Machine Theory of Mind (Deep Mind)

Helmut Wahanik

Waterloo Hydrogeologic
Instituto Nacional de Matemática Pura e Aplicada, Rio de Janeiro - Brazil

IMPA – Rio de Janeiro

-Research Dynamical Systems, Differential Geometry, Applied Mathematics.

-2014 Fields Medal, Artur Avila, work in Dynamical Systems (Ten Martini Problem).

My work:

-Mathematical Physics - Fluid dynamics.

-Riemann problems - Numerical Shock Waves and Rarefactions waves in Gas Dynamics.

-Markov Chain Monte-Carlo methods (Seismic Tomography) – SLB- U. of Cambridge.

-Computational Geometry, U. of Calgary.

IMPA – Rio de Janeiro

-Research Dynamical Systems, Differential Geometry, Applied Mathematics.

-2014 Fields Medal, Artur Avila, work in Dynamical Systems (Ten Martini Problem).

My work:

-Mathematical Physics - Fluid dynamics.

-Riemann problems - Numerical Shock Waves and Rarefactions waves in Gas Dynamics.

-Markov Chain Monte-Carlo methods (Seismic Tomography) – SLB- U. of Cambridge.

-Computational Geometry, U. of Calgary.

Collaboration RJ-MCMC - University of Cambridge - UK (Schlumberger).

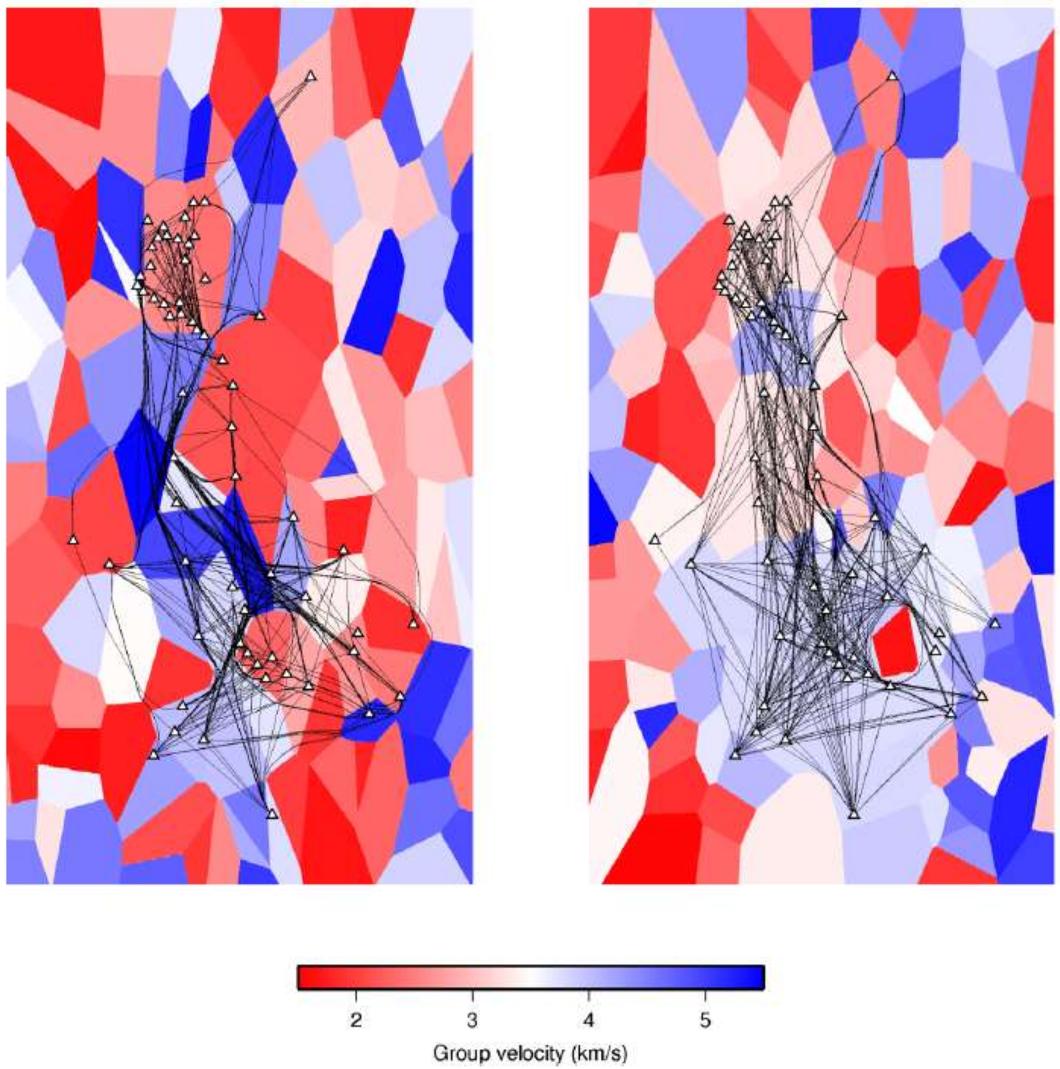
-Travel-times built through Greens function approach and Seismic Ambient Noise.

-Voronoi grids updated across the random walk.

-Minimize difference of theoretical and experimental travel-times.

-Dimension is also variable, and adjust to complexity of the data.

-Samples are accepted or rejected with a modified Metropolis-Hastings algorithm, guiding the samples towards regions of higher probability (e.g. Langevin MCMC MALA).



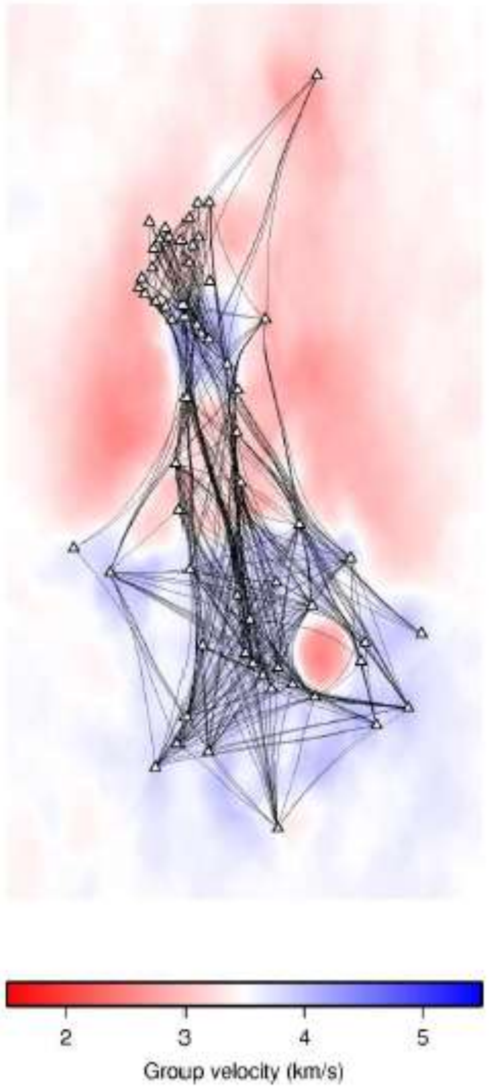
-The 3D point-wise probability distribution across all chains is the final posterior => solution to inverse problem.

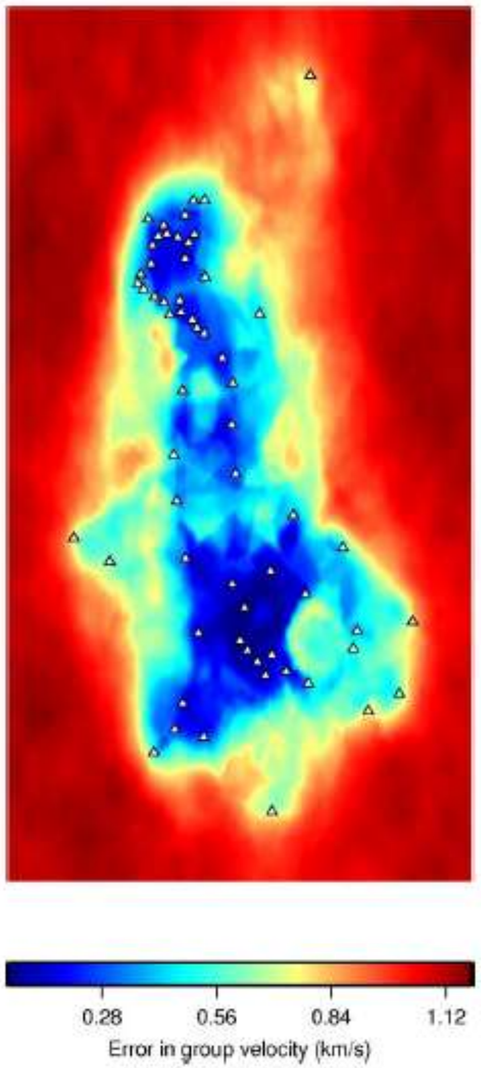
-The uncertainty of the solution can be measured by the spread of the samples.

-Fortran + OpenMPI + Qsub + SLB cluster.

-Parallelization on calculation of seismic travel-times => many seismometers.

-Mapping in GMT – Generic Mapping Tools.





-The 3D point-wise probability distribution across all chains is the final posterior => solution to inverse problem.

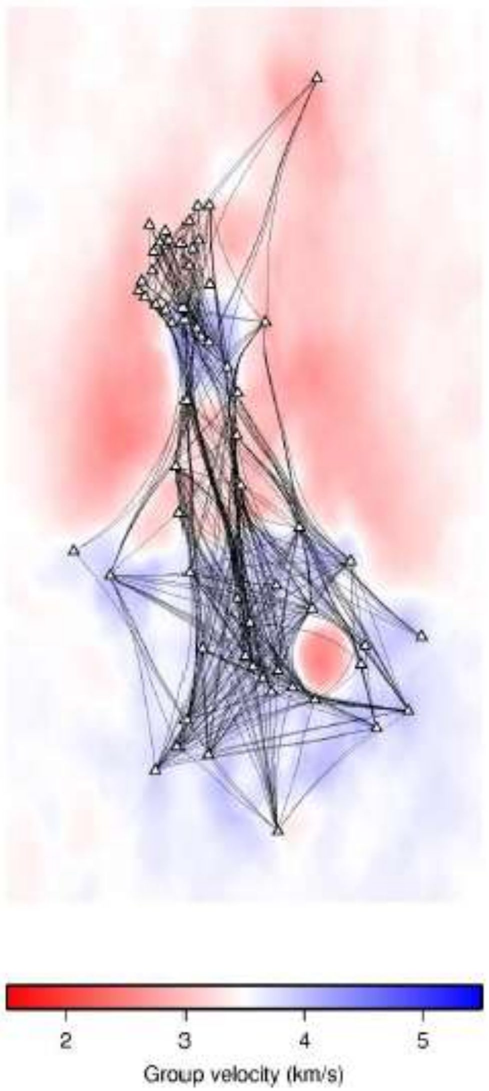
-The uncertainty of the solution can be measured by the spread of the samples.

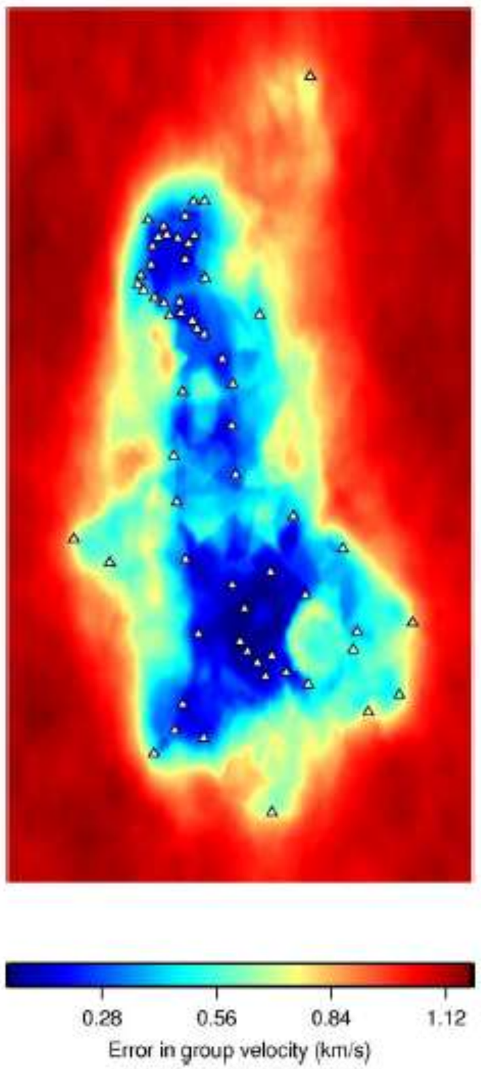
-Fortran + OpenMPI + Qsub + SLB cluster.

-Parallelization on calculation of seismic travel-times => many seismometers.

-Mapping in GMT – Generic Mapping Tools.

Could this be implemented in TensorFlow Probability?





ToM-Net – Theory of Mind Neural Network

Observer: Uses Meta-learning to predict behaviors of agents living in a Grid-World (models other agents).

Objective: To rapidly form predictions about new agents from limited data and behavioral traces.

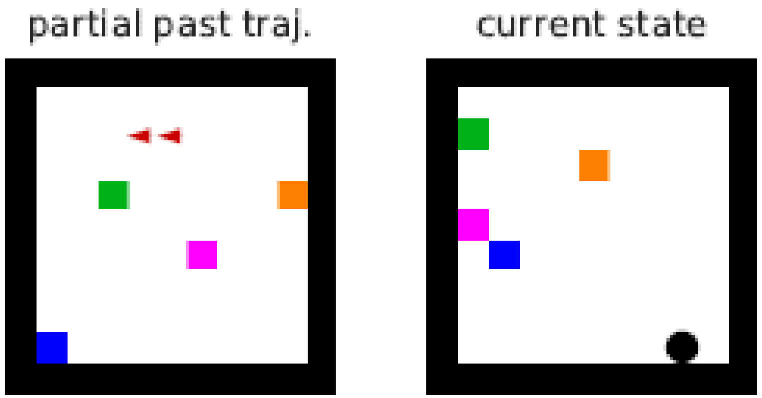
Players: Agents are themselves Deep Reinforcement Learning agents.

Important Feature:

To imitate cognitive predictive patterns of human mind.

-Passes "cognition" tests such as the Sally-Anne test.

Grid-world



Sally-Anne Test

-Developmental psychology test, for measuring a person's social cognitive intelligence: ability to recognize that others have false beliefs about the world.

-Measure of higher intelligence in primates:

3 year old child fails it.

4 year old passes it.

3 year old

Sound-proof
light-proof
scent-proof
barrier

Sally-Anne Test

-Developmental psychology test, for measuring a person's social cognitive intelligence: ability to recognize that others have false beliefs about the world.

-Measure of higher intelligence in primates:

3 year old child fails it.

4 year old passes it.

What's inside?

Sally-Anne Test

-Developmental psychology test, for measuring a person's social cognitive intelligence: ability to recognize that others have false beliefs about the world.

-Measure of higher intelligence in primates:

3 year old child fails it.

4 year old passes it.

Crayons

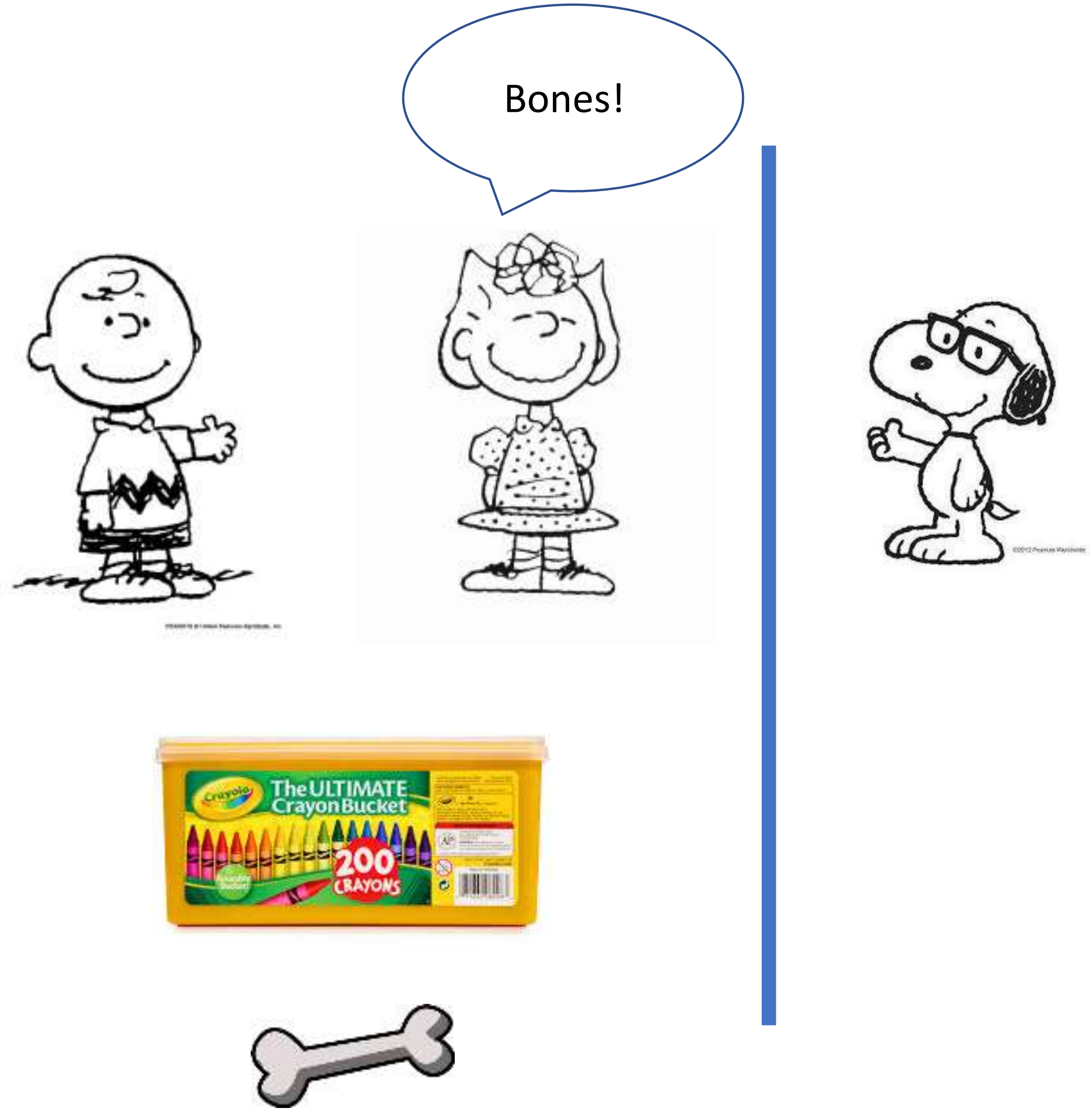
Sally-Anne Test

-Developmental psychology test, for measuring a person's social cognitive intelligence: ability to recognize that others have false beliefs about the world.

-Measure of higher intelligence in primates:

3 year old child fails it.

4 year old passes it.



Sally-Anne Test

-Developmental psychology test, for measuring a person's social cognitive intelligence: ability to recognize that others have false beliefs about the world.

-Measure of higher intelligence in primates:

3 year old child fails it.

4 year old passes it.

Remove
Snoopy-proof wall!

Sally-Anne Test

What does Snoopy think is inside?

-Developmental psychology test, for measuring a person's social cognitive intelligence: ability to recognize that others have false beliefs about the world.

-Measure of higher intelligence in primates:

3 year old child fails it.

4 year old passes it.

Sally-Anne Test

-Developmental psychology test, for measuring a person's social cognitive intelligence: ability to recognize that others have false beliefs about the world.

-Measure of higher intelligence in primates:

3 year old child fails it.

4 year old passes it.

Sally-Anne Test

INCORRECT

-Developmental psychology test, for measuring a person's social cognitive intelligence: ability to recognize that others have false beliefs about the world.

-Measure of higher intelligence in primates:

3 year old child fails it.

4 year old passes it.

Sally-Anne Test

Repeat first 3 steps
with 4 year old

What does Snoopy
think is inside?

-Developmental psychology test, for measuring a person's social cognitive intelligence: ability to recognize that others have false beliefs about the world.

-Measure of higher intelligence in primates:

3 year old child fails it.

4 year old passes it.

Sally-Anne Test

Crayons!

-Developmental psychology test, for measuring a person's social cognitive intelligence: ability to recognize that others have false beliefs about the world.

-Measure of higher intelligence in primates:

3 year old child fails it.

4 year old passes it.

Sally-Anne Test

-Developmental psychology test, for measuring a person's social cognitive intelligence: ability to recognize that others have false beliefs about the world.

-Measure of higher intelligence in primates:

3 year old child fails it.

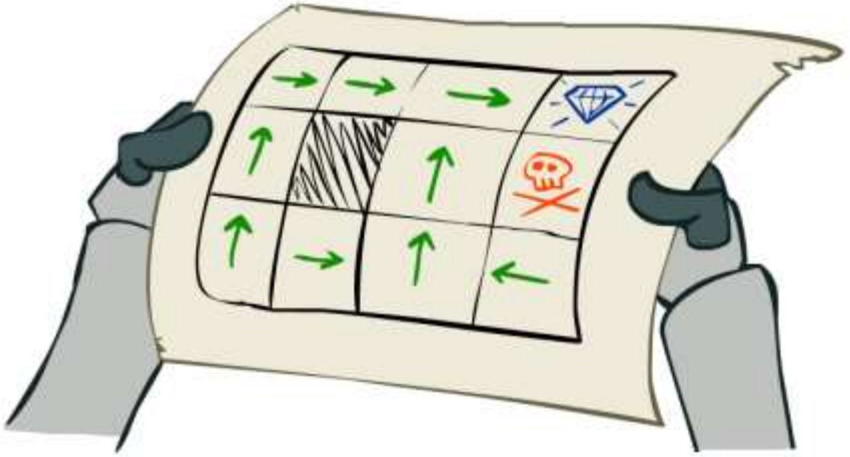
4 year old passes it.

Preliminaries: Markov Decision Process

MDP: Augmented Markov Chain.

(S, A, T, R, γ) such that:

- s states
- $a \coloneqq a(s)$ set of actions available at s.
- $T\left(s_{t+1} \mid s_{t,} a_t\right)$ prob transition if using action a_t at s_t
- $R_{a_t}(s_t, s_{t+1})$ reward given action a_t .
- $\gamma \in [0, 1]$ is a discount factor.

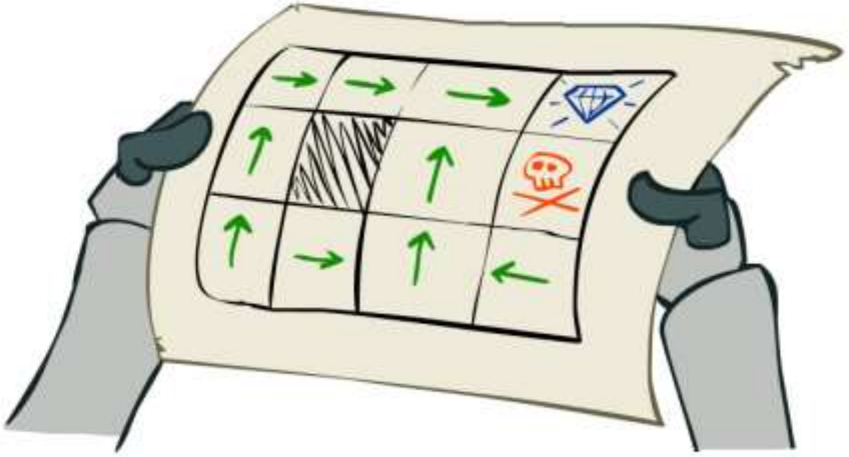


Preliminaries: Markov Decision Process

MDP: Augmented Markov Chain.

(S, A, T, R, γ) such that:

- s states
- $a := a(s)$ set of actions available at s.
- $T(s_{t+1} \mid s_{t,} a_t)$ prob transition if using action a_t at s_t
- $R_{a_t}(s_t, s_{t+1})$ reward given action a_t .
- $\gamma \in [0, 1]$ is a discount factor.



Objective: Find optimal choice (policy) π of actions at all states, maximizing the average discounted reward obtained when starting the chain at any state s.

Preliminaries: Markov Decision Process

Objective: We look for a policy $\pi: S \rightarrow A$ maximizing the discounted average rewards earned starting at state $S: V(s)$.

Preliminaries: Markov Decision Process

Objective: We look for a policy $\pi: S \rightarrow A$ maximizing the discounted average rewards earned starting at state $S: V(s)$.

Under policy π, the expected average reward is recursively defined through:

$$V^{\pi}(s) := R\big(s, \pi(s)\big) + \gamma \sum_{s'} T_{\pi(s)}(s, s')\, V^{\pi}(s')$$

Preliminaries: Markov Decision Process

Objective: We look for a policy $\pi : S \rightarrow A$ maximizing the discounted average rewards earned starting at state $S: V(s)$.

Under policy π, the expected average reward is recursively defined through:

$$V^{\pi}(s) := R\big(s, \pi(s)\big) + \gamma \sum_{s'} T_{\pi(s)}(s, s')\, V^{\pi}(s')$$

The optimal policy π^* is derived from the Bellman Optimality Equation:

$$V^{*}(s) \coloneqq max_a\{\ R(s, a) + \gamma \sum_{s'} T_a(s, s')\, V^{*}(s')\ \}$$

Preliminaries: Markov Decision Process

Objective: We look for a policy $\pi: S \rightarrow A$ maximizing the discounted average rewards earned starting at state $S: V(s)$.

Under policy π, the expected average reward is recursively defined through:

$$V^{\pi}(s) := R\big(s, \pi(s)\big) + \gamma \sum_{s'} T_{\pi(s)}(s, s')\, V^{\pi}(s')$$

The optimal policy π^* is derived from the Bellman Optimality Equation:

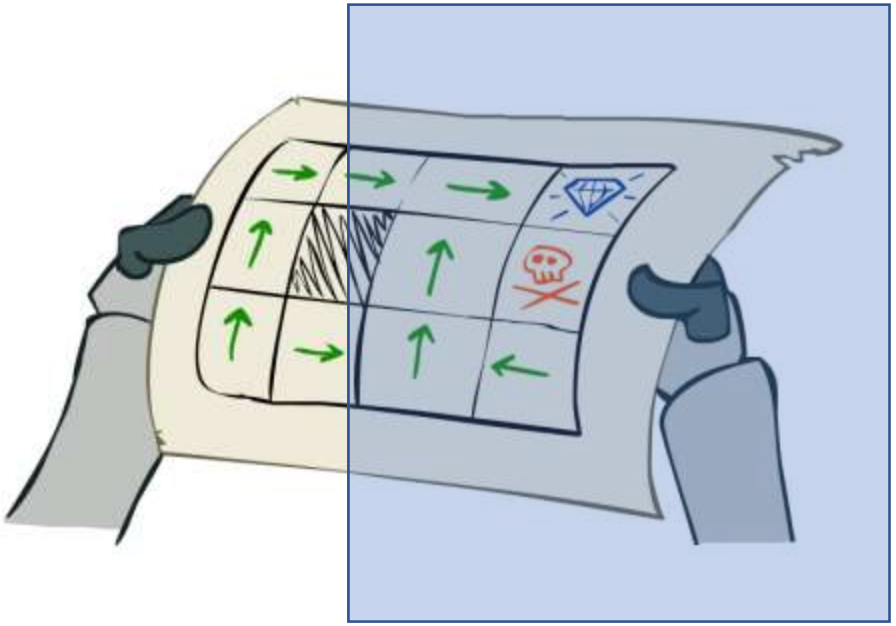
$$V^{*}(s) \coloneqq max_a\{\ R(s, a) + \gamma \sum_{s'} T_a(s, s')\, V^{*}(s')\ \}$$

Argument contraction + fixed point theorem => there exists a unique solution V^* to BOE.

Partially Observable Markov Decision Process

$(S, A, T, R, O, \omega, \gamma)$ such that:

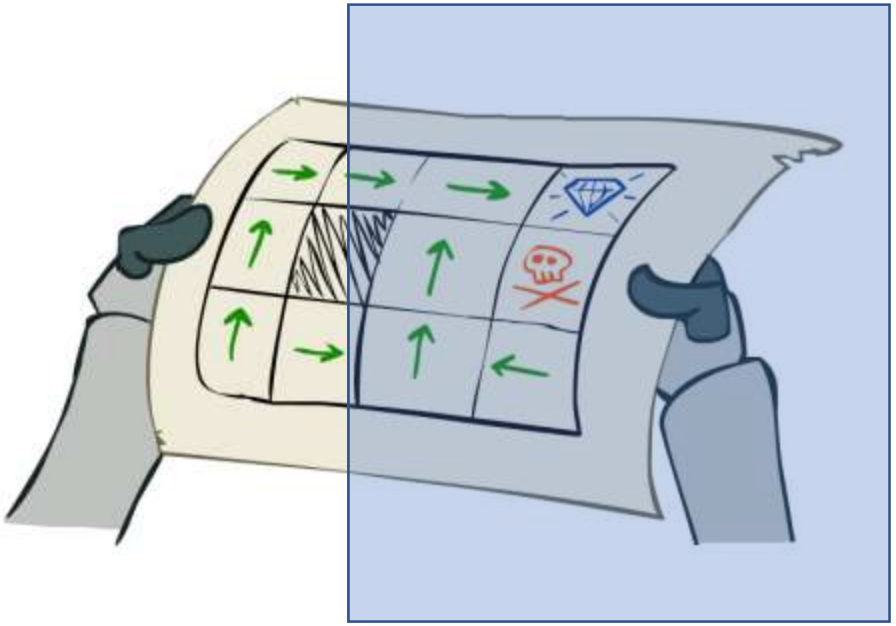
- O observations, o
- ω conditional probability of observations, w.



Partially Observable Markov Decision Process

$(S, A, T, R, O, \omega, \gamma)$ such that:

- O observations, o
- ω conditional probability of observations, w.

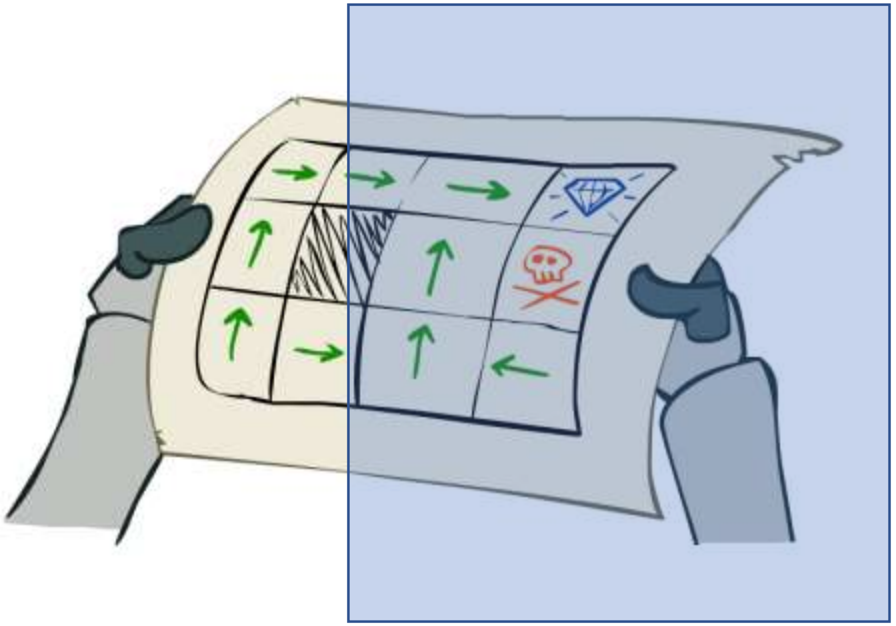


Time $t+1$, if $s \Rightarrow s'$ after a, we receive observation $o \in O$, with probability $w(o \mid s', a)$. Agent updates it's beliefs b about current state.

Partially Observable Markov Decision Process

$(S, A, T, R, O, \omega, \gamma)$ such that:

- O observations, o
- ω conditional probability of observations, w.



Time $t+1$, if $s \Rightarrow s'$ after a, we receive observation $o \in O$, with probability $w(o \mid s', a)$. Agent updates it's beliefs b about current state.

-The agent tries to infer the new state from observations & beliefs.

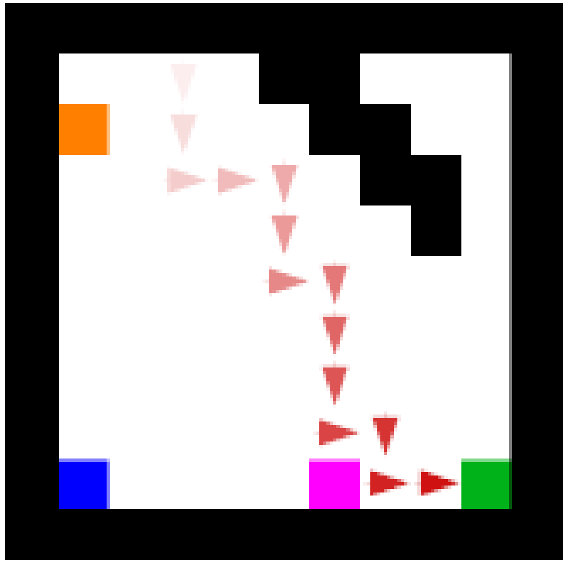
-POMDPS ⇒ MDPs observations equal true states, probability 1.

-For POMDPS the Meta-learning process is evident: agent must learn how to learn to read observations and how to update beliefs: parameters in the probability distributions.

The Machine Theory of Mind Architecture

Family of POMPDs $\boldsymbol{M} = \cup_k M_k$, Mazes (11x11), walls, 4 consumable objects.

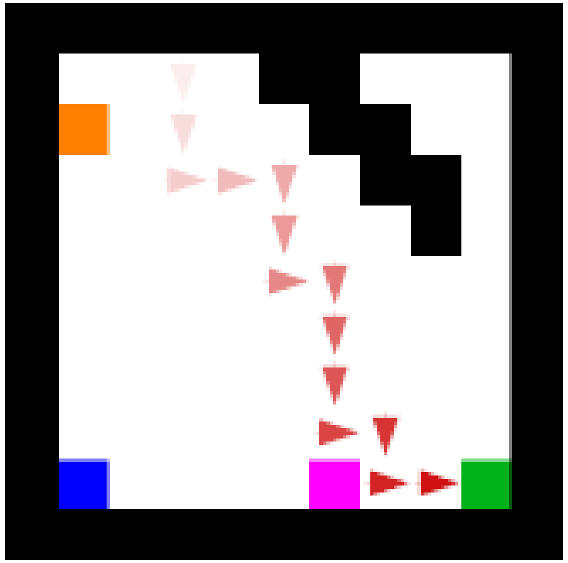
- (S_k, A_k, T_k)



The Machine Theory of Mind Architecture

Family of POMPDs $\boldsymbol{M} = \cup_j M_j$, Mazes (11x11), walls, 4 consumable objects.

- (S_k, A_k, T_k)



Agents:

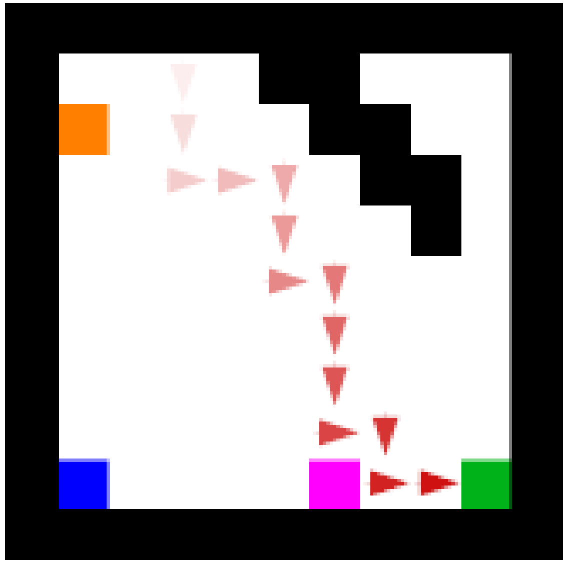
Rewards, discount factors, conditional observation functions, and policies are associated with $\boldsymbol{Agent\ i}$

- $(O_i, w_i, R_i, \gamma_i, \pi_i)$
- Policies might be stochastic, and non-optimal.

The Machine Theory of Mind Architecture

Family of POMPDs $\boldsymbol{M} = \bigcup_j M_j$, Mazes (11x11), walls, 4 consumable objects.

- (S_k, A_k, T_k)



Agents:

Rewards, discount factors, conditional observation functions, and policies are associated with $\boldsymbol{Agent\ i}$

- $(O_i, w_i, R_i, \gamma_i, \pi_i)$
- Policies might be stochastic, and non-optimal.

Observer ToMNet:

- State observation function: $w^{(obs)}: S \rightarrow O^{obs}$
- Action observation function $\alpha^{(obs)}: A \rightarrow A^{obs}$
- $w^{(obs)}(s) = s^{obs}$
- $\alpha^{(obs)}(a) = a^{obs}$

Observer's Architecture

Training:

Observes $\boldsymbol{Agent\ i}$, and a set of past trajectories:

$$\{\tau_{ij}\}_{j=1}^{N_{past}} \rightarrow \{\tau_{ij}{}^{(obs)}\}_{j=1}^{N_{past}}, \qquad \text{where} \qquad \tau_{ij}{}^{(obs)} = \left\{(s_t^{(obs)}, a_t^{(obs)})\right\}_{t=0}^{T}$$

Observer's Architecture

Training:

Observes $\boldsymbol{Agent\ i}$, and a set of past trajectories:

$$\{\tau_{ij}\}_{j=1}^{N_{past}} \to \{\tau_{ij}{}^{(obs)}\}_{j=1}^{N_{past}}, \qquad \text{where} \qquad \tau_{ij}{}^{(obs)} = \left\{(s_t^{(obs)}, a_t^{(obs)})\right\}_{t=0}^{T}$$

- Here $s_t^{(obs)}$ is a **tensor of size 11 x 11 x K.**
- K feature planes, such as walls, objects, agent.

Observer's Architecture

Training:

Observes $\boldsymbol{Agent\ i}$, and a set of past trajectories:

$$\{\tau_{ij}\}_{j=1}^{N_{past}} \rightarrow \{\tau_{ij}{}^{(obs)}\}_{j=1}^{N_{past}}, \qquad \text{where} \qquad \tau_{ij}{}^{(obs)} = \left\{(s_t^{(obs)}, a_t^{(obs)})\right\}_{t=0}^{T}$$

- Here $s_t^{(obs)}$ is a **tensor of size 11 x 11 x K.**
- K feature planes, such as walls, objects, agent.

- Also $a_t^{(obs)}$ is a dimension 5 logit, fully characterizing the action: **[** $\cdot$, $\downarrow$, $\rightarrow$, $\uparrow$, $\leftarrow$**]**
- The trajectory $\tau_{ij}{}^{(obs)}$ is a tensor is of size 11x11x (K + 5).

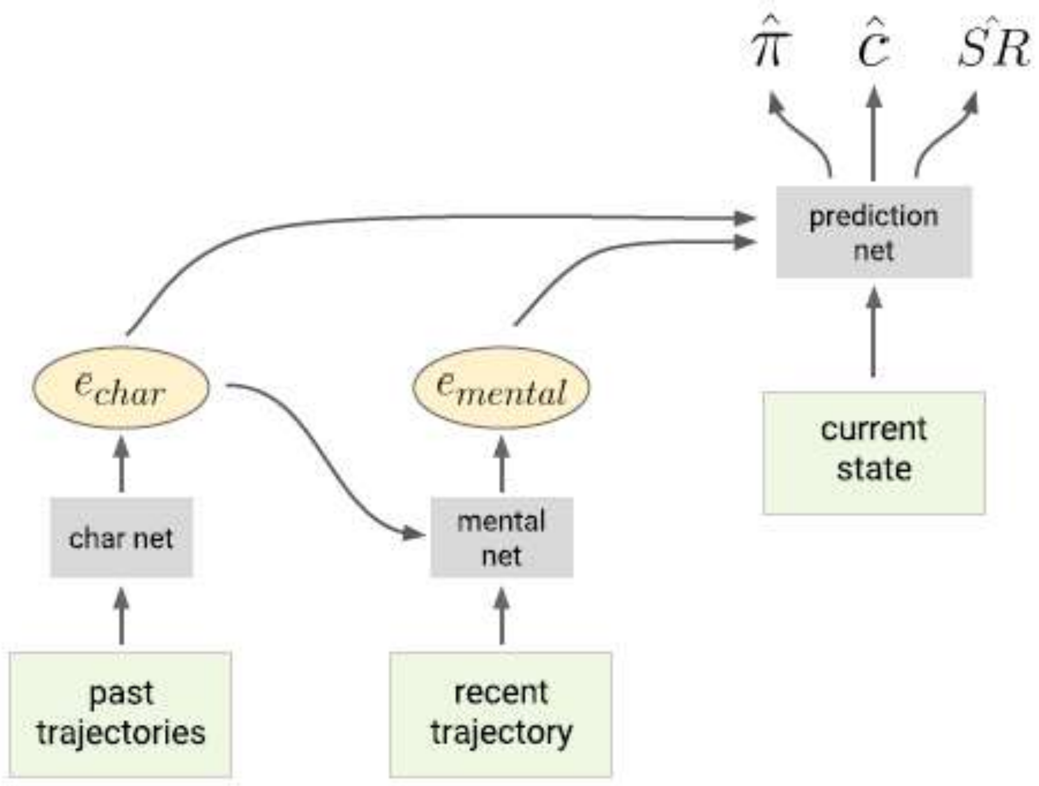
Observer's Neural Net

Character Net: Characterizes the past $\{\tau_{ij}{}^{(obs)}\}_{j=1}^{N_{past}}$

$$\tau_{ij} \xrightarrow{f_\theta} e_{char,ij} \text{ (2D Tensor)}$$

For all agents we add:

$$e_{char,i} = \sum_{j=1}^{N_{past}} e_{char,ij}$$



Observer's Neural Net

Character Net: Characterizes the past $\{\tau_{ij}{}^{(obs)}\}_{j=1}^{N_{past}}$

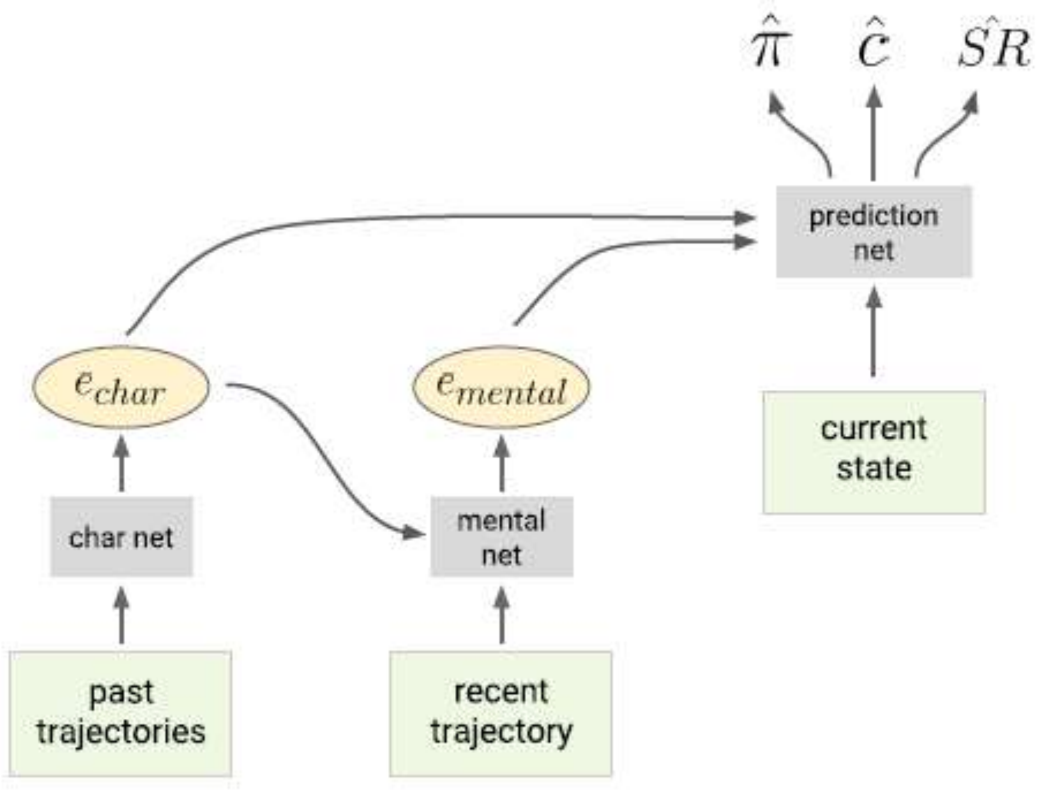
$$\tau_{ij} \xrightarrow{f_\theta} e_{char,ij} \quad \text{(2D Tensor)}$$

For all agents we add:

$$e_{char,i} = \sum_{j=1}^{N_{past}} e_{char,ij}$$

Mental Net: Mentalizes about the CURRENT EPISODE

$$[\tau_{ij}]_{0:t-1}, e_{char,i} \xrightarrow{g_\theta} e_{mental,i}$$



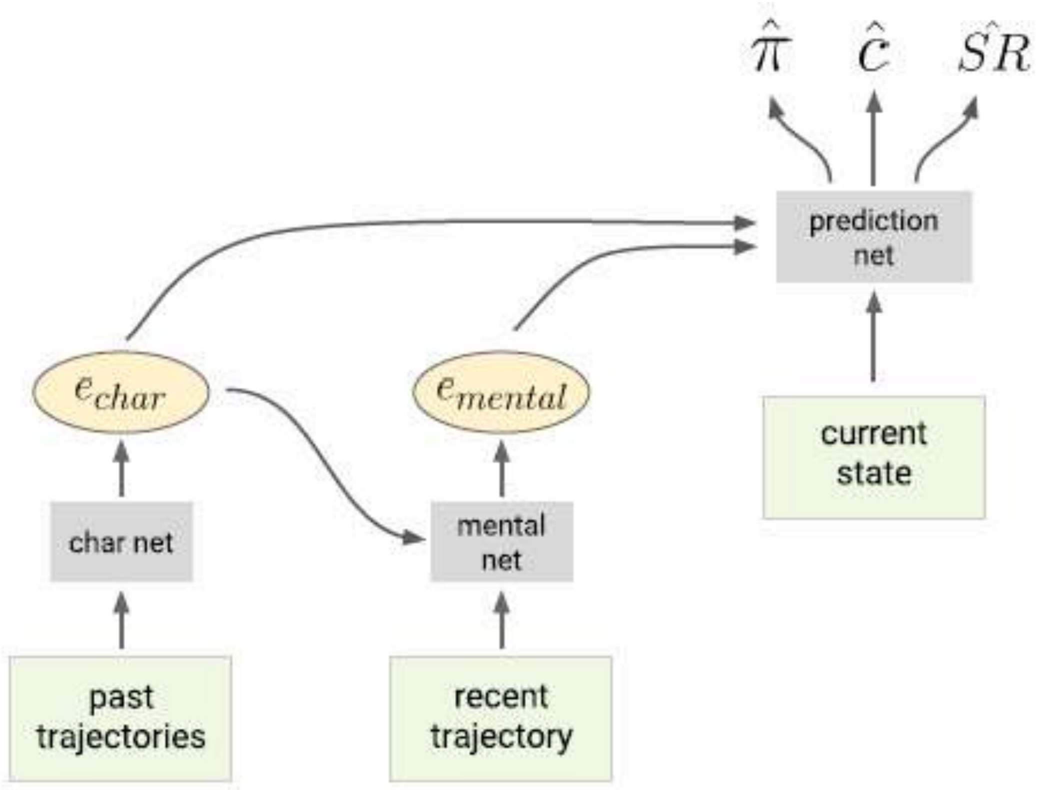
Observer's Neural Net

Character Net: Characterizes the past $\{\tau_{ij}{}^{(obs)}\}_{j=1}^{N_{past}}$

$$\tau_{ij} \xrightarrow{f_\theta} e_{char,ij} \text{ (2D Tensor)}$$

For all agents we add:

$$e_{char,i} = \sum_{j=1}^{N_{past}} e_{char,ij}$$

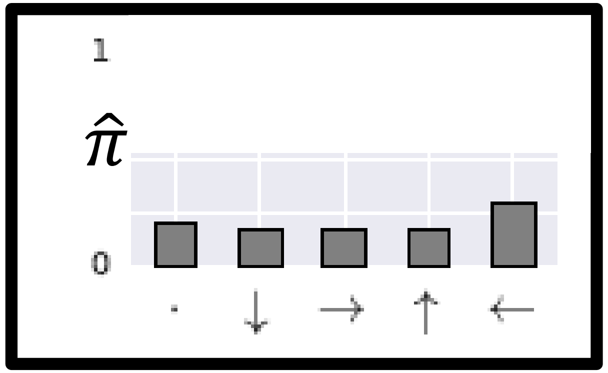


Mental Net: Mentalizes about the CURRENT EPISODE

$$[\tau_{ij}]_{0:t-1}, e_{char,i} \xrightarrow{g_\theta} e_{mental,i}$$

Prediction Net: Current state + Character + Mental to estimate:

- Predicted policy: $\hat{\pi}(\ \cdot \mid s_t^{(obs)}, e_{char}, e_{mental})$
- Probability of consuming an object $\hat{c}$



Experiments

Fully Random agents

- Species of agents.
- 5D stochastic policy vector $\pi_i(\cdot) \coloneqq \boldsymbol{\pi_i}$
- $\boldsymbol{\pi_i} \sim Dir(\alpha)$, Dirichlet distribution. Species can be written as $S(\alpha)$.
- For $\alpha \ll 1$, one-sided deterministic policies. $\alpha \sim 3 \Rightarrow$ uniform distribution.

Training

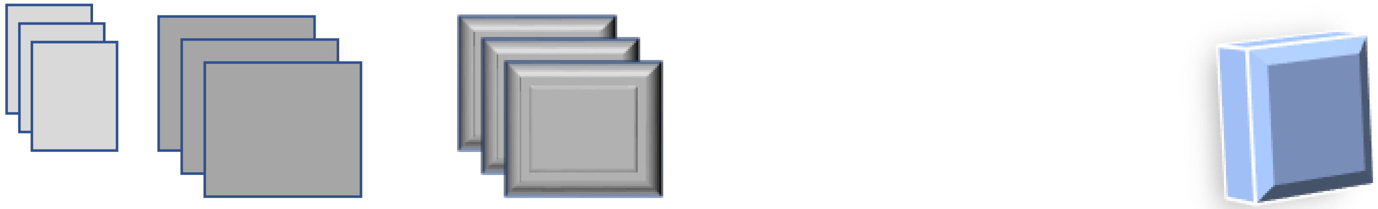
- Observe sample from species $S(\alpha)$, running on Grid-Worlds.
- A set of recent trajectories, with $N_{past} \sim U\{0, 10\}$.
- Length of trajectory = 1.
- Adam optimizer, $\delta = 10^{-4}$, 40K Minibatches of size 16.

Architecture

Fully Random agents:

- Character Net: $\boldsymbol{T}$ tensor for the trajectory, dim (11x11)x(K + 5)

$$\tau_{ij} \rightarrow \boldsymbol{T} \rightarrow 8\ \text{Convnet} \xrightarrow{ReLu} LSTM \xrightarrow{Av.\,Pooling+Fully\ Connected} \quad \xrightarrow{Embedding} e_{char,i}$$

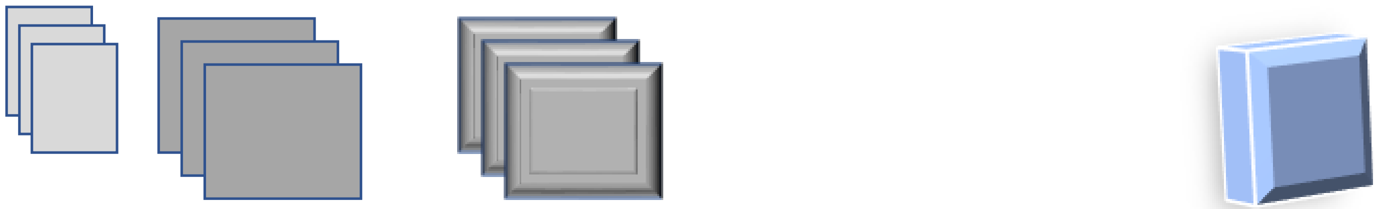


Architecture

Fully Random agents:

- Character Net: $\boldsymbol{T}$ tensor for the trajectory, dim (11x11)x(K + 5)

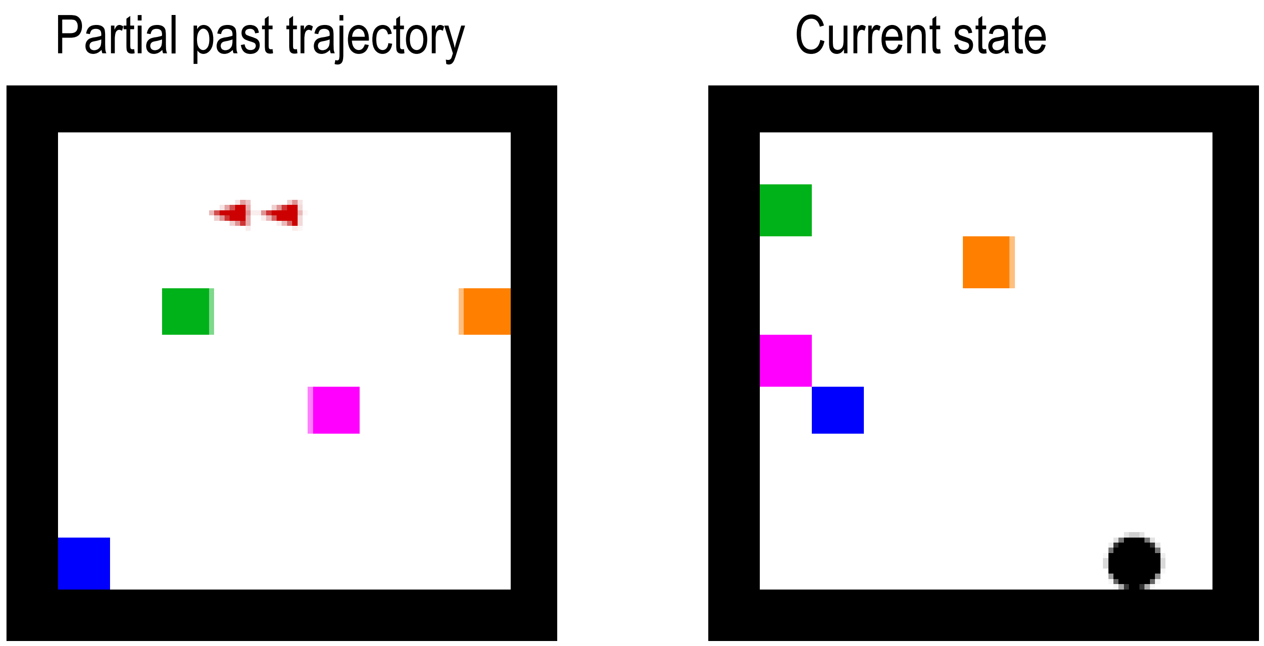
$$\tau_{ij} \rightarrow \boldsymbol{T} \rightarrow 8\text{ Convnet} \xrightarrow{ReLu} LSTM \xrightarrow{Av.\,Pooling+Fully\ Connected} \xrightarrow{Embedding} e_{char,i}$$

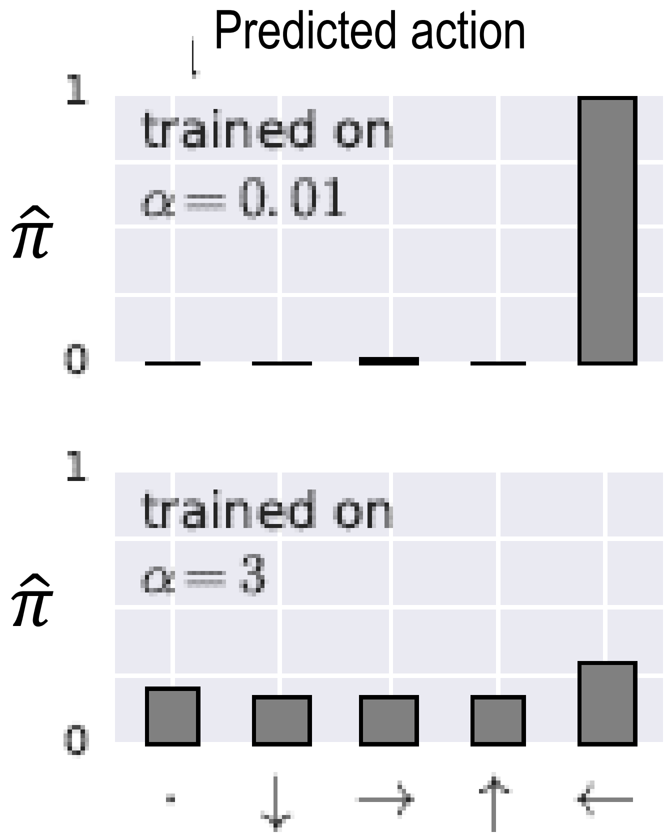


- Mental State: None.
- Prediction Net:

$$\tau_{ij}\,, e_{char,i} \rightarrow T \rightarrow 2\text{ Layer }32\text{ Convnet} \xrightarrow{Av.Pooling+Fully\ Connected} \text{Logits in } R^5 \xrightarrow{Softmax} \hat{\pi}$$

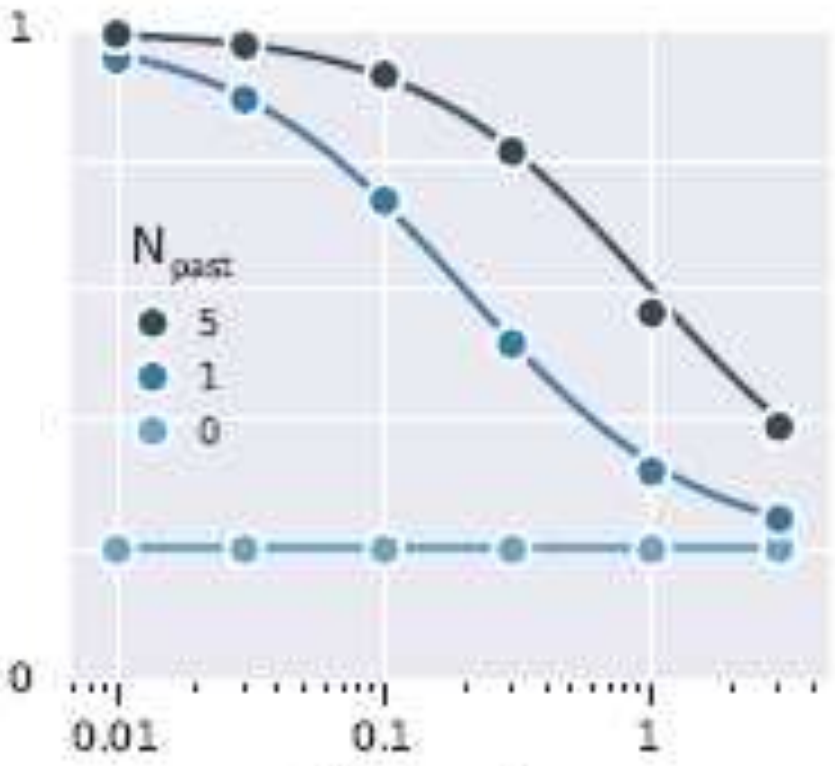
Random agent Training



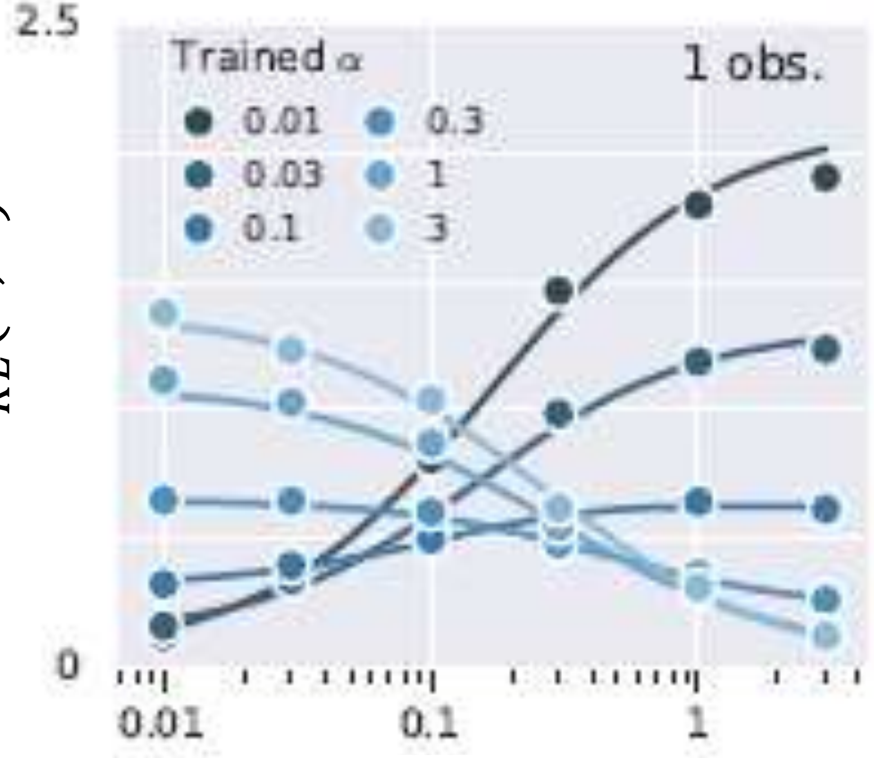


Random agent Training

Estimated prob. of performing an action



Trained α



$D_{KL}(\pi,\hat{\pi})$

Tested α

-ToMNet estimates increase with the number of past observations of that action!
-$D_{KL}(\pi,\hat{\pi})$ is the divergence between the true and estimated stochastic policies.

Inferring goal-directed behaviour

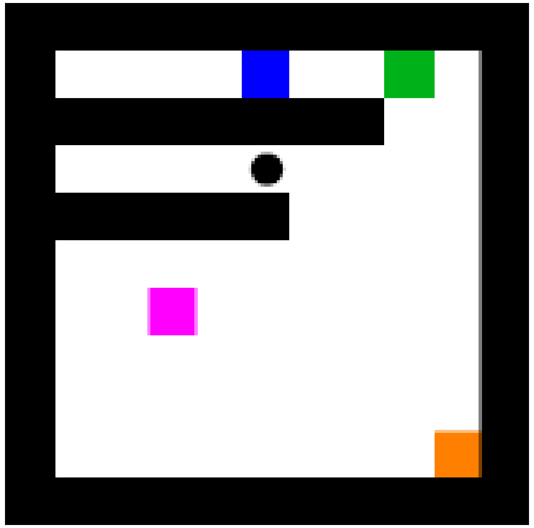
ToMNet learns to infer goals of reward seeking agents.

- 4 consumable objects.
- Agent A_i has a reward function: $r_{i,a} \in (0,1)$ when consuming an object.
- -0,01 for every move.
- Penalty of 0.05 for walking into walls.

- Agent finds optimal policy π_i^* through Bellman equation.

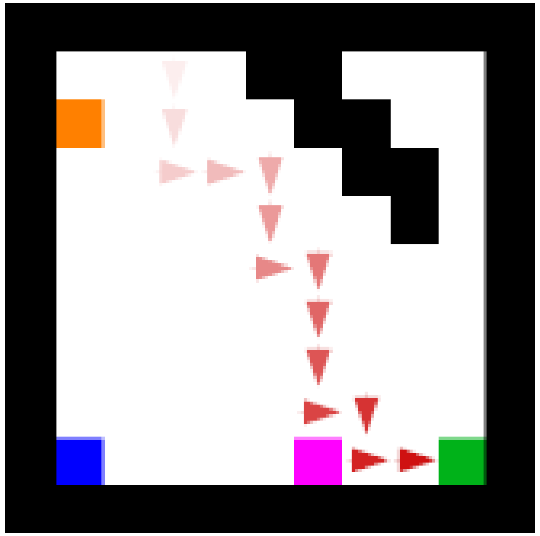
Training: ToMNet observes a single full trajectory of an agent acting on the Grid-World.

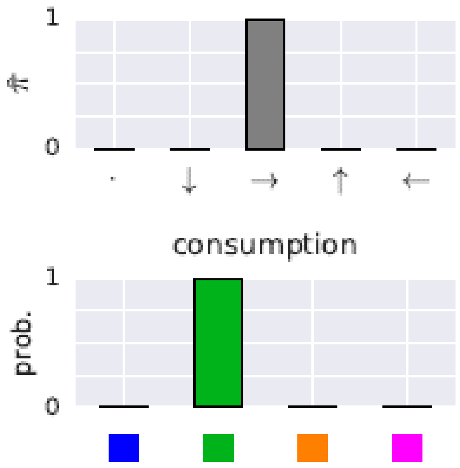
Inferring goal-directed behaviour



Current state

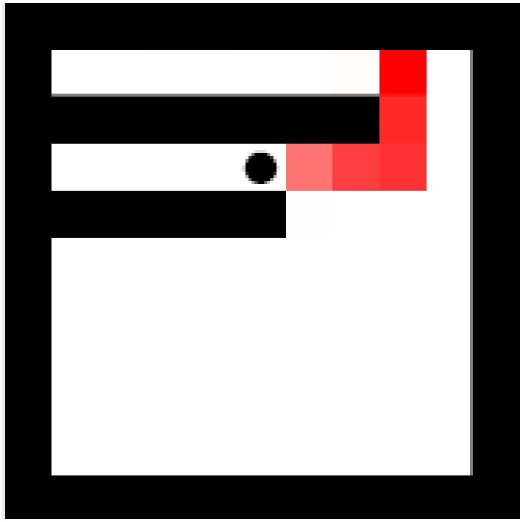
Observe single past MDP





ToMNet prediction of next action

Prediction of successive states



ToMNet vs Sally-Anne Test

ToMNet must pass the Sally-Anne test!

- Create POMDPs, agents 5 x 5 visibility window, where agents have false beliefs.
- We run random changes in the environment that are invisible to the agent.
    - Agent has a goal and a sub-goal.

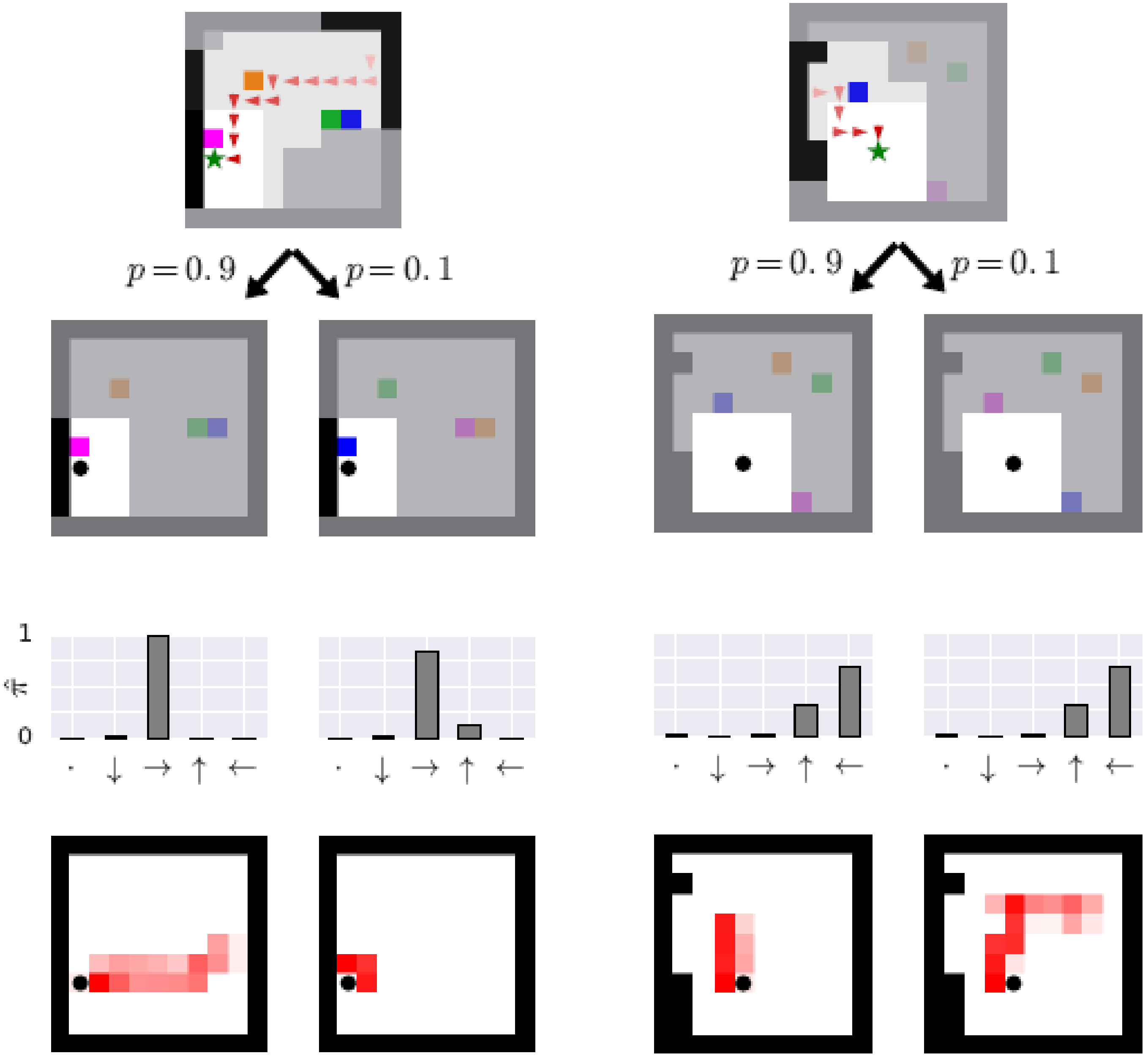
- When obtaining the sub-goal => *swap* the remaining objects, with low probability.

Acting on false beliefs:
Preliminaries

- Sub-goal: star. Goal: blue object.
- Dark grey => not observed.
- Light grey => observed before but NOT during goal consumption.
- Consumption => p=0.01 of *swap* event
- Observe Effect of swap in agent's policies and expected future moves.

Acting on false beliefs: Preliminaries

- Sub-goal: star.   Goal: blue object.
- Dark grey => not observed.
- Light grey => observed before but NOT during goal consumption.
- Consumption => p=0.01 of *swap* event
- Observe effect of swap in agent's policies and expected future moves.



- **Left: Swap event within field of view.**
- **Right: Swap event outside of field of view.**

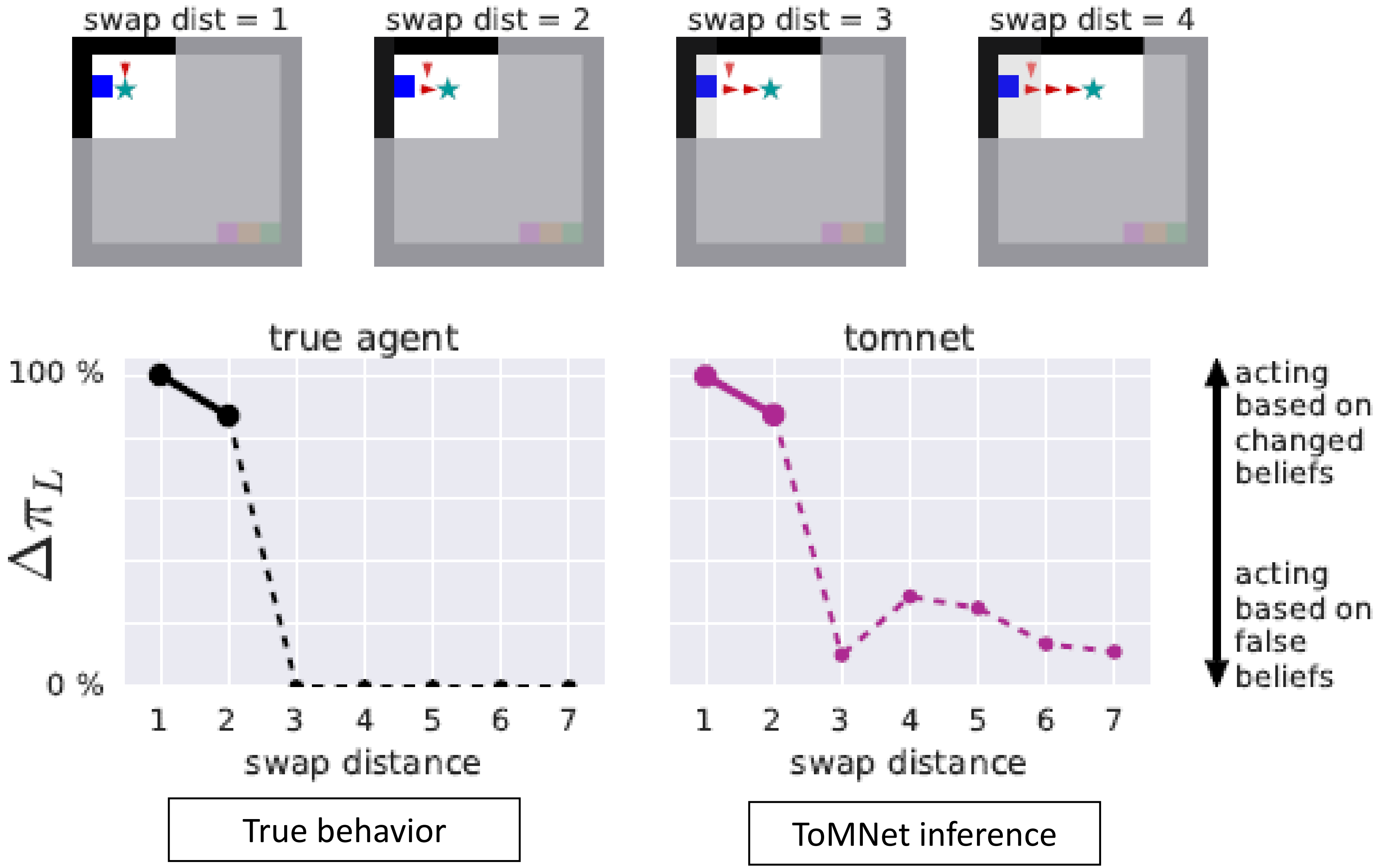
Running the Sally-Anne Test

- Agent has 5 x 5 window, consume star (sub-goal), prefers blue object.
- If we increase distance to swap, it may be invisible.

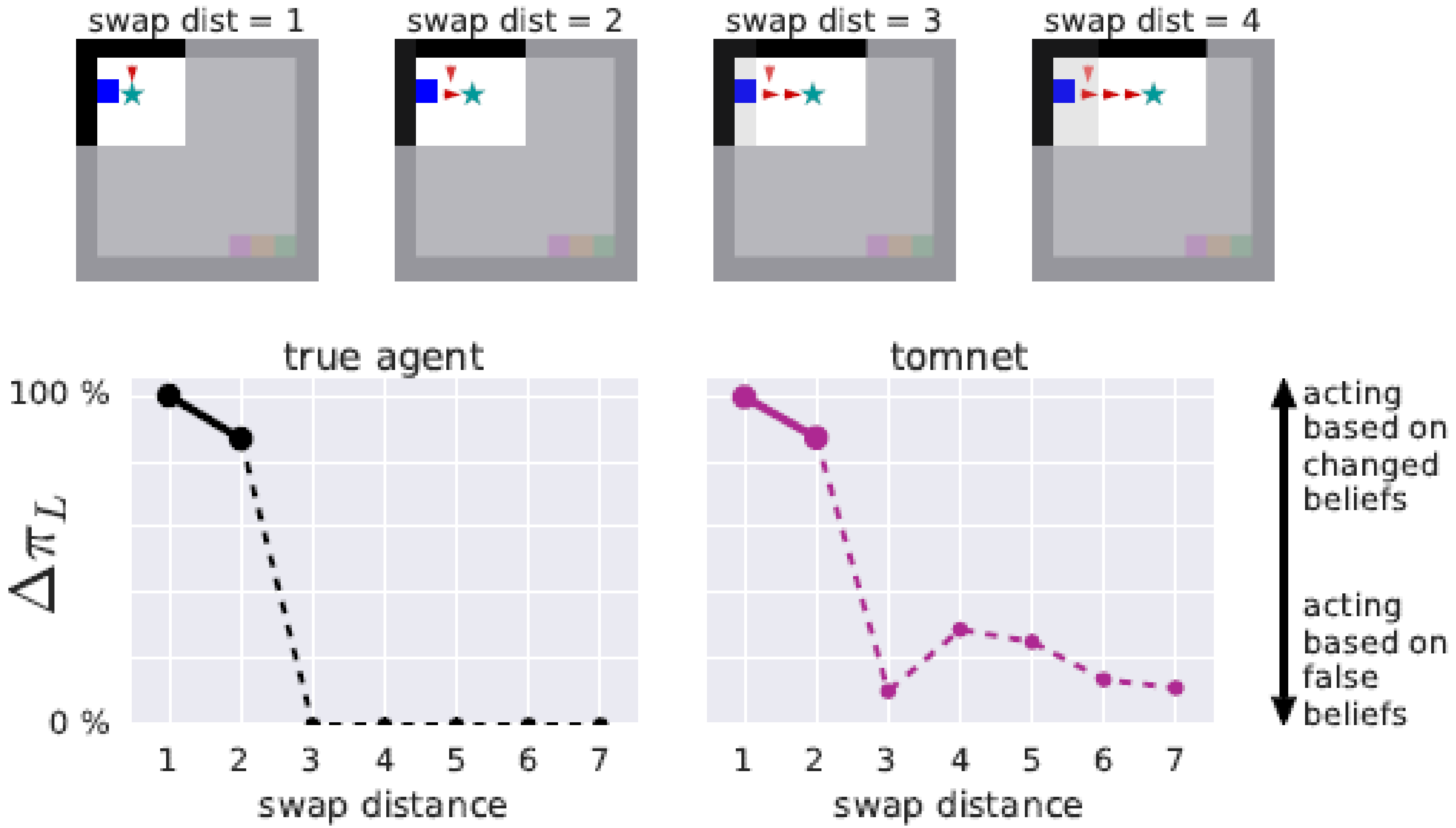
- Agent's policy unchanged for invisible swap.

$$\Delta\pi_L = \frac{\pi(a_L\,|no\ swap)\ -\ \pi(a_l\,|swap)}{\pi(a_L\,|no\ swap)} * \ 100\%$$

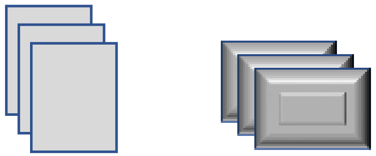
Running the Sally-Anne Test



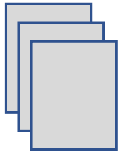
Running the Sally-Anne Test ⇒ It passes! ToMNet ~ 4 year old IQ



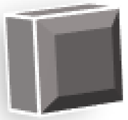
Architecture



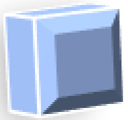
- Character Net*:* *ConvNet + LSTM* f_θ
- Mental State: None.
- Prediction Net:
  - Three predictions, with shared Torso:



    - Policy Prediction: *<u>ConvNet</u>* $a_\theta \Rightarrow \hat{\pi}$
    - Probability Consumption Prediction: *ConvNet* $c_\theta \Rightarrow \hat{c}$
    - Sucessor Representation: *ConvNet* $SR_\theta \Rightarrow \widehat{SR}$



- Deep RL Agents: UNREAL architecture, 100M episodes, cluster 16 CPU



- Belief Prediction Head:
  - ConvNet ⇒ 11x11x5 Dim Logit predicted belief objects present on map.
  - ConvNet ⇒ 11x11x5 Dim Logit predicted belief objects absent from map.

THANK YOU