FastSVERL: Approximating Shapley Explanations in Reinforcement Learning

Daniel Beechey and Özgür Şimşek

University of Bath
BATH RL LAB
art-ai

Background: We need to understand our agents

Reinforcement learning (RL) agents are achieving remarkable success, but their lack of transparency is a significant obstacle to deployment. To trust and debug these agents, we need principled methods to understand them.

A principled approach: Shapley values

The SVERL framework [1, 2] provides a principled and rigorous way to explain RL agents by attributing the influence of state features on an agent's:

- Behaviour (Why this action?)
- Outcome (Why this expected return?)
- Prediction (Why this value estimate?)

What are Shapley-based explanations?

As an example, we focus on explaining behaviour, which SVERL explains by attributing how each feature influences the probability of an agent's action.

This is captured by a **characteristic function**, which measures the expected action probability when only the features in subset $\mathcal{C} \subseteq \mathcal{F}$ are known:

$$\tilde{\pi}_s^a(\mathcal{C}) = \mathbb{E}\left[\pi(S,a) \mid S^{\mathcal{C}} = s^{\mathcal{C}}\right] = \sum_{s \in \mathcal{S}^+} p^{\pi}(s \mid s^{\mathcal{C}})\, \pi(s,a)$$

Shapley values [3] assign credit to each feature i based on its average marginal contribution across all possible feature subsets:

$$\phi^i(\tilde{\pi}_s^a) = \sum_{\mathcal{C} \subseteq \mathcal{F} \setminus \{i\}} \frac{|\mathcal{C}|! \cdot (|\mathcal{F}| - |\mathcal{C}| - 1)!}{|\mathcal{F}|!} [\tilde{\pi}_s^a(\mathcal{C} \cup \{i\}) - \tilde{\pi}_s^a(\mathcal{C})]$$

These values uniquely satisfy axioms formalising fair credit assignment.

The bottleneck: impractical computational cost

The primary obstacle to deploying SVERL is computational cost:

1. Each characteristic value is an expectation over the state space.
2. The Shapley value sums these values over all feature combinations.

The total cost per explanation, $\mathcal{O}(2^{|\mathcal{F}|} \cdot |\mathcal{S}|)$, is infeasible in any complex setting.

Both the characteristic functions and the Shapley value summation must be approximated.

[1] Daniel Beechey, Thomas MS Smith, and Özgür Şimşek. Explaining reinforcement learning with Shapley values. In International Conference on Machine Learning, pages 2003–2014.PMLR, 2023.
[2] Daniel Beechey, Thomas MS Smith, and Özgür Şimşek. A theoretical framework for explaining reinforcement learning with Shapley values. arXiv preprint arXiv:2505.07797, 2025.
[3] Lloyd S Shapley. A value for n-person games. Contributions to the Theory of Games, 2(28):307–317, 1953.

Our contribution: The FastSVERL framework

We introduce FastSVERL, a scalable framework that **learns to approximate** Shapley explanations. FastSVERL trains parametric models to amortise estimation cost across states and features.

Key Features:

- Handles **temporal dependencies** across multi-step trajectories.
- Learns from **off-policy data**.
- Adapts to **evolving agent behaviour** in real-time.

How FastSVERL works: A two-model approach

FastSVERL approximates Shapley explanations by training two models:

1. A parametric model, $\hat{\pi}(s, a \mid \mathcal{C}; \beta)$, is trained to approximate the characteristic function. It minimises the expected squared error:

$$\mathcal{L}(\beta) = \mathop{\mathbb{E}}_{p^{\pi}(s)} \mathop{\mathbb{E}}_{\mathrm{Unif}(a)} \mathop{\mathbb{E}}_{\mathrm{Unif}(\mathcal{C})} \left|\pi(s,a) - \hat{\pi}(s, a \mid \mathcal{C}; \beta)\right|^2$$

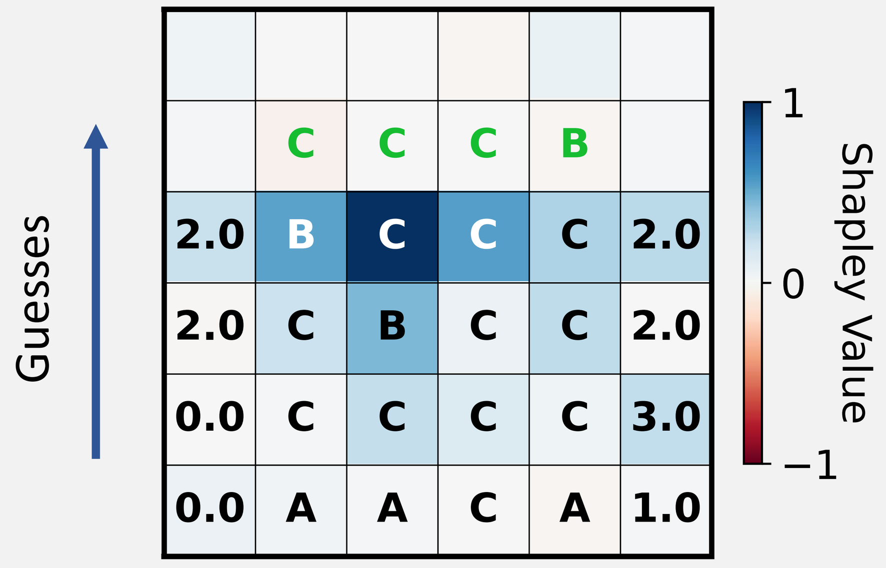
2. Using the characteristic model, a second model, $\hat{\phi}(s, a; \theta)$, estimates the Shapley values by solving a weighted least-squares problem:

$$\mathcal{L}(\theta) = \mathop{\mathbb{E}}_{p^{\pi}(s)} \mathop{\mathbb{E}}_{\mathrm{Unif}(a)} \mathop{\mathbb{E}}_{p(\mathcal{C})} \left|\tilde{\pi}_s^a(\mathcal{C}) - \tilde{\pi}_s^a(\emptyset) - \sum_{i \in \mathcal{C}} \hat{\phi}^i(s, a; \theta)\right|^2$$

Example explanation (Mastermind)

In Mastermind, an agent must guess a hidden 4-letter code, drawn from a 3-letter alphabet. Each guess receives clues for the number of correct letters in the correct position (right column) and wrong position (left column).

Before FastSVERL, Shapley explanations were infeasible due to the scale of this domain.

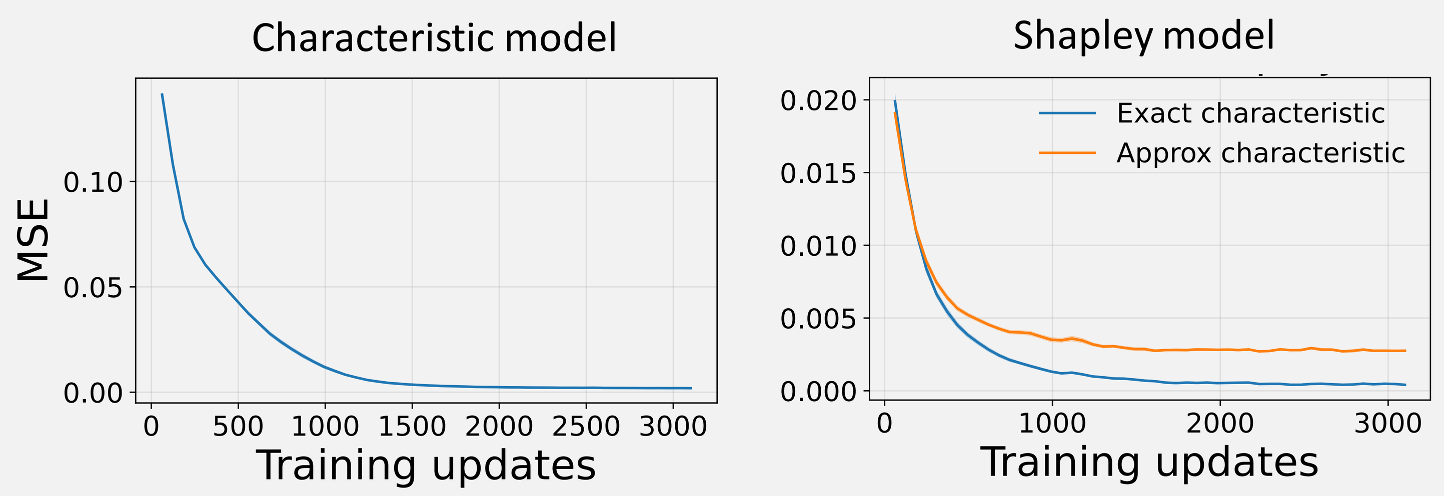


Darker blue cells indicate features *positively* influencing the agent's next move. Note how the explanation focuses on recent, relevant clues (Guesses 2-4) and ignores now redundant ones (Guess 1).

Empirical results: It's accurate and practical

All approximations are validated by comparing to exact Shapley values in a smaller, computationally feasible Mastermind domain.

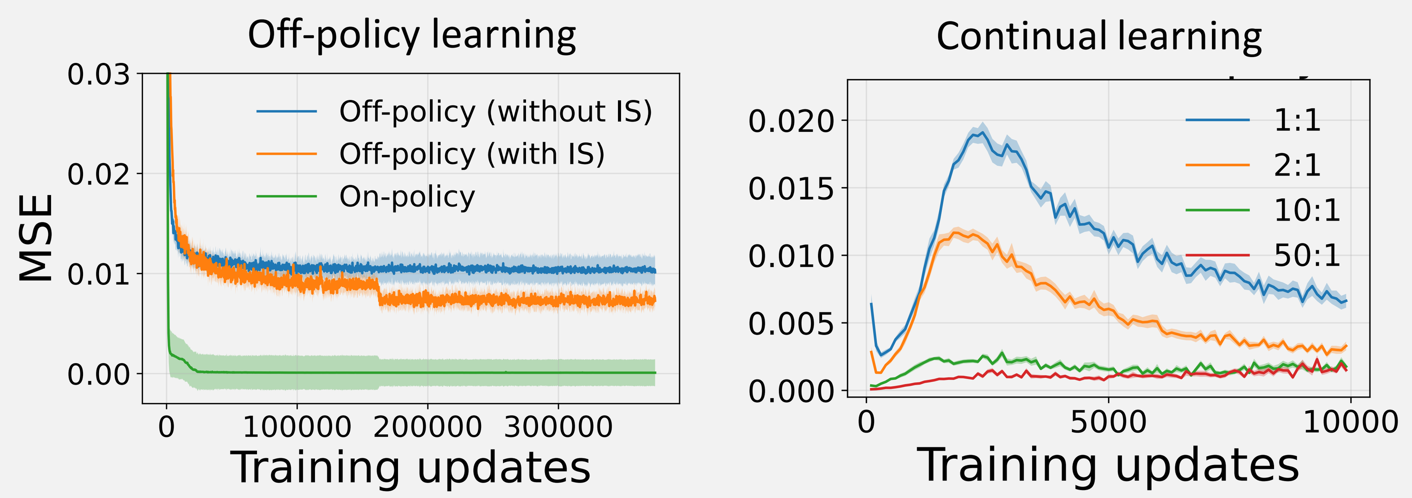
1. Validating the approximation models



Our approximation models converge to low error.

- **(Left):** The characteristic model quickly learns its target.
- **(Right):** The Shapley model also converges accurately. When the Shapley model is trained using the approximate characteristic model versus the ground truth, approximation errors propagate to the final approximations.

2. Designed for practical RL challenges



FastSVERL is designed to handle practical reinforcement learning challenges.

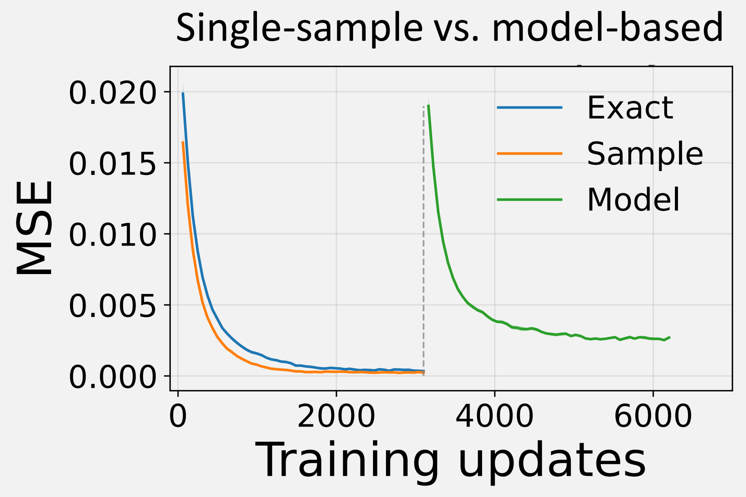
- **(Left):** The framework can learn from **off-policy data**. However, it requires Importance Sampling (IS; orange line) to correct the distributional mismatch and improve accuracy over direct off-policy training (blue).
- **(Right):** Explanations can be trained in parallel with a **non-stationary agent**. Error spikes when the agent's policy shifts significantly, which is mitigated by increasing the explanation model's update ratio (e.g. 10:1) relative to the agent.

Extending FastSVERL: A more efficient approach?

FastSVERL's two-model approach is effective, but it has two main drawbacks:

1. **Cost:** It requires training and storing two separate neural networks.
2. **Approximation errors propagate** from the characteristic model.

We propose one possible extension building on FastSVERL's solid foundation: integrating a noisy single-sample approximation of the characteristic value directly into the Shapley model's loss function.



Shapley models trained with single samples (**orange**) converge as quickly and accurately as using the ground truth (**blue**) and significantly faster than the two-model approach (**green**), which includes pre-training. This method has the potential to **halve computational cost** and **eliminate error propagation**.

Big picture

- **For researchers:** A scalable foundation for rigorous, real-time interpretability.
- **For practitioners:** A practical tool to build trust and enable the deployment of agents in real-world systems.

Future work

- User studies to formally evaluate how these explanations aid human understanding.
- Approximations in continuous state and action spaces.

Try FastSVERL on your own agents!

Code: github.com/djeb20/fastsverl **Email:** djeb20@bath.ac.uk