017a.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017b.

Shvechikov, P., Grishin, A., Kuznetsov, A., Fritzler, A., and Vetrov, D. Joint belief tracking and reward optimization through approximate inference. In *NeurIPS Workshop on Reinforcement Learning under Partial Observability*, 2018.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

Sutton, R. S. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine Learning Proceedings 1990*, pp. 216–224. Elsevier, 1990.

Todorov, E. General duality between optimal control and estimation. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pp. 4286–4292. IEEE, 2008.

Toussaint, M. Robot trajectory optimization using approximate inference. In *International conference on machine learning*, pp. 1049–1056, 2009.

Toussaint, M. and Storkey, A. Probabilistic inference for solving discrete and continuous state markov decision processes. In *Proceedings of the 23rd international conference on Machine learning*, pp. 945–952. ACM, 2006.

Weber, T., Racanière, S., Reichert, D. P., Buesing, L., Guez, A., Rezende, D. J., Badia, A. P., Vinyals, O., Heess, N., Li, Y., et al. Imagination-augmented agents for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, 2017.

Zhang, M., Vikram, S., Smith, L., Abbeel, P., Johnson, M. J., and Levine, S. Solar: Deep structured latent representations for model-based reinforcement learning. In *International Conference on Machine Learning*, 2019.

A. Lower bound derivation

We start by writing the joint distribution over all variables from the agent and environment distributions:

$$p(\mathbf{x}_{1:T}, r_{1:T}, \mathbf{s}_{1:T}, \mathbf{a}_{1:T}, \mathbf{z}_{1:T}, \mathcal{O}_{1:T}) = \prod_{t=1}^{T} p_{\text{a}}(\mathbf{a}_t | \mathbf{a}_{<t}, \mathbf{z}_{\leq t}, \mathbf{x}_{\leq t}, r_{\leq t}) p_{\text{a}}(\mathbf{z}_t | \mathbf{a}_{<t}, \mathbf{z}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}) \cdot p_{\text{e}}(\mathbf{s}_t | \mathbf{s}_{t-1}, \mathbf{a}_{t-1}) p_{\text{e}}(\mathbf{x}_t | \mathbf{s}_t) p_{\text{e}}(r_t | \mathbf{s}_t) p(\mathcal{O}_t | r_t). \tag{11}$$

We will bound the log marginal likelihood of optimality being 1, using an approximate posterior, q, over variables for which we can control sampling, i.e. actions and model latent variables:

$$q(\mathbf{z}_{1:T}, \mathbf{a}_{1:T} | \mathbf{x}_{1:T}, r_{1:T}, \mathcal{O}_{1:T}) = \prod_{t=1}^{T} q(\mathbf{z}_t | \mathbf{z}_{<t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T}) \cdot q(\mathbf{a}_t | \mathbf{z}_{\leq t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T}). \tag{12}$$

We first write the marginal likelihood as the marginalization of the joint distribution (Eq. 11):

$$\begin{aligned}
p(\mathcal{O}_{1:T}) &= \int p(\mathbf{x}_{1:T}, r_{1:T}, \mathbf{s}_{1:T}, \mathbf{a}_{1:T}, \mathbf{z}_{1:T}, \mathcal{O}_{1:T}) d\mathbf{x}_{1:T} dr_{1:T} d\mathbf{s}_{1:T} d\mathbf{a}_{1:T} d\mathbf{z}_{1:T} \\
&= \int p(\mathbf{x}_{1:T}, r_{1:T}, \mathbf{s}_{1:T}, \mathbf{a}_{1:T}, \mathbf{z}_{1:T}, \mathcal{O}_{1:T}) \\
&\qquad \cdot \frac{q(\mathbf{z}_{1:T}, \mathbf{a}_{1:T} | \mathbf{x}_{1:T}, r_{1:T}, \mathcal{O}_{1:T})}{q(\mathbf{z}_{1:T}, \mathbf{a}_{1:T} | \mathbf{x}_{1:T}, r_{1:T}, \mathcal{O}_{1:T})} d\mathbf{x}_{1:T} dr_{1:T} d\mathbf{s}_{1:T} d\mathbf{a}_{1:T} d\mathbf{z}_{1:T} \\
&= \int \prod_{t=1}^{T} p_{\text{a}}(\mathbf{a}_t | \mathbf{a}_{<t}, \mathbf{z}_{\leq t}, \mathbf{x}_{\leq t}, r_{\leq t}) p_{\text{a}}(\mathbf{z}_t | \mathbf{a}_{<t}, \mathbf{z}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}) p_{\text{e}}(\mathbf{s}_t | \mathbf{s}_{t-1}, \mathbf{a}_{t-1}) \\
&\qquad \cdot p_{\text{e}}(\mathbf{x}_t | \mathbf{s}_t) p_{\text{e}}(r_t | \mathbf{s}_t) p(\mathcal{O}_t | r_t) \frac{q(\mathbf{z}_t | \mathbf{z}_{<t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T})}{q(\mathbf{z}_t | \mathbf{z}_{<t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T})} \\
&\qquad \cdot \frac{q(\mathbf{a}_t | \mathbf{z}_{\leq t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T})}{q(\mathbf{a}_t | \mathbf{z}_{\leq t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T})} d\mathbf{x}_{1:T} dr_{1:T} d\mathbf{s}_{1:T} d\mathbf{a}_{1:T} d\mathbf{z}_{1:T} \\
&= \mathbb{E}_{p_{\text{e}}(\mathbf{s}_1, \mathbf{x}_1, r_1)} \mathbb{E}_{q(\mathbf{z}_1|\cdot)} \mathbb{E}_{q(\mathbf{a}_1|\cdot)} \mathbb{E}_{p_{\text{e}}(\mathbf{s}_2 | \mathbf{a}_1, \mathbf{s}_1)} \cdots \left[\prod_{t=1}^{T} p(\mathcal{O}_t | r_t) \frac{p_{\text{a}}(\mathbf{z}_t | \mathbf{a}_{<t}, \mathbf{z}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t})}{q(\mathbf{z}_t | \mathbf{z}_{<t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T})} \right. \\
&\qquad \left. \cdot \frac{p_{\text{a}}(\mathbf{a}_t | \mathbf{a}_{<t}, \mathbf{z}_{\leq t}, \mathbf{x}_{\leq t}, r_{\leq t})}{q(\mathbf{a}_t | \mathbf{z}_{\leq t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T})} \right]
\end{aligned}$$

To obtain the log marginal likelihood, we take the log of both sides. Bringing the log inside of the expectations forms a lower bound, $\mathcal{L}$, using Jensen's inequality:

$$\log p(\mathcal{O}_{1:T}) \geq \mathcal{L} = \mathbb{E}_{\substack{\mathbf{s},\mathbf{x},r \sim p_{\text{e}} \\ \mathbf{z},\mathbf{a} \sim q}} \left[\sum_{t=1}^{T} \log p(\mathcal{O}_t | r_t) - \log \frac{q(\mathbf{z}_t | \mathbf{z}_{<t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T})}{p_{\text{a}}(\mathbf{z}_t | \mathbf{a}_{<t}, \mathbf{z}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t})} - \log \frac{q(\mathbf{a}_t | \mathbf{z}_{\leq t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T})}{p_{\text{a}}(\mathbf{a}_t | \mathbf{a}_{<t}, \mathbf{z}_{\leq t}, \mathbf{x}_{\leq t}, r_{\leq t})} \right]. \tag{13}$$

Evaluating the log marginal likelihood at $\mathcal{O}_t = 1$ for all time steps yields:

$$\log p(\mathcal{O}_{1:T} = \mathbf{1}) \geq \mathbb{E}_{\substack{\mathbf{s},\mathbf{x},r \sim p_{\text{e}} \\ \mathbf{z},\mathbf{a} \sim q}} \left[\sum_{t=1}^{T} r_t - \log \frac{q(\mathbf{z}_t | \mathbf{z}_{<t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T})}{p_{\text{a}}(\mathbf{z}_t | \mathbf{a}_{<t}, \mathbf{z}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t})} - \log \frac{q(\mathbf{a}_t | \mathbf{z}_{\leq t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T})}{p_{\text{a}}(\mathbf{a}_t | \mathbf{a}_{<t}, \mathbf{z}_{\leq t}, \mathbf{x}_{\leq t}, r_{\leq t})} \right]. \tag{14}$$

We can parameterize $p_{\text{a}}(\mathbf{z}_t|\mathbf{a}_{<t}, \mathbf{z}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t})$ as a direct, discriminative mapping. Alternatively, we can parameterize the inverse, generative mapping, using Bayes' rule to express

$$p_{\text{a}}(\mathbf{z}_t|\mathbf{a}_{<t}, \mathbf{z}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}) = \frac{p_{\text{a}}(\mathbf{x}_t, r_t|\mathbf{a}_{<t}, \mathbf{z}_{\leq t}, \mathbf{x}_{<t}, r_{<t}) p_{\text{a}}(\mathbf{z}_t|\mathbf{a}_{<t}, \mathbf{z}_{<t}, \mathbf{x}_{<t}, r_{<t})}{p_{\text{a}}(\mathbf{x}_t, r_t|\mathbf{a}_{<t}, \mathbf{z}_{<t}, \mathbf{x}_{<t}, r_{<t})}. \tag{15}$$

Plugging this expression into the bound yields

$$\begin{aligned}\mathcal{L} = \mathbb{E}_{\substack{\mathbf{s},\mathbf{x},r \sim p_{\text{e}} \\ \mathbf{z},\mathbf{a} \sim q}} \Bigg[\sum_{t=1}^{T} r_t + \log \frac{p_{\text{a}}(\mathbf{x}_t, r_t|\mathbf{a}_{<t}, \mathbf{z}_{\leq t}, \mathbf{x}_{<t}, r_{<t})}{p_{\text{a}}(\mathbf{x}_t, r_t|\mathbf{a}_{<t}, \mathbf{z}_{<t}, \mathbf{x}_{<t}, r_{<t})} - \log \frac{q(\mathbf{z}_t|\mathbf{z}_{<t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T})}{p_{\text{a}}(\mathbf{z}_t|\mathbf{a}_{<t}, \mathbf{z}_{<t}, \mathbf{x}_{<t}, r_{<t})} \\ - \log \frac{q(\mathbf{a}_t|\mathbf{z}_{\leq t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T})}{p_{\text{a}}(\mathbf{a}_t|\mathbf{a}_{<t}, \mathbf{z}_{\leq t}, \mathbf{x}_{\leq t}, r_{\leq t})} \Bigg].\end{aligned} \tag{16}$$

B. Active variational EM algorithm

Algorithm 1: Active Variational Expectation Maximization (EM)

```
Input: Agent-Environment Distribution $p(\mathbf{x}_{1:T}, r_{1:T}, \mathbf{s}_{1:T}, \mathbf{a}_{1:T}, \mathbf{z}_{1:T}, \mathcal{O}_{1:T})$
while $\theta$ not converged do
    $\mathbf{x}_1, r_1, \mathbf{s}_1 \sim p_{\text{e}}(\mathbf{x}_1|\mathbf{s}_1) p_{\text{e}}(r_1|\mathbf{s}_1) p_{\text{e}}(\mathbf{s}_1)$
    for $t = 1 \ldots T$ do
        # E-step
        # internal state inference
        $q(\mathbf{z}_t|\mathbf{z}_{<t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T}) \leftarrow \arg\max_q \mathbb{E}_{p_{\text{e}},q}[\mathcal{L}_{t:T}]$
        $\mathbf{z}_t \sim q(\mathbf{z}_t|\mathbf{z}_{<t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T})$
        # action inference
        $q(\mathbf{a}_t|\mathbf{z}_{\leq t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T}) \leftarrow \arg\max_q \mathbb{E}_{p_{\text{e}},q}[\mathcal{L}_{t:T}]$
        $\mathbf{a}_t \sim q(\mathbf{a}_t|\mathbf{z}_{\leq t}, \mathbf{a}_{<t}, \mathbf{x}_{\leq t}, r_{\leq t}, \mathcal{O}_{t+1:T})$
        # environment interaction
        $\mathbf{x}_{t+1}, r_{t+1}, \mathbf{s}_{t+1} \sim p_{\text{e}}(\mathbf{x}_{t+1}|\mathbf{s}_{t+1}) p_{\text{e}}(r_{t+1}|\mathbf{s}_{t+1}) p_{\text{e}}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$
    end
    # learning (M-step)
    $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{L}$
end
```

C. Planning

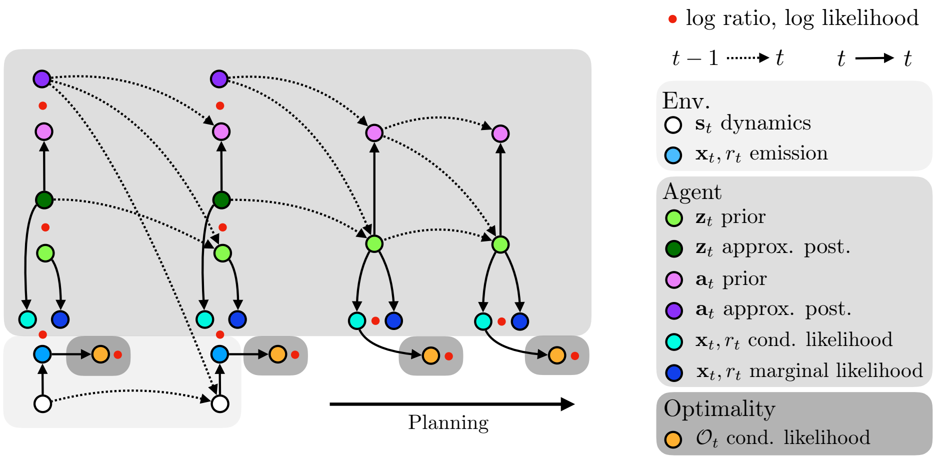
● log ratio, log likelihood
$t-1 \dashrightarrow t$
$t \longrightarrow t$
Env.
○ $\mathbf{s}_t$ dynamics
● $\mathbf{x}_t, r_t$ emission
Agent
● $\mathbf{z}_t$ prior
● $\mathbf{z}_t$ approx. post.
● $\mathbf{a}_t$ prior
● $\mathbf{a}_t$ approx. post.
● $\mathbf{x}_t, r_t$ cond. likelihood
● $\mathbf{x}_t, r_t$ marginal likelihood
Optimality
● $\mathcal{O}_t$ cond. likelihood
Planning

Figure 3. **Unrolled action planning computation graph**. During planning, we replace the future environment, p_{e}, and approximate posterior distributions, q, with the agent's distributions, p_{a}. Rolling out the agent's model provides an estimate of future terms in the objective, which can be used to infer the action approximate posterior.