Supplementary: Sympathy-based Reinforcement Learning Agents

Appendices

A Inaccurate State Prediction Models

In practice it can be difficult to build an accurate state prediction model $M(s, a)$ such as that proposed in our method. However, as the role of the model is primarily to support in acting as a normalisation factor, we expect our method to be robust to a certain degree of inaccuracies. An experiment was conducted to test this robustness where by predictions made by the model were augmented such that the unseen side of the vision field (in the predicted state) was populated with random noise, as shown in Figure 6 - to test how our method responds to inaccuracies in the next state prediction.

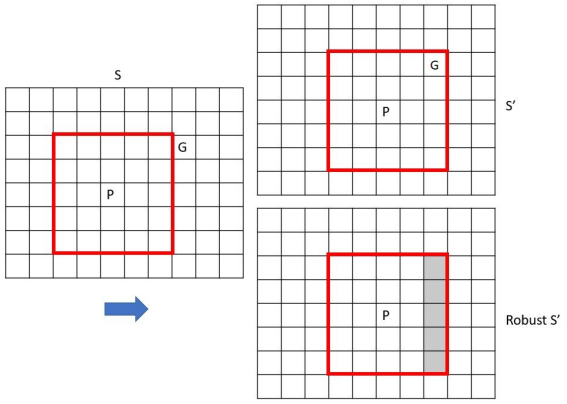
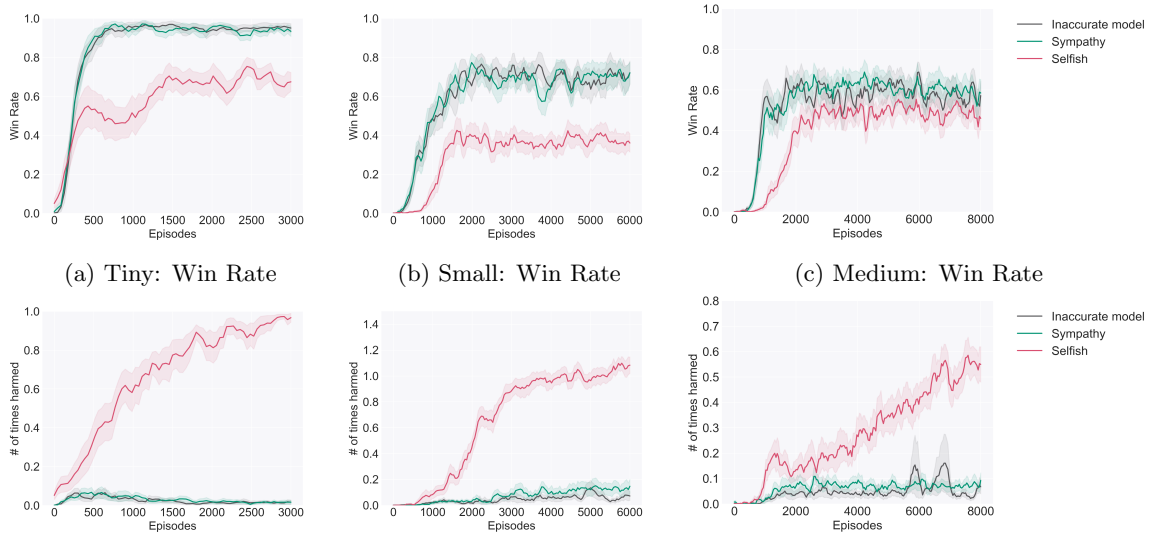


Figure 6: To test the robustness of the framework to inaccuracies in the prediction made by the next state model, experimentation was conducted such that the prediction made by the model was actively altered. This was done by filling the portion of the vision field (that was previously unseen) with random uniform noise.

Figure 7 and Figure 8 shows the results of our Sympathy Framework with an inaccurate model, compared against (1) the framework without the inaccuracies, and (2) a fully selfish agent.

As seem from the results, this injection of inaccuracies resulted in similar performance outcomes to the framework without the inaccuracies.

A.1 Pacman Results

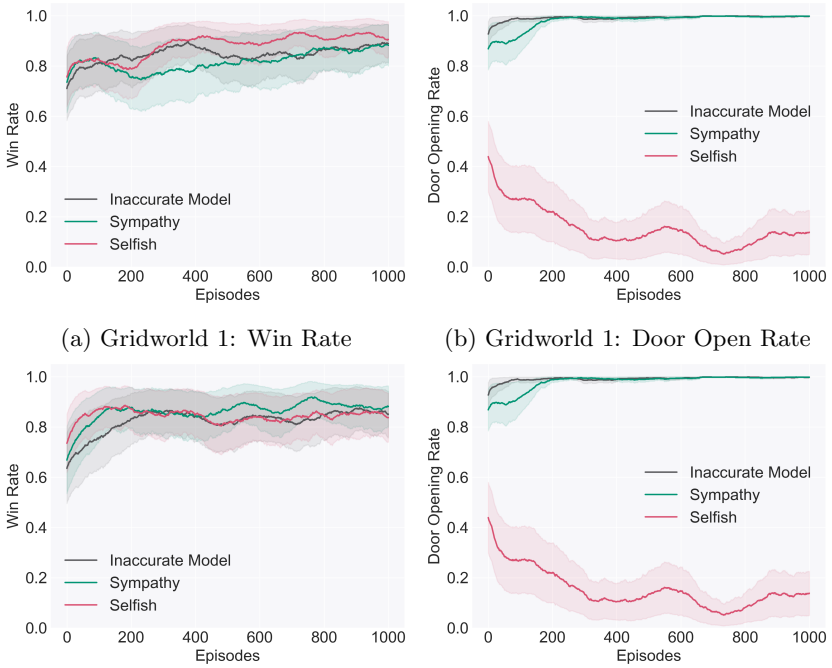


(a) Tiny: Win Rate (b) Small: Win Rate (c) Medium: Win Rate

(d) Tiny: Times Ghost Harmed (e) Small: Times Ghost Harmed (f) Medium: Times Ghost Harmed

Figure 7: Comparison of Sympathy Framework with an inaccurate state prediction model for Pacman games, compared against a selfish agent, and framework without inaccuracies.

A.2 Gridworld Results



(a) Gridworld 1: Win Rate (b) Gridworld 1: Door Open Rate

(c) Gridworld 2: Win Rate (d) Gridworld 2: Door Open Rate

Figure 8: Comparison of Sympathy Framework with an inaccurate state prediction model for Gridworld games, compared against a selfish agent, and framework without inaccuracies.

A.3 Ablations

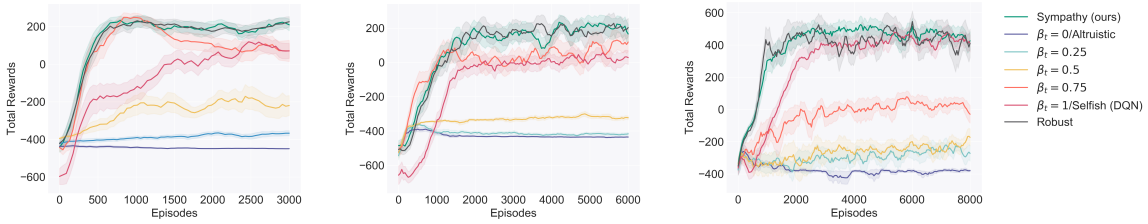
Our work has conducted some implicit ablations of our proposed framework:

- Ablation 1: An inaccurate next-state model ablates the next-state model (Figure 6)
- Ablation 2: Constant values of β_t (these are the results used as benchmarks in our experiments), reflect the results when the sympathy function and next-state model are removed.
- Ablation 3: The Q_{symp} block is ablated in the results where $\beta_t = 0$.

B Game rewards

Total accumulated rewards for Pacman and Gridworld games.

B.1 Pacman



(a) Tiny Pacman (b) Small Classic Pacman (c) Medium Classic Pacman

Figure 9: Total rewards of learning agent in Pacman environments

B.2 Ghost

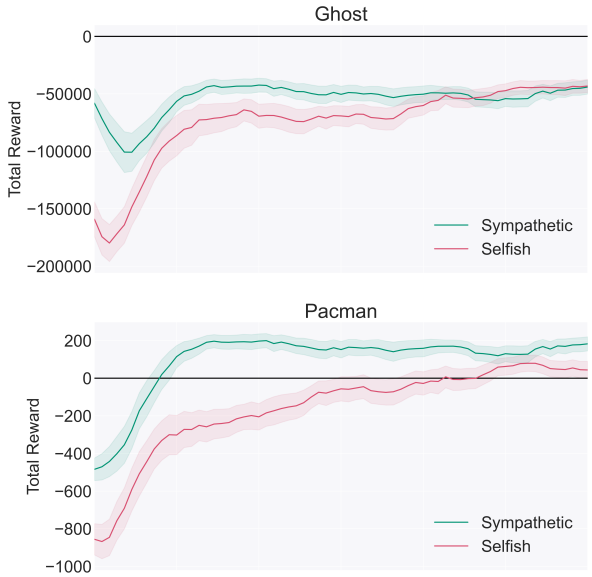
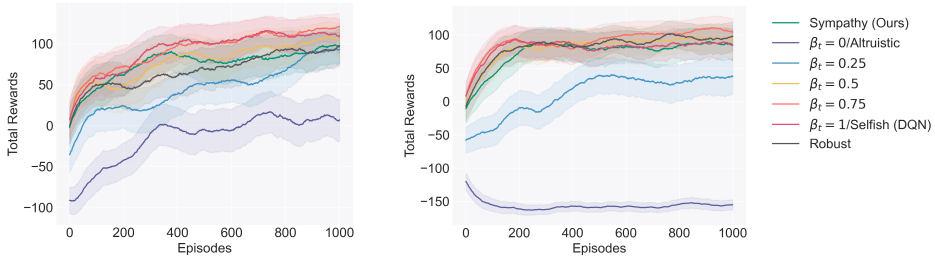


Figure 10: Tiny Pacman Game: Comparison of total rewards of Pacman and Ghost when Pacman is behaving sympathetically compared to selfishly ($\beta_t = 1$). As the ground truth rewards of the Ghost are unknown, the values inferred through IRL are used (as shown in Figure 12a). The rewards obtained from sympathetic behaviour is generally higher or equal for both agents, reflecting a win-win outcome. The strong negative total rewards of the Ghost can be explained by the highly negative value (refer Figure 12) inferred by IRL for the Ghost taking a step.

B.3 Gridworld



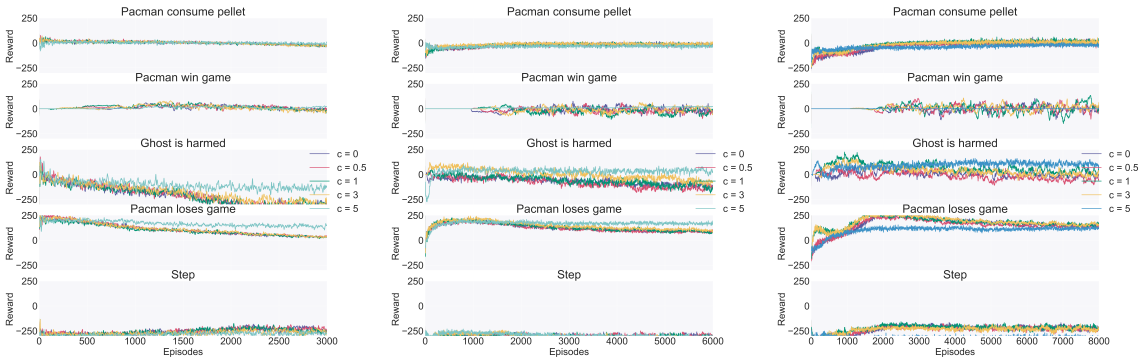
(a) Gridworld 1 (b) Gridworld 2

Figure 11: Total rewards of learning agent in Gridworld environments

C IRL Trend Results

Figures 12 and 13 show the inferred reward function feature weights ($\hat{\mathbf{r}}_{indep}$) through IRL during training (for the independent agent). Weights have been scaled to have $\ell 1$ norm equal to that of the learning agent

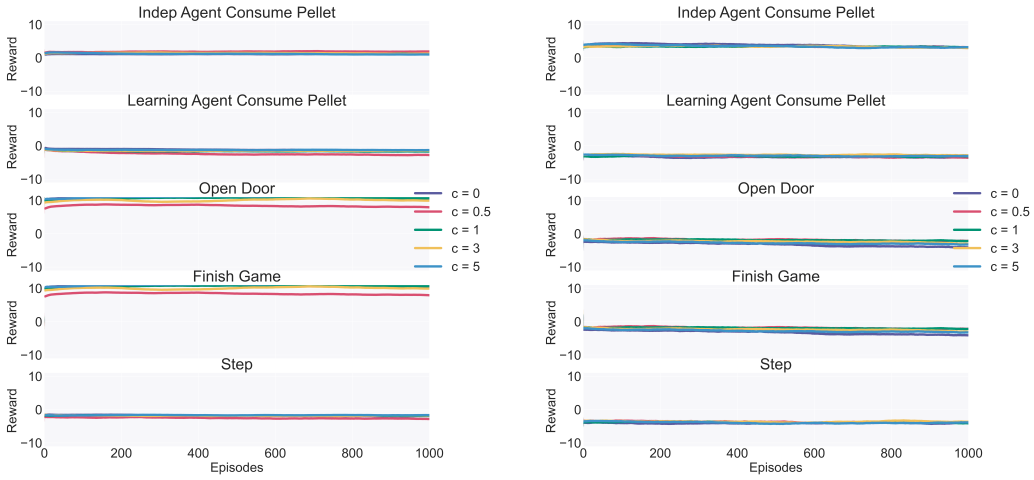
C.1 Pacman



(a) Tiny Pacman (b) Small Classic Pacman (c) Medium Classic Pacman

Figure 12: IRL results for $\hat{\mathbf{r}}_{indep}$ in Pacman games

C.2 Gridworld



(a) Gridworld 1

(b) Gridworld 2

Figure 13: IRL results for $\hat{\mathbf{r}}_{indep}$ in Gridworld games

D Degree of Selfishness trend

Figure 14 and 15 visualise the degree of selfishness outputted by the Sympathy function $\beta(s, a)$. Empirically the value outputted is shown to converge. This is demonstrated on the Tiny Pacman game, as well as the two Gridworld environments for various c values.

D.1 Pacman

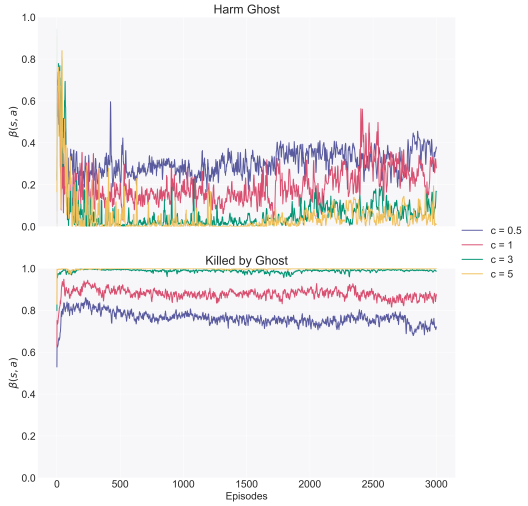
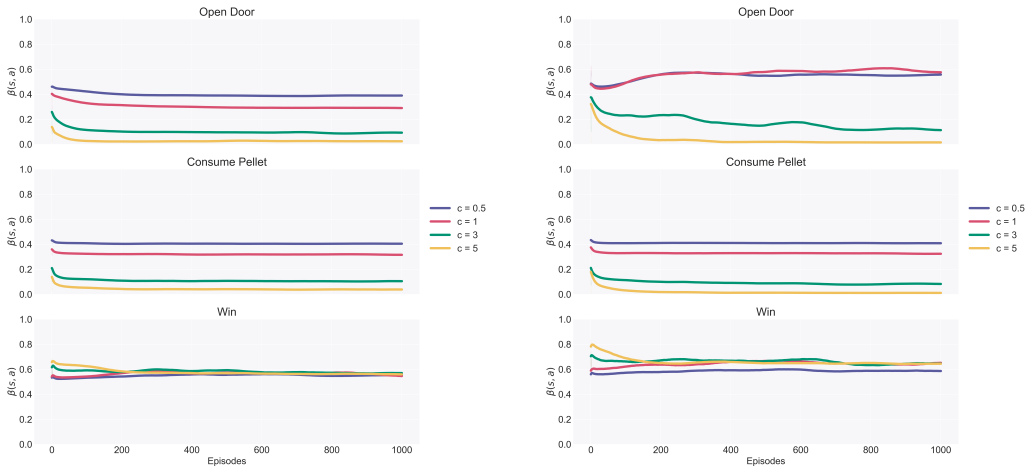


Figure 14: Trend in Tiny Pacman game $\beta(s, a)$ value over time for the events: harming the Ghost, and being killed by the Ghost.

D.2 Gridworld



(a) Gridworld 1 $\beta(s, a)$ trend

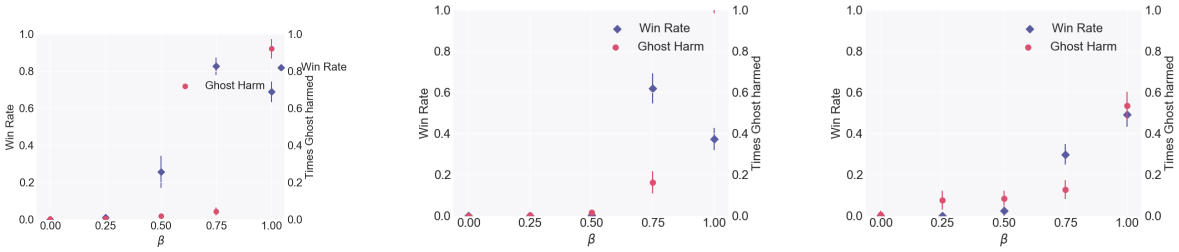
(b) Gridworld 2 $\beta(s, a)$ trend

Figure 15: Trend in Gridworld $\beta(s, a)$ value over time for the events: opening door, consuming a pellet, and winning game

E Performance vs constant beta

The following figures show the performance in each game when, rather than our proposed Sympathy function, the degree of selfishness β_t is held at a constant value throughout the game.

E.1 Pacman



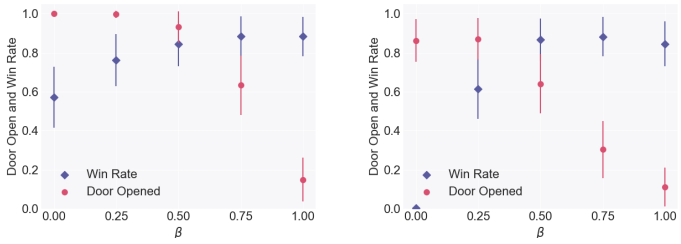
(a) Tiny Pacman

(b) Small Classic Pacman

(c) Medium Classic Pacman

Figure 16: Impact to Pacman performance by varying a constant β_t (applying a constant degree of selfishness throughout the whole game)

E.2 Gridworld

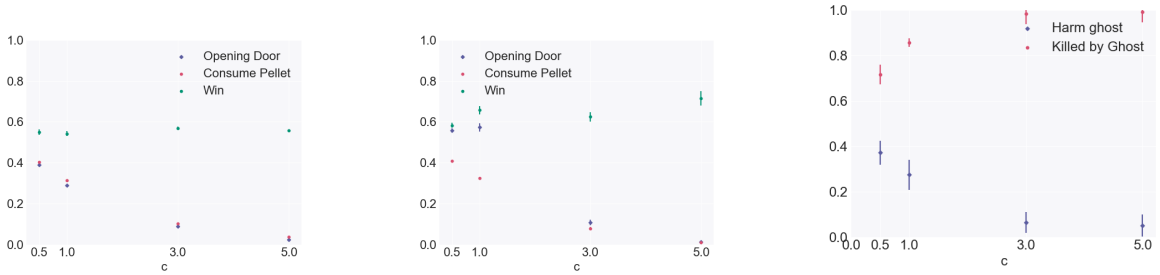


(a) Gridworld 1 (b) Gridworld 2

Figure 17: Impact to Gridworld performance by varying a constant β_t (applying a constant degree of selfishness throughout the whole game)

F Degree of Selfishness Visualised

Figure 18 visualises the manner in which the degree of selfishness β_t (output from the sympathy function) for various events, diverges as the c hyperparameter is increased. These results are averaged from the last 100 episodes of training.



(a) Gridworld 1 (b) Gridworld 2 (c) Tiny Pacman

Figure 18: Impact to the degree of selfishness β_t determined for various events in the game as hyperparameter c in Sympathy Function is altered. In general, as c is increased, β_t values diverge from 0.5.

For the Pacman game, Table 3 displays the converged β_t values for the events (1) Pacman harming the ghost (*Harm*) and (2) The Ghost killing Pacman (*Killed*) for the Small Classic and Medium Classic games, at $c = 1$.

Table 3: Learned β_t values for key events in Small Classic and Medium Classic Pacman environments at $c = 1$

	Small Classic		Medium Classic	
c	*Harm*	*Killed*	*Harm*	*Killed*
1	0.44±0.03	0.78±0.04	0.42±0.02	0.78±0.02

Figure 19 shows the impact on performance of changing c for the Medium classic Pacman game on Pacman's win rate, and the number of times the ghost is harmed.

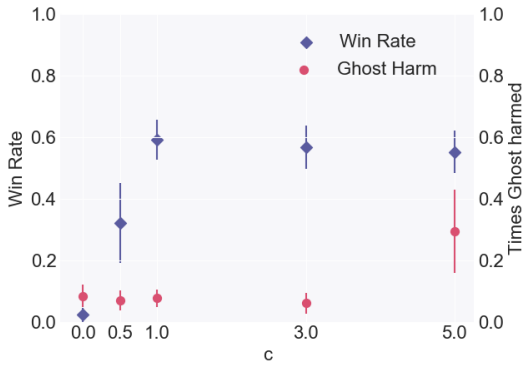


Figure 19: Medium Classic Pacman c vs performance