Affect Control Processes: Probabilistic and Decision Theoretic Affective Control in Human-Computer Interaction.

Jesse Hoey
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, N2L3G1
jhoey@cs.uwaterloo.ca

Tobias Schröder
Centre for Theoretical Neuroscience
University of Waterloo
Waterloo, Ontario, N2L3G1
post@tobiasschroeder.de

Areej Alhothali
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, N2L3G1
aalhotha@cs.uwaterloo.ca

University of Waterloo School of Computer Science Technical Report CS-2013-03.

June 21, 2013

Abstract

Affect Control Theory is a mathematical representation of the interactions between two persons, in which it is posited that people behave in a way so as to minimize the amount of deflection between their cultural emotional sentiments and the transient emotional sentiments that are created by each situation. Affect control theory presents a maximum likelihood solution in which optimal behaviours or identities can be predicted based on past interactions. Here, we formulate a probabilistic and decision theoretic model of the same underlying principles, and show this to be a generalization of the basic theory. The new model, called *BayesAct*, is more expressive than the original theory, as it can maintain multiple hypotheses about behaviours and identities simultaneously as a probability distribution, and can make value-directed action choices. This allows the model to generate affectively believable interactions with people by learning about their identity, predicting their behaviours, and taking actions that are simultaneously goal-directed and affect-sensitive. We demonstrate this generalisation with a set of simulations. We then show how our model can be used as an emotional "plug-in" for systems that interact with humans. We demonstrate human-interactive capability by eliciting knowledge from 37 participants with a survey, and then using this knowledge to build a simple demonstrative intelligent tutoring application. We present results from a pilot study with 20 participants using the application.

1 Introduction

Designers of intelligent systems have increasingly attended to theories of human emotion, in order to build software interfaces that allow its users to experience naturalistic flows of communication with the computer. This endeavour requires a comprehensive mathematical representation of the relations between affective states and actions that captures, ideally, the subtle cultural rules underlying human communication and emotional experience. In this paper, we show that Affect Control Theory (ACT), a mathematically formalized theory of the interplays between cultural representations, interactants' identities, and affective experience [1], is a suitable framework for developing emotionally intelligent agents. To accomplish this, we propose a probabilistic and decision theoretic generalisation of ACT, called *BayesAct*, which we argue is more flexible than the original statement of the theory for the purpose of modelling human-computer interaction. *BayesAct* is formulated as a partially observable Markov decision process or POMDP. The key contributions of this theory are: (1) to represent emotions as probability distributions over a continuous affective space; (2) to allow affective identities to be dynamic and uncertain; (3) to endow agents with the ability to learn affective identities of their interactants; and (4) to introduce explicit utility functions that parsimoniously trade-off emotional and propositional goals for a human-interactive agent. This final contribution is what allows *BayesAct* to be used in human-computer interaction: it provides the computerised *agent* with a mechanism for predicting how the affective state of an interaction will progress (based on Affect Control Theory) and how this will affect the object of the interaction (e.g. the software application being used). The *agent* can then select its strategy of action in order to maximize the expected values of the outcomes based both on the application state and on its emotional alignment with the human. The main contribution of this paper is therefore of a theoretical nature, which we demonstrate in simulation. We have also implemented the theory in a simple tutoring system, and we report the results of an empirical survey and pilot study with human participants.

The remainder of this section gives further detail and intuitions about the model and our contributions. Section 2 explains Affect Control Theory and POMDPs. Section 3 then gives full details of the model. Section 4 discusses simulation and human experiments, and Section 5 concludes. Appendices A, B and C give some additional results and mathematical details that complement the main development. Parts of this paper appeared in a shortened form in [2].

1.1 ACT for HCI

Emotions play a significant role in humans' everyday activities including decision-making, behaviours, attention, and perception [3–6]. This important role is fueling the interest in Affective Computing since its first proposal by Rosalind Picard [7]. Affective Computing research is generally concerned with four main problems: affect recognition (vision-based, acoustic-based, etc.) [8,9], generation of affectively modulated signals such as speech and facial expressions [10,11], the study of human emotions including affective interactions and adaptation [12], and modeling affective human-computer interaction, including embodied conversational agents [13–15]. In this paper, we are proposing to leverage the large body of research in psychology and sociology on Affect Control Theory (ACT) [1] to propose a general-purpose model for the fourth problem: how to integrate affect into computer systems that interact with humans. Social psychological research under the ACT paradigm has generated precise mathematical models of human-human affective interactions based on a single basic principle: minimising the deflection between culturally shared fundamental sentiments about social concepts and transient impressions resulting from the conceptual interpretation of specific situations. Fundamental sentiments and transient impressions are treated mathematically as vectors in a three-dimensional affective space, constituted by the basic dimensions of emotional experience, evaluation, potency, and activity (EPA) [25]. ACT uses the simple deflection minimisation principle, along with predictive equations derived from large-scale experiments across many people, cultures, socio-economic classes, races and genders, to make verifiably accurate predictions about human behaviour. In this paper, we propose a novel and more general formulation that marries Affect Control Theory with a general-purpose probabilistic and decision-theoretic model of propositional (non-affective) interactions between a human and a computer. This marriage combines the empirical predictive power of ACT for emotional states and resulting social actions with the robust and generative power of probabilistic decision-theoretic reasoning. The resulting model is both a generalisation of the original ACT model, and a principled method for integrating affective reasoning into human-interactive computer applications.

This paper does not attempt to address the first two questions posed in the last paragraph addressing the recognition and generation of affective signals. We assume that we can detect and generate emotional signals in affective space, but we allow for the signals to be communicated across a noisy channel. There has been a large body of work in this area and many of the proposed methods can be integrated as input/output devices with our model. Our model gives the mechanism for mapping inputs to outputs based on predictions from ACT combined with probabilistic and decision theoretic reasoning. The probabilistic nature of our model makes it ideally suited to the integration of noisy sensor signals of affect, as it has been used for many other domains with equally noisy signals (e.g. [16]).

We also do not directly address the issue of whether the results of ACT are applicable to the interactions between humans and computers, as the theory is originally developed and measured based on human-human interactions. Shank [17] and Troyer [18] describe experiments to measure EPA fundamental sentiments related to technology and computer terms. Shank gives positive answers to three questions about the use of affect control theory in the modelling of human interactions with technology. He shows that people have shared cultural identity labels for technological actors, and that they share affective sentiments about these labels. He also showed that people view these technological actors as behaving socially, as was previously explored in [19]. The answers to these questions open the doors for the usage of ACT in technology, as we do here.

We do not address the question of how to mathematically model the effects of emotional deflection and a human's identity on a particular application's state. These effects will be application dependent, and require more detailed application-specific study. Our model proposes a general formulation of the affective effects as a generative probabilistic function that can serve as a substrate for applications. To demonstrate, we experiment with some simple ideas about how deflection would have an effect on a student's learning process in an intelligent tutoring application. These ideas are currently being explored in different contexts, including help for persons with ASD [7] and intelligent tutoring systems [20, 21].

Finally, although our model is designed for general purpose interactions, in this paper we only study collaborative and assistive agents. Our model can be used equally well to model competitive and manipulative agents, but this requires a more comprehensive study of the policy generation problem, which is part of our future work.

1.2 Related Work

Research in affective computing has largely focused on using one of two main trends of emotion theories: appraisal theories and dimensional theories. Advocates of appraisal theories define emotions as entities that describe people's personal evaluation for a given stimulus in terms of goals and desires [22, 23]. On the other hand, dimensional theories define emotion as points in a continuous dimensional space where these dimensions typically correspond to evaluation or valence, arousal or activity, and sometimes dominance or potency [24–26]. While the connection of Affect Control Theory with dimensional theories is obvious (since emotions are represented as vectors in a continuous dimensional space), here we examine the connections with appraisal theories in more detail (see also [27, 28]).

Affect Control Theory [1] has important conceptual similarities with appraisal theories. Appraisal theories come in different variants, but generally posit that emotional states are generated from cognitive appraisals of events in the environment in relation to the goals of the agent [29, 30]. As a result, an *agent*'s emotions depend more on its subjective interpretations than on the physical features of an event. Appraisal theorists describe a set of fixed rules or criteria for mapping specific patterns of cognitive evaluations onto specific emotional states. The logic of ACT is quite similar: Emotional states result from (linguistic) interpretations of observed events. The EPA dimensions of affective space can be understood as very basic appraisal rules related to the goal congruence of an event (E), the agent's coping potential (P), and the urgency implied by the situation (A). For discussion of ACT vs appraisal theories see [27]. However, ACT is also more general and more parsimonious than many appraisal theories, since it works without explicitly defining complex sets of rules relating specific goals and states of the environment to specific emotions. Instead, ACT treats the dynamics of emotional states and behaviours as continuous trajectories in affective space. Deflection minimisation is the only prescribed mechanism, while the more specific goals tied to types of agents and situations are assumed to emerge from the semantic knowledge base of the model, as explained in more detail in Section 2.1 below. *BayesAct* further allows *client* goals to be explicitly encoded and optimised.

POMDPs have been used as models for intelligent tutoring systems [31], including with emotional states [21]. Bayesian networks and probabilistic models have also seen recent developments [32] based on appraisal theory [29]. Our work has demonstrated that, by operating completely in a dimensional space, we can surmount computational issues, assure scalability (the state space size only grows with the amount of state necessary to represent the application, not with the number of emotion labels), and we can explicitly encode prior knowledge obtained empirically through a well-defined methodology [1]. Tractable representations of intelligent tutoring systems as POMDPs have recently been explored [31], and allow the modelling of up to 100 features of the student and learning process. Our emotional "plug-in" would seamlessly integrate into such POMDP models, as they also use Monte-Carlo based solution methods [33].

Other researchers in affective computing have looked at using POMDPs and related models, and we now examine in more detail the relationship with our work [21, 32]. Dynamic Bayesian networks (DBNs) and the OCC model of emotions [29] are used by Sabourin *et al.* [32] to model a specific application of a tutoring system. *BayesAct* is similar, but does not commit to any particular emotional labels. Instead, *BayesAct* operates completely in the continuous 81-dimensional space of sentiments and 3-dimensional space of actions of the *agent* as defined by ACT. *BayesAct* also adds decision theoretic constructs to the DBN that allow an intelligent agent to compute theoretically well-grounded policies of action. Conati and Maclaren [21] also use a decision theoretic model, but again rely on sets of labelled emotions and rules from appraisal theories. This is typically done in order to ease interpretability and computability, and to allow for the encoding of detailed prior knowledge into an affective computing application. Finally, *BayesAct* does not require a statically defined *client* (e.g. student) or *agent* (e.g. tutor) identity, but allows the student and tutor to dynamically change their perceived identities during the interaction. This allows for much greater flexibility on the part of the agent to adapt to specific user types "on the fly" by dynamically learning their identity, and adapting strategies based on the decision theoretic and probabilistic model.

2 Basic Models

2.1 Affect Control Theory

Affect Control Theory (ACT) is a comprehensive social psychological theory of human social interaction [1], proposing that peoples' social perceptions, actions, and emotional experiences are governed by a psychological need to minimize deflections between culturally shared fundamental sentiments about social situations and transient impressions resulting from the dynamic behaviours of interactants in those situations.

Fundamental sentiments $\mathbf{f}$ are representations of social objects, such as interactants' identities and behaviours or environmental settings, as vectors in a three-dimensional affective space, hypothesised to be a universal organising principle of human socio-emotional experience [25]. The basis vectors of affective space are called Evaluation/valence, Potency/control, and Activity/arousal (EPA). EPA profiles of concepts can be measured with the semantic differential, a survey technique where respondents rate affective meanings of concepts on numerical scales with opposing adjectives at each end (e.g., {good, nice}↔{bad, awful} for E; {weak, little}↔{strong, big} for P; {calm, passive}↔{exciting, active} for A). Affect control theorists have compiled databases of a few thousand words along with average EPA ratings obtained from survey participants who are knowledgeable about their culture [34]. For example, most English speakers agree that professors are about as nice as students (E), however more powerful (P) and less active (A). The corresponding EPA profiles are $[1.7, 1.8, 0.5]$ for professor and $[1.8, 0.7, 1.2]$ for student (values range by convention from -4.3 to $+4.3$ [34]). In general, within-cultural agreement about EPA meanings of social concepts is high even across subgroups of society, and cultural-average EPA ratings from as little as a few dozen survey participants have been shown to be extremely stable over extended periods of time [34].

Social events can cause transient impressions $\boldsymbol{\tau}$ of identities and behaviours that deviate from their corresponding fundamental sentiments $\mathbf{f}$. ACT models this formation of impressions from events with a minimalist grammar of the form actor-behaviour-object. Extended versions of ACT's mathematical models also allow for representing environmental settings (such as a university or a funeral) and identity modifiers (such as a boring professor or a lazy student) [35–37], but in the interest of parsimony, we will limit our present discussion to the basic actor-behaviour-object scheme. Consider for example a professor (actor) who yells (behaviour) at a student (object). Most observers would agree that this professor appears considerably less nice (E), a bit less potent (P), and certainly more aroused (A) than the cultural average of a professor. Such transient shifts in affective meaning caused by specific events can be described with models of the form $\boldsymbol{\tau} = \boldsymbol{M}\mathbf{f}$, where $\boldsymbol{M}$ is a matrix of statistically estimated prediction coefficients from empirical impression-formation studies where survey respondents rated EPA affective meanings of concepts embedded in a few hundred sample event descriptions such as the example above [1]. Linguistic impression-formation equations exist for English, Japanese, and German [34]. In ACT, the sum of squared Euclidean distances between fundamental sentiments and transient impressions is called deflection:

$$D = \sum_i w_i(\mathbf{f}_i - \boldsymbol{\tau}_i)^2, \tag{1}$$

where w_i are weights (usually set to 1.0).

Affective Deflection is hypothesised to correspond to an aversive state of mind that humans seek to avoid, leading to the *affect control principle* [38]:

Definition 1. *(Affect Control Principle) actors work to experience transient impressions that are consistent with their fundamental sentiments*

ACT is thus a variant of psychological consistency theories, which posit in general that humans strive for balanced mental representations whose elements form a coherent Gestalt [39, 40]. In cybernetic terminology, deflection is a control signal used for aligning everyday social interactions with implicit cultural rules and expectations [1]. For example, advising a student corresponds much better to the cultural expectation of a professor's behaviour than yelling at a student. Correspondingly, the deflection for the former event as computed with the ACT equations is much lower than the deflection for the latter event. Many experimental and observational studies have shown that deflection is indeed inversely related to the likelihood of humans to engage in the corresponding social actions. For example, the deflection-minimization mechanism explains verbal behaviours of mock leaders in a computer-simulated business game [41], non-verbal displays in dyadic interactions [42], and conversational turn-taking in small-group interactions [43].

2.2 Partially Observable Markov Decision Processes

A partially observable Markov decision process (POMDP) [44] is a general purpose model of stochastic control that has been extensively studied in operations research [45, 46], and in artificial intelligence [47, 48]. A POMDP consists of a finite set $\mathcal{X}$ of states; a finite set $\mathcal{A}$ of actions; a stochastic transition model $\Pr : X \times A \rightarrow \Delta(X)$, with $\Pr(x'|x, a)$ denoting the probability of moving from state x to x' when action a is taken, and $\Delta(X)$ is a distribution over $\mathcal{X}$; a finite observation set Ω_x; a stochastic observation model with $\Pr(\omega_x|x)$ denoting the probability of making observation ω_x while the system is in state x; and a reward assigning $R(x, a, x')$ to state transition x to x' induced by action a. A generic POMDP is shown as a Bayesian decision network in Figure 1(a) (solid lines only).

The POMDP can be used to monitor beliefs about the state using standard Bayesian filtering [49]. A *policy* can be computed that maps *belief states* (i.e., distributions over $\mathcal{X}$) into choices of actions, such that the expected discounted sum of rewards is (approximately) maximised. Recent work on so-called "point-based" methods had led to the development of solvers that can handle POMDPs with large state and observation spaces [33, 50–52].

In this paper, we will be dealing with *factored* POMDPs in which the state is represented by the cross-product of a set of variables or features. Assignment of a value to each variable thus constitutes a state.

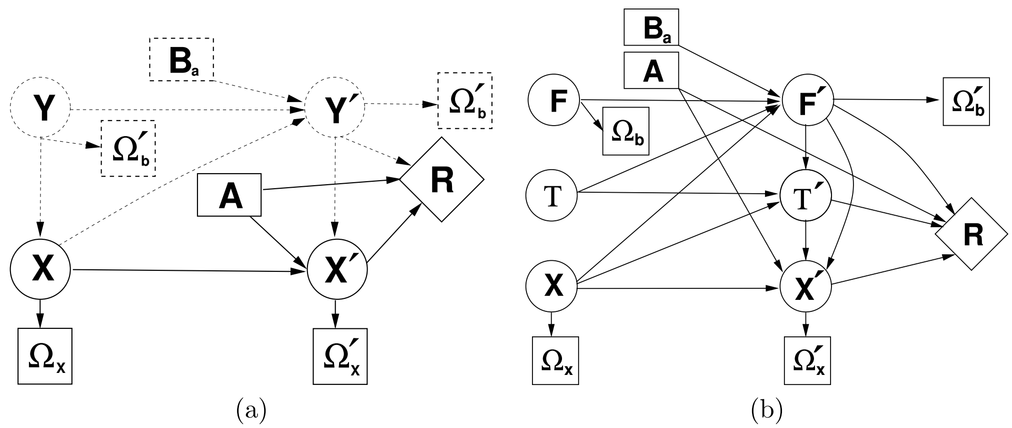


Figure 1: Two time slices of (a) a general POMDP (solid lines) and a POMDP augmented with affective states (dotted lines); (b) a factored POMDP for Bayesian Affect Control Theory;

Factored models allow for conditional independence to be explicitly stated in the model. POMDPs have been used as models for many human-interactive domains, including for intelligent tutoring systems [21, 31], and for human assistance systems [53]. A good introduction to POMDPs and solution methods can be found in [54].

3 Bayesian Formulation of ACT

We are modelling an interaction between an *agent* (the computer system) and a *client* (the human), and will be formulating the model from the perspective of the *agent* (although this is symmetric). We will use notational conventions where capital symbols (F, T) denote variables or features, small symbols (f, τ) denote values of these variables, and boldface symbols $(\mathbf{F}, \mathbf{T}, \mathbf{f}, \boldsymbol{\tau})$ denote sets of variables or values. We use primes to denote post-action variables, so x' means the value of the variable X after a single time step.

In the following we will develop a POMDP model for affect control theory (ACT), which is an extension of the basic POMDP model, as shown with dashed lines in Figure 1(a). It adds a set of variables $\mathbf{Y}$ that describe the emotional state (the sentiments) according to ACT, and a set of emotional actions for the agent, $\mathbf{B_a}$ that affect the emotional state. A human-interactive system can be represented at a very abstract level using a POMDP as shown in Figure 1(a, solid lines). In this case, $\mathbf{X}$ represents everything the system needs to know about both the human's behaviours and the system state, and can itself be factored into multiple correlated attributes. The observations $\boldsymbol{\Omega_x}$ are anything the system observes in the environment that gives it evidence about the state $\mathbf{X}$. The system actions $\boldsymbol{A}$ are things the system can do to change the state (e.g. modify the interface, move a robot) or to modify the human's behaviours (e.g. give a prompt, give an order). Finally, the reward function is defined over state-action pairs and rewards those states and actions that are beneficial overall to the goals of the system-human interaction. Missing from this basic model are the affective elements of the interaction, which can be a significant influence on a person's behaviour. For example, a tutor who imperatively challenges a student *"Do this exercise now!"* will be viewed differently than one who meekly suggests *"here's an exercise you might try..."*. While the propositional content of the action is the same (the same exercise is given), the affective delivery will influence different students in different ways. While some may respond vigorously to the challenge, others may respond more effectively to the suggestion.

3.1 Basic Formulation

Bayesian affect control theory (*BayesAct* for short) gives us a principled way to add the emotional content to a human interactive system by making four key additions to the basic POMDP model, as shown by the dashed lines in Figure 1(a).

1. An unobservable variable, $\mathbf{Y}$, describes sentiments of the *agent* about identities and behaviours. The dynamics of $\mathbf{Y}$ is given by empirical measurements in ACT (see below).

2. Observations $\boldsymbol{\Omega_f}$ give evidence about the part of the sentiments $\mathbf{Y}$ encoding behaviours of the *client*.

3. The actions of the *agent* are now expanded to be $\mathbf{B} = \{A, \mathbf{B_a}\}$. The normal state transition dynamics can still occur based only on A, but now the action space also must include an affective "how" for the delivery "what" of an action. This action space is easily constructed, but may cause significant computational issues for the evaluation of actions during planning (see Section 3.4).

4. The application-specific dynamics of $\mathbf{X}$ now depends on sentiments, $Pr(\mathbf{X}'|\mathbf{X}, \mathbf{Y}', A)$, and will generally follow the original distribution $Pr(\mathbf{X}'|\mathbf{X}, A)$, but now moderated by deflection. For example, $\mathbf{X}$ may move towards a goal, but less quickly when deflection is high.There are different ways to formulate this, and possibly many more to be discovered as this method is applied to other domains. One way will be to simply add noise to the distribution, with the noise being proportional to the amount of deflection between fundamentals and transients.

We assume that time is discrete, and agents take turns acting (so the "turn" is one element of $\mathbf{X}$). This assumption does not limit the generality of the approach, as anything beyond simple turn-taking (e.g. backchannel responses, interruptions) could be included in $\mathbf{X}$, and time steps are defined by the transitions therein. We refer to the modified POMDP model as a Partially Observable Affect Control Process (PO-ACP), shown as a graphical model in Figure 1(b). We will continue to refer to the theory and model as *BayesAct* for short.

Let $\mathbf{F} = \{\mathrm{F}_{ij}\}$ denote the set of fundamental *agent* sentiments about itself where each feature $\mathrm{F}_{ij}, i \in \{a, b, c\}, j \in \{e, p, a\}$ denotes the j^{th} fundamental sentiment (evaluation, potency or activity) about the i^{th} interaction object: actor (*agent*), behaviour, or object (*client*). Let $\mathbf{T} = \{\mathrm{T}_{ij}\}$ be similarly defined and denote the set of transient *agent* sentiments. Variables F_{ij} and T_{ij} are continuous valued and $\mathbf{F}, \mathbf{T}$ are each vectors in a continuous nine-dimensional space. Affect control theory encodes the identities as being for "actor" (A, the person acting) and "object" (O, the person being acted upon). In *BayesAct*, we encode identities as being for "agent" and "client" (regardless of who is currently acting). However, this means that we do need to know who is currently acting, and the prediction equations will need to be "inverted" to handle turn-taking during an interaction. This poses no significant issues, but must be kept in mind if one is trying to understand the connection between the two formulations. In particular, since we are assuming a discrete time model, then the "turn" (who is currently acting) will have to be represented (at the very least) in $\mathbf{X}$. Considering time to be event-based, however, we can still handle interruptions, but simultaneous action by both agents will need further consideration. One method may be to have a dynamically changing environment noise: an agent cannot receive a communication from another agent if it is simultaneously using the same channel of communication, for example.

In the following, we will use symbolic indices in $\{a, b, c\}$ and $\{e, p, a\}$, and define two simple index dictionaries, d_i and d_a, to map between the symbols and numeric indices ({0,1,2}) in matrices and vectors so that, $d_i(a) = 0, d_i(b) = 1, d_i(c) = 2$ and $d_a(e) = 0, d_a(p) = 1, d_a(a) = 2$. Thus, we can write an element of $\mathbf{F}$ as $\mathbf{F}_{bp}$ which will be the k^{th} element of the vector representation of $\mathbf{F}$, where $k = 3d_i(b) + d_a(p)$. We will use a "dot" to represent that all values are present if there is any ambiguity. For example, the behaviour component of $\mathbf{F}$ is written $\mathbf{F}_{b\cdot}$ or $\mathbf{F}_b$ for short. A covariance in the space of $\mathbf{F}$ might be written Σ (a 9×9 matrix), and the element at position (n, m) of this matrix would be denoted $\Sigma_{ij,kl}$ where $n = 3d_i(i) + d_a(j)$ and $m = 3d_i(l) + d_a(k)$. We can then refer to the middle 3×3 block of

Σ (the covariance of $\mathbf{F}_b$ with itself) as $\Sigma_{b \cdot b \cdot}$. Although the extra indices seem burdensome, they will be useful later on as we will see.

The POMDP action will be denoted here as $\mathbf{B} = \{A, \mathbf{B_a}\}$ (behaviour of the agent), but this is treated differently than other variables as the *agent* is assumed to have freedom to choose the value for this variable. The action is factored into two parts: A is the propositional content of the action, and includes things that the agent does to the application (change screens, present exercises, etc), while $\mathbf{B_a}$ is the emotional content of the action. Thus, $\mathbf{B_a}$ gives the affective "how" for the delivery "what" of an action, A. The affective action $\mathbf{B_a} = \{\mathrm{B}_{ae}, \mathrm{B}_{ap}, \mathrm{B}_{aa}\}$ will also be continuous valued and three dimensional: the agent sets an EPA value for its action (not a propositional action label). The client behaviour is implicitly represented in the fundamental sentiment variables $\mathbf{F}_b$, and we make some observations of the behaviour, $\boldsymbol{\Omega_f}$. For example, if the computer application is using a microphone and a speech recognition engine to "hear" what the client is saying, then $\boldsymbol{\Omega_f}$ found by looking up the output of the speech recognizer (a set of keywords, for example) in a database of EPA values for keywords or phrases. This lookup procedure is not defined in this model - it is a way of assigning EPA values to statements by using using existing knowledge bases.

Finally, a set of variables $\mathbf{X}$ represents the state of the system (e.g. the state of the computer application or interface). Variables in $\mathbf{X}$ will normally be discrete-valued, but we can allow for continuous variables as well. We don't assume that the system state or client behaviour are directly observable, and so also use sets of variables $\boldsymbol{\Omega_x}$.

Figure 1(b) shows a graphical model of the ACT model we are proposing. This type of model is a partially observable Markov decision process, or POMDP (see Section 2.2 for basic definitions of the model), as we can make the association of the *state* $\mathbf{S} = \{\mathbf{F}, \mathbf{T}, \mathbf{X}\}$, the observations $\mathbf{O} = \{\boldsymbol{\Omega_x}, \boldsymbol{\Omega_f}\}$, and the action $A = \{A, \mathbf{B_a}\}$. We denote $\mathbf{Y}$=$\{\mathbf{F}, \mathbf{T}\}$, $\boldsymbol{S}$=$\{\mathbf{Y}, \mathbf{X}\}$ and $\boldsymbol{\Omega}$=$\{\boldsymbol{\Omega_f}, \boldsymbol{\Omega_x}\}$.

The *deflection* in affect control theory is a nine-dimensional weighted Euclidean distance measure between fundamental sentiments $\mathbf{F}$ and transient impressions $\mathbf{T}$ (Section 2.1). Here, we propose that this distance measure is the logarithm of a probabilistic potential

$$\varphi(\mathbf{f}', \boldsymbol{\tau}') \propto e^{-(\mathbf{f}'-\boldsymbol{\tau}')^T \Sigma^{-1} (\mathbf{f}'-\boldsymbol{\tau}')}. \tag{2}$$

The covariance Σ is a generalisation of the "weights" (Equation (1) and [1]), as it allows for some sentiments to be more significant than others when making predictions (i.e. their deflections are more carefully controlled by the participants), but also represents correlations between sentiments in general. The empirically derived prediction equations of ACT can be written as $\boldsymbol{\tau}' = \boldsymbol{M}\mathscr{G}(\mathbf{f}', \boldsymbol{\tau}, \mathbf{x})$ where $\mathscr{G}$ is a non-linear operator that combines $\boldsymbol{\tau}$, $\mathbf{f}'$, and $\mathbf{x}$, and $\boldsymbol{M}$ is the prediction matrix (see Section 2.1 and [1]). Putting the deflection potential together with the dynamics, we have the probabilistic generalisation of the affect control principle (Definition 1):

$$Pr(\mathbf{f}'|\mathbf{f}, \boldsymbol{\tau}, \mathbf{x}, \mathbf{b_a}, \varphi) \propto e^{-\psi(\mathbf{f}', \boldsymbol{\tau}, \mathbf{x}) - \xi(\mathbf{f}', \mathbf{f}, \mathbf{b_a}, \mathbf{x})} \tag{3}$$

where $\psi(\mathbf{f}', \boldsymbol{\tau}, \mathbf{x}) = (\mathbf{f}' - \boldsymbol{M}\mathscr{G}(\mathbf{f}', \boldsymbol{\tau}, \mathbf{x}))^T \boldsymbol{\Sigma}^{-1} (\mathbf{f}' - M\mathscr{G}(\mathbf{f}', \boldsymbol{\tau}, \mathbf{x}))$ and ξ represents the temporal dynamics of $\mathbf{f}'$, encoding both the stability of affective identities and the predictive dynamics of affective behaviours (as shown in Figure 1(b))ξ is such that $\mathbf{f}_b'$ is equal to $\mathbf{b_a}$ if the *agent* is acting, and otherwise is unconstrained, and $\mathbf{f}_a', \mathbf{f}_c'$ are likely to be close to $\mathbf{f}_a, \mathbf{f}_c$, respectively. Equation (3) can be re-written as a set of multivariate Gaussian distributions indexed by $\mathbf{x}$, with means and covariances that are non-linearly dependent on $\mathbf{f}, \mathbf{b_a}$ and $\boldsymbol{\tau}$. The full derivation is in Section 3.3.

The other factors in *BayesAct* are as follows:

- $Pr(\boldsymbol{\tau}'|\boldsymbol{\tau}, \mathbf{f}', \mathbf{x})$ is the probability distribution over transient sentiments (computed using the matrix $\boldsymbol{M}$ as in ACT). The simplest form is a deterministic function, and we write $P(\boldsymbol{\tau}'|\boldsymbol{\tau}, \mathbf{f}', \mathbf{x}) = \delta(\boldsymbol{\tau}' - \boldsymbol{M}\mathscr{G}(\mathbf{f}', \boldsymbol{\tau}, \mathbf{x}))$, where $\delta(z) = 1$ if $z = 0$ and $\delta(z) = 1$ otherwise. The operator $\mathscr{G}$ constructs an intermediate vector $\boldsymbol{t}$ from $\boldsymbol{\tau}$ according to a set of empirically derived non-linear combinators. This construction is detailed in [1] (Equation 11.16), but here, we use behaviour sentiments from $\mathbf{f}'$, and settings from $\mathbf{x}$, and $\boldsymbol{M}$ is the prediction matrix ([1] Equation 11.15). Since $\mathscr{G}$ only uses the behaviour component of $\mathbf{f}'$, we can also group terms together and write

$$Pr(\boldsymbol{\tau}'|\boldsymbol{\tau}, \mathbf{f}', \mathbf{x}) = \delta(\boldsymbol{\tau}' - \mathscr{H}(\boldsymbol{\tau}, \mathbf{x})\mathbf{f}'_b - \mathscr{C}(\boldsymbol{\tau}, \mathbf{x})), \tag{4}$$

where $\mathscr{H}$ and $\mathscr{C}$ are 9×3 and 9×1 matrices of coefficients from the dynamics $\boldsymbol{M}$ and $\mathscr{G}$ together. That is, $\mathscr{H}_{ijk}$ is the element at row $3d_i(i) + d_a(j)$, column $d_a(k)$ of $\mathscr{H}$, and the sum of all terms in row $3d_i(i) + d_a(j)$ of $\boldsymbol{M'}\boldsymbol{t}$ that contain B_k is then $\mathscr{H}_{ijk}B_k$. Similarly, the element at row $d_i(i)$ of $\mathscr{C}$, $\mathscr{C}_i$, is the sum of all terms in row $d_i(i)$ of $\boldsymbol{M'}\boldsymbol{t}$ that contain no $B.$ at all. Simply put, the matrices $\mathscr{H}$ and $\mathscr{C}$ are a refactoring of the operators $\boldsymbol{M}\mathscr{G}$, such that a linear function of $\mathbf{f}'_b$ is obtained. Recall that, due to our model being over "agent" and "client", instead of over "actor" and "object", the matrices $\mathscr{H}$ and $\mathscr{C}$ will change depending on whose turn it is, effectively swapping "agent" and "object" in both $\boldsymbol{M}$ and $\mathscr{G}$ for each change of turn. In general, these dynamics may also depend on other aspects of the state (e.g. the state could include the "setting" from ACT [37]), and hence, we include a dependence on $\mathbf{x}$.

- $R(\mathbf{f}, \boldsymbol{\tau}, \mathbf{x})$ is a reward function giving the immediate reward given to the *agent*. We assume an additive function

$$R(\mathbf{f}, \boldsymbol{\tau}, \mathbf{x}) = R_x(\mathbf{x}) + R_s(\mathbf{f}, \boldsymbol{\tau}) \tag{5}$$

where R_x encodes the application goals (e.g. to get a student to pass a test), and

$$R_s \propto -(\mathbf{f} - \boldsymbol{\tau})^2$$

depends on the deflection. The relative weighting and precise functional form of these two reward functions require further investigation, but in the examples we show can be simply defined. The affect control principle only considers R_s, and here we generalise to include other goals.

- $Pr(\mathbf{x}'|\mathbf{x}, \mathbf{f}', \boldsymbol{\tau}', a)$ denotes how the application progresses given the previous state, the fundamental and transient sentiments, and the (propositional) action of the *agent*. The dependence on the sentiments is important: it indicates that the system state will progress differently depending on the affective state of the user and agent. In Section 4.2 we explore this idea further in the context of a specific application by hypothesising that the system state will more readily progress towards a goal if the deflection (difference between fundamental and transient sentiments) is low. The *client* behaviours (explicitly represented as b'_c) may also be included here as part of $\mathbf{x}'$, if they have some effect on the (propositional, non-emotional) state. In general this will be stochastic, but in many cases is will be deterministic, such as when the application changes because of *client* mouse clicks or specific *agent* actions. For the case of deterministic state dynamics (given client behaviour) but stochastic client behaviours, we will write

$$Pr(\mathbf{x}', b'_c|\mathbf{x}, \mathbf{f}', \boldsymbol{\tau}', a) = Pr(b'_c|\mathbf{x}, \mathbf{f}', \boldsymbol{\tau}', a)Pr(\mathbf{x}'|b'_c, \mathbf{x}, \mathbf{f}', \boldsymbol{\tau}', a) \tag{6}$$

$$= Pr(b'_c|\mathbf{x}, \mathbf{f}', \boldsymbol{\tau}', a)\delta(\mathbf{x}' - \mathscr{X}(\mathbf{x}, \mathbf{f}', \boldsymbol{\tau}', a, b'_c)). \tag{7}$$

where $\mathscr{X}(\mathbf{x}, \mathbf{f}', \boldsymbol{\tau}', a, b'_c)$ is a deterministic function giving how the application state proceeds over time given *agent* and *client* behaviours, and the current sentiments.

- $Pr(\boldsymbol{\omega_f}|\mathbf{f}), Pr(\boldsymbol{\omega_x}|\mathbf{x})$ observation functions for the client behaviour sentiment and system state, respectively. These functions are stochastic in general, but may be deterministic for the system state (so that $\mathbf{X}$ is fully observable). It will not be deterministic for the client behaviour sentiment as we have no way of directly measuring this (it can only be inferred from data).

3.2 Transition Dynamics

Overall, we are interested in computing the transition probability over time: the probability distribution over the sentiments and system state given the history of actions and observations. Denoting $\boldsymbol{S} =$

$\{\mathbf{F}, \mathbf{T}, \mathbf{X}\}$, and $\mathbf{\Omega} = \{\mathbf{\Omega}_{\boldsymbol{f}}, \mathbf{\Omega}_{\boldsymbol{x}}\}$, and $\mathbf{\Omega}_t, \mathbf{b}_{\mathbf{a}t}, \boldsymbol{S}_t$ are the observations, *agent* action, and state at time t, we want:

$$b(\boldsymbol{s}_t) \equiv Pr(\boldsymbol{s}_t|\boldsymbol{\omega}_0, \ldots, \boldsymbol{\omega}_t, \mathbf{b}_{\mathbf{a}0}, \ldots, \mathbf{b}_{\mathbf{a}t})$$

which can be written as

$$\begin{aligned}
b(\boldsymbol{s}_t) &= \int_{\boldsymbol{s}_{t-1}} Pr(\boldsymbol{s}_t, \boldsymbol{s}_{t-1}|\boldsymbol{\omega}_0, \ldots, \boldsymbol{\omega}_t, \mathbf{b}_{\mathbf{a}0}, \ldots, \mathbf{b}_{\mathbf{a}t}) \\
&\propto \int_{\boldsymbol{s}_{t-1}} Pr(\boldsymbol{\omega}_t|\boldsymbol{s}_t) Pr(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}, \boldsymbol{\omega}_0, \ldots, \boldsymbol{\omega}_{t-1}, \mathbf{b}_{\mathbf{a}0}, \ldots, \mathbf{b}_{\mathbf{a}t}) Pr(\boldsymbol{s}_{t-1}|\boldsymbol{\omega}_0, \ldots, \boldsymbol{\omega}_{t-1}, \mathbf{b}_{\mathbf{a}0}, \ldots, \mathbf{b}_{\mathbf{a}t}) \\
&= Pr(\boldsymbol{\omega}_t|\boldsymbol{s}_t) \int_{\boldsymbol{s}_{t-1}} Pr(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}, \mathbf{b}_{\mathbf{a}t}) b(\boldsymbol{s}_{t-1}) \\
&= Pr(\boldsymbol{\omega}_t|\boldsymbol{s}_t) \mathbb{E}_{b(\boldsymbol{s}_{t-1})} \left[Pr(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}, \mathbf{b}_{\mathbf{a}t}) \right] \qquad (8)
\end{aligned}$$

where $Pr(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}, \mathbf{b}_t)$ factored according to Figure 1(b):

$$Pr(\boldsymbol{s}_t|...) = Pr(\mathbf{x}'|\mathbf{x}, \mathbf{f}', \boldsymbol{\tau}', a) Pr(\boldsymbol{\tau}'|\boldsymbol{\tau}, \mathbf{f}', \mathbf{x}) Pr(\mathbf{f}'|\mathbf{f}, \boldsymbol{\tau}, \mathbf{x}, \mathbf{b}_{\mathbf{a}}) \qquad (9)$$

This gives us a recursive formula for computing the distribution over the state at time t as an expectation of the transition dynamics taken with respect to the distribution state at time $t-1$. Now, we have that and $Pr(\boldsymbol{\omega}|\boldsymbol{s}) = Pr(\boldsymbol{\omega}_{\boldsymbol{x}}|\mathbf{x}') Pr(\boldsymbol{\omega}_{\boldsymbol{f}}|\mathbf{f}')$ and rewriting $\boldsymbol{s}_t \equiv \boldsymbol{s}'$ and $\boldsymbol{s}_{t-1} \equiv \boldsymbol{s}$, we have:

$$\begin{aligned}
b(\boldsymbol{s}') &= Pr(\boldsymbol{\omega}_{\boldsymbol{x}}|\mathbf{x}') Pr(\boldsymbol{\omega}_{\boldsymbol{f}}|\mathbf{f}') \mathbb{E}_{b(\boldsymbol{s})} \left[Pr(\mathbf{f}', \boldsymbol{\tau}', \mathbf{x}'|\mathbf{f}, \boldsymbol{\tau}, \mathbf{x}, \mathbf{b}_{\mathbf{a}}, a) \right] \\
&= Pr(\boldsymbol{\omega}_{\boldsymbol{x}}|\mathbf{x}') Pr(\boldsymbol{\omega}_{\boldsymbol{f}}|\mathbf{f}') \mathbb{E}_{b(\boldsymbol{s})} \left[Pr(\mathbf{x}'|\mathbf{x}, \mathbf{f}', \boldsymbol{\tau}', a) Pr(\boldsymbol{\tau}'|\boldsymbol{\tau}, \mathbf{f}', \mathbf{x}) Pr(\mathbf{f}'|\mathbf{f}, \boldsymbol{\tau}, \mathbf{x}, \mathbf{b}_{\mathbf{a}}) \right] \qquad (10)
\end{aligned}$$

The first four terms correspond to parameters of the model as explained in the last section, while we need to develop a method for computing the last term, which we do in the next section.

The belief state can be used to compute expected values of quantities of interest defined on the state space, such as the expected deflection

$$\mathbb{E}_{b(\boldsymbol{s})}[D(\boldsymbol{s})] = \int_{\boldsymbol{s}} b(\boldsymbol{s}) D(\boldsymbol{s}), \qquad (11)$$

where $D(\boldsymbol{s})$ is the deflection of $\boldsymbol{s}$. This gives us a way to connect more closely with emotional "labels" from appraisal theories. For example, if one wanted to compute the expected value of an emotion such as "Joy" in a situation with certain features (expectedness, events, persons, times), then the emotional content of that situation would be explicitly represented in our model as a distribution over the E-P-A space, $b(\boldsymbol{s})$, and the expected value of "Joy" would be $\mathbb{E}_{b(\boldsymbol{s})}[Joy(\boldsymbol{s})] = \int_{\boldsymbol{s}} b(\boldsymbol{s}) Joy(\boldsymbol{s})$, where $Joy(\boldsymbol{s})$ is the amount of joy produced by the sentiment $\boldsymbol{s}$. This expectation would be a single number on the same scale as $Joy(\boldsymbol{s})$ that gives the amount of "Joy" being felt.

3.3 Estimating Behaviour Sentiment Probabilities

Here we describe how to compute $Pr(\mathbf{f}'|\mathbf{f}, \boldsymbol{\tau}, \mathbf{x}, \mathbf{b}_{\mathbf{a}})$. This is the *agent*'s prediction of what the *client* will do next, and is based partly on the principle that we expect the *client* to do what is optimal to reduce deflection in the future, given the identities of *agent* and *client*.

We denote the probability distribution of interest as a set of parameters $\boldsymbol{\Theta}_{\boldsymbol{f}}$. Each parameter in this set, $\boldsymbol{\theta}_{\boldsymbol{f}}$, will be a probability of observing a value of $\mathbf{f}'$ given values for $\mathbf{f}, \boldsymbol{\tau}, \mathbf{b}_{\mathbf{a}}$ and $\mathbf{x}$. We write $\boldsymbol{\theta}_{\boldsymbol{f}}(\mathbf{f}'; \mathbf{f}, \boldsymbol{\tau}, \mathbf{x}, \mathbf{b}_{\mathbf{a}}) = Pr(\mathbf{f}'|\mathbf{f}, \boldsymbol{\tau}, \mathbf{x}, \mathbf{b}_{\mathbf{a}}, \boldsymbol{\theta}_{\boldsymbol{f}})$, so that the distribution over $\boldsymbol{\theta}_{\boldsymbol{f}}$ given the knowledge that $\boldsymbol{\tau}'$

and $\mathbf{f}'$ are close according to $\varphi(\mathbf{f},\boldsymbol{\tau})$ is[1]:

$$Pr(\boldsymbol{\theta_f}|\varphi) \propto \int_{\substack{\mathbf{f}',\boldsymbol{\tau}',\mathbf{b_a}\\ \mathbf{f},\boldsymbol{\tau},\mathbf{x}}} Pr(\boldsymbol{\theta_f},\mathbf{f}',\boldsymbol{\tau}',\mathbf{b_a},\mathbf{f},\boldsymbol{\tau},\varphi) \tag{12}$$

$$= \int_{\substack{\mathbf{f}',\boldsymbol{\tau}',\mathbf{b_a}\\ \mathbf{f},\boldsymbol{\tau},\mathbf{x}}} \varphi(\mathbf{f}',\boldsymbol{\tau}')Pr(\boldsymbol{\tau}'|\mathbf{x},\mathbf{f}',\boldsymbol{\tau})Pr(\mathbf{f}'|\mathbf{f},\boldsymbol{\tau},\mathbf{x},\mathbf{b_a},\boldsymbol{\theta_f}) \; Pr(\boldsymbol{\theta_f})Pr(\mathbf{f})Pr(\boldsymbol{\tau})Pr(\mathbf{x})Pr(\mathbf{b_a}) \tag{13}$$

$$= \int_{\substack{\mathbf{f}',\boldsymbol{\tau}',\mathbf{b_a}\\ \mathbf{f},\boldsymbol{\tau},\mathbf{x}}} e^{-(\mathbf{f}'-\boldsymbol{\tau}')^T\Sigma^{-1}(\mathbf{f}'-\boldsymbol{\tau}')} Pr(\boldsymbol{\tau}'|\mathbf{x},\mathbf{f}',\boldsymbol{\tau})\boldsymbol{\theta_f}(\mathbf{f}';\mathbf{f},\boldsymbol{\tau},\mathbf{x},\mathbf{b_a})Pr(\boldsymbol{\theta_f}) \tag{14}$$

where by $\int_z$ we mean $\int_{z\in\mathcal{Z}}$ ($\mathcal{Z}$ is the domain of Z) and we have left off the infinitesimals (e.g. $d\mathbf{f}', d\boldsymbol{\tau}'$). We have assumed even priors over $\mathbf{f}$,$\boldsymbol{\tau}$, $\mathbf{x}$ and $\mathbf{b_a}$. The expression (14) will give us a posterior update to the parameters of the distribution over $\boldsymbol{\theta_f}$. During interaction, we will observe $\boldsymbol{\Omega_f} = \boldsymbol{\omega_f}$, from which a distribution over $\mathbf{F}$ can be inferred

Equation (14) gives the general form for this distribution, but this can be simplified by using the determinism of some of the distributions involved, as described at the start of this section. The deterministic function for the distribution over $\boldsymbol{\tau}'$ will select specific values for these variables, and we find that:

$$Pr(\boldsymbol{\theta_f}|\varphi) \propto \int_{\substack{\mathbf{f}',\mathbf{b_a}\\ \mathbf{f},\boldsymbol{\tau},\mathbf{x}}} e^{-\psi(\mathbf{f}',\boldsymbol{\tau},\mathbf{x})}\boldsymbol{\theta_f}(\mathbf{f}';\mathbf{f},\boldsymbol{\tau},\mathbf{x},\mathbf{b_a})Pr(\boldsymbol{\theta_f}) \tag{15}$$

where

$$\psi(\mathbf{f}',\boldsymbol{\tau},\mathbf{x}) = (\mathbf{f}' - M\mathscr{G}(\mathbf{f}',\boldsymbol{\tau},\mathbf{x}))^T\boldsymbol{\Sigma}^{-1}(\mathbf{f}' - M\mathscr{G}(\mathbf{f}',\boldsymbol{\tau},\mathbf{x})) \tag{16}$$

is the *deflection* between fundamental and transient sentiments.

Equation (15) gives us an expression for a distribution over $\boldsymbol{\theta_f}$, which we can then use to estimate a distribution over $\mathbf{F}' = \mathbf{f}'$ given the state $\{\mathbf{f},\boldsymbol{\tau},\mathbf{x},\mathbf{b_a}\}$ and the known relation between fundamentals and transients, φ (ignoring the observations $\boldsymbol{\Omega_f}$ and $\boldsymbol{\Omega_x}$ for now):

$$Pr(\mathbf{f}'|\mathbf{f},\boldsymbol{\tau},\mathbf{x},\mathbf{b_a},\varphi) \propto \int_{\boldsymbol{\theta_f},\boldsymbol{\tau}'} Pr(\boldsymbol{\theta_f},\mathbf{f}',\mathbf{f},\boldsymbol{\tau}',\boldsymbol{\tau},\mathbf{x},\mathbf{b_a},\varphi) \tag{17}$$

$$= \int_{\boldsymbol{\theta_f}} e^{-\psi(\mathbf{f}',\boldsymbol{\tau},\mathbf{x})}\boldsymbol{\theta_f}(\mathbf{f}';\mathbf{f},\boldsymbol{\tau},\mathbf{x},\mathbf{b_a})Pr(\boldsymbol{\theta_f}|\mathbf{x}) \tag{18}$$

$$= e^{-\psi(\mathbf{f}',\boldsymbol{\tau},\mathbf{x})} \int_{\boldsymbol{\theta_f}} \boldsymbol{\theta_f}(\mathbf{f}';\mathbf{f},\boldsymbol{\tau},\mathbf{x},\mathbf{b_a})Pr(\boldsymbol{\theta_f}|\mathbf{x}) \tag{19}$$

$$= e^{-\psi(\mathbf{f}',\boldsymbol{\tau},\mathbf{x})} \left[\mathbb{E}_{Pr(\boldsymbol{\theta_f}|\mathbf{x})}(\boldsymbol{\theta_f})\right] \tag{20}$$

The first term is a distribution over $\mathbf{f}'$ that represents our assumption of minimal deflection, while the second is the expected value of the parameter $\boldsymbol{\theta_f}$ given the prior. This expectation will give us the most likely value of $\boldsymbol{\theta_f}$ given only the system state $\mathbf{x}$. We know two things about the transition dynamics ($\boldsymbol{\theta_f}$) that we can encode in the prior. First, we know that the behaviour will be set equal to the *agent*'s action if it is the *agent*'s turn (hence the dependence on $\mathbf{x}$). Second, we know that identities are not expected to change very quickly. Therefore, we have that

$$\mathbb{E}_{Pr(\boldsymbol{\theta_f}|\mathbf{x})}(\boldsymbol{\theta_f}) \propto e^{-(\mathbf{f}'-\langle\mathbf{f},\mathbf{b_a}\rangle)^T\boldsymbol{\Sigma}_f^{-1}(\mathbf{x})(\mathbf{f}'-\langle\mathbf{f},\mathbf{b_a}\rangle)} \tag{21}$$

[1]We are postulating an undirected link in the graph between $\boldsymbol{\tau}$ and $\mathbf{f}$. An easy way to handle this undirected link properly is to replace it with an equivalent set of directed links by adding a new Boolean variable, D, that is conditioned by both $\mathbf{T}$ and $\mathbf{F}$, and such that $Pr(D = True|\boldsymbol{\tau},\mathbf{f}) \propto \varphi(\boldsymbol{\tau},\mathbf{f})$. We then set $D = True$ if we have the knowledge that $\mathbf{T}$ and $\mathbf{F}$ are close together according to $\varphi(\mathbf{F},\mathbf{T})$, and the quantity of interest is $Pr(\boldsymbol{\theta_f}|D = True)$. In the text, we use the shorthand $Pr(\boldsymbol{\theta_f}|\varphi)$ to avoid having to introduce D.

where $\langle \mathbf{f}, \mathbf{b_a} \rangle$ is $\mathbf{f}$ for the identities and $\mathbf{b_a}$ for the behaviours

$$\langle \mathbf{f}, \mathbf{b_a} \rangle \equiv \begin{bmatrix} \mathbf{f}_a \\ \mathbf{b_a} \\ \mathbf{f}_c \end{bmatrix} \tag{22}$$

and $\mathbf{\Sigma}_f(\mathbf{x})$ is the covariance matrix for the inertia of the fundamentals, and including the setting of behaviour fundamentals by the *agent* action, and is a set of parameters governing the strength of our prior beliefs that the identities of client and agent will remain constant over time[2]. Thus, $\mathbf{\Sigma}_f$ is a 9×9 block matrix:

$$\mathbf{\Sigma}_f(\mathbf{x}) = \begin{bmatrix} \boldsymbol{I}_3\beta_a^2 & 0 & 0 \\ 0 & \boldsymbol{I}_3\beta_b^2(\mathbf{x}) & 0 \\ 0 & 0 & \boldsymbol{I}_3\beta_c^2 \end{bmatrix} \tag{23}$$

where β_a^2 and β_c^2 are the variances of *agent* and *client* identity fundamentals (i.e. how much we expect *agent* and *client* to change their identities over time), and $\beta_b^2(\mathbf{x})$ is infinite for a *client* turn and is zero for an *agent* turn. Writing $\xi(\mathbf{f}', \mathbf{f}, \mathbf{b_a}, \mathbf{x}) \equiv (\mathbf{f}' - \langle \mathbf{f}, \mathbf{b_a} \rangle)^T \mathbf{\Sigma}_f^{-1}(\mathbf{x})(\mathbf{f}' - \langle \mathbf{f}, \mathbf{b_a} \rangle)$, we therefore have that:

$$Pr(\mathbf{f}'|\mathbf{f}, \boldsymbol{\tau}, \mathbf{x}, \mathbf{b_a}, \varphi) \propto e^{-\psi(\mathbf{f}', \boldsymbol{\tau}, \mathbf{x}) - \xi(\mathbf{f}', \mathbf{f}, \mathbf{b_a}, \mathbf{x})} \tag{24}$$

We can now estimate what the most likely value of $\mathbf{F}'$ by computing

$$\mathbf{F}'^* = \arg\max_{\mathbf{f}'} Pr(\mathbf{f}'|\mathbf{f}, \boldsymbol{\tau}, \mathbf{x}, \mathbf{b_a}, \varphi).$$

Intuitively, this expression will be maximized for exactly the behaviour that minimizes the deflection as given by ψ, tempered by the inertia of changing identities given by ξ. This is the generalisation of the *affect control principle* (Definition 1, see also Appendix B). We can rewrite this by first rewriting the matrix $\mathscr{H}$ as

$$\mathscr{H} = \begin{bmatrix} \mathscr{H}_a \\ \mathscr{H}_b \\ \mathscr{H}_c \end{bmatrix}$$

where $\mathscr{H}_a \equiv \mathscr{H}_{a\cdot\cdot}$ (a 3×3 matrix giving the rows of $\mathscr{H}$ in which $\mathscr{H}_{ijk}$ have $i = a$) and similarly for $\mathscr{H}_b$ and $\mathscr{H}_c$. We also define $\mathbf{I_3}$ as the 3×3 identity matrix and $\mathbf{0_3}$ as the 3×3 matrix of all zeros. We can then write a matrix

$$\mathscr{K} = \begin{bmatrix} \boldsymbol{I}_3 & -\mathscr{H}_a & \mathbf{0}_3 \\ \mathbf{0}_3 & \boldsymbol{I}_3 - \mathscr{H}_b & \mathbf{0}_3 \\ \mathbf{0}_3 & -\mathscr{H}_c & \boldsymbol{I}_3 \end{bmatrix}$$

Using $\mathscr{K}$, we can now write the general form for ψ starting from Equation (16) as:

$$\psi(\mathbf{f}', \boldsymbol{\tau}, \mathbf{x}) = (\mathbf{f}' - \mathscr{H}(\boldsymbol{\tau}, \mathbf{x})\mathbf{f}_b' - \mathscr{C}(\boldsymbol{\tau}, \mathbf{x}))^T \mathbf{\Sigma}^{-1} (\mathbf{f}' - \mathscr{H}(\boldsymbol{\tau}, \mathbf{x})\mathbf{f}_b' - \mathscr{C}(\boldsymbol{\tau}, \mathbf{x})) \tag{25}$$

$$= (\mathscr{K}\mathbf{f}' - \mathscr{C})^T \mathbf{\Sigma}^{-1} (\mathscr{K}\mathbf{f}' - \mathscr{C}) \tag{26}$$

$$= (\mathbf{f}' - \mathscr{K}^{-1}\mathscr{C})^T \mathscr{K}^T \mathbf{\Sigma}^{-1} \mathscr{K} (\mathbf{f}' - \mathscr{K}^{-1}\mathscr{C}) \tag{27}$$

and thus, if we ignore the inertia from previous fundamentals, ξ, we recognize Equation (24) as the expectation of a Gaussian or normal distribution with a mean of $\mathscr{K}^{-1}\mathscr{C}$ and a covariance of $\mathbf{\Sigma}_\tau \equiv \mathscr{K}^{-1}\mathbf{\Sigma}(\mathscr{K}^T)^{-1}$. Taking ξ into account means that we have a product of Gaussians, itself also a Gaussian the mean and covariance of which can be simply obtained by completing the squares to find a covariance, $\mathbf{\Sigma}_n$ equal to the sum in quadrature of the covariances, and a mean, $\boldsymbol{\mu_n}$, that is proportional to a weighted sum of $\mathscr{K}^{-1}\mathscr{C}$ and $\langle \mathbf{f}, \mathbf{b_a} \rangle$, with weights given by the normalised covariances of $\mathbf{\Sigma}_n\mathbf{\Sigma}_\tau^{-1}$ and $\mathbf{\Sigma}_n\mathbf{\Sigma}_f^{-1}$, respectively.

[2] It may also be the case that $\mathbf{b_a}$ can change the *agent* identity directly, so that $\mathbf{b_a}$ is six-dimensional and $\langle \mathbf{f}, \mathbf{b_a} \rangle = [\mathbf{b_a}, \mathbf{f}_c]^T$.

Putting it all together, we have that

$$Pr(\mathbf{f}'|\mathbf{f}, \boldsymbol{\tau}, \mathbf{x}, \mathbf{b_a}, \varphi) \propto e^{-(\mathbf{f}'-\boldsymbol{\mu_n})^T \boldsymbol{\Sigma}_n^{-1}(\mathbf{f}'-\boldsymbol{\mu_n})} \tag{28}$$

where

$$\boldsymbol{\mu_n} = \boldsymbol{\Sigma}_n \mathscr{K}^T(\boldsymbol{\tau}, \mathbf{x})\boldsymbol{\Sigma}^{-1}\mathscr{C}(\boldsymbol{\tau}, \mathbf{x}) + \boldsymbol{\Sigma}_n \boldsymbol{\Sigma}_f^{-1}(\mathbf{x})\langle \mathbf{f}, \mathbf{b_a}\rangle \tag{29}$$

$$\boldsymbol{\Sigma}_n = (\mathscr{K}^T(\boldsymbol{\tau}, \mathbf{x})\boldsymbol{\Sigma}^{-1}\mathscr{K}(\boldsymbol{\tau}, \mathbf{x}) + \boldsymbol{\Sigma}_f^{-1}(\mathbf{x}))^{-1}. \tag{30}$$

The distribution over $\mathbf{f}'$ in Equation (28) is a Gaussian distribution, but has a mean and covariance that are dependent on $\mathbf{f}, \boldsymbol{\tau}, \mathbf{x}$ and $\mathbf{b_a}$ through the non-linear function $\mathscr{K}$. Thus, is is not simple to use this analytically as we will explore further below.

The "optimal" behaviour from [1] is, in fact, only an approximation as the identities are treated as constants when optimising the behaviour in [1] (and similarly for identities: behaviours are held constant). However, ACT hypothesises that the overall deflection will be minimised. Therefore, our equations are the exact version of the approximations in [1]. See Appendix C for details on the connections between our full computation and the approximate one.

3.4 Computing Policies

The goal here is to compute a policy $\pi(b(\boldsymbol{S})) : \Delta(\boldsymbol{S}) \rightarrow \mathcal{A}$ that maps distributions over $\boldsymbol{S}$ into actions, where $b(\boldsymbol{S})$ is the current belief state as given by Equation (10). This policy is a function from functions (distributions) over a continuous space into the mixed continuous-discrete action space. There are two components to this mapping. First, there is the propositional action as defined by the original POMDP, and second there is the affective action defined by affect control theory. Here we consider only the affective action with the understanding that this can be easily expanded to include any actions that effect the state directly at a later stage.

An interesting property of POMDP policies is that they may use "information gathering" actions. In the context of affect control theory, if the *agent* is uncertain about the identity of the *client*, then it can take actions that temporarily increase deflection for example in order to discover something about the *client* identity. The information gained by such an exploratory action may be very worthwhile for the *agent*, as it may help the *agent* better understand the identity of the *client*, and therefore better decrease deflection in the long term.

Policies for POMDPs in general can be computed using a number of methods, but recent progress in using Monte-Carlo (sampling) based methods has shown that very large POMDPs can be solved tractably, and that this works equally well for continuous state and observation spaces [51,52]. POMCP [52] is a Monte-Carlo based method for computing policies in POMDPs with discrete action spaces. POMCP can be generalised to a mixed continuous-discrete action space for *BayesAct* by leveraging the fact the we can *predict* what an agent would "normally" do in any state according to the underlying affect control theory: it is the action that minimises the deflection. Given the belief state $b(\boldsymbol{s})$, we have a probability distribution over the action space giving the probability of each action (see Equation (31)). This normative prediction constrains the space of actions over which the *agent* must plan, and drastically reduces the branching factor of the search space. A sample drawn from the action distribution is used in the POMCP method, and a "rollout" proceeds by drawing a subsequent sample from the distribution over *client* actions, and then repeating the sampling over *agent* actions. This is continued to a maximum depth, and the reward gathered is computed as the value of the path taken. The propositional actions that update $\mathbf{x}$ are handled exhaustively as usual in POMCP.

Denote the "normal" or expected action distribution as $\pi^\dagger(\boldsymbol{s})$:

$$\pi^\dagger(\boldsymbol{s}') = \int_{\mathbf{f}'_a,\mathbf{f}'_c}\int_{\boldsymbol{s}} Pr(\mathbf{f}'|\mathbf{f}, \boldsymbol{\tau}, \mathbf{x}, \varphi)b(\boldsymbol{s}) = \int_{\mathbf{f}'_a,\mathbf{f}'_c}\int_{\boldsymbol{s}} e^{-(\mathbf{f}'-\boldsymbol{\mu}_n^\dagger)^T(\boldsymbol{\Sigma}_n^\dagger)^{-1}(\mathbf{f}'-\boldsymbol{\mu}_n^\dagger)} b(\boldsymbol{s}) \tag{31}$$

where

$$\boldsymbol{\mu_n^\dagger} = \boldsymbol{\Sigma}_n^\dagger \mathscr{K}^T(\boldsymbol{\tau},\mathbf{x})\boldsymbol{\Sigma}^{-1}\mathscr{C}(\boldsymbol{\tau},\mathbf{x}) + \boldsymbol{\Sigma}_n^\dagger(\boldsymbol{\Sigma}_f^\dagger(\mathbf{x}))^{-1}\mathbf{f}, \tag{32}$$

$$\boldsymbol{\Sigma}_n^\dagger = (\mathscr{K}^T(\boldsymbol{\tau},\mathbf{x})\boldsymbol{\Sigma}^{-1}\mathscr{K}(\boldsymbol{\tau},\mathbf{x}) + (\boldsymbol{\Sigma}_f^\dagger(\mathbf{x}))^{-1})^{-1}, \tag{33}$$

and $\boldsymbol{\Sigma}_f^\dagger$ is the same as $\boldsymbol{\Sigma}_f$ given by Equation (23) with $\beta_b^2(\mathbf{x})$ set to infinity (instead of zero) so the behaviour sentiments are unconstrained. Equation (31) computes the expected distribution over $\mathbf{f}'$ given $b(\boldsymbol{s})$ and then marginalises (sums) out the identity components[3] to get the distribution over $\mathbf{f}_b$. A sample drawn from this distribution could then be used as an action in the POMCP method. A POMCP "rollout" would then proceed by drawing a subsequent sample from the distribution over *client* actions, and then repeating the sampling from Equation (31) over *agent* actions. This is continued to some maximum depth, at which point the reward gathered is computed as the value of the path taken. The propositional actions that update the state $\mathbf{x}$ are handled as usual in POMCP by looping over them.

The integration in Equation (31) may be done analytically if $b(\boldsymbol{s})$ is Gaussian, but for the general case this may be challenging and not have a closed-form solution. In such cases, we can make a further approximation that $b(\boldsymbol{s}) = \delta(\boldsymbol{s}^* - \boldsymbol{s})$ where $\boldsymbol{s}^* = \{\mathbf{f}^*, \boldsymbol{\tau}^*, \mathbf{x}^*\} = \mathbb{E}_{b(\boldsymbol{s})}[\boldsymbol{s}] = \int_{\boldsymbol{s}} \boldsymbol{s} b(\boldsymbol{s})$ is the expected state (or one could use $\boldsymbol{s}^* = \arg\max_{\boldsymbol{s}} b(\boldsymbol{s})$ as the most likely state). We will denote the resulting action distribution as $\pi^{\dagger *}(\boldsymbol{s}')$.

In this paper, we don't use the full POMCP solution, instead only taking a "greedy" action that looks one step into the future by drawing samples from the "normal" action distribution in Equation (31) using these to compute the expected next reward, and selecting the (sampled) action $\mathbf{b_a}^{\dagger *}$ that maximizes this:

$$\mathbf{b_a}^{\dagger *} = \arg\max_{\mathbf{b_a}} \int_{\boldsymbol{s}'} \left[Pr(\mathbf{x}'|\mathbf{x}^*, \mathbf{f}', \boldsymbol{\tau}', \mathbf{b_a}) Pr(\boldsymbol{\tau}'|\boldsymbol{\tau}^*, \mathbf{f}', \mathbf{x}^*) Pr(\mathbf{f}'|\mathbf{f}^*, \boldsymbol{\tau}^*, \mathbf{x}^*, \mathbf{b_a}) R(\mathbf{f}', \boldsymbol{\tau}', \mathbf{x}') d\boldsymbol{s}', \quad \mathbf{b_a} \sim \pi^{\dagger *}(\boldsymbol{s}') \right] \tag{34}$$

In practice we make two further simplifications: we avoid the integration over $\mathbf{f}_a$ and $\mathbf{f}_c$ in Equation (31) by drawing samples from the distribution over $\mathbf{f}'$ and selecting the $\mathbf{f}'_b$ components, and we compute the integration in Equation (34) by sampling from the integrand and averaging.

3.5 Sampling

We return now to Equation (8) and consider how we can compute the belief distribution at each point in time. The nonlinearities in the transition dynamics that arise from the dynamics of fundamental sentiments (Equation (28)) prevent the use of a simple Kalman filter. Even the extended Kalman filter (EKF) may run into difficulties due to these nonlinearities. Instead, we will find it more convenient and general to represent $b(\boldsymbol{s})$ using a set of N samples [55]. This will allow us to represent more complex belief distributions, including, but not limited to multi-modal distributions over identities. This can be very useful in cases where the *agent* believes the *client* to be one of a small number of identities with equal probability. In such a case, the *agent* can maintain multiple hypotheses, and slowly shift its belief towards the one that agrees most with the evidence accrued during an interaction. We will write the belief state as [55]:

$$b(\boldsymbol{s}) \propto \sum_{i=1}^{N} w_i \delta(\boldsymbol{s} - \boldsymbol{s}_i), \tag{35}$$

where $\boldsymbol{s}_i = \{\mathbf{f}_i, \boldsymbol{\tau}_i, \mathrm{x}_i\}$ and w_i is the weight of the i^{th} sample and

$$\delta(\boldsymbol{s} - \boldsymbol{s}_i) = \begin{cases} \infty & \text{if } \boldsymbol{s} = \boldsymbol{s}_i \\ 0 & \text{otherwise} \end{cases} \tag{36}$$

Then, we implement Equation (8) using a sequential Monte Carlo method sampling technique, also known as a *particle filter* or *bootstrap filter* [55, 56]. We start at time $t = 0$ with a set of samples and weights

[3]If the agent is able to "set" its own identity, then the integration would be only over $\mathbf{f}'_c$, the client identity.

$\{\boldsymbol{s}_i, w_i\}_{i=1...N}$, which together define a belief state $b(\boldsymbol{s}_0)$ according to Equation (35). The precise method of getting the first set of samples is application dependent, but will normally be to draw the samples from a Gaussian distribution over the identities of the *agent* and *client*, and set all weights to 1.0. The *agent* then proceeds as follows:

1. Consult the policy to retrieve a new action $\mathbf{b_a} \leftarrow \pi(b(\boldsymbol{s}_t))$. If using the approximation in Equation (34), then we first compute the expected value of the state $\boldsymbol{s}_t^* = \sum_{i=1}^N w_i \boldsymbol{s}_i$.
2. Take action $\mathbf{b_a}$ and receive observation $\boldsymbol{\omega}$.
3. Sample (with replacement) unweighted samples from $b(\boldsymbol{s})$ from the distribution defined by the current weights.
4. For each unweighted sample, $\boldsymbol{s}_i$, draw a new sample, $\boldsymbol{s}_i'$ from the posterior distribution $Pr(\cdot|\boldsymbol{s}_i, \mathbf{b_a})$:
   (a) draw a sample $\mathbf{f}'$ from Equation (28) (this is a draw from a multivariate normal, and will likely be a bottleneck for the sampling method),
   (b) draw a sample $\boldsymbol{\tau}'$ from Equation (4) (this is deterministic so is an easy sample to draw),
   (c) draw a sample x' from Equation (7) (this is deterministic if we don't need to model b_c explicitly).
5. Compute new weights for each sample using the observation functions $w_i = Pr(\boldsymbol{\omega}|\boldsymbol{s}_i')$
6. If all weights are 0.0, then resample from the initial distribution.
7. The new state is $b(\boldsymbol{s}')$, goto step 1 with $\boldsymbol{s} \leftarrow \boldsymbol{s}'$.

An example of the sampling step 4 is shown above to be from a proposal that is exactly $Pr(\mathbf{f}'|\mathbf{f}, \boldsymbol{\tau}, \mathbf{x}, \mathbf{b_a}, \varphi)$, but this could be from some other distribution close to this.

We can compute expected values of quantities of interest, such as the deflection, by summing over the weighted set of samples (the Monte-Carlo version of Equation (11)):

$$d(\mathbf{f}, \boldsymbol{\tau}) = \sum_{i=1}^{N} w_i (\mathbf{f}_i - \boldsymbol{\tau}_i)^2 \tag{37}$$

One might be tempted to draw samples from $\mathbf{f}'$ (Equation (21)) and $\boldsymbol{\tau}'$ (Equation (4)) independently, and then re-weight based on both the observations (Equation (9)) and the deflection (Equation (2)). Although drawing samples in this case will be easier, as Equation (28) is avoided, we have no easy way to draw samples over $\mathbf{f}_b'$, and must resort to drawing these according to some broad prior distribution over the space of fundamental behaviour sentiments. As the majority of such samples will have high deflection (low potential according to Equation (2)), many more samples will be needed to locate the true modes of the posterior. Therefore, one must resort to drawing from Equation (28) directly, and cannot factor this in any reasonable way since the different components of $\mathbf{f}'$ are connected by the undirected links to $\boldsymbol{\tau}'$.

The dimensionality of the state space is large enough to warrant some concern about how many samples are needed. However, we have found that we can get very accurate simulations with a reasonable number of particles. This is likely so because of the large amount of determinism in the transition dynamics. It is less clear how this will scale once we are using this with humans, whose behaviours may be less predictable (more dependent on non-modelled factors).

Nevertheless, we have found that, for situations in which the *client* identity is not known, but is being inferred by the *agent*, it is necessary to add some "roughening" to the distribution over these unknown identities [56]. This is because the initial set of samples only sparsely covers the identity space (for an unknown identity), and so is very unlikely to come close to the true identity. Coupled with the underlying assumption that the identities are fixed or very slowly changing, this results in the particle filter getting "stuck" (and collapsed) at whatever initial sample was closest to the true identity (which may still be far off in the EPA space, especially when using fewer particles). Adding some zero-mean white noise (in $[-\sigma_r, \sigma_r]$) helps solve this degeneracy. We add this noise to any unknown identity (*agent* or *client*) after the unweighted samples are drawn in step 3 above. As suggested by [56], we use $\sigma_r = K \times N^{-1/d}$, where K is a constant, N is the number of samples and d is the dimension of the search space (in this case 3 for the unknown identity. We use $K = 1$ in our experiments, and note that we are not using white noise (not Gaussian noise), but that this does not make a significant difference.

This so-called "roughening" procedure is well known in the sequential Monte-Carlo literature, and in particular has been used for Bayesian parameter estimation [55] (Chapter 10). Our situation is quite similar, as the *client* identities can be seen as model parameters that are fixed, but unknown. Finally, it may also be possible to change the amount of roughening noise that is added, slowly reducing it according to some schedule as the *client* identity is learned.

It is also possible to mix exact inference over the application state, **X**, with sampling over the continuous affective space, leading to a Rao-Blackwellised particle filter [57].

3.6 Python Implementation

We have implemented *BayesAct* in Python as a class `Agent` that contains all the necessary methods. Applications can use *BayesAct* by subclassing `Agent` and providing three key application-dependent methods:

- `sampleXvar` is used to draw a sample from **X**
- `reward` produces the reward in the current state of **X**
- `initXvar` is used to initialise **X** at the start of a simulation or run

Sub-classes can also implement methods for input and output mappings. For example, an input mapping function could take sentences in English and map them to EPA values based on some affective dictionary. Applications can also learn these mappings by assuming the human user will be behaving according to the affect control principle: whatever the user says can be mapped to the prediction of the theory (or close to it).

4 Experiments and Results

Our goal in this section is to demonstrate, in simulation, that *BayesAct* can discover the affective identities of persons it interacts with, and that *BayesAct* can augment practical applications with affective dynamics. To establish these claims, we do the following.

First, we verify both analytically and empirically that *BayesAct* can reproduce exactly the affective dynamics predicted by the *Interact* software [1]. The analytical derivation is done by reducing Equation (28) to the equations in [1] as shown in Appendix C. The empirical demonstration is done by running *BayesAct* alongside *Interact* and showing that the identical sentiments and actions are generated. We have found a very close match across a range of different *agent* and *client* identities. These analytical and empirical demonstrations show that *BayesAct* can be used as a model of human affective dynamics to the extent that it has been shown empirically that *Interact* is a close model of human affective dynamics. Once we demonstrate this, then we can use the *BayesAct* software in the same way as the *Interact* software to make predictions about how an agent with a *with a fixed identity* and a *fixed and known identity for the person it is interacting with* will behave. Second, we show how, if we loosen the constraints on the *client* identity being fixed, *BayesAct* can "discover" or learn this identity during an interaction with an *Interact* client. Third, we show how, if both *agent* and *client* do not know the identity of their interactant, they can both learn this identity simultaneously. Fourth, we show that a *BayesActagent* can adapt to a changing *client* identity. What this means is that an affective *agent* has the ability to learn the affective identity of a *client* that it interacts with. We demonstrate this under varying levels of environment noise. Finally, we postulate that, since the *agent* can learn the affective identity of its *client*, it can better serve the *client* in an appropriate and effective manner. We give a preliminary demonstration of this with results from a basic experiments with humans in Section 4.2.

4.1 Simulations

In this section, we investigate two types of simulation. The first concerns agents with nearly fixed (low variance) personal identities that try to learn the identity of another agent. The second shows what happens if one of the agents is changing identity dynamically. To enable comparisons with *Interact*, we use action selection according to our generalised *affect control principle* only, using an average of 100 samples from Equation (31). Our simulations therefore do not directly address how policy computation will affect an application. However, we can show that *BayesAct* can replicate *Interact* as far as deflection minimisation goes, and can find low-deflection solutions for many examples, without requiring identities to be known and fixed. Videos showing dynamics of the simulations can be seen at the following webpage: `https://cs.uwaterloo.ca/~jhoey/research/bayesact/`.

4.1.1 Static identities

We have three conditions. In the first two, the *agent* does not know the identity of the *client*, and the *client* either knows or doesn't know the identity of the *agent* (denoted "agent id known" and "agent id hidden", resp.). In the third case, *agent* and *client* know each other's identities. In all three cases, we run 20 trials for each condition, and in each trial a new identity is chosen for each of *agent* and *client*. These two identities are independently sampled from the distribution of identities in the *Interact* database and are the personal identities for each *agent* and *client*. Then, *agent* and *client BayesAct* models are initialised with $\mathbf{F}_a$ set to this personal identity, $\mathbf{F}_c$ (identity of the other) set to either the true identity (if known) or else to the mean of the identities in the database, $[0.4, 0.4, 0.5]$. $\mathbf{F}_b$ set to zeros, but this is not important as it plays no role in the first update. The simulation proceeds according to the procedure in Section 3.5 for 50 steps. Agents take turns acting, and actions are conveyed to the other agent with the addition of some zero-mean normally distributed "environment" noise, with standard deviation σ_e. Agents use Gaussian observation models with uniform covariances with diagonal variances $\gamma = \max(0.5, \sigma_e)$. We perform 10 simulations per trial with $\beta_c = 0.001$ for both *agent* and *client*. If the *client* knows the *agent* identity, it uses no roughening noise ($\sigma_r = 0.0$), otherwise all *agents* use $\sigma_r = N^{-1/3}$ where N is the number of samples. We use *id-deflection* to denote the sum of squared differences between one *agent*'s estimate of the other *agent*'s identity, and that other *agent*'s estimate of its' own identity. Table 3 in Appendix A shows the full results. Here we summarize some of the key findings.

Figure 2 shows a plot of the mean (over 20 trials) of the average (over 10 experiments) final (at the last step) *id-deflection* and total deflection as a function of the environment noise, σ_e, and sample numbers, N, for the *agent id hidden* case. Also shown are the average of the maximum total deflections for all experiments in a trial. The plots in the left and right columns are for *agent* and *client*, respectively, but these are symmetric (are representing the same thing). We see that only about 50 samples are needed to get a solution that is robust to environment noise up to $\sigma_e = 2.0$. This corresponds to enough noise to make a behaviour of "apprehend" be mis-communicated as "confide in"[4]. Further examples of behaviours for different levels of deflection are shown in Table 2 (Appendix A). Surprisingly, deflection is not strongly affected by environment noise. One way to explain this is that, since the agent has a correct model of the environment noise ($\gamma = \sigma_e$), it is able to effectively average the noisy measurements and still come up with a reasonably low deflection solution. The deterministic program *Interact* would have more trouble in these situations, as it must "believe" exactly what it gets (it has no model of environment

[4]However, we are comparing the expected values of identities which may be different than any mode.

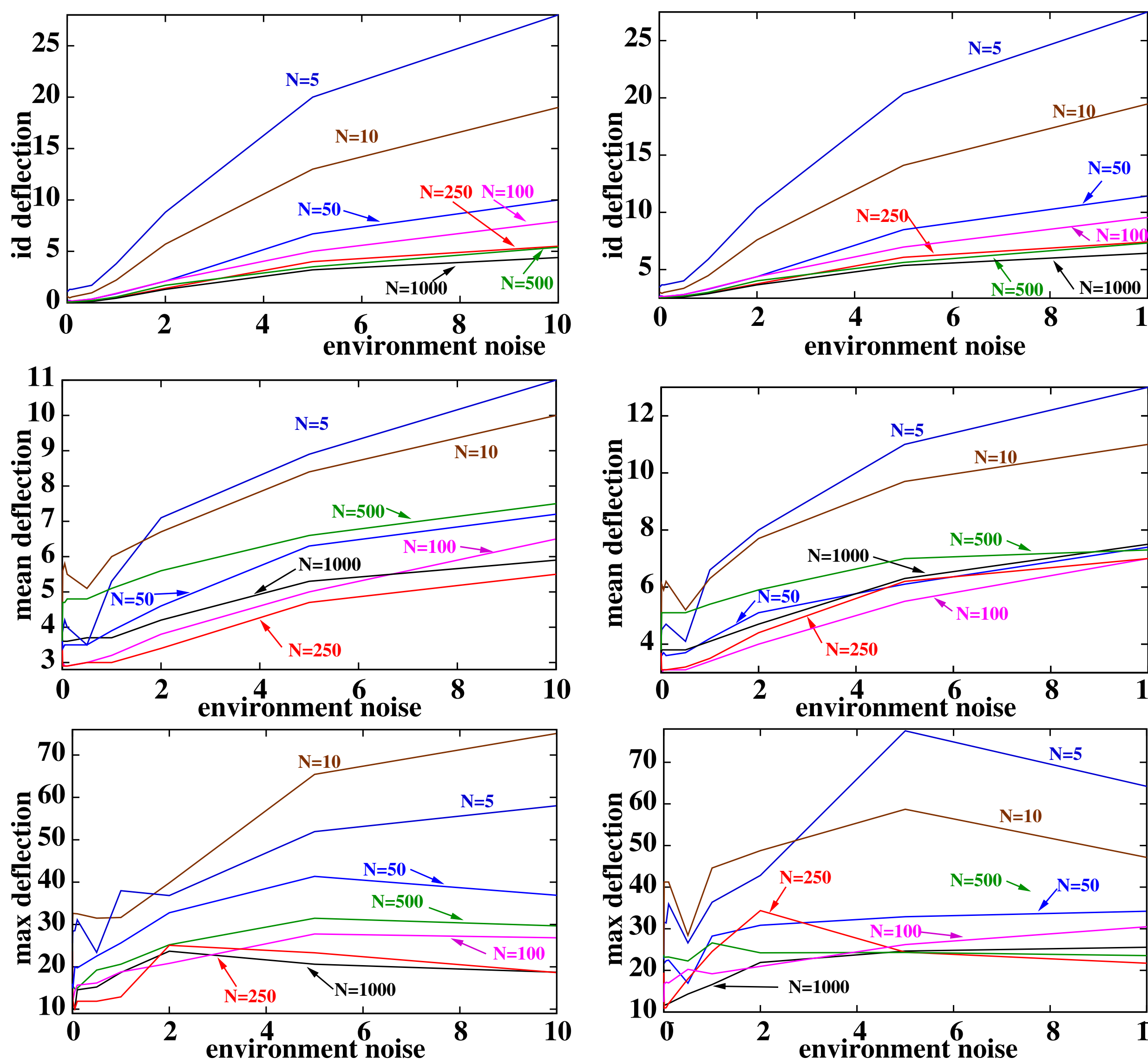


Figure 2: Deflections of identities from simulations with different numbers of samples (N), and environment noise, σ_e. Roughening noise: $\sigma_r = N^{-1/3}$, model environment noise: $\gamma = \max(0.5, \sigma_e)$. Left column: *agent*; right column: *client*. Top row: id-deflection; middle Row: mean deflection; bottom row: max deflection.

noise). Deflection and max deflection also don't seem to decrease monotonically with increasing sample size. However, there are large variances on the deflection results, which we do not show in these figures for clarity reasons. Table 3 shows the full results with standard deviations.

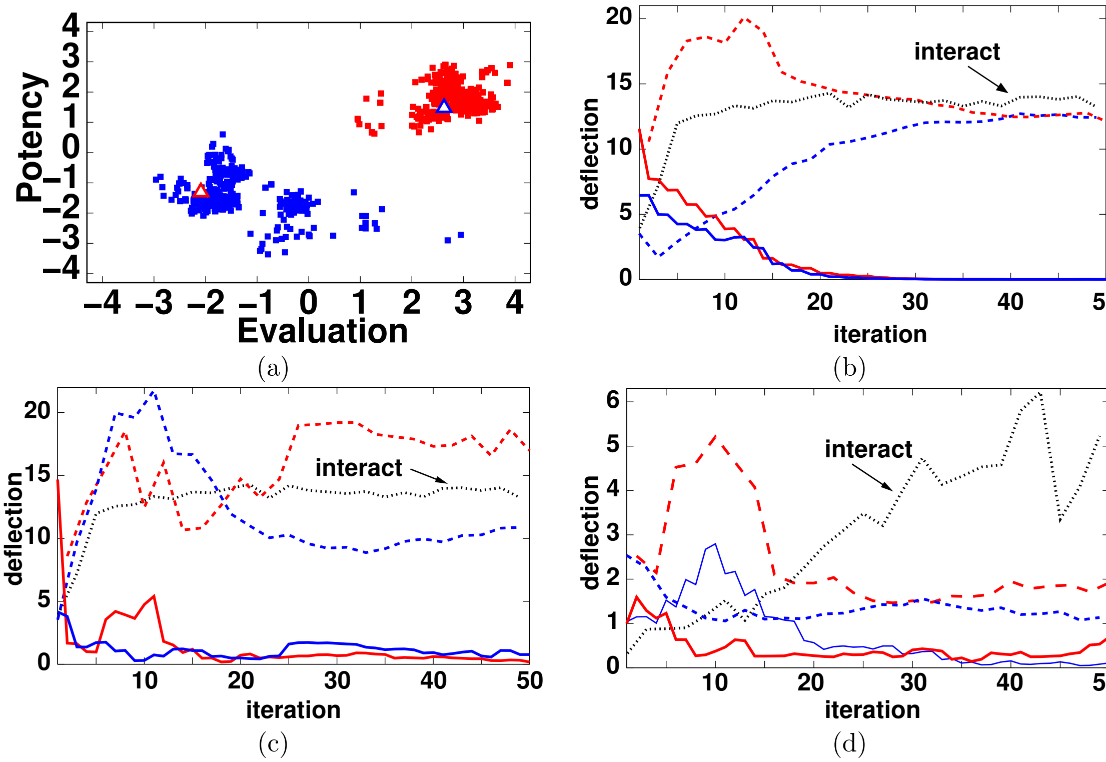


Figure 3: (a) samples (squares) and true identities (triangles) after 7 iterations for one trial. (b-d) Closer look at three 200 sample experiments showing *id-deflection* (solid lines), total deflections (dashed lines) and deflection from *Interact* in dotted black (red=*agent*, blue=*client*). (b) *agent*: $\mathbf{F}_a = [2.7, 1.5, 0.9]$, *client*: $\mathbf{F}_a = [-2.1, -1.3, -0.2]$, $\sigma_e = 0$; (c) as (a) but with $\sigma_e = 1.0$; (d) *agent*: $\mathbf{F}_a = [1.5, 1.5, -0.2]$, *client*: $\mathbf{F}_c = [1.5, 0.3, 0.8]$, $\sigma_e = 1.0$.

Figure 3(a) shows a sample set after 7 iterations of one experiment, clearly showing the multimodal distributions centered around the true identities (triangles) of each interactant[5]. These sample sets normally converge to near the true identities after about 15 iterations or less. Figures 3(b)-(d) look more closely at three of the trials done with $N = 200$ samples and hidden ids for both *client* and *agent*. The red and blue lines show the *agent*- and *client*- *id-deflection* (solid) and *agent* and *client* deflections (dashed), respectively, while the black line shows the deflections using *Interact* (which has the correct and fixed identities for both agents at all times). *BayesAct* allows identities to change, and starts with almost no information about the identity of the other interactant (for both *agent* and *client*). We can see that our model gets at least as low a deflection as *Interact*. In Figure 3(b), the *agent* had $\mathbf{F}_a = [2.7, 1.5, 0.9]$, and the *client* had $\mathbf{F}_a =$ [-2.1, -1.3, -0.2], and $\sigma_e = 0$ (noise-free communication). These two identities do not align very well[6], and result in high deflection when identities are known and fixed in *Interact* (black line). *BayesAct* rapidly estimates the correct identities, and tracks the deflection of *Interact*. Figure 3(c) is the same, but with $\sigma_e = 1.0$. We see, as in Table 3 that *BayesAct* is robust to this level of noise. Figure 3(d) shows a simulation between a "tutor" ($\mathbf{F}_a =$ [1.5, 1.5, -0.2]) and a "student" ($\mathbf{F}_c = [1.5, 0.3, 0.8]$) with

[5] see also videos at `http://www.cs.uwaterloo.ca/~jhoey/research/bayesact/`.

[6] These identities are closest to "lady" and "shoplifter" for *agent* and *client* respectively, but recall that identity labels come from mapping the computed EPA vectors to concepts in ACT repositories [34] and are not used by *BayesAct*.

$\sigma_e = 1.0$. Here we see that *Interact* predicts larger deflections can occur. *BayesAct* also gets a larger deflection, but manages to resolve it early on in the simulation. Identities are properly learned in this case as well.

Figure 4 shows further examples with varying noise levels. Figure 4(a,b) have the same identities as as Figure 3(b,c), but with environmental noise $\sigma_e = 0.5$ and $\sigma_e = 5.0$, respectively. We see that, consistent with Table 3, at $\sigma_e = 5.0$ it becomes much harder for *BayesAct* to estimate identities, however the overall deflection is not necessarily increased. There is a lot of variance in the results for deflection, however (not shown for clarity reasons). Figure 4(c,d) has the same identities as Figure 3(d), but with $\sigma_e = 5.0$. We see the same effect as in Figure 4(b): *BayesAct* is unable to find the true identity. Finally Figure 4(e,f) shows results for identities "hero" for *agent* (EPA: $[2.6, 2.3, 2.1]$) and "insider" for *client* (EPA: $[-0.13, 0.97, 0.2]$). Here we see a rapid convergence to the correct identities for noise-free and noisy communication ($\sigma_e = 0.0$ and $\sigma_e = 0.5$, resp.).

4.1.2 Dynamic (Changing) Identities

We now experiment with how *BayesAct* can respond to agents that change identities dynamically over the course of an interaction. We use the following setup: the *client* has two identities (chosen randomly for each trial) that it shifts between every 20 steps. It shifts from one to the other in a straight line in E-P-A space, at a speed of s_{id}. That is, it moves a distance of s_{id} along the vector from its current identity to the current target identity. It stops once it reaches the target (so the last step may be shorter than s_{id}. It waits at the target location for T steps and then starts back to the original identity. It continues doing this for 200 steps. Our goal here is to simulate an agent that is constantly switching between two identities, but is doing so at different speeds. Table 4 and Table 5 in Appendix A show the full results for these simulations.

We first show that *BayesAct* can respond to a single shift in identity after the first 20 steps (so after that, $T = \infty$). Figure 5(a - solid blue line) shows the results after 100 steps for $\sigma_e = 0.5$. We see that *BayesAct* is able to successfully recover: the *id-deflection* after 100 steps does not keep increasing with increasing s_{id} up to $s_{id} = 2.0$. Figure 5(a - dashed red line) shows the same for the continual identity shifts, after 200 steps using $T = 20$ throughout. Again, *BayesAct* is able to maintain a fairly good estimate of the *client* identity at the end, but the trend appears to be increasing indicating that additional speed may disrupt things further.

Figure 5(b) investigates this further, and shows the mean number of time steps per sequence of 200 steps in which the *id-deflection* of the *agent*'s estimate of the *client*'s identity is greater than a threshold, d_m, for $\sigma_e = 0.5$. The results show that *BayesAct* is able to maintain a low *id-deflection* throughout the sequence when confronted with speeds up to about 0.1. At this setting ($s_{id} = \sigma_e = 0.1$), only 4 frames (out of 200) have an *id-deflection* greater than 1.0.

Figure 6 shows a specific example where the *client* shifts between two identities, for $s_{id} = 0.25$ and $T = 40$. The *agent*'s estimates of $\mathbf{F}_e$ and $\mathbf{F}_p$ are seen to follow the *client*'s changes, alhough the *agent* lags behind by about 50 time steps.

4.2 Tutoring Application

To demonstrate the capability of *BayesAct* to control emotionally plausible behaviours of a computer program interacting with a human, we built a simple tutoring application in which the identities for *agent* and *client* are initially set to "tutor" ($\mathbf{F}_a = [1.5, 1.5, \text{-}0.2]$) and "student" ($\mathbf{F}_c = [1.5, 0.3, 0.8]$), respectively, with low dynamics variances of $\beta_a = \beta_c = 0.01$ and $\sigma_r = 0.0$ (see Section 3.3). Screenshots are shown in Figure 7. The application asks sample questions from the Graduate Record Exam (GRE) educational testing system, and the *client* clicks on a multiple-choice answer. The *agent* provides feedback as to whether the *client*'s answer is correct. The *client* then has the opportunity to "speak" by clicking on a labelled button (e.g., "that was too hard for me!"). The statement maps to a behaviour from

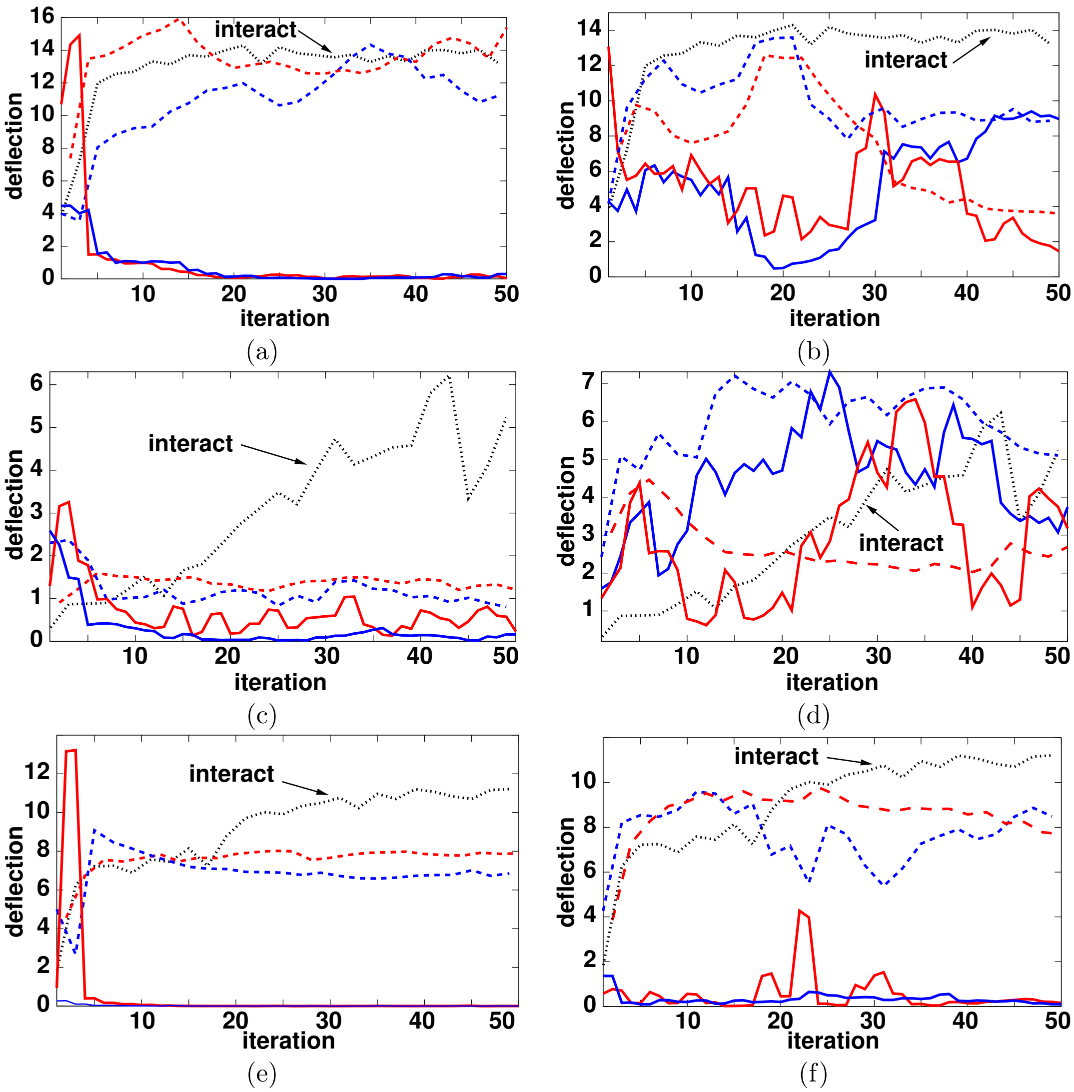
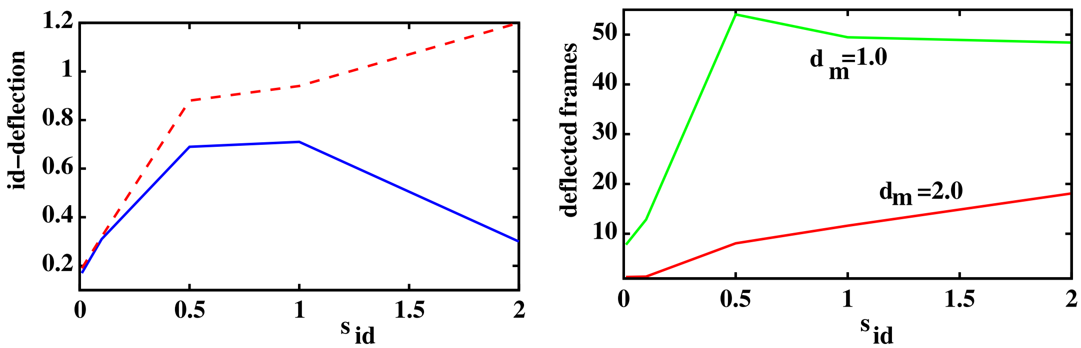
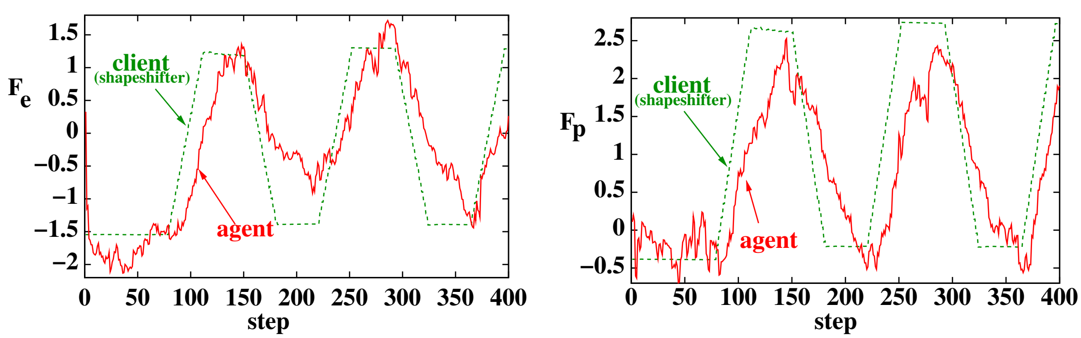


Figure 4: Simulations with different environment noise levels with 200 samples, showing *id-deflection* (solid lines), total deflections (dashed lines) and deflection from *Interact* in dotted black (red=*agent*, blue=*client*). (a) “lady” and “shoplifter”, $\gamma = \sigma_e = 0.5$, (b)“lady” and “shoplifter”, $\gamma = \sigma_e = 5.0$, (c) “tutor” and “student”, $\sigma_e = 0.5, \gamma = 0.5$, (d) “tutor” and “student”, $\sigma_e = 5.0, \gamma = 5.0$, (e) “hero” and “insider”, $\sigma_e = 0.0, \gamma = 0.1$, (f) “hero” and “insider”, $\sigma_e = 0.5, \gamma = 0.5$



(a) (b)

Figure 5: (a-solid blue line) Agent *id-deflection* after 100 steps for different identity shift speeds of *client*, s_id, for the single identity shift experiments, for $N = 100, T = 20$. (a-dashed red line) shows the same but after 200 steps for continual shifting with $N = 100, T = 20$. (b) Number of frames (out of 200) where *id-deflection* is greater than $d_m = 1.0, 2.0$ for continual shift identity experiments, for $N = 100, T = 20$. $\sigma_e = 0.5$ for all cases.



(a) (b)

Figure 6: Fundamental identity sentiments of *client* (green, dashed line) and *agent*'s estimate of client's identity (red, solid line) for an example where the *client* shifts identities at a speed of $s_{id} = 0.25$ and remains at each of two target identities for 40 steps. *agent* id is $[0.32, 0.42, 0.65]$ and *client* ids are $[-1.54, -0.38, 0.13]$ and $[1.31, 2.75, -0.09]$. (a) $\mathbf{F}_e$; (b) $\mathbf{F}_p$.

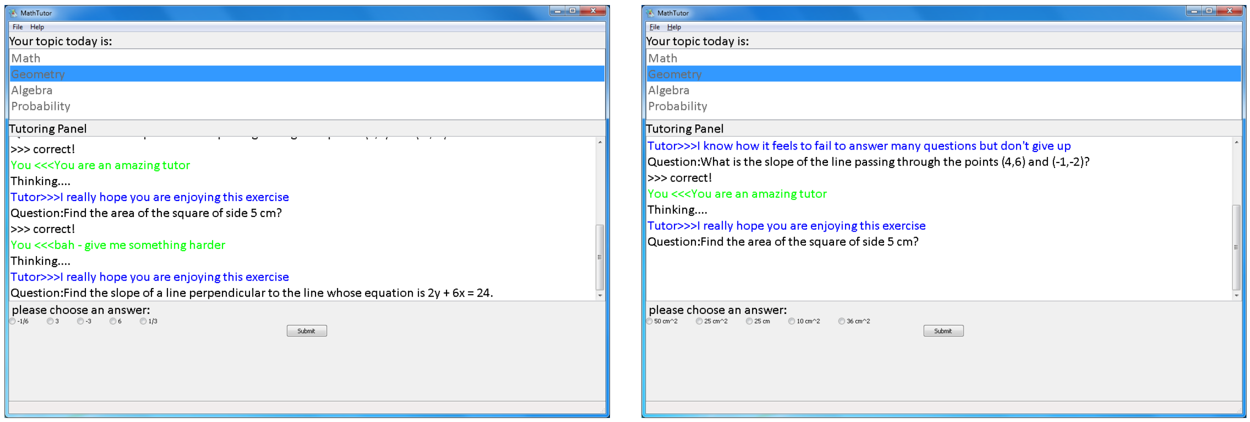


Figure 7: Tutoring interface screenshots

the ACT database (e.g., "whine"). These mappings were determined in an empirical survey described below. The behaviour label in turn maps to the value for $\mathbf{F}_b$ found in the ACT database (in this case [-1.4, -0.8, -0.5]). *BayesAct* then computes an appropriate agent action, i.e. a vector in EPA space, which maps to a behaviour label (e.g., "apologize"), which again maps to a statement (e.g., "Sorry, I may have been too demanding on you.").

The tutor has three discrete elements of state $\mathbf{X} = \{X_d, X_s, X_t\}$ where X_d is the difficulty level, X_s is the skill level of the student and X_t is the turn. X_d and X_s have 3 integer "levels" where lower values indicate easier difficulty/skill. The tutor's model of the student's progress is $P(X'_s = x_s|x_s, \mathbf{f}, \boldsymbol{\tau}') = 0.9$ with the remaining probability mass distributed evenly over skill levels that differ by 1 from x_s. The dynamics for all values where $X'_s \leq x_s$ are then multiplied by $(\mathbf{f}' - \boldsymbol{\tau}')^2/2$ and renormalised. Thus, as deflection grows, the student is less likely to increase in skill level and more likely to decrease. Thus, skill level changes inversely proportionally to deflection. The tutor gets observations of whether the student succeeded/failed ($\Omega_x = 1/0$), and has an observation function $P(\Omega_x|X_d, X_s)$ that favours success if X_d (the difficulty level) matches X_s (the skill level).The reward is the sum of the negative deflection as in Equation (5) and $R_x(\mathbf{x}) = -(\mathbf{x}-2)^2$. It uses the approximate policy given (Section 3.4) by Equation (34). for its affective response, and a simple heuristic policy for its propositional response where it gives an exercise at the same difficulty level as the mean (rounded) skill level of the student 90% of the time, and an exercise one difficulty level higher 10% of the time. Further optimisations of this policy as described in Section 3.4 would take into account how the student would learn in the longer term.

This simple model suffices for our pilot study, but would need to be expanded and made to better reflect actual student development in future versions. In particular, one could expand the state space $\mathbf{X}$ to include more features related to the application and student skills than only the simple 3-valued difficulty and skill levels we have used here. This would require making a more complex model of the transitions in $\mathbf{X}$ (e.g. identifying goals and problem space dimensions [21, 31]), a more complex model of the observations of $\mathbf{X}$ (e.g. from sensors), and a more complex model of the dependence of the sentiments on the state. This last part is the only part that would require more analysis, as the fundamental sentiments of system states would need to be elicited from groups of users, for which clear methodologies exist, as in [17]. It may also be possible to encode more general sentiment mappings based on key words or recognised behaviours [58]. Finally, one could imagining learning the transition functions over $\mathbf{X}$ as the tutor gathered data while interacting with the student.

We thus require a specification of the POMDP, and two mappings, one from the combination of *client* statement button labels and difficulty levels to ACT behaviours (of *client*), and the other from ACT behaviours (of *agent*) to difficulty level changes and statements to the student. We conducted an empirical online survey to establish these mappings.

survey question	BACT	rand.	T	p
Communication was similar to a human.	2.10	1.70	0.87	n.s.
The tutor acted as if it understood my feelings	2.80	2.00	1.92	$< .05$
I felt emotionally connected with MT.	2.50	2.00	1.00	n.s.
I enjoyed interacting with MT.	3.30	3.10	0.44	n.s.
MT acted like it knew what kind of person I am.	2.90	2.11	1.55	$< .10$
MT gave awkward/inappropriate responses (RC)	3.44	4.50	-2.70	$< .01$
I found MT to be flexible to interact with.	3.00	1.90	2.18	$< .05$
Using MT would improve my skills.	3.40	2.00	2.49	$< .05$
The dialogue was simple and natural.	3.50	2.00	3.14	$< .01$
Overall, I am satisfied with this system.	2.70	1.70	2.34	$< .05$

Table 1: User study results. T has df=18 and p is one-tailed. Scales ranged from 1 (=not true) to 5 (=true). (bact=*BayesAct*,MT=MathTutor, RC=reverse-coded)

Participants in the survey were N = 37 (22 female) students (avg. age: 30.6 years). We presented them with four blocks of statements and behaviour labels, two blocks referring to *agent* and *client* behaviours conditional on a correct/incorrect answer of the *client*. We also asked participants to rate the affective meaning of each statement directly using the semantic differential [34]. In total, the survey contained 31 possible *agent* statements and 26 possible *client* statements plus an equal number of possibly corresponding behaviour labels. Participants were supposed to match each statement to one of the available behaviour labels. For 14 *agent* statements and for 13 *client* statements, a clear majority of participants agreed on one specific mapping. For 13 *agent* statements and 5 *client* statements, mappings were split between two dominant options. In these cases, we compared the direct EPA ratings of the statements with EPA ratings of the two behaviour labels in question to settle the ambiguity. We discarded 4 *agent* statements and 8 *client* statements, because participants' response patterns indicated a lack of consensus and/or unsolvable ambiguities in the mappings. As a result, we thus had a list of 27 agent statements and 18 client statements with corresponding mappings to behaviour labels from the ACT database. We implemented these behaviours as the possible actions in the BayesAct tutoring system.

We conducted a pilot experiment with 20 participants (7 female) who were mostly undergraduate students of engineering or related disciplines (avg. age: 25.8). We compared the experiences of 10 users interacting with the *BayesAct* tutor with those of 10 users interacting with a control tutor whose actions were selected randomly from the same set as the *BayesAct* tutor[7]. Participants completed a short survey after using the system for an average of 20 minutes. Results are displayed in Table 1. Users seemed to experience the flow of communication with the *BayesAct* tutor as more simple, flexible, and natural than with the random control tutor. The mean deflection for *BayesAct* was 2.9 ± 2.1 while for random it was 4.5 ± 2.2. We have to treat these results from a small sample with caution, but this pilot study identified many areas for improvement, and the results in Table 1 are encouraging.

Most of the user comments focused on the nature of the GRE tasks (e.g., explain the steps toward a solution, not only just state the solution), not on the emotional aspects of the communication. However, on directly related questions, participants indicated that while they did not feel an emotional connection with the tutor nor thought the communication was similar to communication with a human, they perceived the flow of communication as simple, flexible, and natural. Specific areas for improvement based on the experience with the pilot study include more variation in the tutor's statements (e.g., have different expressions for the same functional behaviours, not just one) and better integration of the emotional responses with explanations of solutions to the GRE tasks.

[7] *BayesAct* used 500 samples, $\beta_a = \beta_c = 0.01$, and took 4 seconds per interaction on an AMD phenom IIx4 955 3.20 GHz with 8GB RAM running Windows 7, while displaying the words "Thinking...". The random tutor simply ignored the computed response (but still did the computation so the time delay was the same) and then chose at random.

5 Conclusions

This paper has presented a probabilistic and decision theoretic formulation of affect control theory called *BayesAct*, and has shown its use for human interactive systems. The paper's main contributions are the theoretical model development, and a demonstration that a computational agent can use *BayesAct* to integrate reasoning about emotions with application decisions in a parsimonious and well-grounded way.

Overall, our model uses the underlying principle of deflection minimisation from Affect Control Theory to provide a general-purpose affective monitoring, analysis and intervention theory. The key contributions of this paper are

1. A formulation of Affect Control Theory as a probabilistic and decision theoretic model that generalises the original presentation in the social psychological literature in the following ways:
   (a) It makes exact predictions of emotions using the equations of Affect Control Theory, generalising the partial updates of ACT
   (b) It removes the assumption that identities are fixed through time and allows an *agent* to model a changing identity
   (c) It removes the assumption that sentiments (of identities and behaviours) are known exactly by modelling them as probability distributions
2. A set of simulations that demonstrate some of the capabilities of this generalised model of ACT.
3. A formulation of a general-purpose model for human-computer interaction that augments the model proposed by affect control theory in the following ways:
   (a) It adds a propositional state vector that models other events occurring as a result of the interaction, and models this state vector's progression as being dependent on the affective *deflection* of the interaction
   (b) It adds a reward function that an *agent* can optimise directly, allowing an *agent* to combine deflection minimisation with goal pursual in a parsimonious and theoretically well-grounded way.
   (c) It proposes a theory of control that uses the reward function and the dependence of the state vector on the affective component of the interaction to compute a policy of action that maximizes expected return for the *agent* in the long term.
4. A description of a simple intelligent tutoring system (ITS) that uses the proposed model to better align itself with the student.
5. A survey of 37 respondents who rated ITS actions in affective (EPA) space
6. Results of a pilot study with 20 participants who used the tutoring system for 20 minutes each. Our study demonstrates some of the basic elements of our model, and uncovers some of the key design considerations for future work in this area.

In future, the measurement of EPA behaviours and the translation of EPA actions requires further study. We plan to develop the planning aspects, and to parallelize the code (for which the sampling method is ideally suited). We plan to investigate usages of the model for collaborative agents in more complex domains, for competitive, manipulative or therapeutic agents, conversational agents, and for social simulations. We also plan to investigate methods for automatically learning the parameters of the prediction equations, and the identity labels. This would allow longer-term learning and adaptation for agents.

Acknowledgements

We would like to thank our study participants. We thank Pascal Poupart for helpful discussions. We acknowledge funding from the Natural Sciences and Engineering Council of Canada (J. Hoey), DFG research fellowship #SCHR1282/1-1 (T. Schröder). A. Alhothali is supported by a grant from King Abdulaziz Univ., Saudi Arabia.

References

[1] D. R. Heise, *Expressive Order: Confirming Sentiments in Social Actions.* Springer, 2007.

[2] J. Hoey, T. Schröder, and A. Alhothali, "Bayesian affect control theory," in *In Proc. of the 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction (ACII 2013)*, 2013, to appear.

[3] A. R. Damasio, *Descartes' error: Emotion, reason, and the human brain.* Putnam's sons, 1994.

[4] R. J. Dolan, "Emotion, cognition, and behavior," *Science*, vol. 298, no. 5596, pp. 1191–1194, 2002.

[5] R. Pekrun, "The impact of emotions on learning and achievement: Towards a theory of cognitive/motivational mediators," *Applied Psychology*, vol. 41, no. 4, pp. 359–376, 1992.

[6] P. Thagard, *Hot thought: Mechanisms and applications of emotional cognition.* MIT Press, 2006.

[7] R. W. Picard, *Affective Computing.* Cambridge, MA: MIT Press, 1997.

[8] R. A. Calvo and S. D'Mello, "Affect detection: An interdisciplinary review of models, methods, and their applications," *IEEE Transactions on Affective Computing*, pp. 18–37, 2010.

[9] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, "A survey of affect recognition methods: Audio, visual, and spontaneous expressions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 1, pp. 39–58, 2009.

[10] S. Hyniewska, R. aw Niewiadomski, M. Mancini, and C. Pelachaud, "Expression of affects in embodied conversational agents," in *Blueprint for Affective Computing: A sourcebook.* Oxford University Press, 2010, pp. 213–221.

[11] M. Schröder, F. Burkhardt, and S. Krstulović, "Synthesis of emotional speech," in *Blueprint for Affective Computing: A sourcebook.* Oxford University Press, 2010, pp. 222–231.

[12] J. Steephen, "HED: A computational model of affective adaptation and emotion dynamics," *IEEE Transactions on Affective Computing*, vol. PP, no. 99, pp. 1–1, 2013.

[13] M. Hoque, M. Courgeon, J.-C. Martin, B. Mutlu, and R. W. Picard, "Mach: My automated conversation coach," in *Proc. of Ubicomp*, Zurich, 2013.

[14] C. Becker-Asano and I. Wachsmuth, "Affective computing with primary and secondary emotions in a virtual human," *Autonomous Agents and Multi-Agent Systems*, vol. 20, no. 1, pp. 32–49, 2010.

[15] J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, Eds., *Embodied Conversational Agents.* MIT Press, 2000.

[16] S. Thrun, "Probabilistic algorithms in robotics," School of Computer Science, CMU, Tech. Rep. CMU-CS-00-126, 2000.

[17] D. B. Shank, "An affect control theory of technology," *Current Research in Social Psychology*, vol. 15, no. 10, pp. 1–13, 2010.

[18] L. Troyer, "An affect control theory as a foundation for the design of socially intelligent systems," in *Proc. AAAI Spring Symp. on Architechtures for Modeling Emotion*, 2004.

[19] B. Reeves and C. Nass, *The media equation.* Cambridge University Press, 1996.

[20] J. Robison, S. McQuiggan, and J. Lester, "Evaluating the consequences of affective feedback in intelligent tutoring systems," in *Proc. of IEEE Conference on Affective Computing and Intelligent Interaction*, 2009.

[21] C. Conati and H. Maclaren, "Empirically building and evaluating a probabilistic model of user affect." *UMUAI*, vol. 19, 2009.

[22] K. R. Scherer, "Appraisal theory," *Handbook of cognition and emotion*, pp. 637–663, 1999.

[23] K. R. Scherer, T. Banziger, and E. Roesch, *A Blueprint for Affective Computing.* Oxford University Press, 2010.

[24] L. F. Barrett, "Solving the emotion paradox: Categorization and the experience of emotion," *Personality and social psychology review*, vol. 10, no. 1, pp. 20–46, 2006.

[25] C. E. Osgood, W. H. May, and M. S. Miron, *Cross-Cultural Universals of Affective Meaning.* University of Illinois Press, 1975.

[26] J. A. Russell and A. Mehrabian, "Evidence for a three-factor theory of emotions," *Journal of research in Personality*, vol. 11, no. 3, pp. 273–294, 1977.

[27] K. B. Rogers, T. Schröder, and C. von Scheve, "Dissecting the sociality of emotion: A multi-level approach," *Emotion Review*, forthcoming.

[28] K. R. Scherer, E. S. Dan, and A. Flykt, "What determines a feeling's position in affective space: A case for appraisal," *Cognition and Emotion*, vol. 20, no. 1, pp. 92–113, 2006.

[29] A. Ortony, G. Clore, and A. Collins, *The Cognitive Structure of Emotions.* Cambridge University Press, 1988.

[30] K. R. Scherer, A. Schorr, and T. Johnstone, *Appraisal Processes in Emotion.* Oxford University Press, 2001.

[31] J. T. Folsom-Kovarik, G. Sukthankar, and S. Schatz, "Tractable POMDP representations for intelligent tutoring systems," *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 2, March 2013.

[32] J. Sabourin, B. Mott, and J. C. Lester, "Modeling learner affect with theoretically grounded dynamic Bayesian networks," in *Proc. Affective Computing and Intelligent Interaction*, 2011.

[33] H. Kurniawati, D. Hsu, and W. Lee, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces," in *Proc. Robotics: Science and Systems*, 2008.

[34] D. R. Heise, *Surveying Cultures: Discovering Shared Conceptions and Sentiments.* Wiley, 2010.

[35] C. P. Averett and D. R. Heise, "Modified social identities: Amalgamations, attributions, and emotions," *Journal of Mathematical Sociology*, vol. 13, pp. 103–132, 1987.

[36] H. W. Smith, "The dynamics of japanese and american interpersonal events: Behavioral settings versus personality traits." *Journal of Mathematical Sociology*, vol. 26, pp. 71–92, 2002.

[37] L. Smith-Lovin, "The affective control of events within settings," *Journal of Mathematical Sociology*, vol. 13, pp. 71–101, 1987.

[38] D. T. Robinson and L. Smith-Lovin, "Affect control theory," in *Contemporary Social Psychological Theories*, P. J. Burke, Ed. Stanford University Press, 2006, ch. 7, pp. 137–164.

[39] F. Heider, "Attitudes and cognitive organization," *Journal of Psychology: Interdisciplinary and Applied*, vol. 21, pp. 107–112, 1946.

[40] P. Thagard, *Coherence in Thought and Action.* MIT Press, 2000.

[41] T. Schröder and W. Scholl, "Affective dynamics of leadership: An experimental test of affect control theory," *Social Psychology Quarterly*, vol. 72, pp. 180–197, 2009.

[42] T. Schröder, J. Netzel, C. Schermuly, and W. Scholl, "Culture-constrained affective consistency of interpersonal behavior," *Social Psychology*, vol. 44, pp. 47–58, 2013.

[43] D. R. Heise, "Modeling interactions in small groups," *Social Psychology Quarterly*, vol. 76, pp. 52–72, 2013.

[44] K. J. Åström, "Optimal control of Markov decision processes with incomplete state estimation," *J. Math. Anal. App.*, vol. 10, 1965.

[45] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* New York, NY.: Wiley, 1994.

[46] W. S. Lovejoy, "A survey of algorithmic methods for partially observed Markov decision processes," *Annals of Operations Research*, vol. 28, pp. 47–66, 1991.

[47] C. Boutilier, T. Dean, and S. Hanks, "Decision theoretic planning: Structural assumptions and computational leverage." *Journal of Artifical Intelligence Research*, vol. 11, pp. 1–94, 1999.

[48] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, pp. 99–134, 1998.

[49] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* San Mateo, CA: Morgan Kaufmann, 1988.

[50] G. Shani, J. Pineau, and R. Kaplow, "A survey of point-based pomdp solvers," *Autonomous Agents and Multi-Agent Systems*, 2012.

[51] J. M. Porta, N. Vlassis, M. T. Spaan, and P. Poupart, "Point-based value iteration for continuous pomdps," *Journal of Machine Learning Research*, vol. 7, pp. 2329–2367, 2006.

[52] D. Silver and J. Veness, "Monte-Carlo planning in large POMDPs," in *Proc. NIPS*, December 2010.

[53] J. Hoey, C. Boutilier, P. Poupart, P. Olivier, A. Monk, and A. Mihailidis, "People, sensors, decisions: Customizable and adaptive technologies for assistance in healthcare," *ACM Trans. IIS*, vol. 2, no. 4, 2012.

[54] P. Poupart, "An introduction to fully and partially observable Markov decision processes," in *Decision Theory Models for Applications in Artificial Intelligence: Concepts and Solutions*, E. Sucar, E. Morales, and J. hoey, Eds. IGI Global, 2011, ch. 1, pp. 1–30.

[55] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo in Practice.* Springer-Verlag, 2001.

[56] N. J. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings F*, vol. 140, no. 2, April 1993.

[57] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-Blackwellised particle filtering for dynamic Bayesian networks," in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, 2000, pp. 176–183.

[58] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and trends in information retrieval*, vol. 2.1, no. 2, pp. 1–135, 2008.

A Tabulated Simulation Results

Table 2 shows examples of behaviours for different levels of deflection. Each row shows the two behaviour labels and their actual *id-deflection*. The first column shows the maximum *id-deflection* searched for.

Table 3 shows the mean (over 20 trials) of the average (over 10 experiments) final (at the last step) *id-deflection* for *agent* and *client* for varying numbers of samples and environment noises. Table 3 also shows the total deflection (Equation (1)) and the maximum deflection across all experiments and time steps, for each agent.

Table 4 shows the results for the experiments with *client* shifting its identity after 10 steps and then staying at the new identity until 100 steps. We see that *BayesAct* is able to successfully recover: the *id-deflection* and *deflection* are both the same at the end of the 100 steps, regardless of s_{id}.

Table 5 shows the mean number of time steps per sequence of 200 steps in which the *id-deflection* of the *agent*'s estimate of the *client*'s identity is greater than a threshold, d_m. The results are shown for a variety of environment noises, σ_e, and iddentity shifting speeds, s_{id}. The results show that *BayesAct* is able to maintain a low *id-deflection* throughout the sequence when confronted with speeds up to about 0.5 and environment noises less than $\sigma_e = 0.5$. At this setting ($s_{id} = \sigma_e = 0.5$), only 12 frames (out of 200) have an *id-deflection* greater than 1.0.

σ_r	id_1	id_2	$\lvert id_1 - id_2 \rvert$
≤ 0.02	-	-	-
0.05	quarrel with	quibble with	0.046
0.05	hoot at	strip	0.033
0.05	criticize	hush	0.028
0.1	make business proposal to	back	0.099
0.1	whip	bite	0.099
0.1	cajole	seduce	0.095
0.1	work	overwhelm	0.095
0.1	bash	distract	0.093
0.2	command	tackle	0.20
0.2	make eyes at	confess to	0.20
0.2	look at	draw near to	0.20
0.2	sue	spank	0.20
0.2	ask out	approach	0.20
0.5	eat with	suggest something to	0.50
0.5	shout at	knock out	0.50
0.5	medicate	caress	0.50
0.5	bully	hassle	0.50
0.5	restrain	contradict	0.50
1.0	borrow money from	peek at	1.0
1.0	join up with	show something to	1.0
1.0	criticize	rib	1.0
1.0	sue	fine	1.0
1.0	nuzzle	convict	1.0
2.0	massage	thank	2.0
2.0	dress	console	2.0
2.0	mind	accommodate	2.0
2.0	apprehend	confide in	2.0
2.0	harass	knock out	2.0
5.0	denounce	care for	5.0
5.0	collaborate with	kill	5.0
5.0	hug	scoff at	5.0
5.0	listen to	abandon	5.0
5.0	educate	pester	5.0
≥ 10.0	steal from	make love to	7.7
≥ 10.0	steal from	sexually arouse	7.6
≥ 10.0	steal from	help	7.5
≥ 10.0	steal from	save	7.4
≥ 10.0	steal from	give medical treatment to	7.4

Table 2: Most different behaviour pairs with a Euclidean distance less than σ_r, the environment noise. A $-$ indicates that there are no behaviours that are closer than the value of σ_r indicated.

		both known				client id known					client id hidden					
		deflection		max. deflection		id-deflection	deflection		max deflection		id-deflection		deflection		max deflection	
σ_e	N	agent	client	agent	client	agent	agent	client	agent	client	agent	client	agent	client	agent	client
0.0	5	3.2 ± 2.9	3.2 ± 2.9	13.94	14.00	0.89 ± 0.42	4.2 ± 2.1	4 ± 2.5	20.61	10.95	0.88 ± 0.46	1.1 ± 0.44	3.8 ± 2.5	4.2 ± 2.3	25.61	22.69
0.0	10	3.1 ± 3.1	3 ± 3.1	11.48	11.52	0.26 ± 0.14	4.3 ± 3	4 ± 3.1	25.68	15.46	0.43 ± 0.39	0.56 ± 0.71	3.6 ± 2.6	3.8 ± 2.3	29.21	23.49
0.0	50	3.3 ± 2.5	3.3 ± 2.5	13.90	13.93	0.2 ± 0.68	3.6 ± 3.2	3.5 ± 3.2	25.96	13.99	0.22 ± 0.42	0.34 ± 0.76	3.7 ± 2.2	3.9 ± 2.2	16.00	22.29
0.0	100	5.1 ± 6.6	5.1 ± 6.6	33.23	33.22	0.11 ± 0.26	4.2 ± 3.5	4 ± 3.5	23.47	15.18	0.12 ± 0.16	0.093 ± 0.13	2.9 ± 1.7	3 ± 1.9	13.35	12.41
0.0	250	4.5 ± 4.3	4.5 ± 4.3	21.28	21.29	0.12 ± 0.49	4.2 ± 3.7	4 ± 3.7	19.42	16.34	0.14 ± 0.3	0.04 ± 0.064	3.7 ± 3.3	3.7 ± 3.3	23.65	19.46
0.0	500	3.6 ± 2.6	3.6 ± 2.6	12.04	12.08	0.065 ± 0.19	3.7 ± 2.8	3.6 ± 2.9	14.57	14.53	0.093 ± 0.23	0.057 ± 0.14	3.6 ± 3.2	3.8 ± 3.3	15.73	23.72
0.0	1000	4.3 ± 3.1	4.3 ± 3.1	12.39	12.39	0.046 ± 0.12	4.5 ± 2.7	4.4 ± 2.7	12.06	10.46	0.038 ± 0.12	0.025 ± 0.053	3.7 ± 3.4	3.6 ± 3.3	17.30	21.07
0.01	5	4.2 ± 3.1	4.2 ± 3.1	12.33	12.43	0.74 ± 0.16	3.2 ± 1.9	2.6 ± 1.7	30.35	6.62	1.1 ± 0.7	1.1 ± 0.35	3.9 ± 2.3	4.5 ± 2.8	28.45	31.59
0.01	10	4 ± 3.7	4 ± 3.7	15.73	15.78	0.33 ± 0.2	4 ± 2.7	3.6 ± 2.7	19.69	11.22	0.55 ± 1	0.59 ± 0.64	5.6 ± 4.1	6.1 ± 4	32.63	41.23
0.01	50	5 ± 4.8	5 ± 4.8	25.08	25.05	0.12 ± 0.26	3.8 ± 2	3.5 ± 2	23.23	8.32	0.2 ± 0.36	0.19 ± 0.53	3.4 ± 1.8	3.6 ± 1.8	14.93	21.74
0.01	100	4 ± 3.6	4 ± 3.6	14.69	14.62	0.11 ± 0.19	4.4 ± 3.2	4.2 ± 3.3	26.03	14.73	0.086 ± 0.14	0.16 ± 0.32	3 ± 1.4	3 ± 1.4	12.22	16.81
0.01	250	3.4 ± 2.5	3.3 ± 2.5	13.60	13.64	0.14 ± 0.31	4.3 ± 3.1	4.1 ± 3.3	16.38	12.12	0.051 ± 0.096	0.034 ± 0.057	2.9 ± 1.7	3.1 ± 1.8	11.98	10.92
0.01	500	4.2 ± 3.5	4.2 ± 3.5	16.45	16.38	0.066 ± 0.19	3.8 ± 1.8	3.6 ± 1.9	19.98	10.32	0.21 ± 0.65	0.071 ± 0.15	4.7 ± 2.6	5.1 ± 3.3	15.46	23.16
0.01	1000	4.3 ± 3	4.3 ± 3	11.43	11.46	0.025 ± 0.067	3.5 ± 2.9	3.4 ± 3	12.49	10.57	0.083 ± 0.28	0.044 ± 0.11	3.6 ± 2.3	3.8 ± 2.3	9.91	11.61
0.05	5	4.2 ± 3.1	4.2 ± 3.1	12.33	12.39	0.89 ± 0.46	3.1 ± 1.7	2.6 ± 1.7	30.32	6.59	1.3 ± 1.1	1.3 ± 0.73	4.2 ± 2.6	4.6 ± 2.7	28.52	31.41
0.05	10	4 ± 3.7	4 ± 3.7	15.73	15.80	0.33 ± 0.19	4 ± 2.7	3.6 ± 2.7	16.73	11.19	0.53 ± 0.83	0.48 ± 0.43	5.8 ± 4.2	5.9 ± 3.7	32.58	41.23
0.05	50	5 ± 4.8	5 ± 4.8	25.08	25.06	0.098 ± 0.16	3.7 ± 2	3.5 ± 2	17.56	8.33	0.17 ± 0.36	0.22 ± 0.63	3.5 ± 1.8	3.7 ± 1.8	19.98	22.34
0.05	100	4 ± 3.6	4 ± 3.6	14.70	14.64	0.11 ± 0.2	4.4 ± 3.3	4.2 ± 3.3	26.05	14.72	0.12 ± 0.2	0.18 ± 0.38	2.9 ± 1.4	3.1 ± 1.5	14.30	17.15
0.05	250	3.4 ± 2.5	3.3 ± 2.5	13.60	13.65	0.17 ± 0.37	4.3 ± 3	4.1 ± 3.3	17.55	12.13	0.051 ± 0.086	0.043 ± 0.089	2.9 ± 1.7	3.1 ± 1.8	10.12	11.05
0.05	500	4.2 ± 3.5	4.2 ± 3.5	16.42	16.38	0.074 ± 0.2	3.8 ± 1.9	3.6 ± 1.9	19.35	10.32	0.18 ± 0.54	0.069 ± 0.15	4.7 ± 2.5	5.1 ± 3.3	14.80	23.19
0.05	1000	4.3 ± 3	4.3 ± 3	11.43	11.47	0.042 ± 0.12	3.5 ± 2.9	3.4 ± 3	12.72	10.57	0.079 ± 0.26	0.051 ± 0.14	3.6 ± 2.3	3.8 ± 2.3	10.20	11.80
0.1	5	4.2 ± 3.1	4.2 ± 3.1	12.30	12.40	0.76 ± 0.24	3.1 ± 1.6	2.6 ± 1.7	28.81	6.59	1.2 ± 0.88	1.3 ± 0.62	4 ± 2.4	4.7 ± 2.7	31.10	35.94
0.1	10	4 ± 3.7	4 ± 3.7	15.73	15.92	0.38 ± 0.34	4.1 ± 2.7	3.6 ± 2.7	18.46	11.20	0.48 ± 0.75	0.57 ± 0.56	5.5 ± 3.9	6.2 ± 4.1	32.52	41.23
0.1	50	5 ± 4.8	5 ± 4.8	25.09	25.08	0.068 ± 0.083	3.8 ± 2	3.5 ± 2	18.07	8.34	0.2 ± 0.37	0.15 ± 0.26	3.5 ± 1.9	3.6 ± 1.9	19.76	22.47
0.1	100	4 ± 3.6	4 ± 3.6	14.69	14.65	0.11 ± 0.15	4.4 ± 3.4	4.2 ± 3.3	26.06	14.72	0.11 ± 0.17	0.17 ± 0.33	2.9 ± 1.4	3.1 ± 1.6	15.68	16.99
0.1	250	3.4 ± 2.5	3.3 ± 2.5	13.59	13.64	0.22 ± 0.49	4.4 ± 3.1	4.1 ± 3.3	18.14	12.13	0.053 ± 0.076	0.043 ± 0.077	2.9 ± 1.7	3.1 ± 1.8	11.87	12.05
0.1	500	4.2 ± 3.5	4.2 ± 3.5	16.40	16.41	0.064 ± 0.16	3.7 ± 1.8	3.6 ± 1.9	19.74	10.35	0.2 ± 0.64	0.079 ± 0.17	4.8 ± 2.6	5.1 ± 3.3	14.96	23.23
0.1	1000	4.3 ± 3	4.3 ± 3	11.44	11.47	0.044 ± 0.14	3.5 ± 2.9	3.4 ± 3	12.71	10.59	0.08 ± 0.25	0.035 ± 0.071	3.6 ± 2.3	3.8 ± 2.3	14.58	12.06
0.5	5	4.2 ± 3.1	4.2 ± 3.1	12.33	12.44	1.2 ± 0.46	3.3 ± 1.8	2.6 ± 1.7	21.88	6.62	1.5 ± 0.54	1.7 ± 0.5	3.5 ± 1.9	4.1 ± 1.8	23.42	26.62
0.5	10	4 ± 3.7	4 ± 3.7	15.88	15.95	0.78 ± 0.26	4.3 ± 2.9	3.6 ± 2.7	19.11	11.26	1 ± 0.7	0.96 ± 0.59	5.1 ± 3.6	5.2 ± 3.6	31.49	28.43
0.5	50	5 ± 4.8	5 ± 4.8	25.10	25.20	0.27 ± 0.092	3.7 ± 2	3.5 ± 2	23.06	8.36	0.42 ± 0.37	0.36 ± 0.38	3.5 ± 1.8	3.7 ± 1.8	22.53	16.93
0.5	100	4 ± 3.6	4 ± 3.6	14.75	14.84	0.26 ± 0.18	4.4 ± 3.4	4.2 ± 3.3	25.07	14.76	0.28 ± 0.15	0.35 ± 0.35	3 ± 1.4	3.1 ± 1.6	16.18	20.26
0.5	250	3.4 ± 2.5	3.3 ± 2.5	13.71	13.65	0.26 ± 0.25	4.3 ± 3.1	4.1 ± 3.3	18.51	12.23	0.23 ± 0.24	0.2 ± 0.17	3 ± 1.7	3.2 ± 1.9	11.88	17.96
0.5	500	4.2 ± 3.5	4.2 ± 3.5	16.47	16.52	0.15 ± 0.13	3.7 ± 1.8	3.6 ± 1.9	19.16	10.43	0.29 ± 0.51	0.21 ± 0.21	4.8 ± 2.4	5.1 ± 3.3	19.23	22.26
0.5	1000	4.3 ± 3	4.3 ± 3	11.52	11.52	0.11 ± 0.083	3.5 ± 2.9	3.4 ± 3	21.50	10.61	0.14 ± 0.15	0.13 ± 0.063	3.7 ± 2.3	3.8 ± 2.4	15.24	14.32
1.0	5	4.2 ± 3.1	4.2 ± 3.1	12.32	12.41	3.8 ± 1.8	4.6 ± 1.7	3 ± 2	32.28	9.61	4.2 ± 2.3	3.8 ± 1.3	5.3 ± 2.2	6.6 ± 2.7	37.97	36.38
1.0	10	4 ± 3.7	4 ± 3.7	15.88	15.87	2 ± 1.2	4.8 ± 3	3.6 ± 2.7	21.89	11.28	2.4 ± 1.1	2.2 ± 0.98	6 ± 3.7	6.3 ± 2.9	31.65	44.60
1.0	50	5 ± 4.8	5 ± 4.8	25.06	25.11	0.82 ± 0.48	3.9 ± 2	3.5 ± 2	18.08	8.36	1 ± 0.57	0.88 ± 0.57	3.9 ± 1.7	4.2 ± 1.8	25.62	28.23
1.0	100	4 ± 3.6	4 ± 3.6	14.70	14.71	0.83 ± 0.6	4.5 ± 3.4	4.2 ± 3.3	27.90	14.75	0.76 ± 0.39	0.92 ± 0.72	3.2 ± 1.3	3.4 ± 1.8	18.80	19.19
1.0	250	3.4 ± 2.5	3.3 ± 2.5	13.66	13.63	0.61 ± 0.59	4.4 ± 3	4.1 ± 3.3	19.39	12.15	0.63 ± 0.49	0.5 ± 0.21	3 ± 1.4	3.5 ± 2.1	12.91	24.54
1.0	500	4.2 ± 3.5	4.2 ± 3.5	16.45	16.43	0.49 ± 0.33	3.9 ± 1.7	3.6 ± 1.9	20.58	10.38	0.57 ± 0.48	0.57 ± 0.31	5.1 ± 2.7	5.4 ± 3.2	20.62	26.60
1.0	1000	4.3 ± 3	4.3 ± 3	11.48	11.49	0.42 ± 0.52	3.5 ± 2.7	3.4 ± 3	21.57	10.60	0.4 ± 0.21	0.44 ± 0.22	3.7 ± 2.2	4.1 ± 2.5	18.67	16.60
2.0	5	4.2 ± 3.1	4.2 ± 3.1	12.32	12.41	7.6 ± 3.2	6.5 ± 3	3 ± 2	57.94	9.56	9.5 ± 3.8	8.8 ± 2.5	7.1 ± 2.2	8 ± 3	36.84	42.81
2.0	10	4 ± 3.7	4 ± 3.7	15.78	15.89	5.4 ± 3.1	5.6 ± 3.3	3.6 ± 2.7	53.66	11.27	5.9 ± 2.7	5.7 ± 1.9	6.7 ± 3.8	7.7 ± 2.9	39.83	48.78
2.0	50	5 ± 4.8	5 ± 4.8	25.05	25.11	2.3 ± 1.4	4.1 ± 1.9	3.5 ± 2	25.90	8.34	2.4 ± 0.93	2.1 ± 0.91	4.6 ± 1.9	5.1 ± 2.3	32.77	30.85
2.0	100	4 ± 3.6	4 ± 3.6	14.66	14.62	2.2 ± 1.1	5.2 ± 3.4	4.2 ± 3.3	33.76	14.74	2 ± 0.64	2.1 ± 0.92	3.8 ± 1.4	4 ± 1.6	20.82	20.96
2.0	250	3.4 ± 2.5	3.3 ± 2.5	13.63	13.66	1.6 ± 0.97	4.7 ± 2.7	4.1 ± 3.3	23.76	12.14	1.6 ± 0.56	1.4 ± 0.38	3.4 ± 1.4	4.4 ± 2.5	25.07	34.36
2.0	500	4.2 ± 3.5	4.2 ± 3.5	16.42	16.41	1.4 ± 0.62	4.1 ± 1.6	3.6 ± 1.9	21.56	10.34	1.4 ± 0.93	1.7 ± 0.7	5.6 ± 2.5	5.9 ± 3.2	25.22	24.22
2.0	1000	4.3 ± 3	4.3 ± 3	11.44	11.48	1.1 ± 0.59	3.8 ± 2.4	3.4 ± 3	23.69	10.59	1.1 ± 0.37	1.3 ± 0.43	4.2 ± 2.4	4.7 ± 2.6	23.69	21.91
5.0	5	4.2 ± 3.1	4.2 ± 3.1	12.29	12.40	19 ± 5.2	9.3 ± 3	3 ± 2	68.72	9.52	19 ± 6.1	20 ± 4.8	8.9 ± 2.5	11 ± 3.5	51.93	77.55
5.0	10	4 ± 3.7	4 ± 3.7	15.72	15.80	14 ± 5	7.2 ± 3.1	3.6 ± 2.7	39.97	11.23	15 ± 4.5	13 ± 3.8	8.4 ± 3.7	9.7 ± 3.2	65.43	58.71
5.0	50	5 ± 4.8	5 ± 4.8	25.11	25.08	6.5 ± 2.6	5.4 ± 1.8	3.5 ± 2	28.77	8.34	6.9 ± 2.2	6.7 ± 2.5	6.3 ± 1.7	6.1 ± 1.3	41.35	32.89
5.0	100	4 ± 3.6	4 ± 3.6	14.66	14.60	5.1 ± 1.6	6 ± 2.3	4.2 ± 3.3	28.69	14.74	5.1 ± 2	5 ± 1.7	5 ± 1.5	5.5 ± 1.6	27.75	26.21
5.0	250	3.4 ± 2.5	3.3 ± 2.5	13.63	13.65	4 ± 1.2	5.9 ± 2.7	4.1 ± 3.3	29.17	12.13	4.4 ± 1.5	4 ± 0.88	4.7 ± 1.5	6.2 ± 2.6	23.32	24.38
5.0	500	4.2 ± 3.5	4.2 ± 3.5	16.43	16.39	3.8 ± 1.2	5.4 ± 1.6	3.6 ± 1.9	19.11	10.33	3.5 ± 1.4	3.5 ± 1.2	6.6 ± 2.3	7 ± 2.4	31.47	24.31
5.0	1000	4.3 ± 3	4.3 ± 3	11.44	11.47	3 ± 0.95	4.9 ± 2.1	3.4 ± 3	22.51	10.58	3.2 ± 1	3.2 ± 1.1	5.3 ± 2.2	6.3 ± 2.5	20.61	24.65
10.0	5	4.2 ± 3.1	4.2 ± 3.1	12.32	12.39	26 ± 5.8	11 ± 4.5	3 ± 2	67.70	9.54	27 ± 7.4	28 ± 9.3	11 ± 2.9	13 ± 4.3	58.03	64.23
10.0	10	4 ± 3.7	4 ± 3.7	15.72	15.78	19 ± 7	8.2 ± 3.6	3.6 ± 2.7	41.59	11.23	22 ± 5.4	19 ± 6.4	10 ± 4.5	11 ± 3.7	75.09	47.14
10.0	50	5 ± 4.8	5 ± 4.8	25.12	25.07	12 ± 5.2	7.7 ± 3.4	3.5 ± 2	62.12	8.34	11 ± 3.4	10 ± 3.1	7.2 ± 1.6	7.4 ± 1.9	36.90	34.22
10.0	100	4 ± 3.6	4 ± 3.6	14.66	14.60	8.1 ± 2.8	7 ± 2.3	4.2 ± 3.3	38.44	14.73	8.4 ± 2.4	7.9 ± 2.5	6.5 ± 1.8	7 ± 1.9	26.87	30.49
10.0	250	3.4 ± 2.5	3.3 ± 2.5	13.62	13.64	6.3 ± 2.2	7 ± 2.5	4.1 ± 3.3	30.96	12.13	5.8 ± 2.3	5.5 ± 1.4	5.5 ± 1.7	7 ± 2.1	18.64	21.72
10.0	500	4.2 ± 3.5	4.2 ± 3.5	16.44	16.39	5.7 ± 2.1	6.3 ± 1.4	3.6 ± 1.9	19.84	10.33	5.3 ± 2.3	5.4 ± 2.3	7.5 ± 2.4	7.3 ± 1.4	29.68	23.55
10.0	1000	4.3 ± 3	4.3 ± 3	11.44	11.47	4.5 ± 2.1	6.1 ± 1.8	3.4 ± 3	20.38	10.57	4.4 ± 2	4.4 ± 1.5	5.9 ± 1.9	7.5 ± 2.2	18.74	25.56

Table 3: Deflections of identities from simulations with different numbers of samples (N), and environment noise, σ_e. Roughening noise: $\sigma_r = N^{-1/3}$, model environment noise: $\gamma = \max(0.5, \sigma_e)$. *id-deflections* in cases where the identity is known are not shown as they are all very small (less than 10^{-3}).

σ_e	s_{id}	id-deflection agent	id-deflection client ($\times 10^3$)	deflection agent	deflection client	num deflected frames $d_m = 1.0$	$d_m = 2.0$	$d_m = 3.0$	$d_m = 5.0$
0.1	0.01	0.065 ± 0.15	0.28 ± 0.14	3.1 ± 2.3	3 ± 2.4	4.12 ± 9.09	0.53 ± 0.96	0.04 ± 0.20	0.00 ± 0.00
0.1	0.1	0.16 ± 0.13	0.27 ± 0.13	3.6 ± 2.4	3.9 ± 2.7	5.14 ± 12.03	0.94 ± 5.71	0.04 ± 0.20	0.00 ± 0.00
0.1	0.5	0.51 ± 0.54	0.28 ± 0.14	3.2 ± 2.3	3.4 ± 2.4	18.11 ± 18.37	1.94 ± 4.52	0.19 ± 1.53	0.00 ± 0.00
0.1	1.0	0.51 ± 0.97	0.28 ± 0.1	3.2 ± 2.3	3.5 ± 2.5	16.80 ± 18.47	3.69 ± 7.14	0.35 ± 1.32	0.00 ± 0.00
0.1	2.0	0.14 ± 0.33	0.29 ± 0.12	3.4 ± 2.4	3.5 ± 2.5	10.36 ± 12.56	2.50 ± 4.72	0.60 ± 2.16	0.04 ± 0.49
0.5	0.01	0.17 ± 0.06	0.31 ± 0.14	3.1 ± 2.3	3 ± 2.4	6.20 ± 6.59	0.84 ± 1.52	0.06 ± 0.29	0.00 ± 0.00
0.5	0.1	0.31 ± 0.16	0.3 ± 0.13	3.6 ± 2.4	3.9 ± 2.7	9.89 ± 13.75	1.28 ± 5.69	0.10 ± 0.57	0.00 ± 0.00
0.5	0.5	0.69 ± 0.48	0.31 ± 0.11	3.2 ± 2.2	3.5 ± 2.5	23.99 ± 20.29	2.83 ± 5.40	0.18 ± 1.03	0.00 ± 0.00
0.5	1.0	0.71 ± 1.1	0.3 ± 0.14	3.2 ± 2.2	3.5 ± 2.5	21.87 ± 19.54	4.96 ± 9.27	0.74 ± 2.68	0.00 ± 0.00
0.5	2.0	0.3 ± 0.32	0.29 ± 0.15	3.3 ± 2.3	3.5 ± 2.5	15.20 ± 15.48	3.21 ± 6.31	0.72 ± 2.52	0.03 ± 0.35
1.0	0.01	0.41 ± 0.24	0.28 ± 0.13	3 ± 2	3 ± 2.4	20.70 ± 19.44	3.23 ± 7.13	0.27 ± 1.14	0.00 ± 0.00
1.0	0.1	0.94 ± 0.99	0.31 ± 0.16	3.3 ± 2.1	3.9 ± 2.7	35.76 ± 23.98	4.32 ± 10.78	0.69 ± 4.56	0.00 ± 0.00
1.0	0.5	1.1 ± 0.74	0.28 ± 0.12	3.1 ± 2.1	3.5 ± 2.5	41.83 ± 23.05	8.47 ± 11.46	0.91 ± 2.92	0.00 ± 0.00
1.0	1.0	1.1 ± 1.2	0.27 ± 0.11	3.2 ± 2.2	3.5 ± 2.5	38.22 ± 24.14	9.12 ± 13.40	1.82 ± 5.33	0.01 ± 0.14
1.0	2.0	0.63 ± 0.51	0.29 ± 0.13	3.3 ± 2.3	3.5 ± 2.5	31.57 ± 20.70	6.37 ± 10.64	1.23 ± 4.22	0.01 ± 0.07

Table 4: Deflections of identities from simulations with different environment noise, σ_e, and shapeshifted id speed, s_{id}. $N = 250$, *agent-id-hidden*.

σ_e	s_{id}	id-deflection agent	id-deflection client ($\times 10^3$)	deflection agent	deflection client	num deflected frames $d_m = 1.0$	$d_m = 2.0$	$d_m = 3.0$	$d_m = 5.0$
0.1	0.01	0.048 ± 0.051	0.88 ± 0.45	2.9 ± 1.7	2.9 ± 1.8	3.60 ± 5.42	0.64 ± 0.89	0.09 ± 0.28	0.00 ± 0.00
0.1	0.1	0.16 ± 0.094	0.82 ± 0.34	2.9 ± 1.6	2.9 ± 1.6	4.31 ± 6.82	0.63 ± 0.86	0.09 ± 0.28	0.00 ± 0.00
0.1	0.5	0.74 ± 0.8	0.83 ± 0.39	3.1 ± 1.4	3.3 ± 1.7	44.06 ± 44.05	5.61 ± 14.75	0.56 ± 2.70	0.00 ± 0.00
0.1	1.0	0.75 ± 1	0.85 ± 0.4	3.1 ± 1.5	3.4 ± 1.6	40.98 ± 46.90	9.68 ± 23.76	2.40 ± 9.69	0.03 ± 0.21
0.1	2.0	0.37 ± 0.94	0.76 ± 0.51	2.8 ± 1.9	2.9 ± 2.1	22.91 ± 36.32	5.87 ± 15.39	1.66 ± 5.69	0.06 ± 0.40
0.5	0.01	0.19 ± 0.07	0.94 ± 0.37	2.9 ± 1.7	2.9 ± 1.8	7.80 ± 8.56	1.31 ± 2.94	0.26 ± 1.26	0.00 ± 0.00
0.5	0.1	0.32 ± 0.14	0.86 ± 0.35	3 ± 1.6	2.9 ± 1.6	12.84 ± 12.71	1.39 ± 3.55	0.26 ± 1.27	0.00 ± 0.00
0.5	0.5	0.88 ± 0.89	0.84 ± 0.4	3 ± 1.5	3.3 ± 1.7	54.06 ± 44.18	8.07 ± 17.32	1.11 ± 4.29	0.01 ± 0.07
0.5	1.0	0.94 ± 1	1 ± 0.46	3 ± 1.6	3.4 ± 1.7	49.47 ± 48.18	11.62 ± 24.86	3.00 ± 11.31	0.08 ± 0.48
0.5	2.0	1.2 ± 1.8	1.1 ± 0.8	3.5 ± 2.4	4.1 ± 3.1	48.40 ± 59.16	18.09 ± 32.08	5.38 ± 12.87	0.32 ± 1.54
1.0	0.01	0.43 ± 0.17	0.92 ± 0.42	2.9 ± 1.7	2.9 ± 1.8	26.59 ± 23.41	2.75 ± 4.66	0.31 ± 0.98	0.00 ± 0.00
1.0	0.1	0.66 ± 0.23	0.91 ± 0.4	3 ± 1.6	2.9 ± 1.6	57.30 ± 31.38	3.92 ± 8.54	0.57 ± 3.65	0.00 ± 0.00
1.0	0.5	1.3 ± 0.99	0.92 ± 0.47	3 ± 1.4	3.3 ± 1.7	89.39 ± 46.56	16.93 ± 26.96	2.65 ± 9.29	0.10 ± 1.16
1.0	1.0	1.3 ± 1.1	0.88 ± 0.44	3.1 ± 1.5	3.5 ± 1.7	79.89 ± 50.71	17.57 ± 29.44	3.67 ± 11.98	0.08 ± 1.00
1.0	2.0	1 ± 0.99	0.86 ± 0.42	3.2 ± 1.5	3.7 ± 2	59.55 ± 45.36	13.04 ± 21.26	2.94 ± 7.96	0.10 ± 0.71

Table 5: Deflections of identities from simulations with different environment noise, σ_e, and shapeshifted id speed, s_{id}. $N = 100$, *agent-id-hidden*, $d_m \equiv$ threshold for frame deflection.

B Derivation of Most Likely Behaviour

We know from Equation (8) and (10) that the belief distribution over the state at time t (denoted $\boldsymbol{s}'$) is given by

$$b(\boldsymbol{s}') = Pr(\boldsymbol{\omega}'|\boldsymbol{s}')\mathbb{E}_{b(\boldsymbol{s})}\left[Pr(\mathbf{x}'|\mathbf{x}, \mathbf{f}', \boldsymbol{\tau}', \mathbf{b_a})Pr(\boldsymbol{\tau}'|\boldsymbol{\tau}, \mathbf{f}', \mathbf{x})Pr(\mathbf{f}'|\mathbf{f}, \boldsymbol{\tau}, \mathbf{x}, \mathbf{b_a})\right] \tag{38}$$

and further, from 20 that

$$Pr(\mathbf{f}'|\mathbf{f}, \boldsymbol{\tau}, \mathbf{x}, \mathbf{b_a}, \varphi) \propto e^{-\psi(\mathbf{f}', \boldsymbol{\tau}, \mathbf{x})}\left[\mathbb{E}_{Pr(\boldsymbol{\theta_f})}(\boldsymbol{\theta_f})\right] \tag{39}$$

Let us first assume that the prior over $\boldsymbol{\theta_f}$ is uninformative, and so only the first expectation remains. Then, if we compare two values for $\boldsymbol{s}'$, say $\boldsymbol{s}'_1$ and $\boldsymbol{s}'_2$, and we imagine that we have deterministic dynamics for the application state $\mathbf{X}$ and the transients $\mathbf{T}$, then we find

$$b(\boldsymbol{s}'_1) - b(\boldsymbol{s}'_2) \propto \mathbb{E}_{b(\boldsymbol{s})}\left(e^{-\psi(\mathbf{f}'_1, \boldsymbol{s}_1)} - e^{-\psi(\mathbf{f}'_2, \boldsymbol{s}_2)}\right) \tag{40}$$

$$\geq e^{-\psi(\mathbf{f}'_1, \mathbb{E}_{b(\boldsymbol{s})}(\boldsymbol{s}_1))} - e^{-\psi(\mathbf{f}'_2, \mathbb{E}_{b(\boldsymbol{s})}(\boldsymbol{s}_2))} \tag{41}$$

where the inequality between (40) and (41) is due to the expectation of a convex function being always larger than the function of the expectation (Jensen's inequality). From (41), we have that the probability of $\mathbf{f}'_1$ will be greater than the probability of $\mathbf{f}'_2$ if and only if:

$$\psi(\mathbf{f}'_1, \mathbb{E}_{b(\boldsymbol{s})}(\boldsymbol{s}_1)) < \psi(\mathbf{f}'_2, \mathbb{E}_{b(\boldsymbol{s})}(\boldsymbol{s}_2)) \tag{42}$$

that is, the deflection caused by $\mathbf{f}'_1$ is less than the deflection caused by $\mathbf{f}'_2$. This demonstrates that our probability measure over $\mathbf{f}'$ will assign higher likelihoods to behaviours with lower deflection, as expected, and so if we wish find the most likely $\mathbf{f}'$ value, we have only to find the value that gives the smallest deflection by, e.g., taking derivatives and setting equal to zero. The probabilistic formulation in Equation (42), however, takes this one step further, and shows that the probability of $\mathbf{f}'$ will assign higher weights to behaviours that minimize deflection, but in expectation of the state progression if it is not fully deterministic.

If the prior over $\boldsymbol{\theta_f}$ is such that we expect identities to stay constant over time, as in Equation (24), we can derive a similar expression to (42), except it now includes the deflections of the fundamentals over identities:

$$\psi(\mathbf{f}'_1, \mathbb{E}_{b(\boldsymbol{s})}(\boldsymbol{s}_1)) + \xi(\mathbf{f}'_1, \mathbf{f}) < \psi(\mathbf{f}'_2, \mathbb{E}_{b(\boldsymbol{s})}(\boldsymbol{s}_2)) + \xi(\mathbf{f}'_2, \mathbf{f}) \tag{43}$$

We see that this is now significantly different than (42), as the relative weights of ξ (β_a and β_c) and ψ (α) will play a large role in determining which fundamental sentiments are most likely. If $\beta_a \gg \alpha$ or $\beta_c \gg \alpha$, then the agents beliefs about identities will change more readily to accommodate observed deflections. If the opposite is true, then deflections will be ignored to accommodate constant identities.

C Connection to Heise's most likely behaviours

In this section, we show that the most likely predictions from our model match those from [1] if we use the same approximations. We begin from the probability distribution of fundamentals from Equation (24), but we assume deterministic state transitions, ignore the fundamental inertia ξ, and use the formula for ψ as given by Equation (27), we get

$$Pr(\mathbf{f}'|\mathbf{f}, \boldsymbol{\tau}, \mathbf{x}, \mathbf{b_a}, \varphi) \propto e^{-(\mathbf{f}' - \mathscr{K}^{-1}\mathscr{C})^T \mathscr{K}^T \boldsymbol{\Sigma}^{-1} \mathscr{K}(\mathbf{f}' - \mathscr{K}^{-1}\mathscr{C})} \tag{44}$$

Now we saw in Section 3.3 that this was simply a Gaussian with a mean of $\mathscr{K}^{-1}\mathscr{C}$, and so gives us the expected (most likely or "optimal" in Heise's terms) behaviours *and* identities simultaneously. We can find these expected fundamentals by taking the total derivative and setting to zero

$$\frac{d}{d\mathbf{f}'}Pr(\mathbf{f}'|\mathbf{f},\boldsymbol{\tau},\mathbf{x},\mathbf{b_a},\varphi) = \left(\frac{d}{d\mathbf{f}'}(\mathbf{f}'-\mathscr{K}^{-1}\mathscr{C})^T\mathscr{K}^T\boldsymbol{\Sigma}^{-1}\mathscr{K}(\mathbf{f}'-\mathscr{K}^{-1}\mathscr{C})\right)e^{-(\mathbf{f}'-\mathscr{K}^{-1}\mathscr{C})^T\mathscr{K}^T\boldsymbol{\Sigma}^{-1}\mathscr{K}(\mathbf{f}'-\mathscr{K}^{-1}\mathscr{C})} = 0$$

which means that $\mathbf{f}' = \mathscr{K}^{-1}\mathscr{C}$ (the mean of the Gaussian), as expected. Heise, however, estimates the derivatives of each of the identities and behaviours separately *assuming the others are held fixed.* This is the same as taking partial derivatives of (44) with respect to $\mathbf{f}_b$ only while holding the others fixed:

$$\left(\frac{\partial}{\partial\mathbf{f}'_b}(\mathbf{f}'-\mathscr{K}^{-1}\mathscr{C})^T\mathscr{K}^T\boldsymbol{\Sigma}^{-1}\mathscr{K}(\mathbf{f}'-\mathscr{K}^{-1}\mathscr{C})\right)e^{-(\mathbf{f}'-\mathscr{K}^{-1}\mathscr{C})^T\mathscr{K}^T\boldsymbol{\Sigma}^{-1}\mathscr{K}(\mathbf{f}'-\mathscr{K}^{-1}\mathscr{C})} = 0 \tag{45}$$

Now, we recall that (writing $\boldsymbol{I}\equiv\boldsymbol{I_3}$ and $\mathbf{0}\equiv\mathbf{0}_3$):

$$\mathscr{K} = \begin{bmatrix} \boldsymbol{I} & -\mathscr{H}_a & \mathbf{0} \\ \mathbf{0} & 1-\mathscr{H}_b & \mathbf{0} \\ \mathbf{0} & -\mathscr{H}_c & \boldsymbol{I} \end{bmatrix}$$

So that

$$\mathscr{K}^{-1} = \begin{bmatrix} \boldsymbol{I} & \mathscr{H}_a(1-\mathscr{H}_b)^{-1} & \mathbf{0} \\ \mathbf{0} & (1-\mathscr{H}_b)^{-1} & \mathbf{0} \\ \mathbf{0} & \mathscr{H}_c(1-\mathscr{H}_b)^{-1} & \boldsymbol{I} \end{bmatrix}$$

and if $\boldsymbol{\Sigma}$ is a diagonal identity matrix, we can write

$$\mathscr{K}^T\boldsymbol{\Sigma}^{-1}\mathscr{K} = \begin{bmatrix} \boldsymbol{I} & -\mathscr{H}_a & \mathbf{0} \\ -\mathscr{H}_a & \mathscr{H}_a^2+(1-\mathscr{H}_b)^2+\mathscr{H}_c^2 & -\mathscr{H}_c \\ \mathbf{0} & -\mathscr{H}_c & \boldsymbol{I} \end{bmatrix}$$

To simplify, we let $a = -\mathscr{H}_a, b = 1-\mathscr{H}_b, c = -\mathscr{H}_c$, and $z = \mathscr{H}_a^2+(1-\mathscr{H}_b)^2+\mathscr{H}_c^2$ we get

$$\mathscr{K}^T\boldsymbol{\Sigma}^{-1}\mathscr{K} = \begin{bmatrix} \boldsymbol{I} & a & \mathbf{0} \\ a & z & c \\ \mathbf{0} & c & \boldsymbol{I} \end{bmatrix}$$

we also have that

$$\mathbf{f}-\mathscr{K}^{-1}\mathscr{C} = \begin{bmatrix} \mathbf{f}_a-\left(\mathscr{C}_a+\mathscr{C}_b\mathscr{H}_a(1-\mathscr{H}_b)^{-1}\right) \\ \mathbf{f}_b-(1-\mathscr{H}_b)^{-1}\mathscr{C}_b \\ \mathbf{f}_c-\left(\mathscr{C}_c+\mathscr{C}_b\mathscr{H}_c(1-\mathscr{H}_b)^{-1}\right) \end{bmatrix} = \begin{bmatrix} y_a \\ y_b \\ y_c \end{bmatrix}$$

where we have used y_a, y_b, y_c to denote the difference between the actor identity, behaviour and object identity and their respective true means as given by the total derivative. Therefore, we have from Equation (45):

$$\frac{\partial}{\partial\mathbf{f}_b}\left(y_a^2+2ay_ay_b+zy_b^2+2cy_by_c+y_c^2\right) = 0$$

$$2ay_a+2zy_b+2cy_c = 0$$

and therefore that the "optimal" behaviour, $\mathbf{f}_b^*$ is

$$\mathbf{f}_b^* = -z^{-1}(ay_a+cy_c)+(1-\mathscr{H}_b)^{-1}\mathscr{C}_b$$

partially expanding out this is

$$\mathbf{f}_b^* = -z^{-1}\left[-\mathscr{H}_a(\mathbf{f}_a-\mathscr{C}_a-\mathscr{C}_b\mathscr{H}_a(1-\mathscr{H}_b)^{-1})-\mathscr{H}_c(\mathbf{f}_c-\mathscr{C}_c-\mathscr{C}_b\mathscr{H}_c(1-\mathscr{H}_b)^{-1})-z\mathscr{C}_b(1-\mathscr{H}_b)^{-1}\right] \tag{46}$$

Now we note that, the terms from Heise's book can be written as follows

$$[\boldsymbol{I}-\boldsymbol{M'}]\boldsymbol{I_\beta g_\beta} = \begin{bmatrix} \mathbf{f}_a - \mathscr{C}_a \\ -\mathscr{C}_b \\ \mathbf{f}_c - \mathscr{C}_c \end{bmatrix}$$

and

$$[\boldsymbol{I}-\boldsymbol{M'}]\boldsymbol{I_\beta S_\beta} = \begin{bmatrix} \mathscr{H}_a \\ \boldsymbol{I} - \mathscr{H}_b \\ \mathscr{H}_c \end{bmatrix}$$

so that

$$\boldsymbol{S}_{\boldsymbol\beta}^T \boldsymbol{I_\beta} \begin{bmatrix} \boldsymbol{I} \\ -\boldsymbol{M'} \end{bmatrix} [\boldsymbol{I}-\boldsymbol{M'}]\boldsymbol{I_\beta g_\beta} = -\mathscr{H}_a(\mathbf{f}_a - \mathscr{C}_a) - \mathscr{C}_b(\boldsymbol{I} - \mathscr{H}_b) - \mathscr{H}_c(\mathbf{f}_c - \mathscr{C}_c) \tag{47}$$

and that

$$\left(\boldsymbol{S}_{\boldsymbol\beta}^T \boldsymbol{I_\beta} \begin{bmatrix} \boldsymbol{I} \\ -\boldsymbol{M'} \end{bmatrix} [\boldsymbol{I}-\boldsymbol{M'}]\boldsymbol{I_\beta S_\beta}\right)^{-1} = \left(\mathscr{H}_a^T \mathscr{H}_a + (\boldsymbol{I} - \mathscr{H}_b)^T (\boldsymbol{I} - \mathscr{H}_b) + \mathscr{H}_c^T \mathscr{H}_c\right)^{-1} = z^{-1}$$

so that $\mathbf{f}_b^*$ is now

$$\begin{aligned} \mathbf{f}_b^* = -\left(\boldsymbol{S}_{\boldsymbol\beta}^T \boldsymbol{I_\beta} \begin{bmatrix} \boldsymbol{I} \\ -\boldsymbol{M'} \end{bmatrix} [\boldsymbol{I}-\boldsymbol{M'}]\boldsymbol{I_\beta S_\beta}\right)^{-1} \times \\ \big[-\mathscr{H}_a(\mathbf{f}_a - \mathscr{C}_a) + \mathscr{H}_a \mathscr{C}_b \mathscr{H}_a (1 - \mathscr{H}_b)^{-1} - \mathscr{H}_c(\mathbf{f}_c - \mathscr{C}_c) + \mathscr{H}_c \mathscr{H}_b \mathscr{H}_c (1 - \mathscr{H}_b)^{-1} \\ - \left(\mathscr{H}_a^T \mathscr{H}_a + (1 - \mathscr{H}_b)^T (1 - \mathscr{H}_b) + \mathscr{H}_c^T \mathscr{H}_c\right) \mathscr{C}_b (1 - \mathscr{H}_b)^{-1}\big] \end{aligned} \tag{48}$$

collecting terms and comparing to Equation (47), this give us exactly Equation (12.21) from [1]:

$$\mathbf{f}_b^* = -\left(\boldsymbol{S}_{\boldsymbol\beta}^T \boldsymbol{I_\beta} \begin{bmatrix} \boldsymbol{I} \\ -\boldsymbol{M'} \end{bmatrix} [\boldsymbol{I}-\boldsymbol{M'}]\boldsymbol{I_\beta S_\beta}\right)^{-1} \boldsymbol{S}_{\boldsymbol\beta}^T \boldsymbol{I_\beta} \begin{bmatrix} \boldsymbol{I} \\ -\boldsymbol{M'} \end{bmatrix} [\boldsymbol{I}-\boldsymbol{M'}]\boldsymbol{I_\beta g_\beta} \tag{49}$$

Similar equations for actor and object identities can be obtained in the same way by computing with partial derivatives keeping all other quantities fixed, and the result is equations (13.11) and (13.18) from [1].