IV-Posterior: Inverse Value Estimation for Interpretable Policy Certificates

Tatiana López-Guevara[1,2] **Michael Burke**[2] **Nicholas K. Taylor**[1] **Kartic Subr**[2]
[1]School of Mathematics and Computer Science, Heriot-Watt University
[2]School of Informatics, University of Edinburgh
`t.l.guevara@ed.ac.uk`

Abstract

Model-free reinforcement learning (RL) is a powerful tool to learn a broad range of robot skills and policies. However, a lack of policy interpretability can inhibit their successful deployment in downstream applications, particularly when differences in environmental conditions may result in unpredictable behaviour or generalisation failures. As a result, there has been a growing emphasis in machine learning around the inclusion of stronger inductive biases in models to improve generalisation. This paper proposes an alternative strategy, inverse value estimation for interpretable policy certificates (IV-Posterior), which seeks to identify the inductive biases or idealised conditions of operation already held by pre-trained policies, and then use this information to guide their deployment. IV-Posterior uses Masked Autoregressive Flows to fit distributions over the set of conditions or environmental parameters in which a policy is likely to be effective. This distribution can then be used as a policy certificate in downstream applications. We illustrate the use of IV-Posterior across a two environments, and show that substantial performance gains can be obtained when policy selection incorporates knowledge of the inductive biases that these policies hold.

1 Introduction

The proliferous development of methods that use reinforcement learning [3, 6, 4, 23] or other strategies [24, 25, 11, 17] for learning robotic manipulation skills has enabled the creation of exciting libraries of potentially viable policies to solve many common tasks. However, an extra layer of interpretability to identify when such policies are likely to succeed is important for their successful deployment, where uncertainty around environmental conditions and parameters may exist. This has motivated the inclusion of strong inductive biases [10, 16, 13] or alternative forms of certification over these models [24].

For example, consider an agent that wants to move an object as shown in Figure-1. The agent can either push or pick and place, but choosing the appropriate strategy depends on the belief around the object being manipulated: i.e pick and place is preferable for lighter objects whereas push is better for bigger or heavier ones. We propose a methodology to enable interpretability of such expert systems that choose from different pre-trained policies. Specifically, we focus on how to infer the posterior probability (q_{ϕ}) of the model parameters and initial conditions (s_0, ξ) given evidence of the rewards (r) from an arbitrarily chosen policy. These posteriors (q_{ϕ}^{push}, $q_{\phi}^{\mathrm{p\&p}}$, etc.) may be used to reason about viability and preferability or to counterfactually query environment settings and conditions in which a policy may fail. We perform inverse value estimation for interpretable policy certificates (IV-Posterior), to identify inductive biases or idealised conditions of operation *already held by pre-trained policies*, and then use this information to guide their deployment.
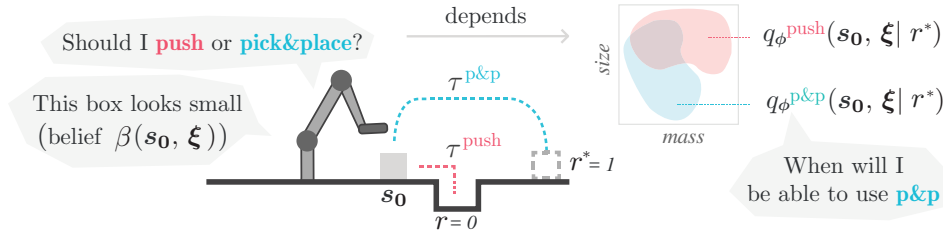
Figure 1: We can use the set of initial conditions and parameters learned using IV-Posterior to select how to move an object based on the agent's belief. It is better to pick and place small objects that start at the left of the hole and push bigger or heavier objects.

2   Related Work

IV-Posterior fits posterior distributions over environment states or settings, conditioned on rewards. This is similar to recent work that attempts to learn models of preconditions for planning [24, 25], by generating samples of valid initial states for a given policy based on the context of a scene. Here, samples are drawn from Gaussian Processes using level sets, so this approach is only applicable to low-data regimes. The use of interpretable policy posteriors for task selection is also related to that of initiation sets within the Options Framework [23, 14, 4] or abstractions in skill segmentation [18, 15]. However, these sets are usually discovered by learning classifiers that identify, from which observable states, an option was successfully triggered, without any notion of uncertainty. In contrast, by fitting a full posterior over environments, IV-Posterior also allows for counterfactual reasoning of the form, 'Where won't this policy work?'.

The Sim2Real (S2R) community [26, 3, 1, 22, 6] has developed a number of methods to iteratively learn a policy and fit distributions over environment parameters that can be used to model a particular instance of a real world, often relying on Gaussianity [22, 6] or independence assumptions [22, 3] regarding the underlying distribution. Of particular interest here is Likelihood Free Inference (LFI), which is often used for simulation alignment to infer which simulation parameter generated a particular real trajectory or observation [7, 8, 22, 12, 19, 21, 9]. Recent advances in LFI have shown exciting results in the quality of the estimated posterior by relying on contrastive losses and sequential estimation [8, 9]. Our work also relies on sequential LFI, but focuses on estimating a full posterior distribution over initial conditions and environment parameters that might be partially observable during deployment. Importantly, by doing so we allow for counterfactual reasoning and the use of a broad range of easily computed metrics that incorporate uncertainty, like the proposed cross entropy criterion for task selection in expert systems. Existing sample-based approaches producing an intractable posterior do not directly allow this. By framing our problem within the LFI framework, we take advantage of recent algorithmic developments in inference and density estimation.

3   IV-Posterior

IV-Posterior is a sequential a method to learn interpretable posteriors over the set of initial conditions or environmental parameters in which a policy is likely to be effective.

3.1   Problem formulation

Let $\mathcal{M}$ define a Markov Decision Process, composed of states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$ and a reward map $R : \mathcal{S} \to \mathbb{R}$. The distribution of possible starting states is defined by $s_0 \sim p(s_0)$. Transitions are determined by $s_{t+1} \sim p(s_{t+1}|s_t, a; \xi)$, where $\xi \sim p(\xi)$ represent the parameters that affect the dynamics of the environment (extrinsic/intrinsic parameters of the objects involved in the task). Note that $p(\xi)$ implicitly defines a *class of objects* from which particular object instances can be sampled (i.e, objects with different shapes, masses, etc.). An agent can act on such environments according to a policy distribution $a_t \sim \pi_{\theta}(a_t|s_t)$ parametrized by θ. Given a initial state s_0, parameter ξ and a policy θ, the agent will observe a trajectory after each interaction as $p(\tau|s_0, \xi, \theta) = \prod_{t=0}^{T} p(s_{t+1}|s_t, \pi_{\theta}(a_t|s_t); \xi)p(s_0)$. The observed rewards will thus be distributed according to $r \sim p(R(\tau)|s_0, \xi, \theta)$. Let $q_{\phi}(s_0, \xi|r)$ represent the variational approximator of the distribution over initial states and parameters conditioned on the reward, i.e an autorregresive flow parametrised by ϕ.

Let $\beta(s_0, \xi)$ represent the belief distribution over initial states and parameters of the object the agent wants to manipulate.

Given a policy, we wish to learn under which initial conditions and object instances the policy is most likely to succeed. Essentially, we are interested in the posterior distribution over instances of the environment that will yield high reward.
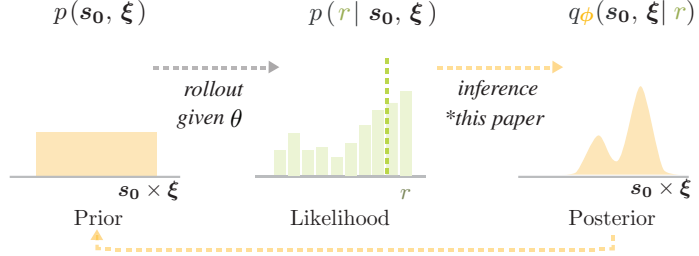

Figure 2: Sequential Estimation of the Conditional Posterior on Environment Parameters.

$$\phi^*(\boldsymbol{\theta}) = \operatorname*{argmin}_{\phi} \mathbb{E}_{\substack{s_0, \xi \sim p(s_0, \xi) \\ r \sim p(r|s_0, \xi, \boldsymbol{\theta})}} \left[- \log q_\phi(s_0, \xi|r) \right] \tag{1}$$

We sequentially narrow down the posterior across $k = 1 : N$ iterations and update the prior of the next iteration with the conditioned posterior learned in the previous (Algorithm 1 / Figure 2). At each iteration, we collect a dataset $\mathcal{D}^{(k)}$ from b observed rollouts $\{(s_0, \xi, r)^{(b)}\}_{b=1}^B$ and optimize Eq 1 to fit the variational approximator of the posterior q_ϕ. We used the Automatic Posterior Transformation [9] algorithm to optimize Eq 1. The form of q_ϕ should be flexible enough to capture complex correlations across parameters (i.e. Figure 3-*Right*). Therefore assuming a diagonal covariance [22] might not be enough. Masked Autoregressive Flow (MAF) [20] is a powerful density estimator method that excels at modelling complex distributions, compared to other neural estimators, for inference [20, 9]. We model our posterior distribution q_ϕ using MAF.

3.2 Posterior for Task Selection

The learned posterior distribution q_ϕ can also be used as a policy certificate in downstream applications like task selection. Consider the case where the agent wants to manipulate an instance of an object for which it has an initial belief $\beta(s_0, \xi)$ (Figure 1). Assume it already knows how to perform a task in different ways $\boldsymbol{\theta}^{(i)}$ and that we learned their respective posteriors $\phi^{(i)}$ as described in Section 3.1.

We can use ϕ to select which manipulation skill to use by computing the cross-entropy with respect to the prior belief of the object:

$$i^* = \operatorname*{argmax}_{i} \mathbb{E}_{s_0, \xi \sim \beta(s_0, \xi)} \left[\log q_\phi^i(s_0, \xi|r^*) \right] \tag{2}$$

4 Experiments and Results

Our experiments aim to answer the following questions: (1) What is the quality of the learned posterior in terms of interpretability? (2) How useful is the posterior over the environment parameters for task selection?

We tested our method on two different environments: Kitchen2D-Pouring [24] with their predefined dense reward of $exp(2 * (x * 10 - 9.5)) - 1$, where x is the proportion of liquid particles that fall outside the target container. Here $\xi = \emptyset$, $s_0 = \{grasp, rel.x, rel.y, dangle\}$, $B = 300$ and $N = 10$. To evaluate the task selection downstream task, we used pre-trained policies for `pickplace` and `push` using Hindsight Experience Replay (HER) [2][1]. We created a new environment in OpenAI [5], FetchBox, where the goal is to move a puck inside a box with a hole at the bottom (Figure 3-*Left, Bottom*). The reward is 1 if the puck is inside the box at the end of the episode and 0 otherwise. We chose $s_0 = \{x, y\}$ to be the initial position of the puck in the table, and the parameters of the puck to be a combination of mass, width, height and friction (Table 1 shows the used ranges) $\xi \subset \{mass, h, w, fr\}$, $B = 500$ and $N = 15$.

---

[1] https://github.com/TianhongDai/hindsight-experience-replay

Figure 3-*Left* shows evolution of the marginals of the posterior at iterations $k = 1, 7, 15$ for $\phi^{pickplace}$ and ϕ^{push} using 5000 samples. We can observe that, even after 7 iterations, the posterior varies significantly between both policies, especially with respect to $\boldsymbol{\xi}_{mass}$ and $\boldsymbol{\xi}_h$. This indicates that $\boldsymbol{\theta}^{pickplace}$ works best for pucks with mid to low mass and $\boldsymbol{\theta}^{push}$ with short pucks due to the obstacle in the box. We also present 1000 samples of the final posterior distribution for the Kitchen2D environment in Figure 3-*Right*. We can observe a multimodal correlation between the relative positions of the containers and where to grasp the source container, as well as the irrelevance of the final angle of the cup for this task.

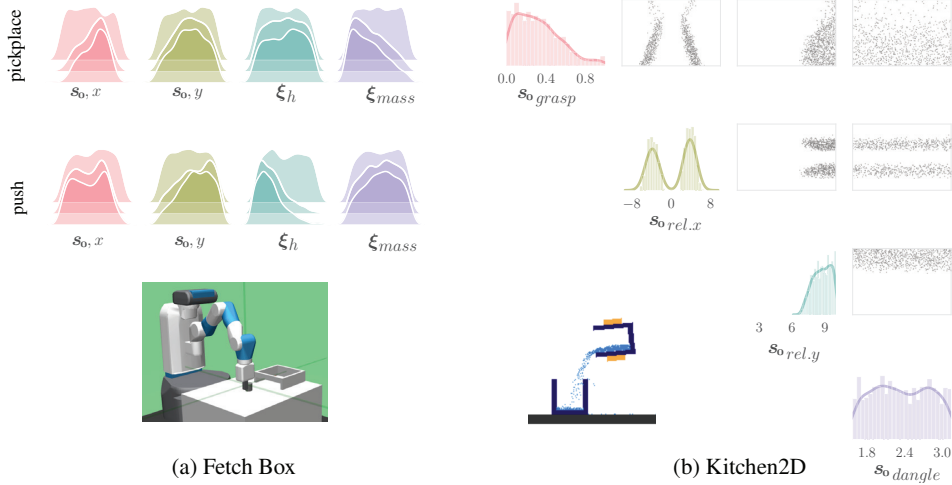(a) Fetch Box   (b) Kitchen2D

Figure 3: (Left) Evolution of the marginals of the posterior at iterations $k = 1, 7, 15$ for pickplace and push in the Fetch Box environment. (Right) Samples from the posterior at the last iteration $k = 10$ in the Kitchen2D environment, visualized as pairwise relations between variables.

To test the usefulness of the posterior for task selection, we used the FetchBox environment to move a puck into a box (Figure 3-*Left*). The goal is to select between $\boldsymbol{\theta}^{push}$ and $\boldsymbol{\theta}^{pickplace}$ policies to move a puck for which the agent has a belief $\beta(\boldsymbol{s_0}, \boldsymbol{\xi})$. We sampled 1000 different beliefs of the puck according to $\beta(\boldsymbol{s_0}, \boldsymbol{\xi}) = \mathcal{N}(\boldsymbol{\xi} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with $\mu^{(j)} = 0.5$ and $\Sigma^{(j,j)} = 0.7$ and off-diagonal 0. The belief is then rescaled to valid ranges according to Table 1. We use random selection as the baseline and also show always pick or always push as guidelines. We show different combinations of parameters with different dimensionalities on the x-axis and the average reward on the y-axis. All configurations of methods and parameters were executed 1000 times. Figure 4 shows the performance gained by using the learned posterior for task selection.

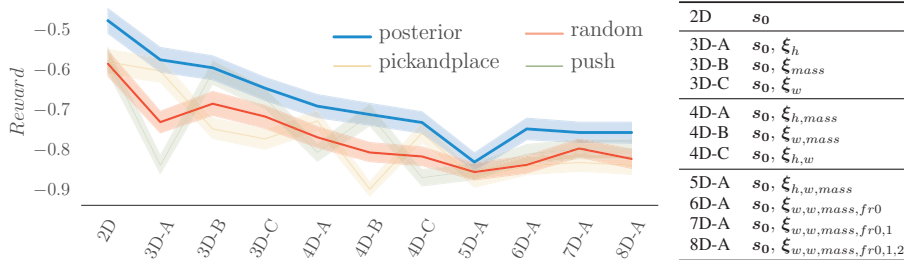| 2D	s_0
3D-A	s_0, ξ_h
3D-B	s_0, ξ_{mass}
3D-C	s_0, ξ_w
4D-A	$s_0, \xi_{h,mass}$
4D-B	$s_0, \xi_{w,mass}$
4D-C	$s_0, \xi_{h,w}$
5D-A	$s_0, \xi_{h,w,mass}$
6D-A	$s_0, \xi_{w,w,mass,fr0}$
7D-A	$s_0, \xi_{w,w,mass,fr0,1}$
8D-A	$s_0, \xi_{w,w,mass,fr0,1,2}$

Figure 4: Average rewards for different task selection methods over the initial position of the puck on the table s_0 and different subsets of parameters $\boldsymbol{\xi}$ for 1k randomly initialized object beliefs.

5   Conclusions

We presented IV-Posterior, a method to learn the inductive biases held by a policy, by fitting a distribution over the set of conditions or environmental parameters in which a policy is likely to be effective. We showed that this posterior provides an additional layer of interpretability to preexisting policies, allows counterfactual reasoning, and facilitates task selection in expert systems without requiring an intractable posterior to be computed using an expectation over samples. In future work we plan to apply IV-Posterior to curriculum learning and domain adapation.

References

[1] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[2] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in neural information processing systems*, pages 5048–5058, 2017.

[3] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

[4] Akhil Bagaria and George Konidaris. Option discovery using deep skill chaining. In *ICLR*, 2020.

[5] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[6] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan D. Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. *International Conference on Robotics and Automation*, 2019.

[7] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *arXiv preprint arXiv:1911.01429*, 2019.

[8] Conor Durkan, Iain Murray, and George Papamakarios. On contrastive learning for likelihood-free inference. *arXiv preprint arXiv:2002.03712*, 2020.

[9] David S. Greenberg, Marcel Nonnenmacher, and Jakob H. Macke. Automatic posterior transformation for likelihood-free inference. In *ICML*, 2019.

[10] Vincent Le Guen and Nicolas Thome. Disentangling Physical Dynamics from Unknown Factors for Unsupervised Video Prediction. In *CVPR*, 2020.

[11] Tatiana López Guevara, Nicholas K Taylor, Michael U Gutmann, Subramanian Ramamoorthy, and Kartic Subr. Adaptable pouring: Teaching robots not to spill using fast but approximate fluid simulation. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2017.

[12] M.U. Gutmann and J Corander. Bayesian optimization for likelihood-free inference of simulator-based statistical models. *Journal of Machine Learning Research*, 17(125):1–47, 2016.

[13] Miguel Jaques, Michael Burke, and Timothy Hospedales. Newtonianvae: Proportional control and goal identification from pixels via physical latent spaces, 2020.

[14] George Konidaris and Andrew G Barto. Skill discovery in continuous reinforcement learning domains using skill chaining. In *Advances in neural information processing systems*, pages 1015–1023, 2009.

[15] George Konidaris, Scott Kuindersma, Roderic Grupen, and Andrew Barto. Cst: Constructing skill trees by demonstration. *Doctoral Dissertations, University of Massachuts, Ahmrest*, 2011.

[16] Yunzhu Li, Hao He, Jiajun Wu, Dina Katabi, and Antonio Torralba. Learning compositional koopman operators for model-based control. In *ICLR*, 2020.

[17] Tatiana López Guevara, Rita Pucci, Nick Taylor, Michael U. Gutmann, Subramanian Ramamoorthy, and Kartic Subr. Stir to pour: Efficient calibration of liquid properties for pouring actions. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2020.

[18] Scott Niekum, Sarah Osentoski, George Konidaris, Sachin Chitta, Bhaskara Marthi, and Andrew G Barto. Learning grounded finite-state representations from unstructured demonstrations. *The International Journal of Robotics Research*, 34(2):131–157, 2015.

[19] George Papamakarios and Iain Murray. Fast ϵ-free inference of simulation models with Bayesian conditional density estimation. *Advances in Neural Information Processing Systems*, 2016.

[20] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pages 2338–2347, 2017.

[21] George Papamakarios, David C Sterratt, and Iain Murray. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. *arXiv preprint arXiv:1805.07226*, 2018.

[22] F. Ramos, Rafael Possas, and Dieter Fox. Bayessim: adaptive domain randomization via probabilistic inference for robotics simulators. *Robotics Science Systems (RSS)*, 2019.

[23] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

[24] Zi Wang, Caelan Reed Garrett, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Active model learning and diverse action sampling for task and motion planning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4107–4114. IEEE, 2018.

[25] Zi Wang, Caelan Reed Garrett, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Learning compositional models of robot skills for task and motion planning, 2020.

[26] Lilian Weng. Domain randomization for sim2real transfer. *lilianweng.github.io/lil-log*, 2019.

Appendix

5.1 IV-PosteriorAlgorithm

Algorithm 1: IV-Posterior: Sequential Refinement of the Posterior

---

Input: $p(s_0, \xi), p(\tau|s_0, \xi, \theta)$
Output: ϕ^*
for $k = 1 : N$ **do**
    for $b = 1 : B$ **do**
1   Sample $s_0, \xi \sim p(s_0, \xi)$
2   Sample $\tau \sim p(\tau|s_0, \xi, \theta)$
3   Compute $r = R(\tau)$
4   Add to dataset $\mathcal{D}^{(k)} = \mathcal{D}^{(k-1)} \cup \{ (s_0, \xi, r) \}$
5   Optimize $\mathcal{L}(\phi(\theta))$ using $\mathcal{D}^{(k)}$ (Eq 1)
6   Get r^* from $\mathcal{D}^{(k)}$
7   Set $p(s_0, \xi) = q_\phi(s_0, \xi|r^*)$

5.2 Range of Parameter Priors used in the Experiments

In all experiments we used a uniform prior for $p(s_0, \xi)$ with the following parameter limits:

	Description	Notation	Name in Sim.	Range	
				ξ_{min}	ξ^*
Kitchen2D*					
	Grasp ratio	s_{0grasp}	grasp_ratio	0.0	1.0
	Relative x	$s_{0rel.x}$	rel_x	-10.0	10.0
	Relative y	$s_{0rel.y}$	rel_y	1.0	10.0
	Final angle	$s_{0dangle}$	dangle	$0.5 * \pi$	π
Fetch OpenAI+					
	Mass	ξ_{mass}	body_mass	1.0	20.0
	Size x	ξ_w	geom_size[0]	0.02	0.045
	Size y	ξ_h	geom_size[1]	0.02	0.03
	Tangential Friction	ξ_{fr0}	geom_friction[0]	0.1	1.0
	Torsional Friction	ξ_{fr1}	geom_friction[1]	0.1	1.0
	Rolling Friction	ξ_{fr2}	geom_friction[2]	0.1	1.0

Table 1: Object Parameter Priors used in *Kitchen2D [24] (Box2D Engine) and +Fetch (Mujoco Engine) environments (Name in Sim corresponds to the name of the parameter in each engine/wrapper.