Published in the proceedings of the Workshop on "Structure & Priors in Reinforcement Learning" at ICLR 2019

VARIATIONAL TASK EMBEDDINGS FOR FAST ADAPTATION IN DEEP REINFORCEMENT LEARNING

Luisa Zintgraf * University of Oxford
Maximilian Igl University of Oxford
Kyriacos Shiarlis Latent Logic
Anuj Mahajan University of Oxford
Katja Hofmann Microsoft Research
Shimon Whiteson University of Oxford, Latent Logic

ABSTRACT

We introduce VATE: a method for fast adaptation in multi-task reinforcement learning settings, where different tasks are different Markov Decision Processes (MDPs) with different reward and/or transition dynamics. We represent an MDP using a stochastic latent variable, and learn to perform approximate inference over this embedding given the agent's experience in an MDP. This posterior expresses a belief over which MDP the policy is in, and can be used by the policy to trade off exploration and exploitation when selecting actions. We present preliminary results showing that VATE outperforms existing approaches, and that the agent uses the approximate task posterior to explore systematically in a gridworld.

1 INTRODUCTION

One key challenge in reinforcement learning (RL) is *sample efficiency*: learning new tasks fast with only few environment interactions. Many successful RL methods require vast amounts of training data and are tailored to a single task, such as playing Atari games (Mnih et al., 2013) or manoeuvring a helicopter (Abbeel et al., 2007). A promising approach to producing more versatile RL agents is meta-learning, or *learning to learn* (Thrun & Pratt, 1998; Schmidhuber, 1987). Here, learning takes place at two different time scales: fast learning on individual tasks, and slow meta-learning across tasks. This enables learning how to *quickly* adapt to a previously unseen task with *little data*.

The optimal action-selection strategy that trades off exploration and exploitation can in principle be computed within the framework of Bayes Adaptive Markov Decision Problems (BAMDPs) (Duff & Barto, 2002). In BAMDPs, the goal is to *learn a distribution over the task, and given a posteriori knowledge of the task, compute an optimal action*. This, however, is hopelessly intractable for most problems. In this paper, we apply the idea of learning a distribution over MDPs in a meta-learning setting. We represent the MDP using a learned, low-dimensional stochastic latent variable, called *variational task embedding* (VATE). Because we grant the policy access to the posterior distribution over those task embeddings, it can learn to take the optimal action *under task uncertainty*.

VATE updates this approximate posterior *online* while the agent interacts with the environment, allowing it to switch naturally between exploration and exploitation. This stands in contrast to methods with separate phases for data collection, policy update, and execution (such as MAML (Finn et al., 2017)). Notably, in a sparse reward setting, VATE is able to start inferring information about the task before collecting rewards.

On a grid-world environment, we illustrate how VATE performs approximate inference over which task it is in, *online* while interacting with the environment. We visualise how the agent uses the approximate task posterior to explore the task space systematically, and show that VATE outperforms existing methods that rely on recurrent policies (Wang et al., 2016; Duan et al., 2016).

*Correspondence to luisa.zintgraf@cs.ox.ac.uk

2 PROBLEM SETTING

We define a single reinforcement learning task as a Markov Decision Process (MDP) $M = (\mathcal{S}, \mathcal{A}, R, T, T_0)$, where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $R(r_{t+1}|s_t, a_t, s_{t+1})$ is a reward function, $T(s_{t+1}|s_t, a_t)$ is a transition function, and $T_0(s_0)$ is an initial state distribution. In a single task setting, the goal is to learn a policy π that maps states $s \in \mathcal{S}$ to actions $a \in \mathcal{A}$ in order to maximise the expected cumulative reward $\mathcal{J}$ under π, $\mathcal{J}(\pi) = \mathbb{E}_{T_0,T,\pi}\left[\sum_{t=0}^{H} \gamma^t R(r_{t+1}|s_t, a_t, s_{t+1})\right]$, where $H \in \mathbb{N}$ is the horizon and $\gamma \in [0, 1]$ is the discount factor. In this paper, we consider a multi-task setting defined by a distribution over MDPs, $p(M)$, where for each $M_i \sim p(M)$ we have $M_i = (\mathcal{S}, \mathcal{A}, R_i, T_i, T_{i,0})$. This distribution is not explicitly known, but we can sample from it to train our policy. The reward and/or transition function can vary across tasks, and typically some structure is shared across tasks. In the multi-task setting, we seek a policy π that can adapt quickly to any of the MDPs from the distribution $p(M)$.

3 VATE

In principle, the exploration-exploitation issue can be settled by taking a Bayesian approach to Reinforcement Learning (Bellman, 1956; Duff & Barto, 2002). Here, an explicit posterior over model parameters is maintained and used for action selection. We make use of this idea in the following.

We choose to represent an MDP M_i by a stochastic latent variable m_i called a *variational task embedding* (VATE). We choose m_i such that we can have transition and reward functions that are *shared* across MDPs, so long as they are conditioned on an MDP's embedding m_i. For a given MDP M_i we thus rewrite

$$R_i(r_{t+1}|s_t, a_t, s_{t+1}) \equiv R'(r_{t+1}|s_t, a_t, s_{t+1}; m_i), \tag{1}$$

$$T_i(s_{t+1}|s_t, a_t) \equiv T'(s_{t+1}|s_t, a_t; m_i), \tag{2}$$

where R' and T' are generalised reward and transition functions that are shared across tasks. In many problems, this embedding has a particular meaning (e.g., a goal position in a maze, or leg length of a humanoid). If we had access to the true m_i, we could train a policy that conditions on this embedding, and is otherwise fixed across tasks.

Instead, we need to *infer* m_i given the agent's experience up until time step t collected in the MDP M_i,

$$\tau_{:t}^{(i)} = (s_0, a_0, r_1, s_1, a_1, r_2, \ldots, s_{t-1}, a_{t-1}, r_t, s_t), \tag{3}$$

i.e., we want to infer the posterior distribution $p(m_i|\tau_{:t}^{(i)})$ over m_i given $\tau_{:t}^{(i)}$. In the following, we will drop the sub- and superscript i for ease of notation.

Recall that our goal is to *learn a distribution over the environment, and given a posteriori knowledge of the environment compute the optimal action*. Given the above reformulation, it is now sufficient to reason about the embedding m, instead of the transition and reward dynamics. Given the posterior distribution the policy can (in theory) optimally trade off exploration and exploitation.

3.1 HYPER-STATES

Given the above formulation, we augment the state s_t at time step t with the posterior over m (e.g., for a normal distribution by representing it by its mean and standard deviation), such that $\pi(a_t|s_t, q(m|\tau_{:t}))$. This formulation is similar to that of the Bayes-Adaptive MDP (Duff & Barto, 2002), with the difference that we learn a distribution over MDP embeddings, instead of the transition/reward function directly. This makes learning easier since the number of parameters to perform inference over is smaller, and we can use data from all tasks to learn the shared reward and transition function. We call augmented states $[s_t, q(m|\tau_{:t})]$ *hyper-states* following the BAMDP literature.

3.2 APPROXIMATE INFERENCE

Estimating the true posterior is typically not possible: we do not have access to the MDP (and hence the transition and reward function), and marginalising over tasks is computationally infeasible.

Consequently, we need to learn a model of the environment $p_\theta(m, \tau_{:H}|a_{:H-1})$, parameterised by θ, together with an amortised inference network $q_\phi(m|\tau_{:t})$, parameterised by ϕ, which allows fast inference at runtime at each timestep t. Note that the action-selection policy is not part of the MDP, so an environmental model can only give rise to a distribution of trajectories when conditioned on actions. We typically draw $a_t \sim \pi$ from our current policy. Our model learning objective is thus

$$\mathbb{E}_{\rho(M,\tau_{:H})}\left[\log p_\theta(\tau_{:H}|a_{:H-1})\right] \tag{4}$$

where $\rho(M, \tau_{:H})$ is the trajectory distribution induced by our action-sampling policy in the environment and we slightly abuse notation by denoting by τ the state-reward trajectories, excluding the actions. In the following, we will drop the conditioning on $a_{:H-1}$ to unclutter notation. To optimise Equation (4), which is intractable, we can use a learned approximate posterior $q_\phi(m|\tau_{:t})$ to find a lower bound which can be estimated by Monte Carlo sampling:

$$\begin{aligned}
\mathbb{E}_{\rho(M,\tau_{:H})}\left[\log p_\theta(\tau_{:H})\right] &= \mathbb{E}_\rho\left[\frac{1}{H}\sum_{t=0}^{H}\mathbb{E}_{q_\phi(m|\tau_{:t})}\left[\log\frac{p_\theta(\tau_{:H},m)q_\phi(m|\tau_{:t})}{p_\theta(m|\tau_{:H})q_\phi(m|\tau_{:t})}\right]\right] \\
&= \mathbb{E}_\rho\left[\frac{1}{H}\sum_{t=0}^{H} KL\left(q_\phi(m|\tau_{:t})\|p_\theta(m|\tau_{:H})\right) + \mathbb{E}_{q_t}\left[\log p_\theta(\tau_{:H}|m)\right] - KL\left(q_\phi(m|\tau_{:t})\|p_\theta(m)\right)\right] \\
&\geq \mathbb{E}_\rho\left[\frac{1}{H}\sum_{t=0}^{H}\mathbb{E}_{q_t}\left[\log p_\theta(\tau_{:H}|m)\right] - KL\left(q_\phi(m|\tau_{:t})\|p_\theta(m)\right)\right] = \mathbb{E}_\rho\left[\frac{1}{H}\sum_{t=0}^{H} ELBO_t\right].
\end{aligned} \tag{5}$$

The inequality in the last line arises due to the KL-divergence being non-negative with equality if and only if $q_\phi(m|\tau_{:t})$ matches the true posterior $p_\theta(m|\tau_{:H})$ for all t. Since the resulting sum over *evidence lower bounds* (ELBOs) is a lower bound to the original objective, it can be used as an optimisation target instead. The term $\mathbb{E}_q[\log p(\tau_{:H}|m)]$ is often referred to as the reconstruction loss, and $p(\tau_{:t}|m)$ as the decoder. The term $KL(q(m|\tau_{:t})||p_\theta(m))$ is the KL divergence between our variational posterior q_ϕ and the prior over the embeddings $p_\theta(m)$. For sufficiently expressive decoders, we are free to choose $p_\theta(m)$. We follow the widely used approach of setting $p_\theta(m) = \mathcal{N}(0, I)$.

The reconstruction term $\log p(\tau_{:H}|m)$ factorises as

$$\begin{aligned}
\log p(\tau_{:H}|m, a_{:H-1}) &= \log p((s_0, r_0, \ldots, s_{t-1}, r_{t-1}, s_t)|m, a_{:H-1}) \\
&= \log p(s_0|m) + \sum_{i=0}^{H-1}\left[\log p(s_{i+1}|s_i, a_i, m) + \log p(r_{i+1}|s_i, a_i, s_{i+1}, m)\right].
\end{aligned} \tag{6}$$

Here, $p(s_0|m)$ is the initial state distribution T'_0, $p(s_{i+1}|s_i, a_i; m)$ the transition function T', and $p(r_{i+1}|s_t, a_t, s_{i+1}; m)$ the reward function R'. From now, we include T'_0 in T' for ease of notation.

3.3 Training Objective

We can now formulate a training objective which allows us to learn the approximate posterior distribution over task embeddings, the policy and the generalised reward and transition functions R' and T'. We use deep neural networks to represent the individual components. These are:

1. The encoder $q_\phi(m|\tau_{:t})$, parameterised by ϕ.
2. An approximate transition function $T'_\theta(s_{i+1}|s_i, a_i; m)$ and an approximate reward function $R_\theta(r_{i+1}|s_t, a_t, s_{i+1}; m)$ which are jointly parameterised by θ.
3. A policy $\pi_\psi(a_t|s_t, q_\phi(m|\tau_{:t}))$ parameterised by ψ (note the dependency on ϕ).

Our overall objective is to maximise

$$\mathcal{L}(\phi, \theta, \psi) = \mathbb{E}_{p(M)}\left[\mathcal{J}(\psi, \phi) + \lambda\mathbb{E}_\rho\sum_{t=0}^{H} ELBO_t(\phi, \theta)\right]. \tag{7}$$

The parameter λ weights the supervised model learning objective against the RL loss. This is necessary because parameters ϕ are shared between the model and the policy. Expectations are approximated by Monte Carlo samples, and the ELBO can be optimised using the reparameterisation trick (Kingma & Welling, 2013). The network architecture is shown in Figure 1.

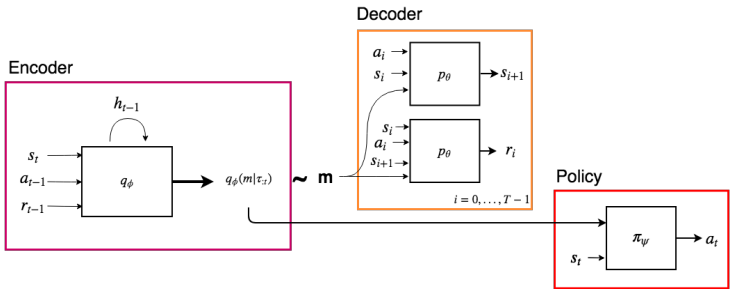


Figure 1: VATE: A trajectory of states, actions and rewards is processed online using an RNN to produce the posterior over task embeddings, $q_\phi(m|\tau)$. The posterior is trained using a decoder which attempts to predict future states and rewards from current states. The policy conditions on the posterior in order to act in the environment and is trained using RL.

4 Related Work

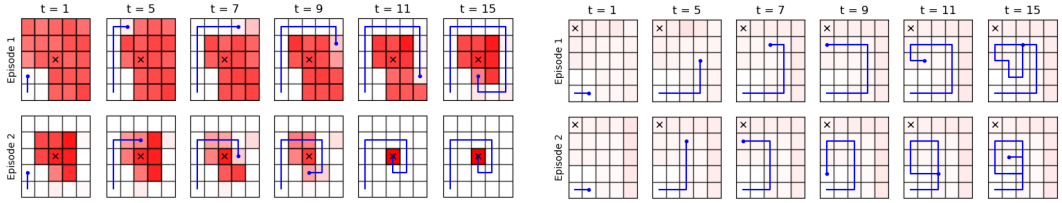
Meta Reinforcement Learning. A prominent model-free approach to meta-learning is to utilise the dynamics of recurrent neural networks for fast adaptation, an idea concurrently proposed by Wang et al. (2016) and Duan et al. (2016) for reinforcement learning. At every time step, the network gets an auxiliary input indicating the agent's action and received reward of the preceding step. This allows learning within a task to happen online, entirely in the dynamics of the recurrent network. Compared to VATE, there is no stochastic latent variable, and no decoder part to reconstruct the transition and reward function. We expect that learning a *distribution* over task embedding can help the policy better trade off exploration and exploitation. The decoder can further act as an auxiliary task which helps learning (Jaderberg et al., 2016). Another popular approach to meta RL is to learn an initialisation of the network parameters, such that at test time, only a few gradient steps are necessary to achieve good performance (Finn et al., 2017; Nichol & Schulman, 2018). A recent extension accounts for the fact that the initial policy needs to explore (Stadie et al., 2018).

Skill / Task Embeddings. Learning task or skill embeddings for meta / transfer reinforcement learning has been done in a variety of settings. Hausman et al. (2018) learn an embedding space of skills using approximate variational inference (using a different lower bound than VATE). At test time the policy is fixed, while a new embedder is learned which can interpolate between already learned skills. Arnekvist et al. (2018) learn a stochastic embedding of optimal Q-functions for different skills, such that the policy can be conditioned on (samples of) this embedding. When learning a new task, adaptation has to be done only in latent space. Co-Reyes et al. (2018) uses the idea of encoding skill embeddings in the setting of hierarchical reinforcement learning. They learn a latent space of low-level skills which can be controlled by a higher-level controller. This embedding is learned using a VAE to encode state trajectories and decode states and actions. In imitation learning, an expert demonstration can be embedded to represent the task, e.g., Wang et al. (2017) who use variational methods or Duan et al. (2017) who learn deterministic embeddings.

In contrast, VATE learns an embedding *of the MDP* (i.e., the reward and transition function). Furthermore, VATE conditions the policy on the posterior *distribution* instead of samples, allowing the policy to reason about task uncertainty and trade off exploration and exploitation online.

Posterior Sampling. Posterior sampling (Thompson, 1933; Strens, 2000; Osband et al., 2013) estimates a posterior distribution over MDPs (i.e., model and reward functions), in the same spirit that VATE does. This posterior is used to periodically sample a single hypothesis MDP (e.g., at the beginning of an episode or after a fixed number of steps), and the policy which is *optimal for that MDP* is followed subsequently. Convergence to the optimal policy is achievable, however compared to the full BAMDP setting this approach can be slow, and does not allow for the policy to take task uncertainty into account. The point at which to re-sample a new hypothesis is hand-chosen, prohibiting online adaptation of the agent's exploration / exploitation strategy. Some recent deep RL methods use stochastic latent variables for structured exploration (Gupta et al., 2018; Rakelly et al., 2019), although with a different (inference) framework that does not directly encode the MDP (reward and transition function).

Figure 2: Agent performance during training, averaged across 10 seeds, when evaluated on 10 random tasks. Goal position is fixed for three episodes, agent position each episode.



(a) VATE (b) Recurrent policy

Figure 3: Hand-picked but representative example test rollouts for VATE (a) and an RNN-based policy (b). The red background indicates the posterior probability of receiving a reward at that cell.

5 Experiments

We present preliminary experiments on a 5×5 gridworld, showing how the VATE policy balances exploration and exploitation. The task is to go to a goal (selected uniformly at random). Crucially, the goal is unobserved by the agent, inducing task uncertainty and necessitating exploration. The goal can be anywhere except around the starting cell, which is at the bottom left. An episode length is 15, with 3 episodes per task. Actions are: *up, right, down, left, stay* (executed deterministically). The agent gets a sparse reward signal: -0.1 on non-goal cells, and $+1$ on the goal cell. The best strategy is to explore until the goal is found, and stay on the goal or return to it when reset to the initial position. We use the architecture of Figure 1, with a Gaussian q and embedding size 10.

We compare VATE to approaches based on recurrent networks (Duan et al., 2016; Wang et al., 2016) by using a similar architecture as in Figure 1, but with deterministic m and no decoder (with embedding sizes 10 and 64). Figure 2 shows the learning curves for both approaches: VATE outperforms RNN-based approaches by a large margin. We also conduct two ablation studies in which we either don't predict the future (i.e. training the decoder only on past transitions) or do not backpropagate the RL loss through the encoder. We find that predicting entire trajectories as opposed to only past and current observations is beneficial. Detaching the RL loss does not make a significant difference for this toy problem. Figure 3a shows the VATE policy's behaviour at test time with deterministic actions (i.e., all exploration is done by the policy). The red background visualises the posterior belief by using the learned reward function. VATE learns the correct prior and adjusts its belief correctly over time. It predicts no reward for cells it has visited, and explores the remaining cells until it finds the goal. Figure 3b shows that the recurrent policy explores less efficiently.

6 Conclusion

In this paper we presented VATE, a novel method for dealing with the problem of balancing exploration and exploitation when performing action selection on new tasks. We use the meta-learning framework to utilise knowledge obtained in related tasks, and perform approximate inference over a learned, low-dimensional latent representation of the MDP. In preliminary experiments on a grid-world, we showed that the policy learns to use the posterior over this latent representation to guide its action selection, exploring the environment strategically. For future work, we are interested in scaling our method up to more challenging tasks such as continuous domains and maze environments, which are frequently used in meta-learning settings.

ACKNOWLEDGEMENTS

We thank Yarin Gal, Sebastian Schulze and Joost van Amersfoort for useful discussions and feedback. The NVIDIA DGX-1 used for this research was donated by the NVIDIA corporation. Luisa Zintgraf is supported by the Microsoft Research PhD Scholarship Program. Anuj Mahajan is sponsored by DeepMind. Maximilian Igl is supported by the UK EPSRC CDT in Autonomous Intelligent Machines and Systems This project has received funding from the European Research Council under the European Union's Horizon 2020 research and innovation programme (grant agreement number 637713).

REFERENCES

Pieter Abbeel, Adam Coates, Morgan Quigley, and Andrew Y Ng. An application of reinforcement learning to aerobatic helicopter flight. In *Advances in neural information processing systems*, pp. 1–8, 2007.

Isac Arnekvist, Danica Kragic, and Johannes A Stork. Vpe: Variational policy embedding for transfer reinforcement learning. *arXiv preprint arXiv:1809.03548*, 2018.

Richard Bellman. A problem in the sequential design of experiments. *Sankhyā: The Indian Journal of Statistics (1933-1960)*, 16(3/4):221–229, 1956.

John D Co-Reyes, YuXuan Liu, Abhishek Gupta, Benjamin Eysenbach, Pieter Abbeel, and Sergey Levine. Self-consistent trajectory autoencoder: Hierarchical reinforcement learning with trajectory embeddings. *arXiv preprint arXiv:1806.02813*, 2018.

Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. RL^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

Yan Duan, Marcin Andrychowicz, Bradly Stadie, OpenAI Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. In *Advances in neural information processing systems*, pp. 1087–1098, 2017.

Michael O'Gordon Duff and Andrew Barto. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*. PhD thesis, University of Massachusetts at Amherst, 2002.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

Abhishek Gupta, Russell Mendonca, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Meta-reinforcement learning of structured exploration strategies. *arXiv preprint arXiv:1802.07245*, 2018.

Karol Hausman, Jost Tobias Springenberg, Ziyu Wang, Nicolas Heess, and Martin Riedmiller. Learning an embedding space for transferable robot skills. 2018.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2018.

Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, pp. 3003–3011, 2013.

Kate Rakelly, Aurick Zhou, Deirdre Quillen, Chelsea Finn, and Sergey Levine. Efficient off-policy meta-reinforcement learning via probabilistic context variables. *arXiv preprint arXiv:1903.08254*, 2019.

Jürgen Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

Bradly C Stadie, Ge Yang, Rein Houthooft, Xi Chen, Yan Duan, Yuhuai Wu, Pieter Abbeel, and Ilya Sutskever. Some considerations on learning to explore via meta-reinforcement learning. *arXiv preprint arXiv:1803.01118*, 2018.

Malcolm Strens. A bayesian framework for reinforcement learning. In *ICML*, volume 2000, pp. 943–950, 2000.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Sebastian Thrun and Lorien Pratt. Learning to learn: Introduction and overview. In *Learning to learn*, pp. 3–17. Springer, 1998.

Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.

Ziyu Wang, Josh S Merel, Scott E Reed, Nando de Freitas, Gregory Wayne, and Nicolas Heess. Robust imitation of diverse behaviors. In *Advances in Neural Information Processing Systems*, pp. 5320–5329, 2017.