Artificial Empathy: Towards a Framework for Unsupervised Agency Detection and Policy Reconstruction

Peter Kuhn
Human Inductive Bias Project
Cambridge, UK
peterkuhn92@gmail.com

Chris Pang
Human Inductive Bias Project
Cambridge, UK
mutatismutandisetplusultra@gmail.com

Sonakshi Chauhan
Human Inductive Bias Project
Cambridge, UK
sonakshichauhan1402@gmail.com

Abstract

We study how an AI system can identify and model other agents in its environment from observation alone, which is a capability necessary for cooperative behaviour in the real world. This problem is less constrained than inverse reinforcement learning and remains largely unexplored. We propose a framework that uses a reinforcement learning agent, trained on an independent task as a prior about agentic dynamics, to perform agency detection and policy reconstruction.

1 Introduction

We define agency detection as the task of distinguishing between agentic behaviour and non-agentic activity in an environment. We define policy reconstruction as the task of inferring the policy of an agent from behaviour alone. The latter task, in isolation, is typically tackled in inverse reinforcement learning (IRL), but there are few prior attempts [for an exception, see Zarncke, 2025] to solve both tasks in conjunction. If we could implement agency detection and policy reconstruction in an unsupervised way, this would be a step towards solving the AI alignment problem, the problem of aligning the goals of AI systems with the goals of their human creators [Bostrom, 2014]. An agent equipped with such a capacity could then develop and act on a joint policy that is a synthesis of its own policy and the policy of the agents in its environment. Such a system would, so to speak, have the propensity to care for other agents. We develop *artificial empathy* (AE) as a framework for solving this task.

Implementing artificial empathy is hard: *a priori*, every random physical event could be the result of agentic activity. Did the apple *want* to fall off the table? To decide this, a vast array of prior knowledge about the nature of apples and the nature of agency is required. Inspired by psychological models of empathy, we suggest that agency detection can work by finding systems that one can model using *one's own* agential dynamics: As the AI system that we want to align is itself a form of agent, it can use its dynamics as a prior about what agency looks like. Thus, a system implementing artificial empathy jointly solves the agency-detection and policy-reconstruction tasks by determining whether parts of the environment can be predicted from variations in its own dynamics, and hence be considered agents. In this paper, we take the first steps towards implementing artificial empathy: We propose an architecture that uses a trained model-based reinforcement learning (RL) agent as a prior to perform agency-detection and policy reconstruction. We demonstrate that this system can be used to differentiate between simple agentic activity and random dynamics.

Preprint.

2 Prior Work

2.1 Psychological Inspirations and Self-Other Overlap

We take inspiration for our approach from simulation theory in cognitive science, which suggests that agents understand and predict others' behaviour by using their own cognition and decision-making rather than an independent theory of mind. Gordon [1986] introduced this in the context of folk psychology, claiming that humans often "put themselves in the other's place" to infer and explain the behaviour of others. The plausibility of simulation-based mechanisms for social understanding is supported by neuroscientific findings, such as mirror neuron responses, in which a population of neurons activates both during execution and observation of an activity [Gallese and Goldman, 1998, Goldman, 2006].

We are also inspired by alignment approaches that reduce self-other overlap in LLM-based agents by reducing the distance between the agent's self-representation and representations of other agents or humans [Carauleanu et al., 2024]. However, even though inducing such overlap can produce behaviour that appears to be aligned, the method is primarily intuition-driven and is predicated on accurately identifying self- and other-representations in activation space. By establishing artificial empathy as a first-principles framework, we hope to overcome these limitations.

2.2 Inverse Reinforcement Learning

Inverse RL (IRL) is a well-established approach that aligns with our notion of artificial empathy. It involves inferring an agent's reward function from observed behaviour, typically given as labelled observation–action pairs [Ng and Russell, 2000]. Extensions using maximum-entropy priors have yielded promising results despite the under-constrained nature of the inference problem [Ziebart et al., 2008]. To reduce goal misgeneralization, IRL has also been proposed as a step towards AI alignment [Russell, 2019]. However, IRL relies on preprocessed observation-action trajectories, limiting its applicability to data-rich settings where agent behaviour is clearly identifiable. Thus, IRL and its variants cannot be complete alignment strategies in themselves.

3 Method

3.1 Overview and Motivation

Our goal is to infer the policy of an agent from observing the environment it inhabits. In our current setup, we will assume full observability of the environment, represented by observations $(o_0, o_1,\ ldots)$, for ease of implementation. Nothing deep hinges on this. We assume that the environment contains an agent, the "other", following some policy π_O. The task of the AE system is to infer π_O from the stream of observations alone. We differentiate between base agents and the AE system, where the former are RL systems trained to play specific games, and the latter aims to solve the AE task. In our setup, the AE task is purely epistemic, i.e. we want it to infer what is an agent and what is it trying to do. The broader aim of AE is the creation of a *joint policy* π_Ω that joins the policy of the agent itself ("self") π_S and the policy of the "other" π_O using some aggregation function f, i.e. $\pi_\Omega = f(\pi_S, \pi_O)$. We use a deep Q-learning framework to perform RL and to represent policies in terms of Q-functions $Q(s, a)$ for some state s and action a [Sutton and Barto, 2018]. As these Q-functions will, in future work, implement complex joint policies learned from other agents, the relevant states s should refer to hidden states of the environment (in the sense of a partially observable Markov decision process, see Puterman [2014]) rather than simple observations. If you want to infer what some agent cares about, you first need to construct a convergent representation of the joint environment. This is why we use a model-based RL setup [Moerland et al., 2023]. We will first describe the base agent that will serve both as the target "other" in our experiments, as well as an agency prior. Note that we provide a diagram in Appendix A.

3.2 Base Agent Architecture

Our base architecture is a deep Q-learner [Mnih et al., 2015] that we have augmented with a trainable world model. The world model is inspired by a variational autoencoder [Kingma and Welling, 2022, Hafner et al., 2025] that infers a probabilistic representation of world states. We

use neural-network-based models that represent probability distributions as Gaussians parametrised by means and logarithmic variances. The architecture consists of a convolutional network encoder $p_\theta(s_t|o_t)$ parametrized by θ that infers world states s_t from observations o_t, a dense transition model $q_\phi(s_{t+1}|s_t, a_t)$ parametrized by ϕ that predicts the next world state s_{t+1} depending on the action taken a_t, and a convolutional, non-probabilistic decoder d parametrized by ω that reconstructs observations based on world states $\hat{o}_{t+1} = d_\omega(\hat{s}_{t+1})$. Here $\hat{s}_{t+1}$ is sampled from the transition model. Finally, the base agent contains a Q-network that predicts Q-values for world states $Q_\rho(s_t)$, parametrized by ρ. Note that $Q(s_t, a_t)$, necessary for the sort of Q-learning we want to do, can be recovered by using the transition model to calculate Q-values over states and actions. In later experiments, we train the base agents using a variant of the experience-replay algorithm described in Mnih et al. [2015] where at every time step t the agent sees two observations, the action it has taken, and a reward $(o_t, a_t, o_{t+1}, r_{t+1})$ and integrates the models by combining three kinds of losses. The reconstruction loss $\mathcal{L}_{recon}$ is defined as the mean squared error between the true next observation o_{t+1} and its reconstruction $\hat{o}_{t+1}$, where $\hat{o}_{t+1}$ is sampled from the model; the transition loss $\mathcal{L}_{trans}$ is the Kullback–Leibler divergence between the modeled distribution $p_\theta(s_{t+1} \mid o_{t+1})$ and the approximate posterior $q_\phi(s_{t+1} \mid s_t, a_t)$ over the next latent state s_{t+1} given the current latent state s_t and action a_t; and the Q-loss $\mathcal{L}_Q$ is the mean squared error between the target value y_i and the predicted value Q_i. With $y_i = r_i + \gamma \max_{a'} Q(s'_i, a')$ and $Q_i = Q(s_i, a_i)$. i indexes transitions in a mini-batch. This results in a combined loss for the base agent:

$$\mathcal{L}_{base} = \alpha\mathcal{L}_{recon} + \beta\mathcal{L}_{trans} + \delta\mathcal{L}_Q \tag{1}$$

We chose the hyperparameters α, β, δ and γ to be 10^{-4}, 10^{-2}, 1 and 0.98 respectively.

3.3 AE Architecture

The AE system uses a trained base agent as an agency-prior. Thus, it is structurally similar to the base agent and also consists of a similar encoder, transition and Q-network combination. Learning proceeds by observing a base agent trained to perform some task. We will call the first base agent the prior agent, and the game it learned the prior game. The target base agent we will call the expert. The environment in which the expert acts generates the stream of observations $(o_0, o_1, ...)$. These are reordered by the AE system and used as training data. The task is to infer the expert's policy by reconstructing its encoder, transition model, and Q-network. We initialise an AE system by loading a prior, consisting of the parameters $(\bar{\theta}, \bar{\phi}, \bar{\rho})$ corresponding to the weights of the prior agent. We will assume a Gaussian prior probability in weight-space for every network with a mean around the prior weights. As is known from a Bayesian analysis of weight decay, a prior Gaussian in weight-space can be enforced by the use of mean squared error [MacKay, 1992]. We will express the prior constraints on our model as set of penalty terms $\mathcal{R}_p$, $\mathcal{R}_q$ and $\mathcal{R}_Q$ for encoder, transition model and Q-network respectively, which are the mean square error between the weights of the prior agent's parameters and the parameters of the AE system.

Conceptually, we want to predict s_{t+1} from s_t (i.e. the dynamics of the world) by assuming that s_{t+1} was the result of an agent observing the environment and acting on it. To do this, we will use networks constrained by the agency prior, which we will distinguish from those of the base agent with a dash. We encode the future observations as before as $p_{\hat{\theta}}(s_{t+1}|o_{t+1})$. We then make a guess about how the agent putatively acting on the environment perceives the world by encoding the current observation as $p'(s_t|o_t)$. We then make predictions about the results the agent we are modelling might expect to find for each action a_i by computing $q'(s_{t+1}|s_t, a_i)$. We can then make a prediction about how much the modelled agent values different states of the world by computing $Q'(s_t, a_t)$. By taking a softmax, we turn this into probabilities for discrete actions taken by the putative agent $P(a_i)$.

Now we have all the ingredients we need to predict the next state of the world, under the assumption that these are generated by an agent (denoted by A) following some policy. To do this, we simply marginalise over all N different actions to obtain a probability distribution over next world states. We use p^* to differentiate it from the results of an encoding:

$$p^*(s_{t+1}|s_t, A) = \sum_{i=0}^{N} P(a_i)q'(s_{t+1}|s_t, a_i) \tag{2}$$

Thus, we can naturally use the Kullback-Leibler divergence between the probability distribution of encoded world-states and the world-states we would expect on the assumptions that there is an agent acting in the environment, parametrized in the way described. Thus the main loss of our AE system is:

$$\mathcal{L}_{KL} = D_{KL}[p(s_{t+1}|o_{t+1})||p^*(s_{t+1}|s_t, A)] \tag{3}$$

Finally, to get the full loss of the artificial empathy system, we have to ensure that our system actually *commits* to some predicted action, rather than just returning a smooth probability distribution and predicting the world dynamics from there. We do this by putting an entropy penalty on $P(a_i)$, i.e. $\mathcal{R}_H = H(P(a_i))$. The combined loss for the AE-system is thus:

$$\mathcal{L}_{AE} = \mathcal{L}_{KL} + \sigma\mathcal{R}_H + \kappa\mathcal{R}_p + \psi\mathcal{R}_q + \upsilon\mathcal{R}_Q \tag{4}$$

In words: We want to infer the world dynamics under the assumption that an agent is present, parametrised similarly to the prior agent, and that takes specific actions. We choose the hyperparameters σ, κ, ψ and υ to be 10, 100, 1000 and 1000 respectively.

4 A Simple Experiment

We validate the feasibility of the proposed architecture as follows. We train two base agents on tasks A and B. We use the agents trained on A as an expert and B as a prior to be loaded into the AE system. After training on fully observable trajectories, the AE system is then evaluated on its ability to imitate the expert by selecting actions via its learned Q-values. Performance in the 'imitation game' is measured by cumulative reward on task A. Experiments on the imitation game are conducted in a 15×15 gridworld with two tasks. In the food-game, the agent must collect three food items, each yielding a reward of 1, while every step incurs a reward of -0.01. In the edge-game, the agent must reach the lower-left edge to obtain a reward of 1, again receiving a reward of -0.01 for every step taken. Base agents are trained for 1200 epochs for stability and use ϵ-decay with 10^5 decay steps starting at 1.

The observations generated by these games are 2x15x15 tensors, where the first channel encodes food item positions (or nothing for the edge-game) and the agent position. We use the food-game as the prior game B and the edge-game as task A. To make the task of the AE-system less simple, we confound its observation stream by adding a decoy 'agent' that merely performs a random walk. If the AE agent can nonetheless learn to imitate the edge-game agent, then it has implicitly solved the task of agency detection. As shown in Figure 1, this is in fact what we observe. Note that this result is far from trivial: To achieve it, the agent has to pick up on the fact that the random dynamics of the decoy are ill-suited for prediction via the prior implicit in the AE-system. It has to utilise implicit prior knowledge about how different actions lead to different world-states, but it has to shift the implicit value of world-states to accommodate the observed behavioural profile. While these are early results, we consider them quite promising.

5 Limitations

Our experiments are conducted in a small, fully observable gridworld with a single confounding decoy agent, so the current results demonstrate feasibility rather than robustness or scale. The AE system currently relies on a hand-selected, structurally similar prior agent; how performance degrades as the prior and the target policy diverge in architecture or task structure remains untested. We also did not yet evaluate settings with multiple genuine agents, partial observability, or non-stationary policies, all of which are more representative of real-world alignment settings. Finally, because the AE loss combines several weighted terms ($\mathcal{L}_{KL}$, $\mathcal{R}_H$, $\mathcal{R}_p$, $\mathcal{R}_q$, $\mathcal{R}_Q$), the sensitivity of results to these hyperparameters has not been systematically characterised.

6 Conclusion and Future Work

We introduced artificial empathy, a framework that jointly performs agency detection and policy reconstruction by using an RL agent's own dynamics as a prior over agentic behaviour. In a simple

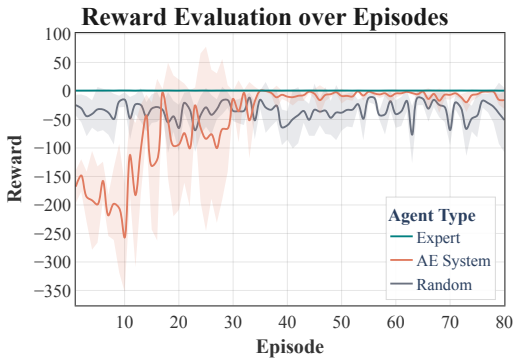

Figure 1: Aggregate of 5 runs of the 'imitation game'. Shaded regions are standard deviations. After 35 episodes the performance of the AE agent is far better than a random walk and approaches that of the base agent.

gridworld imitation task with a random-walk decoy, an AE system initialised from an unrelated prior task learned to imitate an expert agent's policy, suggesting that self-referential priors can support unsupervised agency detection. Future work would extend the framework to conditions of partial observability, scaling to environments with multiple simultaneous agents, relaxing the structural similarity between prior and target agents, using agency priors trained on multiple games, and moving from the purely epistemic AE task described here towards the construction of an explicit joint policy π_Ω that combines self and other.

References

Nick Bostrom. *Superintelligence*. Oxford University Press, London, England, July 2014.

Marc Carauleanu, Michael Vaiana, Judd Rosenblatt, Cameron Berg, and Diogo Schwerz de Lucena. Towards safe and honest ai agents with neural self-other overlap, 2024. URL `https://arxiv.org/abs/2412.16325`.

Vittorio Gallese and Alvin I. Goldman. Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences*, 2(12):493–501, 1998. URL `https://courses.media.mit.edu/2003spring/mas963/Gallese-goldman.pdf`.

Alvin I. Goldman. *Simulating Minds: The Philosophy, Psychology, and Neuroscience of Mindreading*. Oxford University Press, New York, NY, USA, 2006. ISBN 9780195138924.

Robert M. Gordon. Folk psychology as simulation. *Mind & Language*, 1(2):158–171, 1986. doi: https://doi.org/10.1111/j.1468-0017.1986.tb00324.x. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1468-0017.1986.tb00324.x`.

Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models, 2025. URL `https://arxiv.org/abs/2509.24527`.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL `https://arxiv.org/abs/1312.6114`.

David J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992. doi: 10.1162/neco.1992.4.3.448.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015.

Thomas M. Moerland, Joost Broekens, Aske Plaat, and Catholijn M. Jonker. Model-based reinforcement learning: A survey. *Found. Trends Mach. Learn.*, 16(1):1–118, January 2023. ISSN 1935-8237. doi: 10.1561/2200000086. URL https://doi.org/10.1561/2200000086.

Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, page 663–670, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1558607072.

Martin L Puterman. *Markov decision processes*. Wiley Series in Probability and Statistics. Wiley-Interscience, Newy York, August 2014.

Stuart Russell. *Human Compatible: Artificial Intelligence and the Problem of Control*. Viking, 2019.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2018.

Gunnar Zarncke. Foundations of unsupervised agent discovery in raw dynamical systems. AE Studio, July 2025. URL https://github.com/GunnarZarncke/agency-detect/blob/master/docs/unsupervised-agent-discovery.pdf. Date: July 23, 2025.

Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, AAAI'08, page 1433–1438. AAAI Press, 2008. ISBN 9781577353683.

A Artificial Empathy System Diagram

Diagram displaying the structure of the base agent and the AE system. Overlapping layers of boxes represent computation for many possible actions: the system computes one predicted future state for each action and assigns each predicted state a Q-value separately. The state with the maximum Q-value is then selected via argmax in the base architecture, and the states are marginalised via a softmax in the AE system.