# Compositional clustering in task structure learning

Nicholas Franklin[1,2,*], Michael J. Frank[2,3,†]

**1** Department of Psychology, Harvard University, Cambridge, MA, USA
**2** Department of Cognitive, Linguistic & Psychological Sciences, Brown University, Providence, RI, USA
**3** Brown Institute for Brain Science, Brown University, Providence, RI, USA

* nfranklin@fas.harvard.edu
† michael_frank@brown.edu

## Abstract

Humans are remarkably adept at generalizing knowledge between experiences in a way that can be difficult for computers. Often, this entails generalizing constituent pieces of experiences that do not fully overlap, but nonetheless share useful similarities with, previously acquired knowledge. However, it is often unclear how knowledge gained in one context should generalize to another. Previous computational models and data suggest that rather than learning about each individual context, humans build latent abstract structures and learn to link these structures to arbitrary contexts, facilitating generalization. In these models, task structures that are more popular across contexts are more likely to be revisited in new contexts. However, these models predict that structures are either re-used as a whole or created from scratch, prohibiting the ability to generalize constituent parts of learned structures. This contrasts with ecological settings, where some aspects of task structure, such as the transition function, will be shared between context separately from other aspects, such as the reward function. Here, we develop a novel non-parametric Bayesian agent that forms independent latent clusters for transition and reward functions that may have different popularity across contexts. We compare this agent to an agent that jointly clusters both across a range of task domains. We show that relative performance of the two agents depends on the statistics of the task domain, including the mutual information between transition and reward functions in the environment, and the stochasticity of the observations. We formalize our analysis through an information theoretic account of the priors, and develop a meta learning agent that can dynamically arbitrate between strategies across task domains. We argue that this provides a first step in allowing for compositional structures in reinforcement learners, which should be provide a better model of human learning and additional flexibility for artificial agents.

## Author summary

A musician may learn to generalize behaviors across instruments for different purposes, for example, reusing hand motions used when playing classical on the flute to play jazz on the saxophone. Conversely, she may learn to play a single song across many instruments that require completely distinct physical motions, but nonetheless transfer knowledge between them. This degree of compositionality is often absent from computational frameworks of learning, forcing agents either to generalize entire learned policies or to learn new policies from scratch. Here, we propose a solution to this problem that allows an agent to generalize components of a policy independently and compare it to an agent that generalizes components as a whole. We show that the degree to which one form of generalization is favored over the other is dependent on the features of task domain, with independent generalization of task components favored in environments with weak

relationships between components or high degrees of noise and joint generalization of task components favored when there is a clear, discoverable relationship between task components. Furthermore, we show that the overall meta structure of the environment can be learned and leveraged by an agent that dynamically arbitrates between these forms of structure learning.

# Introduction

Compared to artificial agents, humans exhibit remarkable flexibility in our ability to rapidly, spontaneously and appropriately learn to behave in unfamiliar situations, by generalizing past experience and performing symbolic-like operations on constituent components of knowledge [1]. Formal models of human learning have cast generalization as an inference problem in which people learn a (latent) shared task structure across multiple contexts and then infer which causal structure best suits the current scenario [2, 3]. In these models, a context, typically an observable (or partially observable) feature of the environment, is linked to a learnable set of task statistics or rules. Based on statistics and the opportunity for generalization, the learner has to infer which environmental features (stimulus dimensions, episodes, etc.) should constitute the context that signals the overall task structure, and, simultaneously, which features are indicative of the specific appropriate behaviors for the inferred task structure. This learning strategy is well captured by Bayesian nonparametric models, and neural network approximations thereof, that impose a hierarchical clustering process onto learning task structures [3, 4]. A learner infers the probability that two contexts are members of the same task cluster via Bayesian inference, and in novel situations, has a prior to reapply the task structures that have been more popular across disparate contexts, while also allowing for the potential to create a new structure as needed. Empirical studies have provided evidence that humans spontaneously impute such hierarchical structure, which facilitates future transfer, whether or not it is immediately beneficial – and, indeed, even if it is costly – to initial learning [3–5].

These clustering models can account for aspects of human generalization that are not well explained by standard models of learning. This approach to generalization, treating multiple contexts as sharing a common task structure, is similar to artificial agents that reuse previously learned policies in novel tasks when the statistics are sufficiently similar [6–9]. However, a key limitation to these models of both human and artificial learning is that the task statistics are typically generalized as a unit. That is, in a new context, a previously learned policy can either be reused or a new policy must be learned from scratch. This can be problematic as policies are often not robust to untrained variation in task structure [11]. Thus, a previously learned policy can lead to a poor outcome in a new context even if there is a substantial degree of shared structure.

Because task structures are either reused or not as a whole, the ability to reuse and share component parts of knowledge is limited; that is, they are not *compositional*. Compositionality, or the ability to bind (compose) information together in a rule governed way, has long been thought to be a core aspect of human cognition [1, 10]. Importantly, ecological contexts often share a partial structure, limiting the applicability of previously learned policies but nonetheless providing a generalization advantage to a compositional agent.

To provide a naturalistic example, an adept musician can transfer a learned song between a piano and a guitar, even as the two instruments require completely different physical movements, implying that goals can be generalized and reused independently of the actions needed to achieve them. A clustering model that generalizes entire task structures cannot account for this behavior, and would require instead that an agent would need to relearn a song from scratch to play it on a new instrument. Worse, this clustering scheme would predict an unlikely interference effect where the similar outcome of playing the same song on two instruments results in the model incorrectly pooling motor policies across instruments.

This inability to share partial information across contexts is not only a limitation of existing models of human and animal generalization [2–5] but also pervades algorithms within artificial intelligence [6–8]. As such, artificial agents face similar problems when the structure of a contexts

changes in an unexpected way. For example, a robot that has learned to transfer a package from one room to another would need to relearn to do so if an obstacle is put in its way. To effectively generalize, an agent needs to structure their experiences in a useful way, clustering related experiences and segregating unrelated experiences.

Here, we propose a framework to address one aspect of compositionality, by decomposing task structures – and their separable potential for clustering – into reward functions and transition functions. These two independent functions of a Markov decision process are suitable units of generalization: if we assume that an agent has knowledge of a state-space and the set of available actions, then the reward and transition functions are sufficient to determine the optimal policy. In real-world scenarios, a reward function may correspond to the objective of an agent (what it would like to achieve, and the enviromental states that produce these goals). A transition function determines how the agent's actions affect its environment (i.e., the subsequent states). For example, when playing music a reward function might correspond to the desired sequence of notes (a scale, or a song) while the transition function might correspond to the actions needed to produce notes on an instrument. When picking up a new form of guitar, it may be sufficient for a musician to play one or two strings which may then afford inference of the entire tranisiton functions (the tuning: strings and frets needed to obtain each note). Here, we are concerned with how the inference of one (reward or transition) function affects generalization of the other.

We consider two approaches to clustering and compare their relative generalization advantages as a function of environmental statistics. The *independent clustering* agent supports generalization by clustering contexts into orthogonal sets defined by the reward and transition statistics, respectively. In contrast, the *joint clustering* agent clusters contexts into a single set of clusters that binds together the transition and reward functions (hence amounting to previous models of task-set structure [3–5]). Necessarily, independent clustering is compositional and requires the binding of two independent functions.

We show that these two models lead to different predictions depending on the task environment, and we provide an information theoretic analysis to formalize and quantify the bounds of these advantages/disadvantages. In environments where there is a clear, discoverable relationship between transitions and rewards, joint clustering facilitates generalization by allowing an agent to infer one function based on observations that are informative about the other. Nonetheless, we show that independent clustering can lead to superior generalization even in such cases when the transition-reward relationship is weak, difficult to discover, or costly to do so. Finally, we develop a meta-structure learning agent that can infer whether the overall environment is better described by independent or joint statistics.

# Model Description

To provide a test-bed for characterizing the effects compositional structure, we consider a series of navigation tasks by utilizing grid worlds as a simplification of real-world environments. In these grid worlds, an agent learns to navigate by learning transition functions (the consequences of its actions in terms of subsequent states) and separately learns a reward function (the reward values of locations, or goals) as it navigates. At each point in time, the agent is given a state tuple $s =< x, c >$ where $x \in \mathbb{R}^d$ is a vector of state variables (for example, a location vector in coordinate space) and $c \in \mathbb{R}^n$ is a context vector. Here, we define "context" as a vector denoting some mutable property of the world (for example, the presence or absence of rain, an episodic period of time, etc.) that constrain the statistics of the task domain, whereby these task statistics are consistent for each context that cues the relevant task. Formally, for each context we can define a Markov decision process (MDP) with state variables $x \in X$, actions $a \in A$, a reward function mapping state variables and actions to a real valued number $R : X \times \mathcal{A} \to \mathbb{R}$, and a transition function mapping state variables and actions to a probability distribution over successors $T : X \times \mathcal{A} \to \Pi(X)$.

For the purpose of simplicity, we assume that the agent knows the spatial relationship between states (i.e., it has access to a spatial map of its current position and adjacent positions) but has to

learn how its actions take it from one state to another. Specifically, we assume the agent knows a set of *cardinal movements* $A \in \mathcal{A}_{card}$, where each cardinal movement is a vector that defines a change in the state variables with regard to the known spatial structure per unit time. (For example, in a two dimensional grid world we can define North $= \langle dx/dt, dy/dt \rangle = \langle 0, 1 \rangle$ as a cardinal movement). We can thus define a transition function in terms of cardinal movements $T(x, A, x'|c) = \Pr(x'|x, A, c)$ and cast the navigation problem as the learning of some function that maps primitive actions to cardinal movements $\phi_c : \mathcal{A} \rightarrow \mathcal{A}_{card}$, which we assume to be independent of location. This simplifying assumption has the benefit of providing a model of human navigation, whom we assume understand spatial structure. Note that the function mapping motor actions onto cardinal movements can depend on environmental conditions, and thus, context (for example, wind condition can change the relationship between primitive actions and movements in space for an aerial drone). Similarly, we can express the reward function in terms of cardinal movements based on a location in space, $R_c(x, A) = \Pr(r|x, A, c)$. This allows us to consider how the agent receives reward as it moves through coordinate space (as opposed to how it receives reward as a function of its actions). Alternatively, we can express the reward function as $R(x, x')$ or more simply as $R(x')$. The key assumption here is that the reward function is not a function of the agent's actions but is a function of the consequences of those actions.

The task of the agent is to generate a policy (a function mapping state variables to primitive actions, for each context; $\pi^*|c : X \rightarrow \mathcal{A}$) that maximizes its expected future discounted reward [13]. Given a known transition function and reward function, the optimal policy given this task can be defined as:

$$\pi_c^*(x) = \arg\max_a \left[ \sum_{A \in \mathcal{A}_{card}} \phi_c(a, A) \sum_{x' \in X} T_c(x, A, x') \left[ R_c(x') + \gamma V_c(x') \right] \right] \tag{1}$$

where $V_c(x)$ is the optimal value function is defined by the Bellman equation:

$$V_c(x) = \max_A \left[ \sum_{x' \in X} T_c(x, A, x') \left[ R_c(x') + \gamma V_c(x') \right] \right] \ \forall \ x \in X \tag{2}$$

As the relationship between locations in space, $T_c$, is known to the agent, it is sufficient to learn the cardinal *mapping function* $\phi_c(a, A)$ and *reward function* $R_c(x, A)$ to determine an optimal policy.

While the optimal policy is dependent on both the mapping function and reward function, crucially, the optimal value function is not: it is dependent only on the reward function (and the known transition function $T_c$). Consequently, an agent can determine an optimal policy as a function of movements through space:

$$\pi_c^*(x) = \arg\max_A \left[ \sum_{x' \in X} T_c(x, A, x') \left[ R(x') + \gamma V_c(x') \right] \right] \tag{3}$$

This allows the agent to learn how it can take an action to move through space – the mapping function $\phi_c(a, A)$ – independently from the desirability of the consequences of these moves $R_c(x')$. This distinction allows for compositionality during generalization, as we will discuss in the following section.

## Context clustering as generalization

A common strategy to support task generalization is to cluster contexts together, assuming they share the same task statistics, if doing so leads to an acceptable degree of error [7]. This logic underlies models of animal Pavlovian learning and transfer [2], human instrumental learning and transfer [3, 4], and category learning [14]. Clustering models of human generalization typically rely on a non-parametric Dirichlet process, commonly known as the Chinese restaurant process (CRP), which acts as a clustering prior in a Bayesian inference process. Used in this way, the CRP enforces

popularity-based clustering to partition observations, so that the agent will be most likely to reuse those tasks that have been most popular across disparate contexts (as opposed to across experiences; [4]), and has the attractive property of being a non-parametric model that grows with the data [16]. Consequently, it is not necessary to know the number of partitions *a priori* and the CRP will tend to parsimoniously favor a smaller number of partitions.

As in prior work, we model generalization as the process of inferring the assignment of contexts $k = \{c_{1:n}\}$ into clusters that share common task statistics. But here, we decompose these task statistics to consider the possibility that that all contexts $c \in k$ share either the same reward function and/or mapping function, such that $R_k(x, A) = R_c(x, A)\ \forall\ c \in k$ and/or $\phi_k(a, A) = \phi_c(a, A)\ \forall\ c \in k$. (We return to the "and/or" distinction, which affects whether clustering is independent or joint across reward and mapping functions, in the following section). Formally, we define generalization as the inference

$$\Pr(c \in k|\mathcal{D}) \propto \Pr(\mathcal{D}|k)\Pr(c \in k) \tag{4}$$

where $\Pr(\mathcal{D}|k)$ is the likelihood of the observed data $\mathcal{D}$ given cluster $k$, and $\Pr(c \in k)$ is a prior over the clustering assignment. As in previous models of generalization, we use the CRP as the cluster prior. If contexts $\{c_{1:n}\}$ are clustered into $N \leq n$ clusters, then the prior probability for any new context $c_{n+1} \notin \{c_{1:n}\}$ is:

$$\Pr(c_{t+1} \in k|c_{1:n}) = \begin{cases} \frac{N_k}{N+\alpha} & \text{if } k \leq K_n \\ \frac{\alpha}{N+\alpha} & \text{if } k = K_n + 1 \end{cases} \tag{5}$$

where $N_k$ is the number of contexts associated with cluster $k$ and $K_n$ is the number of unique clusters associated with the n observed contexts. If $k \leq K_n$, then $k$ is a previously encountered cluster, whereas if $k = K_n + 1$, then $k$ is a new cluster. The parameter $\alpha$ governs the propensity to assign a new context to a new cluster, that is to create a new task. Higher values of $\alpha$ lead to a greater prior probability that a new cluster is created and favors a more expanded task space overall, leading to reduced likelihood of reusing old tasks. Thus, the prior probability that a new context is assigned to an old cluster is proportional to the number of contexts in that cluster (popularity), and the probability that it is assigned to a new cluster is proportional to $\alpha$. As a non-parametric generative process, the prior allows the number of clusters to grow as new contexts are observed. Importantly, this process is exchangeable, and as such, the order of observation does not alter the inference of the agent [16].

## Independent and joint clustering

As we noted above, there are two key functions the agent learns when navigating in a context: $\phi_c(a, A)$ and $R_c(x, A)$. These functions imply that the agent could cluster $\phi_c(a, A)$ and $R_c(x, A)$ jointly, or it could cluster them independently, such that it learns the popularity of each marginalizing over the other. Formally, the *independent clustering* agent [Fig 1, left] assigns each context $c$ into two clusters via Bayesian inference as in [Eq. 4] and using the CRP prior for each cluster [Eq. 5]. The likelihood function for the two assignments are the mapping and reward functions

$$L(\mathcal{D}|k_\phi) = \phi_k(a, A) \tag{6}$$

and

$$L(\mathcal{D}|k_R) = R_k(x, A), \tag{7}$$

respectively. Conversely, the *joint clustering* agent [Fig 1, right] assigns each context $c$ into a single cluster $k$ via Bayesian inference [Eq. 4] and, like the independent clustering agent, uses the CRP as the prior over assignments [Eq. 5]. The likelihood function for the context-cluster assignment is the product of the mapping and reward functions

$$L(\mathcal{D}|k) = \phi_k(a, A)R_k(x, A) \tag{8}$$

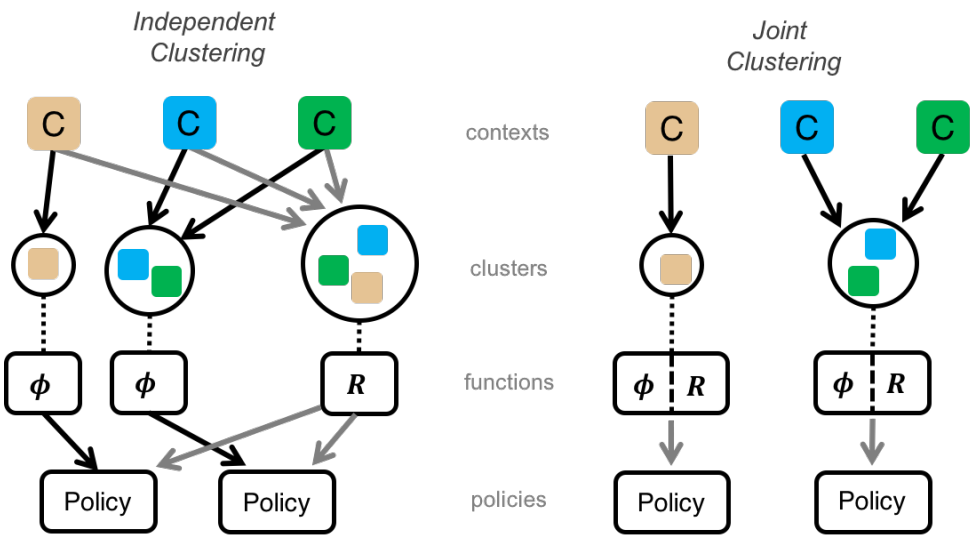


**Fig 1. Schematic example of independent and joint clustering agents.** *Left*: Independent clustering groups each context into two clusters, each associated with a reward ($R$) and mapping ($\phi$) function. Planning combines these functions to generates a policy. Popular clusters will be more likely to associated with new contexts. *Right*: Joint clustering assigns each context into a cluster linked to both functions (i.e., assumes a holistic task structure). In this example, both agents generate the same two policies for the three contexts but independent clustering generalizes the reward function across all three contexts.

The joint clustering agent is highly similar to previous non-compositional models of task structure learning and generalization, which have previously been shown to account for human behavior (but without specifically assessing the compositionality issue) [3, 4]. For the purpose of comparison with the independent clustering agent, here the joint clustering agent separately learns the functions $R$ and $\phi$. Previous models, in contrast, have learned model-free policies directly. In the general case, joint clustering does not require the separate representation of policy components ($R$ and $\phi$) nor does it require a binding operation in the form of planning. However, independent clustering does require the separate representation of policy components that must be bound via planning. Hence, one potential difference between the two approaches is algorithmic complexity, where joint clustering may permit a less complex and computationally costly learning algorithm than independent clustering. In the simulations below, we have equated the agents for algorithmic complexity and examine how inferring the reward and transition functions separately or together effect performance across task domains.

## Grid world simulations

We first consider two sets of simulations to expose the complementary advantages afforded by the two sorts of clustering agents depending on the statistics of the task domain, using a common set of parameters. A third simulation explores how the benefits of clustering can compound in a modified form of the *Rooms* problems previously used to motivate the benefits of hierarchical RL approaches [17, 18]. We show how our independent clustering model with hierarchical decomposition of the state space can facilitate more rapid transfer than that afforded by standard approaches.

Finally, we conduct an information theoretic analysis to formalize the more general conditions under which each scheme is more beneficial. Code for all of the simulations presented here have been made available in our GitHub repository: `https://github.com/nicktfranklin/IndependentClusters`

## Simulation 1: Independence of Task Statistics

In the first set of simulations, we simulated a task domain in which four contexts involving different combinations of reward and transition functions. In every trial, a "goal" location was hidden in a 6x6 grid world. Agents were randomly placed in the grid world and explored action selection until the goal was reached, at which point the trial ended and the next trial began, with the agent again randomized to a new location. The agent's task is to find the goal location (encoded as +1 reward for finding the goal and a 0 reward otherwise) as quickly as possible. The agents had a set of eight actions available to them $\mathcal{A} = \{a_1, ...a_8\}$, which could be mapped onto one of four cardinal movements $\mathcal{A}_{card} = \{\text{North}, \text{South}, \text{East}, \text{West}\}$. The agents were exposed to four trials, in which goal locations and mappings were stationary, for each context. Each of the four contexts had a unique combination of one of two goal locations and one of two mappings [Fig 2, left], and hence knowledge about the reward function or mapping function for any context was not informative about the other function. However, because the reward and mapping functions were each common across two contexts, the independent clustering agent can leverage the structure to improve generalization without being bound to the joint distribution of mappings and rewards.

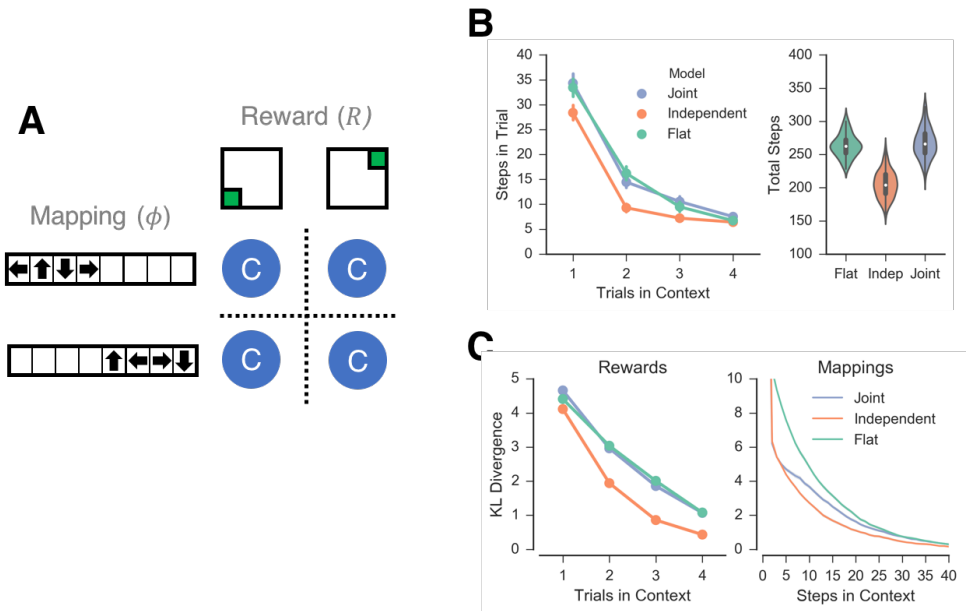


**Fig 2. Simulation 1** *A*: Schematic representation of the task domain. Four contexts (blue circles) were simulated, each paired with a unique combination of one of two goal locations (reward functions) and one of two mappings. *B*: Number of steps taken by each agent shown across trials within a single context (left) and over all trials (right). Fewer steps reflect better performance. *C*: KL-divergence of the models' estimates of the reward (left) and mapping (right) functions as a function of time. Lower KL-divergence represents better function estimates. Time shown as the number of trials in a context (left) and the number of steps in a context collapsed across trials (right) for clarity.

In addition to the independent and joint clustering agents, for comparison, we also simulated a "flat" (non-hierarchical decomposition of "context" and "state") agent that does not cluster contexts

at all and hence has to learn anew in each context. (The flat agent is a special case of both the independent and joint clustering agents such that $k_i = \{c_i\}\ \forall\ i$). We used hypothesis-based inference, where each hypothesis comprised a proposal assignment of contexts in to clusters, $h : c \in k$, defined generatively, such that when a new context is encountered the hypothesis space is augmented. For each hypothesis, maximum likelihood estimation was used to generate the estimates $\hat{\phi}_k(a, A) = \hat{\Pr}(A|a, k)$ and $\hat{R}_k(x) = \hat{\Pr}(r|x)$. To encourage optimistic exploration, $\hat{R}_k(x)$ was initialized to the maximum observable reward ($\Pr(r|x) = 1$) with a low confidence prior using a conjugate beta distribution of $\text{Beta}(0.01, 0)$.

The belief distribution over the hypothesis space is defined by the posterior probability of the clustering assignments [Eq. 4]. Calculating the posterior distribution over the full hypothesis space is computationally intractable, as the size of the hypothesis space grows combinatorially with the number of contexts. As an approximation, we pruned hypotheses with small probability (less than 1/10x posterior probability of the maximum *a posteriori* (MAP) hypothesis) from the hypothesis space. We further approximated the inference problem by using the MAP hypothesis, rather than sampling from the entire distribution, during action selection [3, 19]. Value iteration was used to solve the system of equations defined by [Eq. 3] using the values of $\hat{\phi}_k(a, A)$ and $\hat{R}_k(x)$ associated with the MAP hypothesis(es). A state-action value function, defined here in terms of cardinal movements

$$\hat{Q}_k(x, A) = \hat{R}_k(x, A) + \gamma \sum_{x' \in X} T(x, A, x')\hat{V}_k(x') \tag{9}$$

was used with a softmax action selection policy to select cardinal movements:

$$\Pr(A|x, k) = \frac{e^{\beta \hat{Q}_k(a,A)}}{\sum_{A \in \mathcal{A}_{card}} e^{\beta \hat{Q}_k(a,A)}} \tag{10}$$

where $\beta$ is an inverse temperature parameter that determines the tendency of the agents to exploit the highest estimated valued actions or to explore. Lower level primitive actions (needed to obtain the desired cardinal movement) were sampled using the mapping function:

$$\Pr(a|x, k) = \sum_{A \in \mathcal{A}_{card}} \hat{\phi}_k(a, A) \Pr(A|x, k) \tag{11}$$

We first simulated the independent and joint clustering agents as well as the flat agent on 150 random task domains using the parameter values $\gamma = 0.75$, $\beta = 5.0$ and $\alpha = 1.0$ (below we consider a more general parameter-independent analysis). Each of the four contexts was repeated 4 times for a total of 20 trials. The independent clustering agent completed the task more quickly than either other agent, completing all trials in an average of 205.2 (s=20.2) steps in comparison to 267.4 (s=22.4) and 263.5 (s=17.4) steps for the joint clustering and flat agents, respectively [Figure 2, B]. (We confirmed here and elsewhere that these differences were highly significant (e.g., here, the relevant comparisons are a minimum of $p < 1e{-}77$)).

In this case, the performance advantage of independent clustering is largely driven by faster learning of the reward function, as indexed by the KL-divergence between the agents' estimates of the reward function compared to a flate learner [Fig 2, C, left]. In contrast, both joint and independent clustering show generalization benefit when learning the mapping function [Figure 2, C, right]. This difference reflects an information asymmetry: in a new context, more information is available earlier to an agent about the mappings than the rewards, given that the latter are largely experienced when reaching a goal. (For example, in these environments, the first action in a novel context yields 2 bits of information about the mappings and an average of 0.07 bits of information about the rewards). As a consequence of this asymmetry, observing an element of a mapping can facilitate generalization of the rest of the mapping via the likelihood function, whereas observing unrewarded squares in the grid world tells the agent little about the location of rewarded squares.

In sum, as expected, independent clustering exhibited advantages over joint clustering in a task environment for which the transition and reward functions were orthogonally linked across contexts.

## Simulation 2: Dependence in Task Statistics

We next simulated all three agents on separate task domain in which there was a discoverable relationship between the reward and mapping functions across contexts, such that knowledge about one function is informative about the other. There were with four orthogonal reward functions and four orthogonal mappings across eight contexts, with each pairing of a reward and mapping function repeated across two contexts, permitting generalization [Figure 3, A]. As before, 150 random task domains were simulated for each model using the parameter values $\gamma = 0.75$, $\beta = 5.0$ and $\alpha = 1.0$. Each of the eight contexts was repeated 4 times for a total of 20 trials. In these simulations, both clustering agents show a generalization benefit, completing the task more quickly than the flat agent [Fig 3, B]. The joint clustering agent showed the largest generalization benefit, completing all trials in average of 384.2 (s=23.6) steps in comparison to 445.1 (s=33.4) for the independent clustering agent and 526.0 (s=26.4) steps for the flat agents. Again, these differences were highly significant.

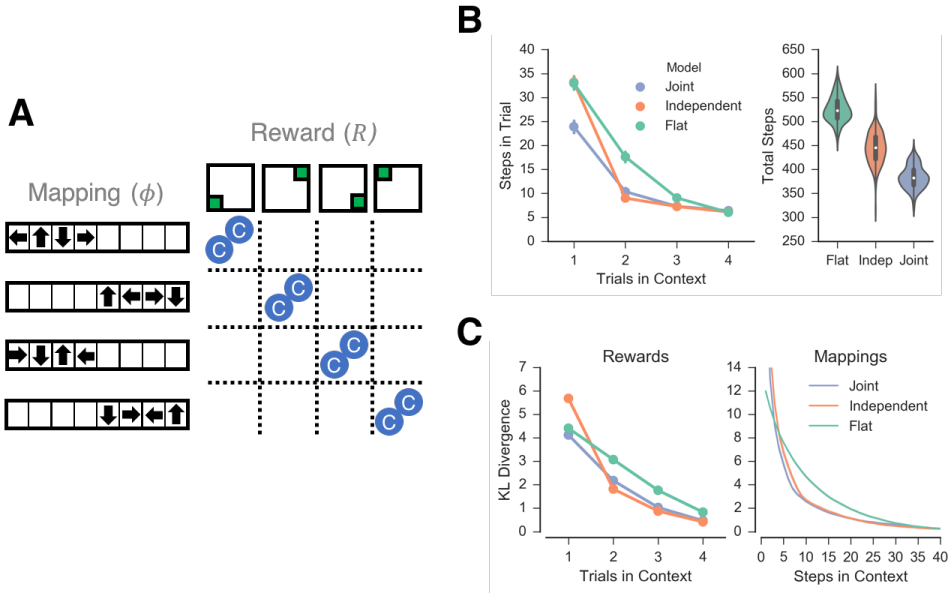


**Fig 3. Simulation 2** A: Schematic representation of the second task domain. Eight contexts (blue circles) were simulated, each paired with a combination of one of four orthogonal reward functions and one of four mappings, such that each pairing was repeated across two contexts, providing a discoverable relationship. B: Number of steps taken by each agent shown across trials within a single context (left) and over all trials (right). C: KL-divergence of the models' estimates of the reward (left) and mapping (right) functions as a function of time.

As is the previous simulations, the differences in performance between the clustering agents was largely driven by learning of the reward function. Both the independent and joint clustering agents had similar estimates of mapping functions across time [Fig 3, C, right] whereas the independent clustering agent uniquely shows an initial deficit in generalization of the reward function, as measured by KL divergence in the first trial in a context [Fig 3, C, left]. The difference in performance between the two clustering models largely occurs for the first trial in a new in a new context, during which time the joint clustering agent had a better estimate of the reward function. As before, this reflects an information asymmetry between mappings and rewards.

## Meta-agent

In the simulations above, independent clustering outperforms joint clustering when there is no relationship between mappings and goals, whereas joint clustering is the better of the strategies when there is a clear relationship. A natural question is whether a single agent could arbitrate its choice of strategy depending on its experience in the environment. In other words, can an agent infer whether the overall statistics are more indicative of a joint or independent structure and capitalize accordingly? Here, we address this question by implementing a meta-agent that dynamically allocates action selection to one of the lower order agents based on reinforcement learning (based on how well each of these agents predict rewards [3, 20, 21]).

The logic of the meta-agent is thus: each agent, joint and independent clustering, generates reward predictions at each time-step that can be used to evaluate the expected value of each strategy. Hence, we implement the meta-agent with a reinforcement learning process where on each trial either a joint or independent clustering agent is chosen to complete the trial [Fig 4, A]. At the beginning of each trial, an actor is chosen to complete the trial by sampling a softmax choice function:

$$p_{\text{actor}} \propto e^{\beta * Q_{\text{actor}}} \tag{12}$$

where $p_{\text{actor}}$ is the probability an actor is selected, $\beta$ is a noise parameter, and $Q_{\text{actor}}$ is the value of the actor. $Q_{\text{actor}}$ is updated on each time step by the equation:

$$Q_{\text{actor}} \leftarrow Q_{\text{actor}} + \eta(r_t - \hat{r}_{<x,c>}) \tag{13}$$

where $\eta$ is a learning rate, $r_t$ is the reward experienced at time $t$ and $\hat{r}_{<x,k>}$ the predicted reward by the actor for the location $x$ and the actor's context-cluster assignment $c_t \in k$. The predicted reward is determined by the MAP estimate of the model and is equal to 0 if the MAP estimate corresponds to a novel location for a novel cluster.

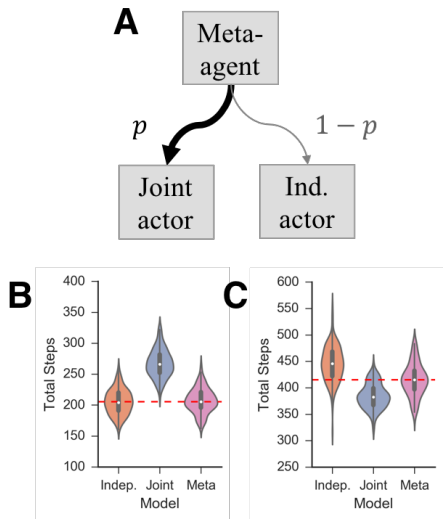


**Fig 4. Meta-Agent** *A*: On each trial, the meta-agent chooses either a joint or independent actor based on its predictive utility, learned via RL. Both agents are updated at each time step *B*: Overall performance of independent, joint and meta agents on simulation 1 *C*: Overall performance of independent, joint and meta agents on simulation 2

We simulated the meta-agent on the task domains used in simulations 1 and 2 with the parameter values $\beta = 1.0$ and $\eta = 0.1$. In the first task domain, where independent clustering results in better performance than joint clustering, the performance of the meta-agent more closely

matched the performance of the independent clustering agent [Fig 4, B]. The meta-agent completed the task in an average of 207.2 (s=19.7) steps compared to 205.2 (s=20.2) and 267.5 (s=22.4) steps for the independent and joint clustering agents. In the second task domain, where joint clustering outperformed independent clustering, the meta-agent completed the task in an average of 425.7 (s=32.1) steps compared to an average of 384.2 (s=21.2) and 443.2 (s=35.8) steps for the joint and independent clustering agents, respectively. Thus, while the meta-agent did not equal the performance of the best agent in either environment, it outperformed the worse of the two agents in both environments. Normatively, this is a useful property if an agent cares about minimizing the worst possible outcome across unknown task domains (as opposed to maximizing their performance within a single domain), similar to a minimax decision rule in decision theory [22]. This can be advantageous if agent has little information about the distribution of task domains and, as as is the case in the next section, if the costs of choosing the wrong strategy are large.

## Exploration costs in the "diabolical rooms" problem

A primary benefit of generalization is a reduction of exploration costs: if an agent generalizes effectively, it can determine a suitable policy without fully exploring a new context. Consequently, a good measure of generalization is the reduction in exploration cost. In the above simulations, exploration costs were uniform across all contexts. Here, we consider a set of task domains in which each context has a different exploration cost and that cost of exploration increases across time. In real world situations, the cost of exploring in a domain with multiple subgoals can compound as a person progresses through a task. For example, if a person is baking a cake and uses vinegar instead of vanilla, they may have to start over from scratch. Thus, the benefits and costs of generalization can compound in task with a sequential structure in ways that are not often apparent in a more restricted task domain.

We define a modified 'rooms' problem as a task domain in which an agent has to navigate a series of rooms (individual grid worlds) to reach a goal location in the last room [Fig 5, A]. In each room, the agent must choose one of three doors, one of which will advance the agent to the next room, whereas the other two doors will return the agent to the starting position of the very first room (hence the 'diabolical' descriptor). Additionally, the mappings that link actions to cardinal movements can vary from room to room, such that the agent has to discover this mapping function separately from the location of the reward (goal door). All of the rooms are visited in order such that if an agent chooses a door that returns it to the starting location, it will need to visit each room before it can explore a new door. Consequently, the cost of exploring a door in a new room increases with each newly encountered room.

Botvinick et al. [18] have previously used the original 'rooms' problem introduced by Sutton et al. [17] to motivate the benefit of the "options" hierarchical RL framework as a method of reducing computational steps. However, in the traditional options framework, there is no method for reusing options across different parts of the state-space (for example, from room to room). Each option needs to be independently defined for the portion of the state-space it covers. In contrast, hierarchical clustering agents that decompose the state space can facilitate generalization in the rooms problem by reusing task structures when appropriate [**?**]. However, because it was a joint clustering agent, this previous work would not allow for separate re-use of mapping and reward functions.

In this new, diabolical, variant of the rooms problem, we have afforded the opportunity for reuse of subgoals across rooms, but have modified the task not only to allow for different mapping functions, but where there is a large cost when the appropriately learned subgoal (choosing the correct door) is not reused, and where this cost is varied parametrically by changing the size or number of the rooms. (The rooms problem here is qualitatively similar to the "RAM combination-lock environments" used by Leffler and colleagues [9] to show that organizing states into classes with reusable properties (analogous to clusters presented in the present work) can drastically reduce exploration costs. In the RAM combination-lock environments, agents navigated through a linear series of states, in which one action would take agents to the next state, another to

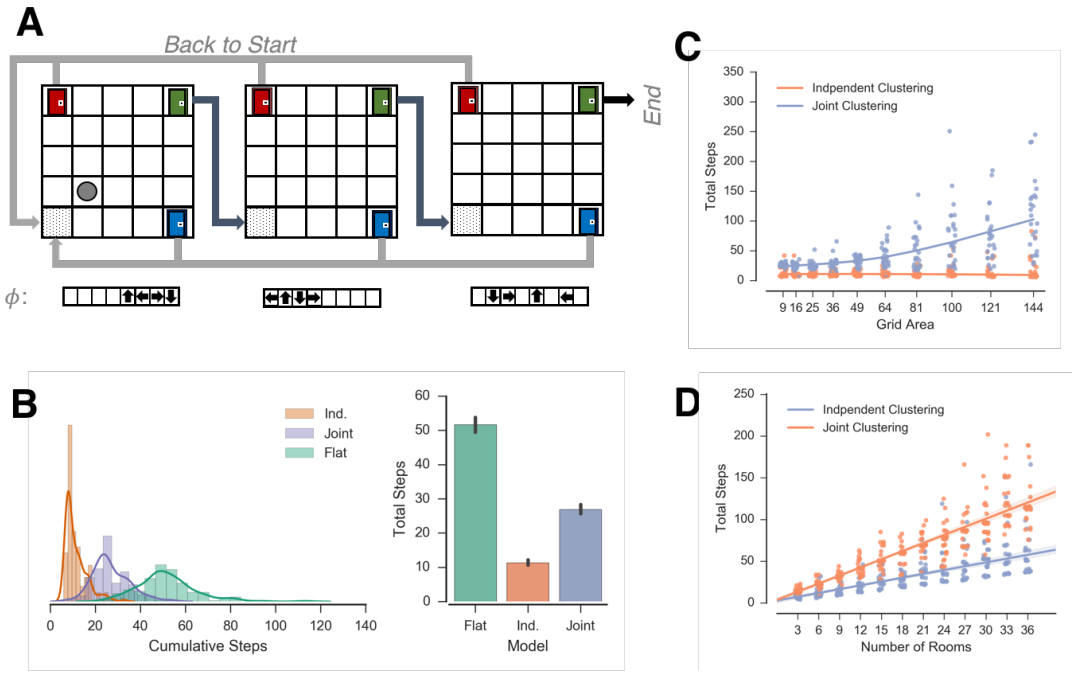


**Fig 5. "Diabolic Rooms Problem"** *A*: Schamatic diagram of rooms problem. Agents enter a room and choose a door to navigate to the next room. Choosing the correct door (green) leads to the next room while choosing the other two doors leads to the start of the task. The agent learns three mappings across rooms *B*: Distribution of steps taken to solve the task by the three agents (left) and median of the distributions (right). *C,D*: Regression of the number of steps to complete the task as a function of grid area (C) and the number of rooms in the task (D) for the joint and independent clustering agents.

a goal state, and all others back to the start. The rooms task environment presented here is highly similar but allows us to vary the cost of exploring each room parametrically by varying its size.)

We simulated an independent clustering agent, a joint clustering agent, and a flat agent on series of rooms problems with the parameters $\alpha = 1.0$, $\gamma = 0.80$ and $\beta = 5.0$. Each room was represented as a new context (for example, to simulate differences in surface features). There were three doors in the corners of the room, and (for simplicity) the same door advanced the agent to the next room for every room. (Without loss of generality, this arrangement still favors independent clustering over joint clustering). Agents received a binary reward for selecting the door that advanced to the next room.

In the first set of simulations, we simulated the agents in 6 rooms, with each room comprising a 6x6 grid world, with three mappings $\phi_c \in \{\phi_1, \phi_2, \phi_3\}$, each repeated once. Because the cost of exploration compounds as an agent progresses through a task, the ordering of the rooms affects the exploration costs. For simplicity, we simulated a fixed order of the mappings encountered by rooms, defined by the sequence $X^\phi = \phi_1\phi_1\phi_2\phi_2\phi_3\phi_3$. In this task domain, both clustering agents show a generalization benefit as compared to the flat agent [Fig 5B], with the flat agent completing the task in approximately 1.9x and 4.5x more total steps than joint and independent clustering, respectively.

We further explored how these exploration costs change parametrically with the geometry of the environment. First, we varied the dimensions of each room from 3x3 to 12x12. While both clustering models show increased exploration costs as the area of the grid world increases, the exploration costs for the joint clustering model grow at a faster exponential rate than the independent clustering model [Fig 5, C]. Similarly, we can increase the exploration costs by

increasing the number of rooms in the task domain. We varied the number of rooms in the rooms in the task domain from 3 to 27 in increments of three. As before, the same door in all rooms advanced the agent and three mappings were repeated across the rooms. The order of the mappings encountered is defined by the sequence $X^{\phi} = \phi_1^{(k)}\phi_2^{(k)}\phi_3^{(k)}$ for $k \in \{1, 2, ..., 9\}$, where $k$ is the number of times a mapping is repeated. Again, both clustering agents experience an increased cost of exploration as the number of rooms increases, but the cost of exploration increases at a faster linear rate for the joint clustering agent than the independent clustering agent [Fig 5, D].

Thus, in environments where the benefits of generalization compound across time, difference between strategies can be dramatic. Here, we have simulated an environment in which independent clustering leads to better generalization than joint clustering but we could equivalently create an example in which joint clustering leads to better performance (for example, joint clustering would do better if each mapping uniquely predicted the correct door). Consequently, a strategy that assumes a fixed relationship between transition dynamics and rewards will have a worse case outcome that can grow exponentially with the complexity of the task domain. This suggest a normative basis for arbitrating between strategies: by learning about the relationship between transition dynamics and goals, an agent may be mitigate the magnitude of this cost.

# Information Theoretic Analysis

Thus far, we have examined the performance of hierarchical clustering variants using specific assumptions about how their choices are generated (e.g., planning and exploration, etc.). However, these agents were not constructed by inverting the generative process of the tasks that they confronted. As such, neither of the clustering agents is strictly optimal for any of the above of the tasks considered in isolation (and certainly not across the set of tasks, given that the ranked model performances reversed across them). Over a larger set of tasks, optimality can be a nebulous construct as it depends on the assumptions embedded in the model about the generative process, and any agent's performance will depend in part on whether those assumptions are congruent with reality. Indeed, we can expect that there exists an algorithm for each task that outperforms the models presented here. However, we are more concerned with the suitability of generalization across ecological environments, rather than the specific task domains we have simulated. To properly assess this question, one requires a definition and characterization of ecological environments, a problem well outside the scope of the current work.

Nevertheless, we can more formally and generally assess when, and under what conditions, each of the clustering models might be more suitable than the other. In the simulations thus far, we have imposed strong assumptions about the task domain, the agents' knowledge of its structure, exploration policies, and planning. While these assumptions are useful for the purpose of demonstration, they are less likely to be valid in real-world applications (such as a robot navigation) or as a model of human or animal learning. To make a more general normative claim, it is desirable to abstract away the implementation and strictly address the normative basis of context-popularity based clustering as a generalization algorithm by itself.

To do so, we can frame generalization as a classification problem and quantify how well an agent correctly identifies the cluster in which a context belongs without regard to learning the associated task statistics. This simplifying assumption allows us to examine the CRP as a mechanism for generalization agent and abstracts away the effect of the likelihood function on generalization. Let $k \in \mathcal{K}$ be a cluster associated with a Markov decision problem and let context $c \in \mathcal{C}$ be a context experienced by the agent. Given a history of experienced contexts and associated clusters $\{c_{1:n}, k_{1:m}\}$, we cast the problem of generalization as learning the classification function $k = f(c)$ that minimizes the risk of misclassification.[1]

[1] Misclassification risk here is loosely defined: some misclassification errors are worse than others and misidentifying a cluster (and consequently, its MDP) may or may not result in a poor policy if the underlying MDPs are sufficiently similar. As such, the loss function in ecological settings can be a highly non-linear, domain specific function (as we demonstrate with the diabolical rooms problem).

More formally, we define risk as the expectation $R(p, f) = \mathbb{E}[L(p, f)]$, where $L(p, f)$ is the loss function for misclassification and $p$ is the generative distribution over new contexts $p = \Pr(k|c_{t+1})$. For our purposes here, we will abstract away domain specificity in the loss function $L(p, f)$. Because the CRP is a probability distribution, a reasonable domain-independent loss function is the information gain between the CRP's estimate of the probability of $k$ and the realized outcome, or

$$L(p, f) = -\log_2 q_K \tag{14}$$

where $q_M$ is the CRP's estimate of the probability of the observed cluster $k$ in context $c$. The misclassification risk is thus the cross entropy between the CRP and the generative distribution:

$$R(p, f) = \mathbb{E}_p[-\log_2 q] = H(p, q) \tag{15}$$

Thus, by casting generalization as classification and assuming information gain as a domain-general loss function, we are in effect evaluating the degree to which the CRP estimates the generative distribution. Risk is minimized when $q = p$, that is, when the CRP perfectly estimates the generative process. There is no upper bound to poor performance, but in many task domains it is possible to make a naïve guess over the space of clusters (for example, a uniform distribution over a known set of clusters). Because any useful generalization model will be better than a naïve guess, we can evaluate whether the CRP will lead to lower information gain than a naïve estimate in different task domains. Thus, we can ask what properties of the task domain affect the quality of the estimate.

## CRP performance as a function of task structure

One question we can ask is how well the CRP prior generalizes as a function of the underlying structure of the task domain. Intuitively, we might expect that a generalization agent should show a generalization benefit in highly structured task domains and show a less of a generalization benefit in unstructured domains. We can thus define a set of restricted task domains that vary in their structure. Specifically, we can vary the predictability of the generative process $p$ and evaluate the agent as a function of this predictability.

Let $\mathcal{K} = \{A, B, C, D\}$ be the set of possible clusters in a task domain with probabilities $\mathcal{P} = \{p_A, p_B, p_C, p_D\}$. Let $X = ABCD$ represent a sequence of contexts and cluster identities experienced by an agent, such that cluster $A$ is experienced in context $c_1$, B is experienced in context $c_2$, etc. We assume that the cluster identity is observable from the statistics of the associated MDP and that the agent knows the members of the set $\mathcal{K}$.

We want to evaluate the ability of the CRP prior to predict each cluster in the sequence, conditional on its own history. We do so by calculating the expected loss experienced by the CRP over the sequence $X$. We define our loss function over sequence $X$ as

$$L(p, f(X)) = -\frac{1}{||X||} \sum_X \log_2 q_k^{(t)} \tag{16}$$

where $q_k^{(t)}$ is the CRP's probability estimate for the value $k = X_t$ given $X_{1:t-1}$. As noted above, this loss function is equivalent to the cross entropy $H(q, p)$ between the CRP and the generative process. That is, $H(q, p)$ is the average degree of unpredictability (in bits of information content) of experiencing each MDP given the estimate $q$. We can assess $H(q, p)$ for the CRP by updating the predictive distribution in each context and probing its estimate for the subsequent context. As the CRP is exchangeable [16], $H(q, p)$ is invariant to the order of the sequence.

We can similarly quantify the degree of predictability of $X$ by evaluating the entropy of the sequence, defined $H(X) = -\sum_{\mathcal{P}} p_x \log_2 p_x$. Here, we define a sequence $X^{(n)}$ to allow us to monotonically decrease $H(X)$ with $n$. Let $X^{(n)}$ be the sequence $A^{(n)}BCD$, where $n$ denotes the number of times $A$ appears in the sequence. For example, $X^{(1)}$ is the sequence $ABCD$ and $X^{(3)}$ is the sequence $AAABCD$. For simplicity, we assume the probability distribution $\mathcal{P}$ over the

ensemble $\mathcal{K}$ is exchangeable and that the probabilities over the members of its ensemble are proportional to their frequency in the sequence such that $p_k = N_k/||X||$ where $N_k$ is the number of times $k$ appears in sequence $X$. Consequently, the entropy of the sequence $X^{(1)}$ is $H(X^{(1)}) = 2$ bits and the entropy of the sequence $X^{(3}$ is $H(X^{(3)}) \approx 1.79$bits. As $n$ approaches infinity, the entropy $H(X^{(n)})$ asymptotically approaches 0 bits. Intuitively, as $A$ is repeated more often in the sequence, the sequence is more predictable (lower entropy). It is important to note that the sequence predictability does not depend on order. Because the CRP is exchangeable, it will have the same predictive error for the sequences ABCDABCD and ADBCDBAC. Order-dependent predictability is beyond of the scope of the current work.

We evaluated the CRP on $X^{(n)}$ for $n = [1, 100]$ and compared it to a naïve guess (uniform distribution over $\mathcal{M}$). Because the CRP is parameterized by its tendency to generate a new cluster, the value of its $\alpha$ parameter alters the predictive distribution. To establish an upper limit on the performance of the CRP, we used numerical optimization to determine $\alpha$ for each value of $n$. In addition, we also evaluated the performance of an agent with a fixed $\alpha = 1$, which we believe is a more accurate reflection of a generalizer in an unknown environment. As expected, as we increase the structure of the sequences (lower entropy), the CRP advantage over a naïve guess increases [Fig 6, left, green line]. Similarly, the optimal value of $\alpha$ declines with the sequence structure, such that it is more advantageous to cluster as the sequence becomes more predictable [Fig 6, right]. However, this benefit is minimal for very unstructured sequences, and for fixed values of $\alpha$, clustering for highly unstructured sequences ($H(X) \lesssim 1.45$bits) yields worse CRP performance compared to a naïve guess.

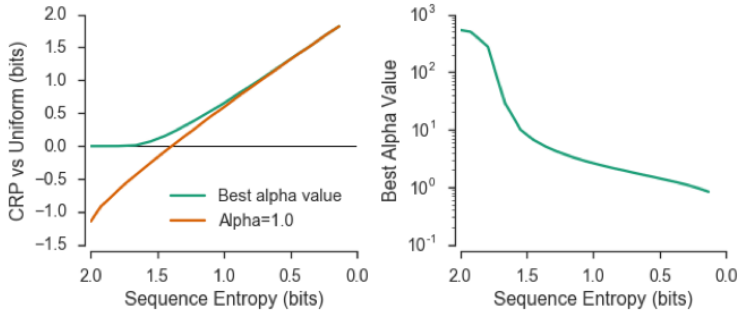


**Fig 6.** Performance of the CRP as a function of task domain structure. *Left*: Relative information gain of a naïve guess over the CRP as function of sequence entropy for a CRP with an optimized alpha parameter (green) or fixed at $\alpha = 1.0$. *Right*: Optimized alpha value (log scale) as a function of sequence entropy.

## Independent vs Joint clustering

Having confirmed these intuitive properties of the CRP, we can now ask under what conditions is it better to cluster aspects of the task structure (such as reward functions and transition or mapping functions) independently or jointly. To do so, we rely on the assumption that a reward function is typically substantially sparser than a transition function: as an agent interacts with a task it will gain more information about the transition function early in the task than it will about the reward function (unless the environment is very rich with rewards at most locations, in which case any random agent would perform well).

Consequently, for an agent that clusters rewards and transitions together, the information gained about transitions will dominate the likelihood function, such that the inference of rewards can be thought of as approximately conditional on knowledge of the transition structure. Conversely, an agent that clusters rewards and transition independently will not consider the mapping information when it predicts the reward function. Thus, we can compare the two agents

by evaluating the consequences of clustering rewards conditional on transitions as compared to clustering rewards independent of transitions. Formally, we can consider the comparison of independent and joint clustering as the comparison between two different classifiers, $R = f(c)$ and $R = f(c, T)$, one of which classifies reward functions solely as a function of contexts and the other as a function of contexts and observed transition statistics.

Interestingly, this approximation leads to the conclusion that independent clustering is a simpler statistical model than joint clustering. Estimating a marginal distribution is a simpler statistical problem than estimating its composing set of conditional distributions. As such, we might expect a trade off where independent clustering provides better generalization with little experience whereas joint clustering provides better generalization asymptotically. We can evaluate the latter claim by noting that given random variables $R$ and $T$, $H(R|T) \leq H(R)$. This statement implies that given a known joint distribution between two random variables, knowledge of one of the random variables cannot increase the uncertainty of the other; an agent can simply learn when there is no relation between the two, in which case the joint distribution doesn't hurt. Intuitively, this claim is based on the notion that more information is always better (or at least, no worse) in the long run.

Note that this relationship is only guaranteed if the true generative process is known and as such, experience in the task domain plays an important role. As we discuss in the following section, there needs to be sufficient experience to determine whether the joint distribution is useful or not (and in fact, assuming conditional dependence when there is no such relationship can slow down learning dramatically). Nonetheless, the CRP prior will converge asymptotically on the conditional and marginal distributions, $\Pr(R|T)$ and $\Pr(R)$, for the joint and independent clustering agents, respectively. If we consider the CRP to be an estimator, we can define its bias as

$$\text{Bias}_p\left[q\right] \equiv \mathbb{E}_p\left[q\right] - p = \frac{-\alpha}{\alpha + N_c} \tag{17}$$

where $N_c$ is the total number of contexts observed and $\alpha$ is the concentration parameter governing new clusters. Asymptotically, the CRP is unbiased as $\lim_{N_c \to \infty} \text{Bias}_p[q] = 0$ and the CRP converges to the generative distribution. As a consequence, joint clustering has lower information gain than independent clustering asymptotically as the CRP for a joint clustering agent will converge to the conditional distribution $\Pr(R|T)$ whereas the CRP for an independent clustering agent will converge to the marginal distribution $\Pr(R)$.

### Mutual Information

As alluded to above, that joint clustering is guaranteed to produce a better estimate is only true as $N_c \to \infty$ whereas here we are concerned with task domains in which an agent has little experience. Intuitively, we might expect independent clustering to be favorable in task domains where there is no relationship between transitions and rewards. Conversely, we might expect joint clustering to be more favorable when there is relationship between transition and rewards. While we considered two extremes of these cases in simulations 1 and 2, we can also vary the relationship parametrically. Formally, we can consider the relationship between transitions and rewards with mutual information. Let each context $c$ be associated with a reward function $R \in \mathcal{R}$ and a transition function $T \in \mathcal{T}$ and let $\Pr(R, T)$ be the joint distribution of $R$ and $T$ in unobserved contexts. The mutual information between $R$ and $T$ is defined

$$I(R;T) = \sum_{R \in \mathcal{R}, T \in \mathcal{T}} \Pr(R,T) \log_2 \frac{\Pr(R,T)}{\Pr(R)\Pr(T)} \tag{18}$$

Mutual information represents the degree to which knowledge of either variable reduces the uncertainty (entropy) of the other, and can be used as a metric of the degree to which a task environment should be considered independent or not. As such, it satisfies $0 \leq I(R;T) \leq \min\left(H(R), H(T)\right)$

To evaluate how mutual information affects the relative performance of independent and joint clustering, we constructed a series of task domains that allow us to monotonically increase $I(R;T)$

by with a single parameter $m$ while holding all else constant. We define $\mathcal{R} = \{A, B\}$ and $\mathcal{T} = \{1, 2\}$ and we define two sequences $X^R$ and $X^T$ such that $X_i^R$ and $X_i^T$ are the reward and transition function for context $c_i$. We define the sequence $X^R = A^{(2n)}B^{(2n)}$, where $A^{(k)}$ refers to a $k$ repeats of $A$, and the sequence $X^T = 1^{(n+m)}2^{(2n)}1^{(n-m)}$, where $1^{(k)}$ refers to $k$ repeats of 1. To provide a concrete example, if $n = 2$ and $m = 0$, the sequence $X^R = AAAABBBB$ and the sequence $X^T = AABBBBAA$. Similarity, if $n = 2$ and $m = 2$, then the sequence $X^R$ is unchanged while $X^T$ is now $AAAABBBB$. Critically, these sequences have the property that $I(R;T) = 0$ if and only if $m = 0$ and that $I(R;T)$ monotonically increases with $m$. In other words, the residual uncertainty of $R$ given $T$, $H(R|T)$, declines as a function of $m$. This allows us to vary $I(R;T)$ independently of all other factors by changing the value of $m$.

We evaluated the relative performance of the independent and joint clustering agents by using the CRP to predict the sequence $X^R$, either independent of $X^T$ (modeling independent clustering), or conditionally dependent on $X^T$ (modeling joint clustering) for values of $n = 5$ and $m = [0, 5]$. For these simulations, we first assume both sequences $X^R$ and $X^T$ are noiseless, but below we show that noise parametrically affects these conclusions. For low values of $m$ ($m \leq 2$), independent clustering provides a larger generalization benefit [Fig 7, left]. This has the intuitive explanation that as the features of the domain becomes more independent, independent clustering provides better generalization. Importantly, there are cases in which independent clustering provides a better evidence even thought there is non-zero mutual information [$0 < I(R;T) \lesssim 0.2$bits].

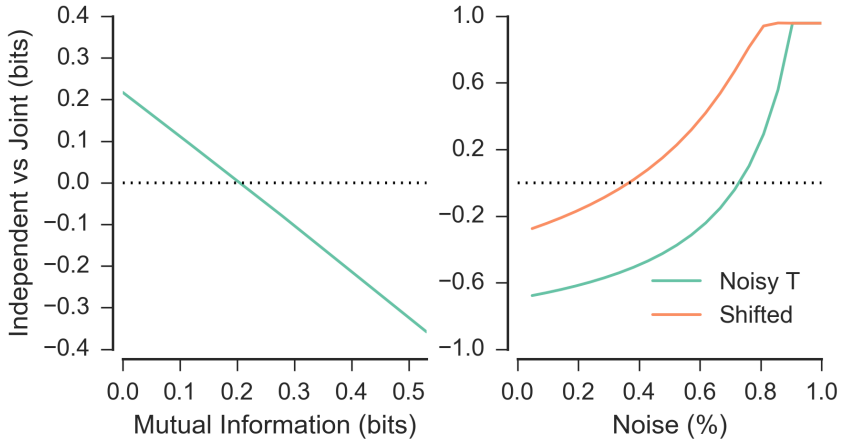


**Fig 7.** Performance of independent vs. joint clustering in predicting a sequence $X^R$, measured in bits of information gained by observation of each item. *Left*: Relative performance of independent clustering over joint clustering as a function of mutual information between the rewards and transitions. *Right*: Noise in observation of $X^T$ sequences parametrically increases advantage for independent clustering. Green line shows relative performance in sequences with no residual uncertainty in $R$ given $T$ (perfect correspondence), orange line shows relative performance for a sequence with residual uncertainty $H(R|T) > 0$bits.

### Noise in observation of transition functions

Above, we assumed that transition functions were fully observable with no uncertainty. But in many real-world scenarios, the relevant state variables are only partially observable and the transition functions may be stochastic. In this section, we therefore relax this noiseless assumption to characterize the effect of noisy observations on inference and generalization. As before, we model generalization as the degree of predictability of $X_{i+1}^R$ given $X_{1:i}^R$ either independent of $X^T$ (independent clustering) or conditionally on $X^T$ (joint clustering).

We first construct reward and transition sequences in which knowledge of the transition function completely reduces the uncertainty about the reward function ($I(R;T) = H(R)$, and hence $H(R|T) = 0$). Consider $X^R = A^{(20)}BCD$ and $X^T = 1^{(20)}234$, where $A^{(20)}$ and $1^{(20)}$ refer to 20 repeats of $A$ and 1, respectively. As such, we would expect joint clustering to produce better generalization than independent clustering if observations are noiseless. To simulate noise / partial observability, we assume each observation of $X_i^T$ is mis-identified as some new function $T^* \notin \{1, 2, 3, 4\}$ with some probability. Importantly, this simulation has the desideratum that noise does not affect $I(R;T)$ or $H(R)$ themselves (the true generative functions). We compare the inference of $X^R$ independent of $X^T$ (modeling independent clustering) or conditionally dependent on $X^T$ (modeling joint clustering) for various noise levels $\sigma = [0,\ 1.0]$. When noise is sufficiently high ($\sigma > 0.71$), independent clustering produces a better estimate of $X^R$ than joint clustering [Fig 7, right, green line] even for this extreme case where the two functions have perfect mutual information.

Next, we assessed how mutual information and noise interact by decreasing the correspondence of the sequences. As noted above, in the sequences used above ($X^R = A^{(20)}BCD$ and $X^T = 1^{(20)}234$), there is no residual uncertainty of $R$ given $T$. We can decrease the correspondence between the two sequences by shifting $X^T$ by one to $X^T_{\text{new}} = 1^{(19)}2341$, and thus decrease the mutual information [$I(X^R; X^T) \approx 0.77$; $I(X^R; X^T_{\text{new}}) \approx 0.52$]. As we have previously noted, this decline in mutual information reduces the benefit of joint clustering, *ceteris paribus*. Nonetheless, there is still a strong relationship between rewards and transition that can be leveraged by joint clustering.

Simulating independent and joint clustering on these new sequences as a function of $\sigma = [0,\ 1.0]$ reveals a lower level of noise needed to see a benefit of independent clustering [Fig 7, right]. As expected, joint clustering provides a better estimate in the no-noise case, as well as for low noise levels ($\sigma < 0.33$), as it can take advantage of the shared structure, while independent clustering results in a better estimate for larger noise levels. Importantly, these effect are cumulative; in ecological settings where observations are noisy and there is only weak mutual information, independent clustering will likely provide a better estimate of the prior over rewards.

# Discussion

In this paper, we provide two alternative models of context-based clustering for the purpose of generalization, a *joint clustering* agent generalizes reward and transition functions together and an *independent clustering* agent that separately generalizes reward and transition functions. These models are motivated by human learning and performance, which is thought to be structured and compositional [23–25], to a degree that previous models of generalization are unable to capture.

Generalization can be seen as a solution to a dimensionality problem. In real-world problems, perceptual space is typically high-dimensional. In order to learn a policy, agents need to learn a mapping between the high-dimensional perceptual space and the effector space. Learning this mapping can require a large set of training data, perhaps much larger than a human would have access to [12, 18]. Clustering can reduce the dimensionality by projecting the perceptual space onto a lower dimensional latent space in which multiple percepts share the same latent representation. Thus, an agent does not need to learn a policy over the full state space but over the lower dimensional latent space. This is an explicit assumption of the models presented here as well as in other clustering models of human generalization, allowing agents to collapse across irrelevant features and preventing interference between stimulus-response mappings across latent distinct rules [3, 20]. Related principles have been explored in lifelong learning [6–8], object-based, and symbol-based approaches [11, 28–31].

Incorporating compositionality takes this argument further, as multiple policies often share component features. For example, playing the saxophone involves the use of the same movements to produce the same notes for different effect in different songs. Learning a policy as a direct mapping from the low level effector space to reward values fails to take advantage of the structure, even if

that policy can be reused as a whole with another instrument. Thus, learning at the component level as opposed to the policy level reduces a high-dimensional problem into multiple lower-dimensional problems. While this adds the additional complexity of the choice of a good set of component features, here we argue the Markov decision problem provides a natural decomposition into rewards and transition functions. This does not preclude further decomposition. For example, we might expect the transition function to be further structured into options or skills but we would expect the choice of these components to depend on the generative environment [26, 27].

Regardless of the choice of component features, a compositional generalization model needs to make assumptions about the relationship between components. We argue here that the proper choice depends on the generative structure, which as an empirical matter, is largely unknown for the ecological environments faced by humans and artificial agents. As we demonstrated in the grid-world simulations above, when there is a strong relationship between components, an agent that assumes as much outperforms an agent that assumes no relationship, and vice-versa [Figs. 2 & 3]. With sufficient (and stationary) experience, we might expect a model that assumes a learnable relationship between components (joint clustering) to perform better in new contexts, since assuming a potential relationship between goals and mappings can be no worse asymptotically than assuming independence (i.e., the agent can simply learn that the correlation is zero). Nonetheless, how much experience is sufficient for joint clustering to provide a better model is difficult to define in general, and will depend on the statistics of the relationships and the combinatorial explosion of the state space that arises. Furthermore, noise or partial observability further complicates the picture: even when there is exploitable mutual information, independent clustering can yield a better estimate when experience is limited [Fig 7]. While beyond the scope of the current work, we would expect a similar pattern if skills or options were considered as a further factorization. If two options lead to stochastic outcomes, generalizing them separately may result in better performance even if their applicability is related.

Why is this the case? It may appear puzzling given the asymptotic assurances that joint clustering will be no worse in stationary environments. Here, the comparison between classification and generalization is instructive. We can think of joint clustering in terms of estimating a joint distribution of the generative process and independent clustering in terms of estimating the marginal distribution for each component independently (similar to a naïve Bayes classifier). In this interpretation, independent clustering trades off an asymptotically worse estimate of the generative process for lower variance, with a bias equal to the mutual information between mappings and goals. In problems with limited experience, such as the type presented here, a biased classifier will often perform better than an asymptotically more accurate estimator because misclassification risk is more sensitive to variance than bias [32]. Thus, by ignoring the correlation structure and increasing the bias to generalize goals that are most popular overall, independent clustering may minimize its overall loss.

We can think of joint clustering as being potentially overly sensitive to noise in the correlation structure and it may be better to ignore real correlations in the environment in order to be robust to this source of noise. To provide a concrete example, we can imagine that if a person is getting food from a buffet, how hungry they are may inform the actions needed to carry the food back to the table. If they are hungry, they may carry more food on a plate and need to execute a subtly different set actions. Nonetheless, the generalization benefit afforded by assuming this relationship may be trivial in comparison to the cost needed to learn the association, especially in other environments where the risk of poor generalization is severe.

Moreover, a person may want to carry heavy plates in a completely different domain (e.g., to bring food to a friend), and if they learned that carrying heavy plates is strongly associated with being hungry, they would need to relearn these actions. The flexibility to compose a novel policy in a new domain by recombining previously learned representations, also known as *systematicity* or *generativity*, has long been thought to be a key feature of cognition [25, 34]. While beyond the scope of the current work, a strong relationship between component properties may impede the potential to generalize task components in domains where that correlation isn't relevant, preventing

systematicity. Consequently, independent clustering may provide a better framework for systematic reinforcement learning.

An altogether different possibility is that a mix of strategies is appropriate. It is not known what a human learner would typically consider to constitute a higher order context variable separate from lower order state variables [3, 33]. Ecologically, the number of contexts a human could potentially encounter is quite high, in which case they would be able to form a more accurate estimate of the correlation structure between components over time. If this speculation is true, then one potential adaptive strategy would be to assume a weak relationship between components early in learning and increasingly relying on the correlation structure as the evidence supports it. Furthermore, high dimensional representations can be useful for decision making [35] (at the expense of generalization) as well as binding operations [36]. This suggests a tension between low-dimensional representations, which are easier to learn and generalize, and high-dimensional representations, which may be more flexible for complex thoughts and behavior.

Finally, while we have presented independent clustering as motivated by human capabilities for generalization, the question of whether human learning is better accounted for by independent or joint clustering remains an open question. Both models are a generalization of previous models used to account for human behavior [2–4]. Future studies will be required to disambiguiate the predictions of these two models.

# References


1. Marcus G, Marblestone A, Dean T. The atoms of neural computation. Science. 2014 Oct 31;346(6209):551-2.

2. Gershman SJ, Blei DM, Niv Y. Context, learning, and extinction. Psychological review. 2010 Jan;117(1):197.

3. Collins AG, Frank MJ. Cognitive control over learning: creating, clustering, and generalizing task-set structure. Psychological review. 2013 Jan;120(1):190.

4. Collins AG, Frank MJ. Neural signature of hierarchically structured expectations predicts clustering and transfer of rule sets in reinforcement learning. Cognition. 2016 Jul 31;152:160-9.

5. Collins AG, Cavanagh JF, Frank MJ. Human EEG uncovers latent generalizable rule structure during learning. Journal of Neuroscience. 2014 Mar 26;34(13):4677-85.

6. Rosman B, Hawasly M, Ramamoorthy S. Bayesian policy reuse. Machine Learning. 2016 Jul 1;104(1):99-127.

7. Mahmud MM, Hawasly M, Rosman B, Ramamoorthy S. Clustering markov decision processes for continual transfer. arXiv preprint arXiv:1311.3959. 2013 Nov 15.

8. Wilson A, Fern A, Tadepalli P. Transfer learning in sequential decision problems: A hierarchical bayesian approach. In Proceedings of ICML Workshop on Unsupervised and Transfer Learning 2012 Jun 27 (pp. 217-227).

9. Leffler BR, Littman ML, Edmunds T. Efficient reinforcement learning with relocatable action models. In AAAI 2007 Jul 22 (Vol. 7, pp. 572-577).

10. James W. The principles of psychology (Vol. 1). New York: Holt. 1890;474.

11. Kansky K, Silver T, Mély DA, Eldawy M, Lázaro-Gredilla M, Lou X, Dorfman N, Sidor S, Phoenix S, George D. Schema Networks: Zero-shot Transfer with a Generative Causal Model of Intuitive Physics. arXiv preprint arXiv:1706.04317. 2017 Jun 14.

12. Lake BM, Ullman TD, Tenenbaum JB, Gershman SJ. Building machines that learn and think like people. Behavioral and Brain Sciences. 2016 Nov:1-01.

13. Kaelbling LP, Littman ML, Moore AW. Reinforcement learning: A survey. Journal of artificial intelligence research. 1996;4:237-85.

14. Sanborn AN, Griffiths TL, Navarro DJ. Rational approximations to rational models: alternative algorithms for category learning. Psychological review. 2010 Oct;117(4):1144.

15. Shafto P, Kemp C, Mansinghka V, Tenenbaum JB. A probabilistic model of cross-categorization. Cognition. 2011 Jul 31;120(1):1-25.

16. Aldous DJ. Exchangeability and related topics. In École d'Été de Probabilités de Saint-Flour XIII—1983 1985 (pp. 1-198). Springer, Berlin, Heidelberg.

17. Sutton RS, Precup D, Singh S. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. Artificial intelligence. 1999 Aug 1;112(1-2):181-211.

18. Botvinick MM, Niv Y, Barto AC. Hierarchically organized behavior and its neural foundations: a reinforcement learning perspective. Cognition. 2009 Dec 31;113(3):262-80.

19. Sanborn AN, Griffiths TL, Navarro DJ. A More Rational Model of Categorization. Proc 28th Annu Conf Cogn Sci Soc. 2006;1–6.

20. Frank MJ, Badre D. Mechanisms of hierarchical reinforcement learning in corticostriatal circuits 1: computational analysis. Cereb Cortex 2012 Mar

21. Doya K, Samejima K, Katagiri K, Kawato M. Multiple model-based reinforcement learning. Neural Comput. 2002 Jun;14(6):1347–69.

22. Berger JO. Statistical decision theory and Bayesian analysis. 2nd ed. New York, NY: Springer-Verlag; 1985.

23. Griffiths TL, Chater N, Kemp C, Perfors A, Tenenbaum JB. Probabilistic models of cognition: exploring representations and inductive biases. Trends Cogn Sci. 2010;14(8):357–64.

24. Reverberi C, Görgen K, Haynes J-D. Compositionality of Rule Representations in Human Prefrontal Cortex. Cereb Cortex. 2012;22(6):1237–46.

25. Kriete T, Noelle DC, Cohen JD, O'Reilly RC. Indirection and symbol-like processing in the prefrontal cortex and basal ganglia. Proc Natl Acad Sci. 2013;110(41):16390–5.

26. Machado MC, Bellemare MG, Bowling M. A Laplacian Framework for Option Discovery in Reinforcement Learning. arXiv preprint arXiv:1703.00956. 2017 Mar 2.

27. Stachenfeld KL, Botvinick MM, Gershman SJ. The hippocampus as a predictive map. Nat Neurosci. forthcoming

28. Diuk C, Cohen A, Littman ML. An object-oriented representation for efficient reinforcement learning. InProceedings of the 25th international conference on Machine learning 2008 Jul 5 (pp. 240-247). ACM.

29. Konidaris G, Barto A. Building portable options: Skill transfer in reinforcement learning. IJCAI Int Jt Conf Artif Intell. 2007;895–900.

30. Konidaris G. Constructing Abstraction Hierarchies Using a Skill-Symbol Loop. In: IJCAI International Joint Conference on Artificial Intelligence. 2016. p. 1648–1654.

31. Solway A, Diuk C, Córdova N, Yee D, Barto AG, Niv Y, et al. Optimal Behavioral Hierarchy. PLoS Comput Biol [Internet]. 2014;10(8):e1003779.

32. Friedman JH. On Bias , Variance , 0 / 1 — Loss , and the Curse-of-Dimensionality. Data Min Knowl Discov. 1997;77:55–6.

33. Collins AGE, Cavanagh JF, Frank MJ. Human EEG Uncovers Latent Generalizable Rule Structure during Learning. J Neurosci. 2014;34(13):4677–85.

34. Fodor J, Pylyshyn Z. Connectionism and cognitive architecture: A critical analysis. Cognition. 1988

35. Fusi S, Miller EK, Rigotti M. Why neurons mix: High dimensionality for higher cognition. Current Opinion in Neurobiology. 2016.

36. Smolensky P. Tensor product variable binding and the representation of symbolic structures in connectionist systems. Artif Intell. 1990 Nov; 46(1–2):159–216.