Learning Theory of Mind via Dynamic Traits Attribution

Dung Nguyen  
A²P², Deakin University  
Geelong, Australia  
dung.nguyen@deakin.edu.au

Phuoc Nguyen  
A²P², Deakin University  
Geelong, Australia  
phuoc.nguyen@deakin.edu.au

Hung Le  
A²P², Deakin University  
Geelong, Australia  
thai.le@deakin.edu.au

Kien Do  
A²P², Deakin University  
Geelong, Australia  
k.do@deakin.edu.au

Svetha Venkatesh  
A²P², Deakin University  
Geelong, Australia  
svetha.venkatesh@deakin.edu.au

Truyen Tran  
A²P², Deakin University  
Geelong, Australia  
truyen.tran@deakin.edu.au

ABSTRACT
Machine learning of Theory of Mind (ToM) is essential to build social agents that co-live with humans and other agents. This capacity, once acquired, will help machines infer the mental states of others from observed contextual action trajectories, enabling future prediction of goals, intention, actions and successor representations. The underlying mechanism for such a prediction remains unclear, however. Inspired by the observation that humans often infer the character traits of others, then use it to explain behaviour, we propose a new neural ToM architecture that learns to generate a latent trait vector of an actor from the past trajectories. This trait vector then multiplicatively modulates the prediction mechanism via a ‘fast weights’ scheme in the prediction neural network, which reads the current context and predicts the behaviour. We empirically show that the fast weights provide a good inductive bias to model the character traits of agents and hence improves mindreading ability. On the indirect assessment of false-belief understanding, the new ToM model enables more efficient helping behaviours.

KEYWORDS
Theory of Mind; False Belief; Multiplicative Interaction

ACM Reference Format:
Dung Nguyen, Phuoc Nguyen, Hung Le, Kien Do, Svetha Venkatesh, and Truyen Tran. 2022. Learning Theory of Mind via Dynamic Traits Attribution. In Proc of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022, IFAAMAS, 9 pages.

1 INTRODUCTION
The capacity of a social agent to predict and interpret the behaviours of others is essential for it to thrive. A basis for this mindreading, also known as theory of mind (ToM), is the ability to attribute transient mental states – knowledge, emotions, beliefs, desires and intention – to others and use the attribution to reason about their actions [12, 16, 17, 31, 35]. We also often attribute stable character traits to others in order to interpret their behaviours [1, 22, 25]. While these two distinct attribution skills have been known to be related, it remains unclear how they are integrated into a coherent system. A recent theory has been pushed forward, suggesting that temporally stable character traits can be used to generate a prior probability distribution for hypotheses about mental states [41]. A computational theory to realise the hypothesis remains open.

Different from these works, we employ the fast weight concept [2, 36] to represent the character traits of actors in a behaviour prediction network. Unlike standard slow weights which are fixed after training, fast weights are computed on-the-fly at inference time, conditioned on the observations. More specifically, these fast
weights are generated by a hypernetwork \([18, 20]\) using past behaviours. We use these character trait weights to modulate the mind prediction in a multiplicative manner. A ToM observer will have the freedom to change the weights of the prediction networks independently for each character trait. We show in the experiments that this capacity helps the ToM observer correctly predict the actor’s goal, intention, and trajectories and perform better in complex tasks. To the best of our knowledge, our paper is the first work implementing and analysing the idea of fast weights and hypernetworks for mindreading tasks. Our main contribution is introducing a new type of Trait-based ToM (Trait-ToM) agent (the observer), which can represent the character traits of other actors and use it to make behaviour predictions of others. We verify the predictive power of Trait-ToM on a suite of tasks in a key-door environment and demonstrate promising results.

2 PROBLEM FORMULATION

We consider a family of partially observable Markov decision processes (POMDPs) \(W = \cup_j W_j\), where each environment is a tuple \(W_j = (S_j, A_j, T_j)\) of the state space \(S_j\), the action space \(A_j\) and the transition functions \(T_j\). Acting on the environments is a family of actors \(\mathcal{A} = \cup_i \mathcal{A}_i\), each of which has its own observation space \(O_i\), observation function \(\Omega_i : O_i \times S_j \times A_j \rightarrow [0,1]\), reward function \(R_i\), and a policy \(\pi_i\), i.e. \(\mathcal{A}_i = (O_i, \Omega_i, R_i, W_i, \pi_i)\). In the simplest form, the type of each actor is defined by its perception ability, preferences, and strategy. Finally, we consider an observer who can observe the behaviours of the actors. Here, the behaviour of the actor \(i\) in the environment \(W_j\) is represented by the trajectory \(\tau_{ij} = (s(t), a(t))_t^{T-1}\) with \(s(t) \in S_j, a(t) \in A_j\), and \(T\) is the length of the trajectory.

The observer or theory of mind (ToM) agent first observes a set of \(N_{past}\) past trajectories \(\{\tau_{ij}\}_{j=1}^{N_{past}}\) of an actor \(i\) in different environments \(W_j\) with \(j = 1, \ldots, N_{past}\). We hypothesise that these past behaviours exhibit the character of this actor, allowing the formation of a good prior for predicting its behaviours. We then ask the ToM agent to predict the behaviours of this actor in the current environment by observing the current trajectory, including the state and action pairs up to the query time step, \( \tau_{i, current} = \{s(t), a(t)\}_{t=0}^{T_q-1}\) and the current state of the world \(s(T_q)\). Here, \(T_q\) is the time step that the ToM agent is queried to make prediction. The behaviour of actors that we would like our model to answer includes preferences, one step ahead actions, intentions and future visit state (via successor representations). To reduce the notation load, we will drop the \(i\) notion if there is no confusion. In this paper, we use the observer and the ToM agent interchangeably; the actor refers to the observed agent.

3 METHOD

3.1 Trait-based Theory of Mind Architecture

In this section, we introduce the architecture of Trait-based Theory of Mind (Trait-ToM) model for the observer. There are three modules: (1) Prediction Model, (2) Trait Model, and (3) Mental Model. In this architecture, the Trait Model captures the long-term trait of actors in the past trajectories while the Mental Model represents the recent behaviours in the current trajectory. The outputs of the two modules will be used by the prediction module through our proposed fast weight mechanism. In the following, we detail the operation of each module. The architecture is shown in Figure 1.

Prediction Model. Let \(s\) be the current environmental state and \(e_{\text{mental}}\) be the current estimated mental state vector of an actor.
Based on this information and the past behaviours, we wish to predict the four outputs of the actor: preference, intention, action, and successor representation. The prediction network will generate an output vector $\mathbf{e}_{\text{pred}}$ as follows:

$$h = \sigma \left( \mathbf{W}_l f_{\text{pred}}(s, \mathbf{e}_{\text{mental}}) + \mathbf{b}_l \right), \quad (1)$$

$$\mathbf{e}_{\text{pred}} = \mathbf{W}_h h + \mathbf{b}_2, \quad (2)$$

for some feature extractor $f_{\text{pred}}(\cdot)$ which is a neural network, activation function $\sigma(\cdot)$, and weights $\mathbf{W}_l$, $\mathbf{b}_l$, $\mathbf{W}_h$, and $\mathbf{b}_2$. The mental state $\mathbf{e}_{\text{mental}}$ is generated from the mental model in Eq. (6). The output vector $\mathbf{e}_{\text{pred}}$ then serves as input for four prediction heads, which are goal, intention, action, and successor representation.

In Eqs. (1,2), the key to our formulation is that the fast weights $\{\mathbf{W}_l, \mathbf{b}_l : l \in \{1, 2\}\}$ represent the **individual characteristics** or the **character traits**, which are functions of the past behaviours. Unlike ToMnet, which uses a vector to represent the trait as the input to the prediction net, our fast-weight traits are higher in capacity and directly modulate the function of the prediction net through multiplicative mechanisms. These weights are computed by our trait model, which we present next in Eqs. (4,5). In other words, the traits modulate the prediction path from the current mental and environmental states to the outcomes.

**Trait Model.** The trait network takes past trajectories of an actor in $N_{\text{past}}$ environment $\{\mathbf{t}_j\}_{j=1}^{N_{\text{past}}}$ as inputs and generates the trait vector of the actor. For each past trajectory $j$, it maintains a dynamic state vector at each time step $t$ by a long short-term memory network [19] as follows:

$$h_j^{(t)} = \text{LSTM}_{\text{trait}} \left( \mathbf{x}_j^{(t)}, h_j^{(t-1)} \right),$$

where $\mathbf{x}_j^{(t)}$ is the features extracted from the state-action pair using a neural network, i.e., $\mathbf{x}_j^{(t)} = f_{\text{trait}}(s_j^{(t)}, a_j^{(t)})$. The trait embedding vector is computed by averaging over all the past trajectories:

$$\mathbf{e}_{\text{char}} = \frac{1}{N_{\text{past}}} \sum_{j=1}^{N_{\text{past}}} \text{ReLU} \left( \text{Linear} \left( h_j^{(T_j)} \right) \right). \quad (3)$$

To directly modulate the prediction path and create the multiplicative interaction between past behaviours and the mental embedding, we use a hypernetwork which takes $\mathbf{e}_{\text{char}}$ as input to generate the weights $\mathbf{W}_l$ and biases $\mathbf{b}_l$, $l \in \{1, 2\}$, of the prediction network in Eqs. (1,2):

$$\mathbf{W}_l = \sigma \left( \text{Linear} \left( [\mathbf{e}_{\text{char}}, \mathbf{e}_l] \right) \right), \quad (4)$$

$$\mathbf{b}_l = \sigma \left( \text{Linear} \left( [\mathbf{e}_{\text{char}}, \mathbf{e}_l] \right) \right), \quad l \in \{1, 2\} \quad (5)$$

where the one-hot vector $\mathbf{e}_l = \text{onehot}(l)$ is used as an additional input to indicate which layer $l$ to generate weights. The set of weights $\{\mathbf{W}_l, \mathbf{b}_l : l \in \{1, 2\}\}$ serves as the **representation of dynamic traits of the actor**, which changes whenever the behaviour history is updated. We hypothesise that the dynamic traits is critical to capture diverse behaviours of multi-agents. Each agent should triggers a signature fast weight to determine the prediction for its future behaviour.

**Mental Model.** The mental model reads the current trajectory and estimates the mental state using the following dynamics:

$$h_{\text{mental}}^{(t)} = \text{LSTM}_{\text{mental}} \left( \left[ h_{\text{mental}}^{(t)}, \mathbf{e}_{\text{char}} \right], h_{\text{mental}}^{(t-1)} \right),$$

where $\mathbf{e}_{\text{char}}$ is the trait embedding computed in Eq. (3), and $\mathbf{x}_{\text{mental}}^{(t)}$ denotes features extracted from the current state-action pair, i.e., $\mathbf{x}_{\text{mental}}^{(t)} = f_{\text{mental}}(s_{\text{current}}^{(t)}, a_{\text{current}}^{(t)})$. The initial hidden state is computed from the trait embedding $h_{\text{mental}}^{(0)} = \text{Linear}(\mathbf{e}_{\text{char}})$. The mental embedding vector is computed as:

$$\mathbf{e}_{\text{mental}} = \text{ReLU} \left( \text{Linear} \left( h_{\text{mental}}^{(t)} \right) \right). \quad (6)$$

It serves as an input for the prediction network in Eq. (1).

### 3.2 Loss functions

We feed $\mathbf{e}_{\text{pred}}$ computed from Eq. (2) to different heads corresponding to different targets that we want to predict. To train the Trait-based ToM, we use the following losses:

$$L = L_{\text{pref}} + L_{\text{intention}} + L_{\text{action}} + L_{\text{SR}}.$$ 

The four component losses are as follows:

**Preference Prediction.** Each actor has its own preference, e.g., a colour. The negative log-likelihood of the preference of the actor is therefore:

$$L_{\text{pref}} = \sum_{\text{pref}} - \log p \left( \text{pref} \mid \mathbf{e}_{\text{pred}} \right),$$

where $p_{\text{pref}}$ is modelled by a neural network that takes $\mathbf{e}_{\text{pred}}$ as its input.

**Intention Prediction.** In our setting, at each time step, the actor maintains a sub-plan (intention) such as going to a place or finding objects. We record the intention of actors at every time step, and compute the negative log-likelihood of the intention of the actor:

$$L_{\text{intention}} = \sum - \log \pi \left( a_t \mid \mathbf{e}_{\text{pred}} \right).$$

**Action Prediction.** We use the negative log-likelihood of the true action of the actor:

$$L_{\text{action}} = - \log \pi \left( a_t \mid \mathbf{e}_{\text{pred}} \right).$$

**Successor Representation Prediction.** We use an empirical successor representation [11] (SR) to compute the SR loss

$$L_{\text{SR}} = \sum_{\text{S}_{\text{SR}}} \sum_s - \text{SR}_{\text{SR}}(s) \log \hat{\text{SR}}_{\text{SR}}(s),$$

with

$$\text{SR}_{\text{SR}}(s) = \frac{1}{Z} \sum_{t=0}^{T-t} \gamma^t \delta(s_{t+t'}, s),$$

where $T$ is the episode length, $t$ is the time at which the successor representation is computed, $Z$ is a normalisation constant, $\gamma \in (0, 1)$ is the discount factor and $\delta(s_{t+t'}, s)$ is an indicator function, which returns 1 if $s_{t+t'} = s$ and 0 otherwise.
Figure 2: Performance of ToMnet and Trait-ToM after being trained on mixed (M) and sequential settings (S1 and S2). The y-axis shows the accuracies of action (top row) and intention prediction (bottom row) on the random-actor population (left column) and the hypo-actor population (right column), conditioned on the intention of the actors (find, goto, pickup).

Figure 3: The y-axis shows the action prediction accuracy (top row) and intention prediction accuracy (bottom row) of ToMnet and Trait-ToM on the random-actor population (left column) and the hypo-actor population (right column), conditioned on two groups of move actions: (1) change direction (turn-left and turn-right) and (2) move forward. Our Trait-ToM predicts better when the actor change its direction.

4 CASE STUDY: KEY-DOOR ENVIRONMENT

4.1 Experiment Settings

Environment. In this section, we conduct experiments on the Key-Door Environment using the gym-minigrid framework [9]. In this environment, there are two types of object (key, door) in four different colours {red, green, blue, yellow}. An actor has its own preference for the colour. An episode is terminated when the actor picks up the key and goes to the door in its preferred colour.

Goal-directed Actors. We construct the actors that have a consistent goal during one episode, e.g. picking up the key and going to the door in the preferred colour. We assume each actor has beliefs about the positions of all objects in the scene, as well as the ability to memorise all visited cells. The actor is able to detect its false belief and update its belief according to recent observations. At each time step, it has an intention to either find(), goto(), or pickup(). The actors have the belief-desires-intentions (BDI) architecture [7, 15] and dynamically switching between three plans find(), goto(), or pickup() can be considered as changing intentions.

Actors are different in the strategy they use to execute the intention find(object): (1) the random-actor that can only take random walks to find its preferred object; and (2) the wiser actor that maintains a hypothesis about the position of its preferred key and door. The latter type of actor tests its hypothesis by going to these positions and seeking for the object. If the actor cannot find the object there, it will make a new guess about the object position. For this reason, we call this actor a hypo-actor (shorthand for hypothesis testing actor). In this paper, we only explore the salient traits (e.g. smart - not smart or hypo - random) [34] and leave other traits, e.g. finding constraints on the value of information and risk aversion, for future work. Each actor partially observes a square area which indicates the field of view (FoV). In sum, there are 32 actors which are characterised by combinations of 4 preferences, 2 traits, and 4 FoVs, e.g. {red, green, blue, yellow}×{random, hypo}×{3×3, 5×5, 7×7, 9×9}.

4.2 Actions and Intention Predictions

In the first experiment, we analyse the behaviour of ToMnet [33] and our proposed Trait-ToM in predicting action and intention while learning from the mixed and sequential population. In the mixed setting (M), actors in one batch are i.i.d sampled from 32 actors. In sequential settings (S), every actor sees a new type of actors and never meets the previous actors again. The actors come sequentially in increasing field of view order of hypo-then random-actors (S1) or vice versa (S2). As a result, there are totally three experimental settings in this task. Amongst all, S1 and S2 are more realistic scenarios, e.g. the observer can only see
| Stream | Model  | Action (Full Observation) | Intention (Full Observation) | Action (Partial Observation) | Intention (Partial Observation) |
|--------|--------|--------------------------|----------------------------|----------------------------|--------------------------------|
| S1     | ToMnet | 37.80 (0.64)             | 53.21 (2.83)               | 54.26 (0.94)               | 53.43 (3.18)                   |
|        |        |                          |                            | 49.65 (0.96)               | 51.94 (2.71)                   |
|        | Trait-ToM | 39.46 (0.47)        | 55.64 (3.02)               | 54.37 (1.09)               | 56.45 (2.48)                   |
|        |        |                          |                            | 50.93 (1.10)               | 55.21 (2.43)                   |
| S2     | ToMnet | 32.48 (1.71)             | 36.60 (3.84)               | 50.29 (2.32)               | 46.59 (2.67)                   |
|        |        |                          |                            | 44.30 (2.45)               | 43.96 (3.09)                   |
|        | Trait-ToM | 38.22 (1.25)        | 47.26 (2.77)               | 52.85 (0.81)               | 54.60 (2.35)                   |
|        |        |                          |                            | 48.80 (1.20)               | 53.32 (2.30)                   |
| M      | ToMnet | 39.30 (0.34)             | 88.14 (1.05)               | 59.55 (1.46)               | 75.92 (2.35)                   |
|        |        |                          |                            | 54.13 (1.23)               | 74.71 (1.83)                   |
|        | Trait-ToM | 40.84 (0.42)        | 90.95 (0.57)               | 60.53 (2.15)               | 82.10 (1.52)                   |
|        |        |                          |                            | 55.52 (1.71)               | 79.85 (1.11)                   |

Table 1: The accuracy of ToMnet and Trait-ToM on predicting action and intention of (the 1st and 2nd column) random-actors, hypo-actors in fully observable environments, (the 3rd column) hypo-actors in environments that have unobserved obstacles to the observer (partial observation). Each cell contains the mean (std.) over predictions of 6 runs.

Figure 4: Knowledge transfer in ToMnet and Trait-ToM during the process of learning from different sequences of actors, assessed by the positive forward transfer ability (FWT) on predicting intention. Our Trait-ToM can learn patterns that are useful to predict intention of different types of actors from the beginning of each sequence (higher FWT).

one type of actor at a time. Here, the observer can only learn to predict behaviours of each type of actor for $T_{stream} = 30,000$ iterations. The observer can see a batch of $B = 16$ actors with the same FoV and trait at each iteration. We assume that during the training process, the actor will explicitly provide its intention (find, goto, or pickup) to the observer as training signals for the observer to predict its preferences, actions and intentions. The actor reveals its individual characteristics such as trait and FoV via its past behaviours ($N_{past} = 3$).

Results. Fig. 2 shows the performance of ToMnet and Trait-ToM on described tasks. To make a fair comparison between the two methods, we designed two networks with roughly a similar numbers of parameters (Trait-ToM has slightly less parameters than ToMnet and they are different in the prediction networks). For each architecture, we trained and reported the results over predictions of 6 runs. In general, our trait-based ToM are able to predict the behaviour of the actors better than ToMnet in all settings (M, S1 and S2). Both can predict precisely the preference of actors since it is shown clearly in the past trajectories. Training observers on the mixed population helps the models predict actor’s intention find() much better.

Fig. 3 shows the action prediction and intention prediction of ToMnet and Trait-ToM conditioned on move actions. There are two groups of move actions: (1) change direction, which includes turn-left and turn-right; (2) move forward. While both models can predict equally well when the actors move forward, our Trait-ToM can predict better when the actors will change their direction. Note that the moving forward action can be easily predicted by looking at the current trajectory, however, the change direction action depends on the individual characteristics such as trait and field of view of the actors. This illustrates that ToMnet heavily relies on the information of the current trajectory, whereas our model uses the information revealed during past trajectories.

Table 1 summarises the action and intention prediction accuracy of ToMnet and Trait-ToM. We also evaluate both observers in predicting the behaviour of hypo-actors, when both of them can only partially observe the environment. These environments contain obstacles that are only observed by the actors. As shown in Table 1, the trait-based observer outperforms the ToMnet in predicting the intention of actors. Especially, our Trait-ToM, with the fast weight mechanism, outperforms ToMnet by a larger margin when the actor streams are more realistic (S1 and S2).

To further understand the effect of the fast weight mechanism on the learning process, we measure the knowledge transfer during this learning process from two sequential settings (S1 and S2). Each stream will provide for the training process a sequence $(A_i)_{i=1...M}$ of $M$ goal-directed actors which are different in their fields of view.
Figure 5: Dynamic intentions of hypo-actor in key-door environment with swap event (bottom) and no swap event (top). The actor has a 3x3 field of view and prefers red colour. After seeing the red door, the actor finds and collects the red key. The actor then comes back to red door position that she last saw and believes the red door is still there. When there is no swap event, the actor successfully reaches the door (top). When there is a swap event, the actor realises that the red door is not at the previous place, but a yellow door instead. She changes her belief then tries to find and reach the red door (bottom).

Intuitively, $FWT$ [24] indicates how learning to predict the behaviour of new actors affects the performance on future unseen actors. Fig. 4 shows that our Trait-ToM can learn useful knowledge to transfer and predict the intention of actors sooner than ToMnet, as shown by a higher $FWT$.

4.3 Direct Assessment of False-Belief Understanding

The false belief in human can be assessed by various methods [6]. A common method to evaluate the computational theory of mind [28, 33] is by constructing experiments of Sally-Anne Test - a classic false-belief task [5, 42]. In this task, the subject observes a scene in which there are two dolls named by Sally and Anne. Sally first puts her toy into the basket then goes out. While Sally is outside, Anne takes the toy from Sally’s basket and put in Anne’s box. The observer will be asked where will Sally finds the toy when she comes back. Here we set up the key-door scenario with a swap event. Fig. 5 illustrates the trajectories of an actor with a 3x3 field of view who prefers red colour in the key-door environment with swap or no swap event. First, the actor looks for the red door. After seeing it, the actor finds and collects the red key. The actor then comes back to the red door position and supposes it is still there. When there is no swap event, the actor successfully reaches the red door. However, when there is a swap event, the actor sees a yellow door instead and realises that the red door has been swapped. She changes her belief then goes to find the red door.

In this experiment, both ToMnet and Trait-ToM are trained on the mixed population of hypo-actors and random-actors. The environment in which actors operate may contain a swap event. In addition to the preference, intention, and action prediction, the models are queried successor representations to make a prediction about the long-term behaviours of the actors. The ability to predict the difference in long-term behaviours between the swap and non-swap events indicates the understanding of the false belief.

Fig. 6 shows the Jensen–Shannon divergence between successor representations when the actor picks up the key in swap vs. no swap situation (left most) of the hypo-actor, (middle) predicted by ToMnet, and (right most) predicted by Trait-ToM. The x-axis is the distance from the preferred key to the target door. Statistical analysis is shown in Table 2.
Figure 7: Visualisation of weights of the prediction model which is generated by the hypernetworks, projected into 2D using t-SNE. The colours indicate the preferences (left), the traits (middle), and the fields of view (right) of the actors. While the preferences learning is supervised, the trait and field of view are learnt in unsupervised manner. With the 9×9 field of view, both hypo- or random-actors have similar behaviours, therefore, these clusters are close (purple clusters at the right figure).

purple line graph in Fig. 6 (left)) does not change much compared to actors with other fields of view.

We calculate the Pearson correlation coefficients between the ground-truth and the prediction of both models, shown in Table 2. Both methods can predict that there are differences in successor representations of the hypo-actor between the swap and no swap environments, \( \tau_{\mathrm{ToMnet}} = 0.929044 \pm 0.00 > \tau_{\mathrm{Trait-ToM}} = 0.935927 \pm 0.00 > r_{0.05}(700) = 0.074004 \) and \( \tau_{\mathrm{Trait-ToM}} = 0.911540 \pm 0.00 = 0.074004 \) for predicting actors with 3×3 field of view. To test whether the predictions of Trait-ToM are significantly better than those of ToMnet, we use Steiger method. This yields \( t = 2.12 \pm 3.42e-2 > t_{0.05}(700) = 1.64 \), hence the null hypothesis is rejected. We obtained similar conclusions about the prediction of Trait-ToM and ToMnet on actors with other FoVs (Table 2).

**Visualisation.** We project the weight space of Trait-ToM into 2D using t-SNE to have a better visual understanding our networks, c.f. Fig. 7. There are clear clusters of the preferences (left) confirming that this information is explicitly coded in the training data. The Trait-ToM produces different clusters given different traits (middle) and FoVs of the actors (right) despite the fact that we do not train Trait-ToM using this information. Especially, there is a smooth transition between clusters of actors with different FoVs. We notice that random-actors and hypo-actors with the furthest field of view (9×9 in this setting behave similarly; thus, the generated weights of our hypernetworks for these types of actors form nearby clusters.

### 4.4 Indirect Assessment of False-Belief Understanding

Developmental psychologists indirectly assess the ability to understand false belief by three experimental settings: (1) Violation of expectation (VoE) [30]; (2) Anticipatory looking [10]; and (3) Active helping [8, 23]. These settings have inspired research in AI to construct benchmarks to test the ToM models, e.g. [13, 38] uses VoE. We choose to indirectly assess our observer by the ability to help other agents, i.e., the active helping setting which the belief attribution will appear within the context of intention attribution. To pass this test, e.g. deliver proper helping behaviour, children need to know that (1) others have goals, and at the same time (2) others have false beliefs and thus may fail to achieve goals. Inspired by this experiment, we implement a scenario in which an actor has a false belief and an assistant needs to help with her (hidden) goal. The closest setting to ours is [32]. However, there the helper in that work only watches demonstrations to infer goals, and their theory of mind model does not make any online prediction about the behaviour of others. In our scenarios, we are able to investigate action-based usages of theory of mind skills.

In this scenario, there are two agents: (1) the assistant (with or without theory of mind) tries to assist, and (2) the actor who pursues its own goal. The assistant received full observations in the factored representation from each environment and (optionally) predictions from the theory of mind model. It can choose to give assistance directly in action forms (the same as the actions which the actor can take in the environment) or not to give any assistance and let the actor act on its own. Note that in training RL agents, we assume any assistance is costly, which means the agent needs to learn to take efficient actions. The intention prediction is important in determining whether to assist or not. A more realistic scenario in which there are more than one agent acting...
on the same environment and the helper needs to select who to help is left for future work. We especially highlight here the scenario in which the actor can hold the false belief about the position of the door. We first let the actor know the initial position of the preferred door then create a swap event right after the actor collects the key. The assistant needs to understand the perspective of the actor to provide assistance. In addition, we consider an obedient actor which would follow any assistance if given and act on its own strategy otherwise to achieve its goal.

**Algorithm 1:** The Procedure Help Policy

| Input | The theory of mind model ToM(·) $\pi^*(a_t|s_t, g)$ the optimal policy |
|-------|--------------------------------------------------------------------------|
| Output| The assistance given to the actor |
| 1     | Predict the action of the actor $a_t, g \leftarrow \text{ToM}(s_t)$; |
| 2     | Compute the optimal action $\hat{a}_t \leftarrow \pi^*(a_t|s_t, g)$; |
| 3     | if $\hat{a}_t = a_t$ then |
| 4     | return no assistance; |
| 5     | else |
| 6     | return $\hat{a}_t$; |
| 7     | end |

We construct two types of ToM-augmented help policy: (1) Procedure Policy; and (2) Reinforcement Learner. The algorithm of the procedure policy is shown in Algorithm 1. To implement the procedure policy, we need to pre-train the goal-conditioned policies that can make near-optimal decisions based on full observations of the environment $\pi^*(a_t|s_t, g)$. The procedure policy will give the near-optimal actions which is computed by $a_t^* = \pi^*(a_t|s_t, g)$ as an assistance to the actor if the optimal action is different from the predicted action of the ToM model. Our intuition is that if the action of the actor is optimal, then it does not need to be assisted. The constructions of reinforcement learner is shown in Fig. 8. Since the assistant can have access to full observations of the scene, we do not need to use a memory mechanism such as one in LSTM or even external memory to implement the policy. The reinforcement learning (RL) agent assists the actor based on goal, action, and intention predictions of the theory of mind model. We simply concatenate these outputs of the theory of mind models to create the feature vector of states observed from the environment. All information is given to two heads, the actor and the critic, in order to predict the action and the expected value, respectively. We trained the RL agents with this actor-critic structure by Proximal Policy Optimisation (PPO) algorithm [37].

Fig. 9 compares the performance of ToM-augmented policies in the helping task. For each ToM-augmented agent, we report the mean and standard deviation measures over 6 runs. Since TraitToM can give better prediction for both types of policies, it can help agents to achieve higher success rates than ToMnet helped agents. Although both ToMnet and Trait-ToM can assist actors achieve its task more frequent than acting on their own (all bars higher than the horizontal black dot line in the most left figure), Trait-ToM can help actors to complete tasks faster, i.e. the average episode length is smaller (the middle figure). Comparing two groups of assistants, the reinforcement learners perform better than the procedure policies on this task, both in term of success rates and times to completion. Interestingly, although the reinforcement learning agent with Trait-ToM does not need to give as much assistance as ToMnet, it still maintains higher success rates (the far left figure) and lower time to achieve the task (the far right figure). This highlights the importance of understanding false belief in helping other agents where accurate false-belief understanding assists better and is more efficient.

5 CONCLUSIONS

We have proposed a new Trait-based Theory of Mind (Trait-ToM) model to equip social observers with the ability to infer the mental states and goals of other actors through observing their past and current behaviours. Central to our model is the idea that stable character traits hold the key prior information that influences the transient mental states. We hypothesise that such an influence has a multiplicative nature – traits may modulate the prediction path from the mental states to the future behaviours. We realised these ideas through the concept of ‘fast weights’, in that the weights of the prediction network are determined by the latent traits, which are functions of the past behaviours, forming a hypernet architecture for Trait-ToM. We designed and conducted a suite of experiments over a key-door environment, in which actors have varied preference and intention traits. The results showed that the multiplicative interactions between the past and the present help make
more accurate future predictions, especially when trained in varying settings such as a mixed or sequential population. Trait-ToM based assistant can also achieve a better performance in providing helping behaviours, known as indirect assessment of false-belief understanding.

REFERENCES

[1] Daniel L. Ames, Susan T. Fiske, and Alexander T. Todorov. 2011. Impression Formation: A Focus on Others' Intent. The Oxford Handbook of Social Neuroscience (2011), 419.

[2] Jimmy Ba, Geoffrey E. Hinton, Volodymyr Mnih, Joel Z. Leibo, and Catalin Ionescu. 2016. Using Fast Weights to Attend to the Recent Past. In Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain. 4331–4339.

[3] Chris Baker, Rebecca Saxe, and Joshua Tenenbaum. 2011. Bayesian theory of mind: Modeling joint belief-desire attribution. In Proceedings of the annual meeting of the cognitive science society. Vol. 33.

[4] Chris L. Baker, Julian Jarz-Etinger, Rebecca Saxe, and Joshua B Tenenbaum. 2017. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. Nature Human Behaviour 1, 4 (2017), 1–10.

[5] Simon Baron-Cohen, Alan M. Leslie, and Uta Frith. 1985. Does the autistic child have a “theory of mind”?: Cognition 21, 1 (1985), 37–46.

[6] Cindy Beaudoin, Elizabel Leblanc, Charlotte Gagner, and Miriam H Beauschamp. 2020. Systematic review and inventory of theory of mind measures for young children. Frontiers in psychology 10 (2020), 2905.

[7] Michael Bratman et al. 1987. Intention, plans, and practical reason. Vol. 10. Harvard University Press Cambridge, MA.

[8] David Buttelmann, Malinda Carpenter, and Michael Tomasello. 2009. Eighteen-month-old infants show false belief understanding in an active helping paradigm. Cognition 112, 2 (2009), 337–342.

[9] Maxime Chevalier Buivert, Lucas Willems, and Suman Pal. 2018. Minimalistic GridWorld Environment for OpenAI Gym. https://github.com/maximch/gym-minigrid.

[10] Wendy A. Clements and Josef Penner. 1994. Implicit understanding of belief. Cognitive development 9, 4 (1994), 377–395.

[11] Peter Dayan. 1993. Improving generalization for temporal difference learning: The successor representation. Neural Computation 5, 4 (1993), 613–624.

[12] Vittorio Gallese and Alvin Goldman. 1998. Mirror neurons and the simulation theory of mind-reading. Trends in cognitive sciences 2, 12 (1998), 493–501.

[13] Kanishk Gandhi, Galia Stojnic, Brenden M. Lake, and Moira R. Dillon. 2021. Baby Intuitions Benchmark (BIB): Discerning the goals, preferences, and actions of others. CoRR abs/2102.11938 (2021). https://arxiv.org/abs/2102.11938

[14] Christopher W. Grubb and Robert P. Goldman. 2009. A probabilistic plan recognition algorithm based on plan tree grammars. Artificial Intelligence 173, 11 (2009), 1101–1132.

[15] Michael P. Georgeff, Barney Pell, Martha E. Pollock, Milind Tambe, and Michael J. Wooldridge. 1998. The Desire-Desire-Intention Model of Agency: In Intelligent Agents V, Agent Theories, Architectures, and Languages, 5th International Workshop, ATAL ’98, Paris, France (Lecture Notes in Computer Science, Vol. 1555). Springer, 1–39.

[16] Alison Gopnik and Henry M Wellman. 1992. Why the Child’s Theory of Mind Really Is a Theory. Mind & Language 7, 1-2 (1992), 145–171.

[17] Robert M Gordon. 1986. Folk psychology as simulation. Mind & language

[18] David Ha, Andrew M. Dai, and Quoc V. Le. 2017. HyperNetworks. In Proceedings of the 35th Conference on Neural Information Processing Systems, ICLR 2017, Toulon, France. http://proceedings.mlr.press/v80/ha17a.html

[19] Reuth Mirsky, Sarah Keren, and Christopher Geib. 2021. Introduction to Symbolic Plan and Goal Recognition. Synthesis Lectures on Artificial Intelligence and Machine Learning 16, 1 (2021), 1–190.

[20] Pol Moreno, Edward Hughes, Kevin R. McKe, Bernardo Avila Pires, and Théophane Weber. 2021. Neural Recursi ve Belief States in Multi-Agent Reinforcement Learning. arXiv preprint arXiv:2102.02274 (2021).

[21] Thuy N. Nguyen and Cletolide Gonzalez. 2021. Theory of Mind From Observation in Cognitive Models and Humans. Topics in Cognitive Science (2021).

[22] Alain Gopnik and Renée Baillargeon. 2005. Do 15-month-old infants understand false beliefs? science 308, 5191 (2005), 255–258.

[23] David Premack and Guy Woodruff. 1978. Does the chimpanzee have a theory of mind? Behavioral and brain sciences 1, 4 (1978), 515–526.

[24] Xavier Puig, Tianmin Shu, Shuang Li, Zilin Wang, Joshua B Tenenbaum, Sanja Fidler, and Antonio Torralba. [n.d.]. Watch-and-help: A challenge for social perception and human-AI collaboration. In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austin, May 3-7, 2021.

[25] Neil C. Rabinowitz, Frank Perbet, H. Francis Song, Chiyuan Zhang, S. M. Ali Eslami, and Matthew Botvinick. 2018. Machine Theory of Mind. In Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmassan, Stockholm, Sweden, July 10-15, 2018 (Proceedings of Machine Learning Research, Vol. 80). PMLR, 4215–4224. http://proceedings.mlr.press/v80/rabinowitz18a.html

[26] Shirin Sohrabi, Anton V. Riabov, and Octavian Udrea. 2016. Plan Recognition with ProbabilisticRecursive Networks. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016. IJCAI/AAAI Press, 3258–3264. http://proceedings.mlr.press/v75/rabinowitz18a.html

[27] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. CoRR abs/1707.06347 (2017). http://arxiv.org/abs/1707.06347

[28] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. CoRR abs/1707.06347 (2017). http://arxiv.org/abs/1707.06347

[29] Shirin Sohrabi, Anton V. Riabov, and Octavian Udrea. 2016. Plan Recognition as Planning Reversal. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016. IJCAI/AAAI Press, 3258–3264.

[30] Ying Wen, Yaodong Yang, Rui Luo, Jun Wang, and Wei Pan. 2019. Probabilistic recursive reasoning for multi-agent reinforcement learning. arXiv preprint arXiv:1901.09070 (2019).

[31] Evan Westra. 2018. Character and theory of mind: An integrative approach. Philosophical Studies 175, 5 (2018), 1217–1241.

[32] Heinz Wimmer and Josef Penner. 1983. Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children’s understanding of deception. Cognition 13, 1 (1983), 103–128.