Inferring Actions, Intentions, and Causal Relations in a Neural Network

Keno Juechems
Department of Experimental Psychology, University of Oxford, Oxford, UK
St John's College, Oxford, UK
keno.juchems@psy.ox.ac.uk

Andrew M. Saxe
Department of Experimental Psychology, University of Oxford, Oxford, UK
CIFAR Azrieli Global Scholars program, CIFAR, Toronto, Canada
andrew.saxe@psy.ox.ac.uk

Abstract

From a young age, we can select actions to achieve desired goals, infer the goals of other agents, and learn causal relations in our environment through social interactions. Crucially, these abilities are productive and generative: we can impute desires to others that we have never held ourselves. These abilities are often captured by only partially overlapping models, each requiring substantial changes to fit combinations of abilities. Here, in an attempt to unify previous models, we present a neural network underpinned by the linearly solvable Markov Decision Process (LMDP) framework which permits a distributed representation of tasks. The network contains two pathways: one captures the desirability of states, and another encodes the passive dynamics of state transitions in the absence of control. Interactions between pathways are bound by a principle of rational action, enabling generative inference of actions, goals, and causal relations supported by gradient updates to parts of the network.

Keywords: Inverse reinforcement learning; Social causal learning; Multitask LMDP

Introduction

Being able to reason accurately about the intentions and actions of other agents is critical for social animals like humans. For instance, even from a young age, we can infer likely goals of agents and generalize this knowledge to predict their actions in novel contexts (Heyes & Frith, 2014; Shafto, Goodman, & Frank, 2012). While there exists striking descriptive evidence of these abilities across human development and in many other species, it is unclear what computations underpin them. Some elements of these abilities have been captured by the powerful Bayesian Theory of Mind formalism (Baker, Saxe, & Tenenbaum, 2009; Shafto et al., 2012; Rabinowitz et al., 2018), in which actions, goals, and causal world structure are linked together through approximately rational planning, allowing them to be inferred from limited evidence using Bayesian inference. These models are effortlessly productive–they can plan actions to achieve novel combinations of goals, infer an agent's intentions from their actions, and react instantly to changed environmental structure–and they account for a variety of behavioral data. However, they are often computationally intensive and difficult to scale to large state and hypothesis spaces.

By contrast, deep reinforcement learning systems have recently been shown to scale to large real-world problems, e.g. (Mnih et al., 2015), but they are often not productive (in their model-free incarnation), or similarly require computationally intensive rollouts through internal models (model-based RL).

Moreover, inferring an agent's goals from their actions in both deep RL and Bayesian frameworks often requires solving the inverse RL problem. Current IRL algorithms repeatedly solve reinforcement learning problems in their inner loop. For this reason, little work has focused on generic goal inference in neural network models, and still less has addressed complex phenomena like social causal learning (but see Rabinowitz et al. (2018)). One of the key problems in the standard MDP framework inherent in these approaches is that the optimal actions associated with different tasks do not typically compose together linearly. To take a spatial navigation example, suppose an agent learns two tasks to navigate to goal locations A and B respectively. The optimal policy for the composite task "navigate to goal A or B" is not a simple linear sum of the optimal actions for the component tasks. To begin to address this problem, we propose a neural network architecture that permits a distributed representation of tasks based on work relying on the LMDP formulation, which at the cost of additional constraints on the problem formulation enjoys a form of compositionality across tasks (Todorov, 2009; Saxe, Earle, & Rosman, 2017; Piray & Daw, 2019). The network can express an infinite set of tasks as weighted combinations of a finite set of prior 'basis' tasks, analogously to how an infinite set of images may be represented as a distributed pattern of activity across a basis of V1-like edge detectors. Leveraging this distributed representation, the network can represent novel tasks as a combination of previously learned tasks, immediately knowing how to act optimally once the task blend is known; or infer a novel goal for an agent by expressing it as a particular combination of previously learned tasks. So long as a novel input lies within the span of these basis tasks, it can be represented exactly with a distributed code, even if it has never been seen before. Here, we apply this network to a popular set of cognitive science tasks across action selection, goal inference, and social causal learning scenarios, and show that the resulting model can behave approximately like sophisticated Bayesian models (and human subjects), while retaining the flexibility of model-based reinforcement learning.

A Neural Network with an Intentional Stance

We base our formulation on the multitask linearly-solvable Markov decision process (MLMDP) framework introduced by (Todorov, 2009; Saxe et al., 2017), which we briefly re-

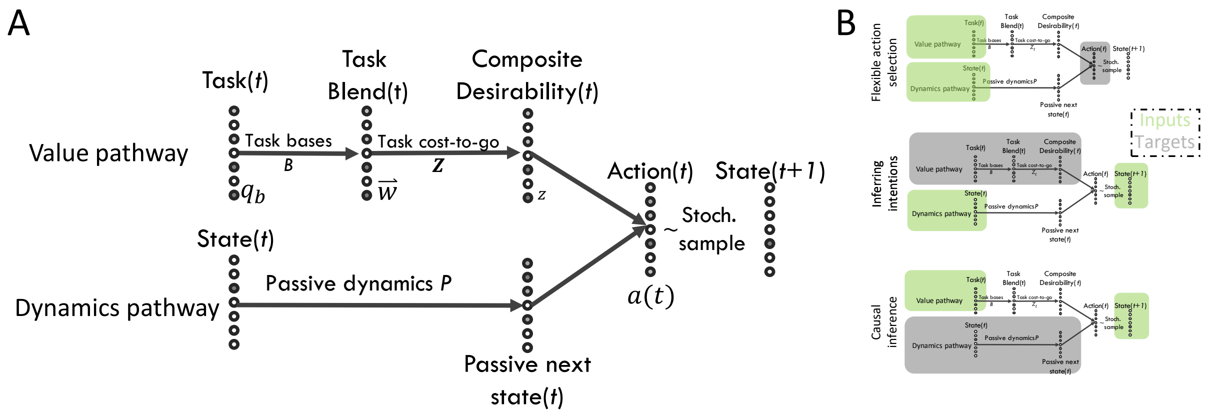

Figure 1: A neural network for intentional action reasoning. (A) The network contains a value pathway (top) and a dynamics pathway (bottom). The value pathway decomposes instantaneous rewards for reaching each terminal state using a set of task bases B. The resulting distributed task blend determines the composite desirability function for each state, i.e., the total expected future returns. The dynamics pathway receives a one-hot vector denoting the current state of the agent, and predicts the distribution over next states in the absence of exerting control (the 'passive' dynamics). The composite desirability and next state distribution are multiplied and renormalized to obtain a desired action, which is a distribution over next states. Sampling from this distribution yields the state of the agent at the next time point. The interaction of these elements is governed by a principle of rational action (see text), such that sampling from the action distribution will maximize expected reward. (B) Illustration of inputs given (green) and targets (grey) to be computed by the network for flexible action selection (top panel), inferring intentions (middle panel) and causal inference (bottom panel).

view here for completeness. The agent exists in a world with N discrete states. These may represent different spatial positions in a navigation task, different block configurations in the Tower of Hanoi task, or different equations in an algebraic task. We represent the current state of the agent at time t as a one-hot column vector $s(t) \in R^N$ with a one in the index corresponding to the current state of the agent, and a zero otherwise. The environment offers instantaneous rewards for arriving in each state, which can be collected into the column vector $r \in R^N$. If the agent does not exert control effort, the state transitions will evolve according to passive dynamics specified by the matrix $P \in R^{N\times N}$, where P_{ij} specifies the probability of transitioning from state j to state i.

Rather than choosing from a discrete set of actions as in the standard MDP framework, the agent's action in the LMDP framework is a choice of distribution over next states. This distribution can be written as the column vector $a(t) \in R^N$. The LMDP framework penalizes actions which deviate from the passive dynamics, thereby rewarding efficient action selection. In particular, the instantaneous reward for arriving in state s and taking action a is

$$\mathcal{R}(s,a) = r^T s - \lambda \mathrm{KL}(a||Ps) \tag{1}$$

where $KL(x||y)$ denotes the Kullback-Leibler divergence between the distributions x and y, and λ is a parameter controlling the strength of this KL control cost. We note in passing that the discrete actions available to the agent are usually recoverable from the non-zero entries in P, as these reflect "links" between states.

Here we adopt the finite exit formulation of the MDP, such that the set of states is divided into N_i internal states and N_b absorbing boundary states. The agent transitions through internal states accruing reward, and when the agent enters a state in the set of absorbing states $\mathcal{A}$, the episode terminates. The goal of the agent is to select actions which maximize total expected future reward,

$$a^* = \mathrm{argmax}_a \mathbb{E}_{\substack{s(t+1)\sim a(t)\\ \varepsilon=\min\{t:s(t)\in\mathcal{A}\}}} \left\{ r^T s(\varepsilon) + \sum_{t=0}^{\varepsilon-1} \mathcal{R}(s(t),a(t)) \right\}. \tag{2}$$

Following (Todorov, 2009), we define the *desirability* of a state as the exponentiated cost-to-go function of that state, which we collect into the vector $z \in R^N$ with components $z_k = e^{v_k/\lambda}$ where v_k is the cost-to-go (expected reward from state k if acting optimally). Analogously, we denote the exponentiated instantaneous rewards as the vector $q \in R^N$ with components $q_k = e^{r_k/\lambda}$. Further, we make use of the decomposition into N_i internal states and N_b absorbing terminal states and similarly partition the desirability, reward, and transition functions into their constituent internal and absorbing parts, e.g. z_i and z_b denote the instantaneous and boundary desirabilities, respectively.

As shown in (Todorov, 2009), this formulation has the benefit that the Bellman equation reduces to a linear system,

$$\left(I - M_i P_i^T\right) z_i = M_i P_b^T q_b, \quad (3)$$

where $M_i = diag(q_i)$. We may then solve for z_i explicitly; or iteratively in an analog of value iteration,

$$z_i \leftarrow M_i P_i^T z_i + M_i P_b^T q_b; \quad (4)$$

or through online exploration and a feedback-driven update known as Z-Learning (the LMDP analogue to Q-Learning), described in detail in (Todorov, 2009).

With the desirability function in hand, the optimal action in any state can be shown to be (Todorov, 2009)

$$a^*(t) \propto z \circ [Ps(t)] \quad (5)$$

where $\circ$ denotes the Hadamard (element-wise) product. That is, we take the product of the desirability of each state with its probability under the next state distribution for the passive dynamics. We then renormalize to ensure the action is a probability distribution, e.g. using the softmax function analogously to action sampling in standard RL. In this way, the passive dynamics are shifted toward states which offer greater expected future reward. The fact that the optimal policy takes this particularly simple form is a distinctive property of the LMDP framework.

So far, we have explained how optimal actions can be selected for one fixed reward structure in the LMDP. Now following Saxe et al. (2017), we exploit the linearity of the Bellman equation to enable acting towards diverse goals. We suppose we have learned optimal actions for a library of $\tau = 1, \cdots, N_\tau$ tasks, where each task shares P, q_i, and λ but differs in its boundary rewards q_b^τ. We can form the $N_b \times N_\tau$ task exponentiated reward matrix $Q = \left[q_b^1 \; q_b^2 \; \cdots \; q_b^{N_\tau}\right]$, which collects these boundary rewards in its columns, and compute the matrix of associated task desirability functions $Z_i = \left[z_i^1 \; z_i^2 \; \cdots \; z_i^{N_\tau}\right]$ as

$$Z_i = \left(I - M_i P_i^T\right)^{-1} M_i P_b^T Q. \quad (6)$$

Now suppose we receive a new task to perform, with immediate boundary rewards r, and suppose that we can express these exponentiated rewards $q = \exp(r/\lambda)$ as a linear combination of the task exponentiated reward matrix,

$$q = Qw \quad (7)$$

for some $w \in R^{N_\tau}$. Then we can compute the optimal desirability function z_i by simply blending the desirability functions of the component tasks linearly,

$$\begin{aligned} z_i &= \left(I - M_i P_i^T\right)^{-1} M_i P_b^T q & (8)\\ &= \left(I - M_i P_i^T\right)^{-1} M_i P_b^T Q w & (9)\\ &= Z_i w. & (10) \end{aligned}$$

Hence, this multitask network can optimally perform any task that lies within the span of its exponentiated rewards (we note that tasks must also satisfy the technical requirement $Qw \geq 0$ due to the exponential relationship between r and q).

To find appropriate task blend weights w for a given new task, define the task basis matrix $B = Q^\dagger$ where † denotes the pseudoinverse. We can take the exponentiated task rewards q, and pass them through the task basis matrix to obtain the required weights, $w = Bq$. The matrix B thus decomposes new reward functions into a weighted combination of previously learned tasks.

Overall then, the computation of optimal actions in the multitask LMDP framework can be cast as a particular neural network architecture, as shown in Fig. 1A. The network contains two pathways (values and dynamics). The value pathway receives the task to execute, specified as a vector q of instantaneous boundary rewards. These pass through synaptic weights corresponding to the task basis matrix B to come up with activations representing the current task blend w. This task blend is passed through synaptic weights encoding the desirability matrix Z to obtain the composite desirability function z, which constitutes the output of the value pathway. In essence, the value pathway transforms instantaneous rewards into total expected future rewards under the optimal policy for each state. The dynamics pathway receives the current state, encoded as a one-hot vector $s(t)$, and computes the distribution over next states using the passive dynamics matrix P. The dynamics pathway thus predicts how the state might evolve one time step into the future. Finally, the value and dynamics pathways converge to construct the optimal action, which is given by element-wise multiplication of the representations in each pathway followed by sum-normalization. In contrast to many neural network schemes, our architecture optimally performs any task that lies in the span of the task basis. Further, the 'synaptic weight' parameters in the pathways (B, Z, P) are linked by specific relations (the Bellman equation) which guarantee this optimality and build in a principle of rational action.

In the following we will illustrate three key benefits of this framework that all capitalise on the explicitly compositional architecture of the network. As depicted in Fig. 1B, we will show how the model can harness this architecture to fill in the gaps if one of the inputs is missing and how it responds to perturbations of the value pathway.

Flexible Action Selection

The choice problem we consider here is best described by differences in the reward function: How can the agent make decisions in novel situations when goals or reward have changed? Here, the intentional action network enables the agent to formulate novel policies by composing and blending previously learned tasks. Due to the linearity of the LMDP, the agent can harness its existing task base to behave optimally as long as the new optimal reward function lies within the span of the previously learned tasks. Importantly, this simple linear combination of previous tasks is not commonly optimal in standard MDP formulations. One notable excep-

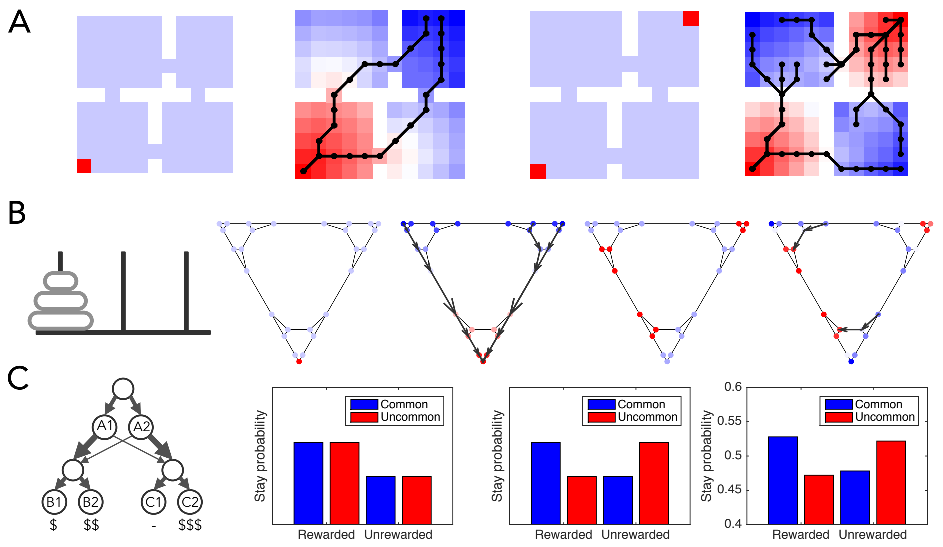

Figure 2: Flexible action selection. (A) After latent learning of the transition structure in a maze, the agent can immediately navigate to a new rewarded location (red squares). Sample trajectories shown in black are overlaid on the inferred composite desirability function. (B) Tower of Hanoi problem. The agent can solve the typical task (reward given for moving the tower to the right-most peg). The rules of the game allow only some transitions between states (edges between points in the graph). Novel goals such as 'place the large block on the middle peg' correspond to a collection of states being rewarded. The agent can optimally act toward this new goal by decomposing it into the pathways illustrated in Fig.1. (C) Revaluation in sequential choice. The classic 'two-step' task tests for the ability of agents to rapidly adapt to new reward contingencies (diagram to the left). It yields different patterns of responses for model-free (left bar plot) and model-based behaviour (middle bar plot) after an uncommon (low probability, in red) transition from the A states to following states. The multitask LMDP network (right bar plot) behaves like model-based systems, immediately adapting to changes in terminal reward, as it does contain a model but does not explicitly roll out possible futures.

tion to this is the Successor Representation and the associated Successor Features framework, where a new reward function may be represented linearly over previously learned policies (Momennejad, 2020; Borsa et al., 2019; Tomov, Schulz, & Gershman, 2021). However, unlike the MLMDP framework, these composite policies evaluate the value function for a specific policy, not the optimal policy, and are therefore not guaranteed to be optimal for new tasks.

For action selection, our results closely follow those in Saxe et al. (2017), but we apply the model to commonly studied tasks in cognitive science. We illustrate the network's behavior in Fig. 2 using three classical tasks from the cognitive sciences. Firstly, in the four rooms environment, the agent can immediately execute optimal policies to rewards at several locations once it has acquired knowledge about navigating to specific locations. Secondly, we consider the Tower of Hanoi task, where the agent needs to move a tower of discs onto a different peg (typically the right most peg), but can only place smaller pieces onto larger ones. Having learned the dynamics (in this case: the allowed transitions between states) and policies to individual states, the model can immediately compute the optimal policy for new targets (e.g. move the largest disc to the middle peg). Lastly, we demonstrate that the model can adapt to novel reward contingencies in the classic sequential two-step task (Doll, Simon, & Daw, 2012). Here, the network behaves like a model-based system (Piray & Daw, 2019). However, unlike many other model-based systems, the model does not engage in active multi-step planning, but rather contains the necessary information to adapt to new rewards in its value and dynamics pathways.

Inferring Intentions

To infer intentions, we make use of the fact that the log probability of a controlled trajectory transitioning through a specific sequence of states $s_1, \cdots, s_N$ has a simple expression (Dvijotham & Todorov, 2010). For our multitask setting, it is

$$L[w,P] = \sum_{n=1}^{N} \left[-\log w^T Z^T s_{n+1} + \log w^T Z^T P s_n + \log s_{n+1}^T P s_n \right]. \tag{11}$$

To infer the task blend that an agent is using, therefore, we can find the task blend weights w that maximize the probability of the observed trajectory. To do this, we perform gradient ascent on w,

$$\Delta w = \eta \frac{\partial L}{\partial w} = \eta \sum_{n=1}^{N} \left[\frac{Z^T s_{n+1}}{w^T Z^T s_{n+1}} - \frac{Z^T P s_n}{w^T Z^T P s_n} \right] \tag{12}$$

where η is a small learning rate. In essence, this procedure attempts to find the task blend most likely to generate the observed sequence of state transitions.

The gradient update in Eq. (12) requires observing and memorizing the complete trajectory for performing the inference. We can also create an approximate online scheme, in which we update the task blend as a running average of updates after observing each new state transition. After observing the n^{th} transition, we make the update

$$\Delta w = -\alpha w + \eta \left[\frac{Z^T s_{n+1}}{w^T Z^T s_{n+1}} - \frac{Z^T P s_n}{w^T Z^T P s_n} \right] \tag{13}$$

where the parameter α controls the length of the running average. This procedure is fully online, showing how approximate goal inference could be a relatively cheap and automatic operation once Z and P are known.

We illustrate the network's inference behavior in a task introduced by Baker et al. (2009). Human participants observed a sequence of state transitions in a grid maze and had to infer, after different numbers of steps, which of three possible goal locations the agent was aiming for (Fig. 3A, top panel). To simulate this, we presented the model several steps of a trajectory. The model then infers a task blend over the three possible goal locations, either using the entire observed trajectory (batch update, Fig. 3A, middle panel), or in a step-by-step fashion (online update, Fig. 3A, bottom panel). The results of the two update schemes are remarkably similar, with the differences in predicted goals slightly more pronounced in the batched case. Note that a sparsity constraint could be added here to associate more weight with the most likely goal. Importantly, once the task blend w is inferred, by switching back to selecting actions, the network can generate forward predictions of where the agent is likely to go next without needing to re-solve this forward problem for the inferred task weight blend (Fig. 3B grey). Here, the forward prediction is *greedy* with respect to controlled dynamics for the purpose of illustration. Note that the forward prediction depends on all entries in w, not just the highest weight. For instance, during the early phase of observation in the top panel in B, the inferred behaviour is indifferent between all three goals and the network may thus predict navigating to the closest (rather than the marginally most likely) goal. Fig. 3C shows a reproduction of Figure 4a-c of (Baker et al., 2009) across six example trajectories with our model fits added in the bottom row. Our results are qualitatively very similar to both observed participant behaviour and the Bayesian model applied in (Baker et al., 2009). Note, however, that our model would need to be augmented with a prior to capture that participants' inferences are not uniformly distributed across goals at the first judgment points.

Social Causal Learning

In social settings where the goal of the agent is known, we can also observe controlled state transitions in order to infer 'causal' structure, here formalized as the passive dynamics P that describes how states evolve in the absence of control input. We use the word 'causal' in line with prior work (Goodman, Baker, & Tenenbaum, 2009), but note that this may differ from other definitions of causality.

To perform this inference, we start again from the probability of observing a controlled state sequence given by Eq. (11). Now we observe the task blend w and a sequence of states $s_1, \cdots, s_N$, and we find the state transition matrix P which maximizes the probability of the sequence through projected gradient ascent, $\Delta P = \eta \frac{\partial L}{\partial P}$, where after each update we project P back to the probability simplex by clipping negative elements to zero and renormalizing each column to sum to one (we also note that when taking the derivative, Z depends on P).

The key feature of interest of this formulation is that the network's inferences depend critically on the observed task blend w. For instance, consider the case of observing an agent operate a machine whose mechanism (its passive dynamics P) is unknown and needs to be inferred. Settings of this type have been studied in developmental psychology, where children learn causal relations in the environment through social observation (Waismeyer, Meltzoff, & Gopnik, 2014). We illustrate such a setup in Fig. 4A. If the model (or child) knows the agent's intentions (via the task blends w), it can infer likely causal relationships under the assumption that the agent acts *rationally*, i.e. they behave optimally with respect to their intentions. For instance, if the network knows that the agent wants the tortoise, and sees the agent place both tokens in the machine, it infers that the machine only dispenses the tortoise toy when given both tokens (not one or the other) (Fig. 4B). This inference can be made even before the machine in fact produces the tortoise. However, if the agent's intentions are unknown to the model (e.g. uniform weights w), the model will make no such attributions (Fig. 4C).

Goal-directed Induced Movement

Finally, formulating the multitask LMDP as a neural network allows us to make links to neural recording and stimulation studies. For example, (Graziano, Aflalo, & Cooke, 2005) stimulate locations in monkey pre-motor cortex and find that this elicits limb movements that end in a stereotyped location or limb configuration irrespective of starting configuration as if these neurons induced behaviour towards a 'goal'. We simulate this here by assuming that a 2D space is mirrored in the neural code, such that adjacent locations in physical space are also represented in adjacent locations on the cortical surface, similar to retinotopic maps in V1. We then stimulate a given location (e.g. the centre point) in the value pathway, elevating its desirability and that of surrounding points, and pass this perturbed activation through the network. This elicits a feedback control loop towards the stimulated location, Fig. 5A. By contrast, randomly perturbing the weight vector w (to reflect intrinsic noise) elicits random movements similar to those observed after 'mock-stimulation' (where no current is applied) in (Graziano et al., 2005), Fig. 5B.

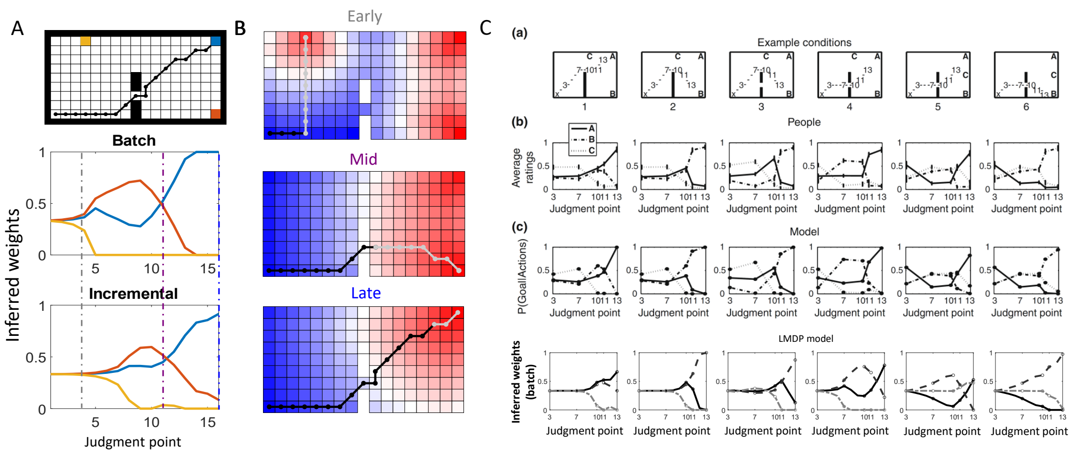

Figure 3: Model predicted intentions given a sequence of observations. (A) Task blends across possible goals predicted by the model at each judgment point. The top panel shows the example trajectory, starting from the bottom left and ending at the top right. The middle panel shows the results from a batch update (entire trajectory), whereas the bottom panel shows the predictions using an incremental, step-by-step update. (B) Model predicted (optimal) desirability functions overlaid with observed (black) and predicted (grey) trajectories at early, mid, and late stages of observations. This illustrates how the model can generate optimal desirability functions from the inferred task blends and use these to make forward predictions. (C) Comparison of our model (bottom row) in the task employed by (Baker et al., 2009) (top three rows). Row (a) shows decision setting and judgment points (numbered), row (b) shows averaged participant inferences over goals, row (c) shows the model fits from the original paper, and the bottom row shows the model fits obtained from our model in eq. 12. Reproduced with permission according to creative commons license CC BY-NC-SA 3.0 from: `http://hdl.handle.net/1721.1/60852`

Discussion

Here, we presented a network model based on a recent multi-task extension of the LMDP framework (Saxe et al., 2017) which gives rise to compositionality across learned tasks. This enables distributed representations of task goals, which can be blended linearly to produce novel, optimal task policies so long as they lie within the span of learned tasks. In the context of action selection, this scheme shares some similarities with other recent advances, for instance the successor representation (Momennejad, 2020), successor features (Borsa et al., 2019), and the default representation (Piray & Daw, 2019). Going beyond this, the network model can also capture well-known effects from social and causal learning by extending the inverse optimal control described in (Dvijotham & Todorov, 2010) to the multi-task case. Notably, inferring the intentions of other agents is achieved through a cheap and simple iterative scheme that updates the likelihoods of each hypothesis in proportion to the sum of two simple ratios, which only depend on the passive dynamics of the environment and the previously learned task bases. The network can readily be combined with other recent work (Earle, Saxe, & Rosman, 2018) to scale the network to larger state spaces using hierarchical decompositions of task bases. The present work only considers the simple case in which most aspects of the task are known and have been learned perfectly. Future work could build on our results using well-established Z-learning algorithms to investigate the development of these abilities, as they are learned in an incremental fashion. Another exciting future direction would be to use function approximators at every layer of the network to avoid the need to pre-specify the state space and task bases. Together, these directions would add flexibility to the framework.

Acknowledgments This work was supported by a Sir Henry Dale Fellowship to A.S. from the Wellcome Trust and Royal Society (grant number 216386/Z/19/Z), and CIFAR Azrieli Global Scholar programme.

References

Baker, C., Saxe, R., & Tenenbaum, J. (2009). Action understanding as inverse planning. *Cognition*, *113*(3), 329–49.

Borsa, D., Barreto, A., Quan, J., Mankowitz, D., Munos, R., Van Hasselt, H., ... Schaul, T. (2019). Universal successor features approximators. *7th International Conference on Learning Representations, ICLR 2019*, 1–24.

Doll, B. B., Simon, D. A., & Daw, N. D. (2012). The ubiquity of model-based reinforcement learning. *Current Opinion in Neurobiology*, *22*(6), 1075–1081.

Dvijotham, K., & Todorov, E. (2010). Inverse optimal control with linearly-solvable mdps. *Proceedings of the 27th International Conference on Machine Learning*.

Earle, A., Saxe, A., & Rosman, B. (2018). Hierarchical subtask discovery with non-negative matrix factorization. In *International conference on learning representations.*

Goodman, N., Baker, C., & Tenenbaum, J. (2009). Cause and intent: Social reasoning in causal learning. In *Proceedings*

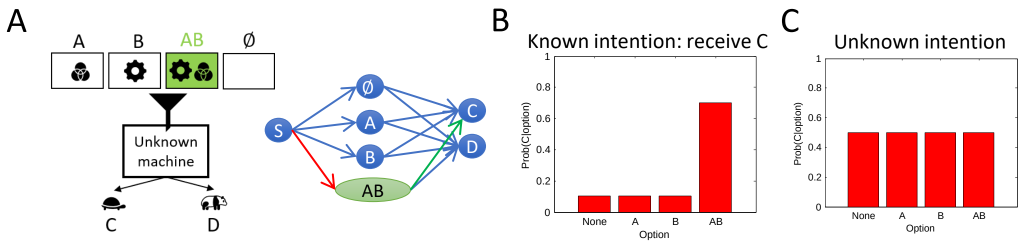

Figure 4: Network model applied to social causal learning. (A) The observer sees a machine with unknown mechanisms which can be fed with one of four combinations of tokens and can return one of two different toys. The associated LMDP state space is shown to the right. (B) If the observer knows the agent's intention (in this case: to receive outcome C) and observes that the agent feeds both tokens A and B to the machine, the model infers that both tokens are necessary to receive the desired outcome, as illustrated by the higher probability mass associated with a transition from state AB to C. (C) By contrast, if the model observes the same state transition, but does not know the agent's intentions, it makes no specific causal attribution.

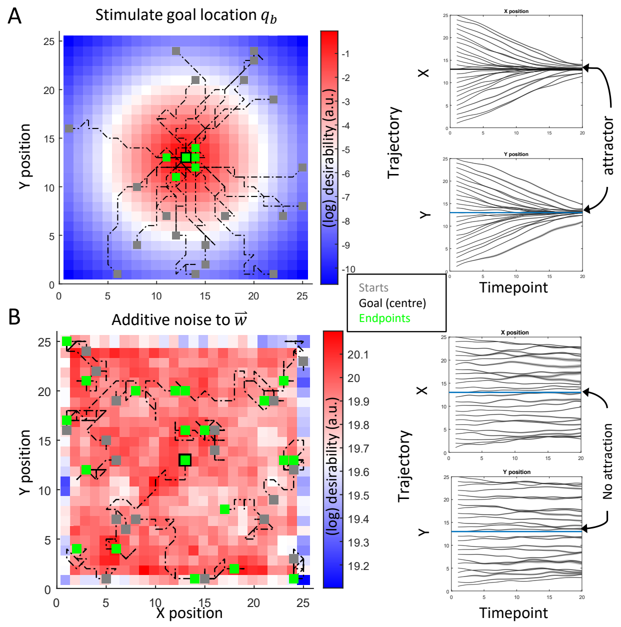

Figure 5: Perturbation analysis of network. (A) Perturbation applied to a spatial location, increasing desirability of stimulation site and surrounding locations. The resulting vector q_b is then simply passed through the network to yield trajectories in 2D space. (B) Random noise control analysis, where the weight vector w is perturbed by Gaussian noise. This results in random walks through 2D space, with no single attractor. Simulations based on (Graziano et al., 2005)

of the thirty-first annual conference of the cognitive science society (p. 2759–2764).

Graziano, M. S., Aflalo, T. N., & Cooke, D. F. (2005). Arm movements evoked by electrical stimulation in the motor cortex of monkeys. *Journal of Neurophysiology*, *94*, 4209–4223.

Heyes, C. M., & Frith, C. D. (2014). The cultural evolution of mind reading. *Science*, *344*(6190), 1243091–1243091.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., . . . et al. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*, 529–533.

Momennejad, I. (2020). Learning structures: Predictive representations, replay, and generalization. *Current Opinion in Behavioral Sciences*, *32*, 155–166.

Piray, P., & Daw, N. D. (2019). A common model explaining flexible decision making, grid fields and cognitive control. *bioRxiv*.

Rabinowitz, N. C., Perbet, F., Song, H. F., Zhang, C., Eslami, S. M. A., & Botvinick, M. (2018). Machine theory of mind. *arXiv:1802.07740 [cs]*.

Saxe, A., Earle, A., & Rosman, B. (2017). Hierarchy through composition with multitask lmdps. *International Conference on Machine Learning*.

Shafto, P., Goodman, N. D., & Frank, M. C. (2012). Learning from others: The consequences of psychological reasoning for human learning. *Perspectives on Psychological Science*, *7*(4), 341–351.

Todorov, E. (2009). Efficient computation of optimal actions. *Proceedings of the National Academy of Sciences of the United States of America*, *106*(28), 11478–11483.

Tomov, M. S., Schulz, E., & Gershman, S. J. (2021). Multitask reinforcement learning in humans. *Nature Human Behaviour*, 1–10.

Waismeyer, A., Meltzoff, A. N., & Gopnik, A. (2014). Causal learning from probabilistic events in 24-month-olds: an action measure. *Developmental science*, 1–8.