USEFUL EMPOWERMENT APPROXIMATION FOR ITS USE IN LEARNING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Process (closed-loop) empowerment is a relevant measure for agency, with promising uses for intrinsic motivation and reinforcement learning. However, its prohibitive computational cost, especially in discrete environments, precludes many of its applications to anything but toy examples and the smallest environments. We introduce a reformulation of process empowerment for Markovian dynamics under which the empowerment computation becomes tractable, and derive from it a variant of the Alternating Blahut-Arimoto algorithm whose per-iteration cost is polynomial in the number of states, actions and horizon compared to previous exponential in the horizon per-iteration cost; making empowerment a viable signal in more complex environments. Moreover, we introduce even faster approximations of empowerment via early-stopping, showing that few iterations are enough to compute empowerment to a reasonable precision, and a warm-start initialisation reusing other states' empowerment information to hasten computation. Across six Minigrid environments, the resulting approximations reach a STARC distance, a reward comparison metric, of 0.025 from exact empowerment making the approximation behaviourally interchangeable with the exact signal at almost no cost. Our approach achieved this in between 1.0 and 6.4 iterations per state on average compared to between 75 and 1275 for the exact computation.

1 INTRODUCTION

Intelligent agents are a kind of AI system which is characterised by existing in an environment with which it interacts (Russell et al., 1995). These agents are expected to achieve some goal or behaviour, but often, this behaviour must be learnt autonomously, for example, through *Reinforcement Learning (RL)*.

However, to achieve these goals, it is often prerequisite to learn the dynamics of the environment the agent populates. This is a non-trivial task, as it requires exploration. Moreover, learning with a reward signal for a goal strongly biases the exploration, and switching tasks or goals can require re-training everything. For this, curiosity, intrinsic motivation, and other surrogate goals are often used within the intriniscally-motivated learning community.

Empowerment is one such goal(Klyubin et al., 2005; Salge et al., 2014), a desirable target of agency. Empowerment is defined state-wise as the capacity of an agent to meaningfully alter their future through its following actions. Intuitively, empowerment is high if the agent has the affordances, or is in a situation where its actions matter (Klyubin et al., 2005; Salge et al., 2014). Training an agent to maximise the empowerment signal means that the agent will learn to converge to such states; from which exploration may be more meaningful and goals easier to achieve (Volpi & Polani, 2023).

There is, however, a strong limitation precluding the application of empowerment to most agent tasks: its computational cost. Current algorithms are either costly and potentially biased lower bounds or are exact computations exponential in time (Tiomkin et al., 2025), and cannot be sensibly applied to any but toy examples and small environments.

In this work, we strive to find more computationally efficient bounds for process-empowerment, one of the variant calculations which is the most informed. We achieve this by using three tricks. Firstly, we introduce the prevalently used Markovian assumption (*ie.* that state transitions are independent on previous history), turning the exponential algorithm for empowerment (Alternating

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Blahut-Arimoto (Tiomkin et al., 2025)) polynomial with respect to state-space and action-space size, as well as the idiosyncratic horizon. Secondly, noticing how the differences in empowerment value become minimal after few iterations of the algorithm, we propose an early-stopping policy that cuts the time even further without compromising the accuracy of the value. Finally, we propose a warm-start method for computing the empowerment of some states by re-using information from its parent states. We validate these findings on 6 different Minigrid environments, showing that very few iterations are needed to capture the useful signal for RL empowerment has.

2 Background and related work

2.1 Background

Empowerment is an information theoretic measure quantifying the influence an agent has on the environment from a certain state. It is established as the channel capacity between actions the agent can carry over a certain temporal horizon and future states . Moreover it is local, universal and task-independent. (Klyubin et al., 2005; Salge et al., 2014). With locality being relevant for its computation.

Two empowerment variants can be found, open-loop empowerment and process or closed-loop. Intuitively process empowerment measures influence assuming the agent can react to the outcomes of its individual actions while open-loop assumes it cannot, for this theoretic and experimental results have shown open-loop is a lower-bound of process empowerment and underestimates real influence in stochastic environment (Tiomkin et al., 2025).

For a given horizon $T+1$ process empowerment is formalized by Tiomkin et al. (2025) $\mathcal{E}^{PR}(s_0) = \max_{\{\pi(a_t|s_0^t,a_0^{t-1})\}_{t=0}^{T}} I[S_{T+1}; A_0^T, S_1^T \mid s_0]$. That is the channel capacity (*ie.* maximum mutual information), between a sequence of $T+1$ actions A_0^T, the intermediate states in which they land the agent S_1^T, and the final state S_{T+1}. Each π in the set is a probing policy dependant on the timestep and is different in general to the policy on which the agent acts, in that it serves to compute the potential influence of the agent not how it should act. The probing policies set may be also called probing policies or policies.

The maximization computation of process empowerment in non-necessarily Markovian environments is carried by Alternating Blahut-Arimoto (ABA) (Tiomkin et al., 2025), a variation of Blahut-Arimoto for process empowerment. ABA is shortly described as this work will later introduce a polynomial cost per iteration variant for environments with Markovian transitions. Along with ABA self-consistent equations for $Q(s_1^T, a_0^T \mid s_0, s_{T+1})$ and for the policy set $\{\pi(a_t|s_0^t, a_0^{t-1})\}_{t=0}^{T}$ are introduced (Tiomkin et al., 2025):

$$\forall t' : \pi\Big(a_{t'} \mid a_0^{t'-1},\, s_0^{t'}\Big) \propto \exp\big(\mathcal{E}^{\mathrm{FPR}}_{\mathrm{NM}}(a_0^{t'}, s_0^{t'}) + \mathcal{E}^{\mathrm{PPR}}_{\mathrm{NM}}(s_1^{t'}, a_0^{t'} \mid s_0)\big)$$

$$Q\big(s_1^T,\, a_0^T \mid s_0,\, s_{T+1}\big) = \frac{P\big(s_{T+1} \mid s_1^T,\, a_0^T,\, s_0\big)\; P\big(s_1^T,\, a_0^T \mid s_0\big)}{\sum\limits_{\substack{s_1^T \\ a_0^T}} P\big(s_{T+1} \mid s_1^T,\, a_0^T,\, s_0\big)\; P\big(s_1^T,\, a_0^T \mid s_0\big)}$$

Where [1]:

$$\mathcal{E}^{\mathrm{FPR}}_{\mathrm{NM}}\big(a_0^{t'},\, s_0^{t'}\big) := \sum_{\substack{s_{t'+1}^{T+1} \\ a_{t'+1}^{T}}} P\Big(s_{T+1}, a_{t'+1}^T, s_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\Big) \log\left(\frac{Q\Big(s_{t'+1}^T, a_{t'+1}^T \mid a_0^{t'}, s_0^{t'}, s_{T+1}\Big)}{P\big(a_{t'+1}^T, s_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\big)}\right)$$

$$\mathcal{E}^{\mathrm{PPR}}_{\mathrm{NM}}\big(s_1^{t'},\, a_0^{t'} \mid s_0\big) := \sum_{\substack{s_{t'+1}^{T+1} \\ a_{t'+1}^{T}}} P\Big(s_{T+1}, a_{t'+1}^T, s_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\Big) \log\Big(Q\Big(s_1^{t'}, a_0^{t'} \mid s_0, s_{T+1}\Big)\Big)\,.$$

[1] Tiomkin et al. (2025) denote these by $\mathcal{E}^{\mathrm{FPR}}[S_{T+1}; A_{t'+1}^T, S_{t'+1}^T \mid a_0^{t'}, s_0^{t'}]$ and $\mathcal{E}^{\mathrm{PPR}}[S_{T+1}; s_1^{t'}, a_0^{t'} \mid s_0]$.

Notice NM stands for non-Markovian. ABA is then used to iterate over the self-consistent equations:

Algorithm 1 Alternating Blahut-Arimoto (ABA)

1: **Initialize** $\{\pi(a_t \mid a_0^{t-1}, s_0^t)\}_{t=0}^T$
2: **repeat**
3: update $Q(s_1^T, a_0^T \mid s_0, s_{T+1})$
4: for every t' update $\pi(a_{t'} \mid a_0^{t'-1}, s_0^{t'})$
5: **until** convergence
6: **Output** $\mathcal{E}^{\mathrm{PR}}(s_0) = I[S_{T+1}; S_1^T, A_0^T \mid s_0]$ at the converged policy set

The enumeration of all possible trajectories, makes each iteration of ABA exponential in the horizon, with $|\mathcal{A}|$ times the number of the reachable states in $T+1$ steps at the base. This makes it severely impractical for RL-training scenarios.

2.2 Related Work

For the computation of process empowerment in discrete environments, the only available algorithm is ABA (Tiomkin et al., 2025). There has been research in the use of variational estimations to estimate closed-loop empowerment via options (Gregor et al., 2017). However, this is sometimes biased in stochastic settings (Kwon, 2021), and the gap between variational lower bounds and the true value is unknown (Song & Ermon, 2020). For open-loop computation in discrete environments Blahut-Arimoto is used (Blahut, 1972; Arimoto, 1972; Klyubin et al., 2005; Salge et al., 2014), despite its similar exponential cost with respect to the horizon (Tiomkin et al., 2025). Variational estimation has also been widely studied however it has similar problems to those it has in process empowerment (Mohamed & Jimenez Rezende, 2015; Karl et al., 2022). On the open-loop side, an efficient reliable estimation can be done estimating Gaussian channel and later water-filling (Zhao et al., 2021), however it is restricted to environments with latent Gaussian transition function and policies.

Research has been done in the use of empowerment in RL for its use as an intrinsic reward (Tiomkin et al., 2024) and in combination with externally specified goal directed behaviour (Volpi & Polani, 2023). More recently variational estimations of empowerment have been used to estimate the empowerment of Language Model Agents showing it correlates with average task performance and that states and actions with high empowerment are often pivotal ones (Song et al., 2026).

3 Useful empowerment approximations for RL

This section introduces a variant of Alternating Blahut-Arimoto for obtaining useful process empowerment approximations. The variant is built in 3 steps. First, the cost of each iteration is taken from exponential with respect to the horizon to polynomial for a Markovian environment. Then early-stop is used to avoid long tails of iterations that just add precision and not usefulness. Finally, warm-starting from predecessor states is used to initialize ABA closer to useful approximations of empowerment, further reducing the required iteration to reach them.

3.1 Polynomial-cost ABA algorithm for the Markov case

We start by developing a polynomial time per iteration variant of Alternating Blahut-Arimoto for computing exact process empowerment in environments with Markov transitions. First, with proposition 1 it is proved that there is always a set of Markov policies for which the empowerment value is reached. This allows to go from the high-dimensional space of policies over all trajectories to the space of Markov policies. Then in theorem 2, the self-consistent equations in Alternating Blahut-Arimoto are adapted for it to compute empowerment in the space of Markovian policies. Finally, a practical note is given for the evaluation of the self-consistent equations in polynomial time and thus render ABA iterations polynomial.

Proposition 1 (Existence of an optimal set of Markov probing policy)**.** *Let P be Markovian transition dynamics of an environment. Then there exists a set of Markovian probing policies $\{\pi(a_t \mid s_t)\}_{t=0}^{T}$ maximizing mutual information and therefore giving the value of empowerment:*

$$\max_{\{\pi(a_t|s_t)\}_{t=0}^{T}} I\big[S_{T+1}; S_1^T, A_0^T \mid s_0\big] \;=\; \mathcal{E}^{\mathrm{PR}}(s_0).$$

Where X_i^j refers to the sequence of all variables X from X_i to X_j.

Proof. The proof has two steps. First, under the Markov assumption mutual information $I\big[S_{T+1}; S_1^T, A_0^T \mid s_0\big]$ depends on the policy only through $P(s_T, a_T \mid s_0)$. Second, by induction over T we show that any $P(s_T, a_T \mid s_0)$ reachable by a history-dependent policy is also reachable by a Markov policy.

Using the chain rule for mutual information it can be decomposed in the following expression:

$$I\big[S_{T+1}; S_1^T, A_0^T \mid s_0\big] = I[S_{T+1}; S_T, A_T \mid s_0] + \underbrace{I\big[S_{T+1}; S_1^{T-1}, A_0^{T-1} \mid S_T, A_T, s_0\big]}_{=0}$$

For a Markovian transition function $S_{T+1} \perp\!\!\!\perp S_1^{T-1}, A_0^{T-1} \mid S_T, A_T, s_0$, thus the second term is 0.

$$I[S_{T+1}; S_T, A_T \mid s_0] = \sum_{s_{T+1}, a_T, s_T} P(s_{T+1}|s_T, a_T) P(s_T, a_T|s_0) \log\left(\frac{P(s_{T+1}|a_T, s_T)}{P(s_{T+1}|s_0)}\right)$$

Since $P(s_{T+1} \mid s_0) = \sum_{s_T, a_T} P(s_{T+1} \mid s_T, a_T)\, P(s_T, a_T \mid s_0)$, mutual information depends on the policy only through $P(s_T, a_T \mid s_0)$.

For $T = 0$, history dependent policies are trivially Markov, so the claim holds. For $T \geq 1$, the $P_{\mathrm{NM}}(s_T|s_0)$ of a set of non-Markovian policies is reached by the set of Markov policies of the induction hypothesis, marginalising over the Markovian transitions $P(s_T|s_{T-1}, a_{T-1})$. Extending this set with $\pi(a_T|s_T) = P_{\mathrm{NM}}(a_T|s_T, s_0)$ gives $P_{\mathrm{NM}}(s_T, a_T|s_0) = P_{\mathrm{NM}}(a_T|s_T, s_0)P(s_T|s_0) = \pi(a_T|s_T)P(s_T|s_0)$, so it is reachable with a set of Markov policies. Applied to an optimal set of policies, by the first step a set of Markov policies attains $\mathcal{E}^{\mathrm{PR}}(s_0)$.

□

Theorem 2 (Markovian ABA self-consistent equations)**.** *Let P be the Markovian transition dynamics of an environment. Then an optimal set of Markovian probing policies satisfies these self-consistent equations:*

$$\forall t' \in \{0, \ldots, T\} : \pi(a_{t'} \mid s_{t'}) \propto \exp\big(\mathcal{E}^{\mathrm{FPR}}(a_{t'}, s_{t'}) + \mathcal{E}^{\mathrm{PPR}}(a_{t'}, s_{t'} \mid s_0)\big)$$

$$Q\big(s_1^T, a_0^T \mid s_0, s_{T+1}\big) = \frac{P(s_{T+1} \mid s_T, a_T)\, P\big(s_1^T, a_0^T \mid s_0\big)}{\sum_{s_1^T, a_0^T} P(s_{T+1} \mid s_T, a_T)\, P\big(s_1^T, a_0^T \mid s_0\big)}$$

where $P\big(s_1^T, a_0^T \mid s_0\big) = \prod_{t=0}^{T-1} P(s_{t+1} \mid s_t, a_t) \prod_{t=0}^{T} \pi(a_t \mid s_t)$

$$\mathcal{E}^{\mathrm{FPR}}(a_{t'}, s_{t'}) := \sum_{\substack{s_{t'+1}^{T+1} \\ a_{t'+1}^{T}}} P\big(s_{T+1}, a_{t'+1}^T, s_{t'+1}^T \mid a_{t'}, s_{t'}\big) \log\left(\frac{Q\big(s_{t'+1}^T, a_{t'+1}^T \mid a_{t'}, s_{t'}, s_{T+1}\big)}{P\big(a_{t'+1}^T, s_{t'+1}^T \mid a_{t'}, s_{t'}\big)}\right),$$

$$\mathcal{E}^{\mathrm{PPR}}(a_{t'}, s_{t'} \mid s_0) := \sum_{s_{T+1}} P(s_{T+1} \mid a_{t'}, s_{t'}) \log(Q(s_{t'}, a_{t'} \mid s_0, s_{T+1}))$$

are the Markov counterparts of the future and past process empowerment terms of Tiomkin et al. (2025). $\mathcal{E}^{\mathrm{FPR}}$ equals the original future term, while $\mathcal{E}^{\mathrm{PPR}}$ differs from the original past term by a residual that does not depend on $a_{t'}$. Moreover, if ABA is initialized with Markov policies, all its iterates are Markov, so it inherits the convergence of ABA to $\mathcal{E}^{\mathrm{PR}}(s_0)$ (Tiomkin et al., 2025).

Proof. For the Q equation, the channel of the original posterior equation reduces to $P(s_{T+1} \mid s_T, a_T)$ by the Markov property, and the source factorizes under Markov dynamics and policies.

For the π equation, the policies on the right-hand side are Markov and Q is their posterior. By Proposition 1 a Markov maximizer exists, and it satisfies the original self-consistent equations (Tiomkin et al., 2025). We apply the Markov property to $\mathcal{E}^{\mathrm{FPR}}_{\mathrm{NM}}$ and $\mathcal{E}^{\mathrm{PPR}}_{\mathrm{NM}}$ for an arbitrary history and a fixed t', then substitute them into the original policy equation, which becomes Markov (Tiomkin et al., 2025).

$$\mathcal{E}^{\mathrm{FPR}}_{\mathrm{NM}}\left(a_0^{t'}, s_0^{t'}\right) = \sum_{\substack{s_{t'+1}^{T+1} \\ a_{t'+1}^{T}}} P\left(s_{T+1},\, a_{t'+1}^{T},\, s_{t'+1}^{T} \mid a_0^{t'},\, s_0^{t'}\right) \log\left(\frac{Q\left(s_{t'+1}^{T},\, a_{t'+1}^{T} \mid a_0^{t'},\, s_0^{t'},\, s_{T+1}\right)}{P\left(a_{t'+1}^{T},\, s_{t'+1}^{T} \mid a_0^{t'},\, s_0^{t'}\right)}\right)$$

$$\overset{\text{Markov}}{=} \sum_{\substack{s_{t'+1}^{T+1} \\ a_{t'+1}^{T}}} P\big(s_{T+1}, a_{t'+1}^{T}, s_{t'+1}^{T} \mid a_{t'}, s_{t'}\big) \log\left(\frac{Q\big(s_{t'+1}^{T}, a_{t'+1}^{T} \mid a_{t'}, s_{t'}, s_{T+1}\big)}{P\big(a_{t'+1}^{T}, s_{t'+1}^{T} \mid a_{t'}, s_{t'}\big)}\right)$$

$$= \mathcal{E}^{\mathrm{FPR}}(a_{t'}, s_{t'}) \tag{1}$$

For $\mathcal{E}^{\mathrm{PPR}}_{\mathrm{NM}}\big(s_1^{t'},\, a_0^{t'} \mid s_0\big) = \sum_{\substack{s_{t'+1}^{T+1} \\ a_{t'+1}^{T}}} P\Big(s_{T+1}, a_{t'+1}^{T}, s_{t'+1}^{T} \mid a_0^{t'}, s_0^{t'}\Big) \log\Big(Q\Big(s_1^{t'}, a_0^{t'} \mid s_0, s_{T+1}\Big)\Big)$ applying the chain rule to Q , and Markov to the term outside the logarithm and to the Q second factor:

$$\sum_{\substack{s_{t'+1}^{T+1} \\ a_{t'+1}^{T}}} P\big(s_{T+1}, a_{t'+1}^{T}, s_{t'+1}^{T} \mid a_{t'}, s_{t'}\big) \Big[\, \log Q(s_{t'}, a_{t'} \mid s_0, s_{T+1}) + \log Q\Big(s_1^{t'-1}, a_0^{t'-1} \mid s_0, a_{t'}, s_{t'}\Big) \,\Big] \tag{2}$$

Then we apply distributivity, denote the first term $\mathcal{E}^{\mathrm{PPR}}\big(a_{t'},\, s_{t'} \mid s_0\big)$ and for the second term:

$$\sum_{\substack{s_{t'+1}^{T+1} \\ a_{t'+1}^{T}}} P\big(s_{T+1}, a_{t'+1}^{T}, s_{t'+1}^{T} \mid a_{t'}, s_{t'}\big) \log Q\Big(s_1^{t'-1}, a_0^{t'-1} \mid s_0, a_{t'}, s_{t'}\Big) \tag{3}$$

$$= \log Q\Big(s_1^{t'-1}, a_0^{t'-1} \mid s_0, a_{t'}, s_{t'}\Big) \overbrace{\sum_{\substack{s_{t'+1}^{T+1} \\ a_{t'+1}^{T}}} P\big(s_{T+1}, a_{t'+1}^{T}, s_{t'+1}^{T} \mid a_{t'}, s_{t'}\big)}^{=1} \tag{4}$$

$$\overset{\text{Markov policy}}{=} \log Q\Big(s_1^{t'-1}, a_0^{t'-1} \mid s_0, s_{t'}\Big) =: \text{residual}\Big(s_1^{t'-1}, a_0^{t'-1}, s_0, s_{t'}\Big) \tag{5}$$

We then get $\mathcal{E}^{\mathrm{PPR}}_{\mathrm{NM}}\big(s_1^{t'},\, a_0^{t'} \mid s_0\big) = \mathcal{E}^{\mathrm{PPR}}\big(a_{t'},\, s_{t'} \mid s_0\big) + \text{residual}\Big(s_1^{t'-1}, a_0^{t'-1}, s_0, s_{t'}\Big)$

We can then substitute both into the original self-consistent by Tiomkin et al. (2025):

$$\begin{aligned}
\pi\Big(a_{t'} \mid a_0^{t'-1}, s_0^{t'}\Big) &\propto \exp\Big(\mathcal{E}^{\mathrm{FPR}}_{\mathrm{NM}}\Big(a_0^{t'}, s_0^{t'}\Big) + \mathcal{E}^{\mathrm{PPR}}_{\mathrm{NM}}\Big(s_1^{t'}, a_0^{t'} \mid s_0\Big)\Big) \\
&= \exp\Big(\mathcal{E}^{\mathrm{FPR}}(a_{t'}, s_{t'}) + \mathcal{E}^{\mathrm{PPR}}(a_{t'}, s_{t'} \mid s_0) + \text{residual}\Big(s_1^{t'-1}, a_0^{t'-1}, s_0, s_{t'}\Big)\Big) \\
&\propto \exp\big(\mathcal{E}^{\mathrm{FPR}}(a_{t'}, s_{t'}) + \mathcal{E}^{\mathrm{PPR}}(a_{t'}, s_{t'} \mid s_0)\big)
\end{aligned}$$

where $\propto$ denotes normalization over $a_{t'} \in \mathcal{A}$, and the last step holds because the residual does not depend on $a_{t'}$. Then for any history $\pi\Big(a_{t'} \mid a_0^{t'-1}, s_0^{t'}\Big)$ only depends on $s_{t'}$, and so it is Markov.

Therefore, in ABA if the initial policy set is Markov the policy set resulting from each iteration will be Markov, and thus it inherits the ABA's convergence from Tiomkin et al. (2025) with all of its iterates being Markov. □

For its computation, in practice, Q is never formed over whole trajectories: only $\mathcal{E}^{\mathrm{FPR}} + \mathcal{E}^{\mathrm{PPR}}$ enters the policy equation, and it follows from one forward pass and one backward pass akin to policy evaluation.

Let $\mathcal{S}_{T+1}(s_0)$ be the set of reachable states form s_0 in at most $T+1$ time steps. Then each iteration therefore costs $\mathcal{O}\big(T\,|\mathcal{S}_{T+1}(s_0)|^2|\mathcal{A}|\big)$ time (Appendix A). For comparison the original ABA iteration costs $\mathcal{O}\big((|\mathcal{S}_{T+1}(s_0)||\mathcal{A}|)^T\big)$. Therefore, for Markovian environments, we now have a polynomial-iteration-cost algorithm optimizing in a much lower dimensional probing policy space that computes process empowerment exactly.

3.2 ABA WITH EARLY STOP

A process empowerment approximation's usefulness is not in its numerical precision but in reaching one that is as useful as needed for its use to train learning agents. A good proxy for usefulness is therefore interchangeability in RL training settings with the true value of empowerment. Additionally, usefulness also requires for the approximation to be as cheap as possible in computational terms. Using as a starting point the ABA variant from the previous section, we now introduce a variation that aims to produce a useful empowerment approximation at little cost.

The following variation is built upon two premises. The first premise is that, as seen for the classical Blahut-Arimoto algorithm (Fasoulakis et al., 2025), for ABA convergence to values close to the empowerment value happens in the first iterations while then many iterations are needed for precision. The second one is that there is little usefulness gain between this close to exact values and the exact empowerment value.

Our method incorporates an early stop to our polynomial-iteration-cost ABA to avoid wasting iterations to gain the unuseful precision to transition from quickly obtained close-to-exact empowerment values to the exact ones. Both premises are validated in the experimentation section allowing for a major reduction in computational cost. In the discussion section tolerance is recommended as an early stopping criterion while also being possible to use iteration count.

3.3 ABA WITH WARM START

ABA is an iterative optimization algorithm with global convergence to a unique optimal value from any strictly positive initialisation (Tiomkin et al., 2025). Therefore, a good choice of starting policy close to the empowerment probing policy can allow the algorithm to need less iterations to reach a useful approximation of empowerment.

This reuse is possible because reachable sets nest. Any state reachable from s_0 in t actions is reachable from s_{-1} in $t+1$ actions, by first moving to s_0. The probing policy of s_{-1} at timestep $t+1$ has therefore already been optimized on every state that s_0 can reach at timestep t. In Markovian ABA the probing policies depend only on the timestep and the current state, so this policy transfers to s_0 directly. Processing states along trajectories, as it is commonly done in RL, or in any order that solves a predecessor first, makes a warm-start policy available for every state except the first.

We now incorporate to our variant of ABA warm-start from previous states. For this the probing policy at timestep t for computing the empowerment in a state s_0 is initialized as follows: use the one for timestep t found for empowerment from s_{-1} if available and at $t+1$ otherwise, this last one is guaranteed to be available except for $t = T$ because of the previous reasoning. If an initializing policy is not strictly positive this is fixed using Laplace smoothing. When no warm-start is possible, which can only happen at timestep T, initialization is done to a strictly positive random or uniform distribution.

Warm start can increase the number of iterations for some states. This happens when a state reachable from s_0 at timestep T gives access to many distinct states, which we call the magic portal situation (Figure 1). The optimal probing policies of s_0 steer towards the portal, while those of s_{-1} give it no special weight because its actions lie beyond their horizon. The warm initialization

then starts far from the optimum. The effect grows as the horizon shrinks, because the last timestep makes up a larger share of the probing policies.

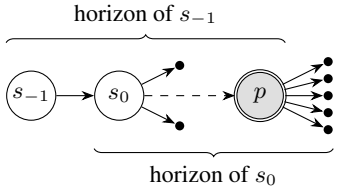

Figure 1: The magic portal situation. Within the horizon of s_{-1}, state p is one final state among the branches of s_0, so the optimal probing policy of s_{-1} has no reason to give the branch leading to p more weight than the others. Within the horizon of s_0, the actions of p lead to many distinct states, so the optimal policy of s_0 steers towards p. Warm start for s_0 can thus be far from the optimum.

4 Experiments and results

In the previous section, we introduced a polynomial-iteration-cost variant of ABA for Markovian environments. This method is not experimentally checked against the original exponential with respect to the horizon one by Tiomkin et al. (2025) for two reasons: its computational cost being much higher, and unfeasible (because of the exponential cost per iteration and search over non-Markovian policies) and because it provides no additional information compared to our polynomial time variation.

In these experiments, we seek to validate: (1) the computational feasibility of our polynomial ABA variant to compute the exact value of process empowerment, (2) demonstrating that implementing early-stop and warm-start in our exact polynomial-iteration ABA preserves most usefulness of empowerment as an internal reward and reduces drastically the time invested in computing it.

To evaluate computational efficiency, the number of iterations of the algorithm, as well as wall-clock time, are used (in Tables 1 and 2). Evaluating the preservation of usefulness is harder, as it is not a problem that can be reduced to comparing values. Taking the assumption that empowerment is used as a reward, a tool from the community of reward-learning is selected: the STARC distance (Skalse et al., 2024), which is a standard way to compare if two rewards signal are similar. STARC distance takes values in $[0, 2]$ with 0 iff two rewards are equivalent, in our experimentation we choose 0.025 as a threshold for almost interchangeable. Taking into account that these signal establishes a relation of ordering or preference between states, we also use Kendall's tau (Kendall, 1938; Kruskal, 1958) as a metric. Both of these metrics are used to compare against the most exact calculation we have of $\mathcal{E}$ (using tolerance $5 \cdot 10^{-8}$) and of approximated empowerment values $\tilde{\mathcal{E}}$. For STARC, these are transformed to reward functions $\mathcal{R}(s, a, s') = \mathcal{E}(s')$ and $\tilde{\mathcal{R}}(s, a, s') = \tilde{\mathcal{E}}(s')$, then we measure STARC distance between. For Kendall's tau, we compare the orderings induced by both $\mathcal{E}$ and $\tilde{\mathcal{E}}$.

The method is compared across three versions. *Exact* corresponds to using the Markovian ABA, with using as a starting policy a random uniform distribution (the least biased) until convergence. *Early Stopping (ES)* does the same, but cuts the convergence ... Finally, *Warm-Start (WS+ES)* does the same as *ES*, but instead of starting from the uniform distribution as a policy, the *best policy* of the parent as was proposed in Section 3.3. No comparison is done with alternate algorithms or approximations in the literature, as none exist to the best of our knowledge.

To experimentally validate these results, we use environments form Minigrid (Chevalier-Boisvert et al., 2023), using as the transition function those taken from the BRIDGE dataset (Laidlaw, 2024; Laidlaw et al., 2023; 2024). We select this environment due to the availability of a computed transition function, removing the need to train one such function. To add a degree of stocasticity, we choose to make the environments stochastic by using sticky actions with $P_{sticky} = 0.25$ as is done in (Laidlaw et al., 2024).

Table 1: Mean ABA iterations per state to compute the empowerment landscape exactly, with early stopping (ES), and with warm start and early stopping (WS+ES); in parentheses, speedup over Exact. Dash (–): no warm start was run as the landscape at iteration 0 is already within STARC distance 0.025. †: 0.1–18.1% of states stopped at the 2000-iteration cap before reaching tolerance

		Horizon $T+1$			
Environment	Method	2	4	6	8
MultiRoom-N2-S4	Exact	112.8	288.7	411.9	598.1†
	ES	1.0 (113×)	1.9 (149×)	2.7 (150×)	3.0 (201×)
	WS+ES	–	1.0 (289×)	1.0 (412×)	1.0 (598×)
SimpleCrossingS9N2	Exact	97.8	310.7	606.8	1275.4†
	ES	1.0 (98×)	3.3 (96×)	3.0 (201×)	4.1 (313×)
	WS+ES	–	1.0 (311×)	1.2 (503×)	1.0 (1274×)
KeyCorridorS3R1	Exact	76.4	250.0	358.7†	555.5
	ES	1.0 (76×)	3.3 (76×)	3.5 (101×)	3.2 (173×)
	WS+ES	–	1.0 (250×)	1.0 (359×)	1.0 (554×)
DoorKey-5x5	Exact	88.6	267.1	402.9	623.6†
	ES	1.0 (89×)	3.1 (86×)	3.4 (118×)	4.0 (156×)
	WS+ES	–	1.0 (267×)	1.0 (403×)	1.0 (619×)
FourRooms	Exact	97.7	318.2	613.4	1112.3†
	ES	1.1 (90×)	3.9 (81×)	2.9 (211×)	3.0 (371×)
	WS+ES	2.1 (46×)	1.5 (207×)	2.5 (249×)	1.3 (872×)
Unlock	Exact	113.6	372.5	564.5	1072.7†
	ES	5.3 (21×)	10.2 (36×)	5.9 (96×)	6.9 (156×)
	WS+ES	6.4 (18×)	2.7 (139×)	2.3 (241×)	1.9 (564×)

5 DISCUSSION

The results highlight the faster efficiency of the technique without compromising results for the chosen environments. The modifications with Early Stopping and Warm Start reduce the time substantially more, making it practical to compute empowerment for every state (which in some cases, such as FourRooms with horizon 8, would mean 2.2 versus 0.003 seconds per state). In particular, the reduced costs enable much higher empowerment horizon values. The near total reduction in error premise in the first iterations is verified as it can be seen in 2. Both STARC 3 and Kendall's τ 4 also show most usefulness gains in the first iterations.

In our experimentation, we set as a stopping criterion the comparison with respect to the accurate computation. This is not possible in a practical setting, so instead a training hyperparameter for number of iterations (the numbers suggesting around 10 at most) or, better yet, tolerance (minimum change in value to perform a new iteration).

A particular noteworthy result is that warm-start does not always improve a uniform policy 5. This happens only when considering low horizon values, as the tendency appears to become the expected one at higher horizon. We hypothesise the reason why this happens is that the uniform policy is already good (being in the centre of the optimisation space, closer to all options); and the warm-start policy may be further away. With low horizon, adding the extra level of exploration (from taking the next state) may drastically change which is the best policy; whereas longer horizons will already have a larger exploration tree with states using a correct policy for most states. Warm-start always incurs a risk coming from having high-value states at the end that were not visible to the predecessor state (a 'magic portal' state where each action almost determines the next state, giving very high empowerment).

Beyond the direct results of this paper, one of the limitations of computing the empowerment is this way is requiring the transition function, and this goes against using empowerment to learn. However, we envision that, with the lowered computational cost, empowerment can be used as an exploration signal *during* training. In this supposition, empowerment is computed after every few updates of

Table 2: Mean wall-clock time per state (ms) to compute the empowerment landscape exactly, with early stopping (ES), and with warm start and early stopping (WS+ES); in parentheses, speedup over Exact. Dash (–): no warm start was run as the landscape at iteration 0 is already within STARC distance 0.025. †: 0.1–18.1% of states stopped at the 2000-iteration cap before reaching tolerance

		Horizon $T+1$			
Environment	Method	2	4	6	8
MultiRoom-N2-S4	Exact	1.54	14.6	83.9	350†
	ES	0.181 (8.5×)	0.158 (92×)	0.708 (119×)	1.91 (183×)
	WS+ES	–	0.110 (132×)	0.378 (222×)	0.834 (420×)
SimpleCrossingS9N2	Exact	0.774	18.3	214	1671†
	ES	0.0212 (37×)	0.278 (66×)	1.17 (183×)	5.17 (323×)
	WS+ES	–	0.150 (122×)	0.626 (341×)	2.44 (685×)
KeyCorridorS3R1	Exact	0.597	13.1	80.2†	385
	ES	0.0198 (30×)	0.233 (56×)	0.885 (91×)	2.41 (160×)
	WS+ES	–	0.111 (117×)	0.412 (195×)	1.47 (261×)
DoorKey-5x5	Exact	0.851	13.4	82.3	420†
	ES	0.0213 (40×)	0.260 (52×)	0.858 (96×)	2.93 (143×)
	WS+ES	–	0.123 (109×)	0.349 (236×)	1.10 (382×)
FourRooms	Exact	0.762	20.5	247	2221†
	ES	0.0424 (18×)	0.366 (56×)	1.48 (168×)	6.47 (343×)
	WS+ES	0.0600 (13×)	0.192 (107×)	1.24 (200×)	3.32 (669×)
Unlock	Exact	0.871	21.4	184	1461†
	ES	0.111 (7.8×)	0.722 (30×)	2.15 (85×)	10.5 (139×)
	WS+ES	0.0691 (13×)	0.261 (82×)	1.04 (176×)	3.67 (398×)

the transition function as the latter is learnt. Then, empowerment updates the reward signal, guiding the exploration of the agent once again. This would have been impossible due to the previous cost, but with the current one it is much more reasonable. This remains something to be studied in future work, but Volpi & Polani (2023) already explore mixing the empowerment with other rewards to explore this relationship.

6 Conclusion

Empowerment is a measure that may provide much value to the intelligent agents community, especially for intrinsic motivation purposes. In this paper, we have presented key optimisations to the computation of Process (closed-loop) Empowerment for non-continuous environments. Our contributions are as follows:

- We prove that by assuming the Markov Property, the search space of the policies goes from exponential to polynomial with respect to the horizon.
- We propose *early-stopping* techniques, as the gains of accuracy are not worth the time for iterations past the very first ones.
- We also propose a *warm-start* to the estimation algorithm via re-using elements used in computing empowerment of previous states.
- We validate these findings experimentally: even as the modifications drastically cut down time, the approximated empowerment values maintain the order of the exact and are nearly interchangeable with exact as reward signals.

The method proposed in this paper provides an extremely cheap way to compute empowerment. We believe this efficiency will remove a bottleneck that has been precluding its application to training learning agents with RL, and experimenting for further uses of this measure in new, complex environments.

AI USE STATEMENT

In this work, we used generative AI tools for providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, provide feedback on research methodology or experiments, implement methods, clean and reformat dataset and formulate mathematical claims (mainly by helping refine the proposition and theorem statement) retrieval of related work and interpret results. We have not used generative AI tools for generating synthetic data sets, help develop theoretical models or conceptual frameworks, propose or refine hypothesis, assist with translation. Additionally, we used generative AI tools to help rewrite sections of this work and draft Appendix B on the experimental details. The use of generative AI tools to Support qualitative and thematic data analysis is not applicable to this work. We have reviewed all AI-assisted work. All mathematical claims, statements and proofs by AI have been verified, reviewed and rewritten by the authors, implemented methods have gone through tests including comparisons to known empowerment values in certain environments, cleaning and reformatting datasets was done using code methods to reformat them, verifications were carried using appropriate tests. Tests for empowerment computation included checking against toy examples analytically solved, for dataset reformatting (BRIDGE MDPs state expansion), this was checked against public bridge information on the characteristics of this. Feedback on research methodology or experiments has been used occasionally, treated critically and used to help refine experiments to the best standards, result interpretation advice has been discussed by the authors of this work before being incorporated to it, related work retrieved with AI has been checked to verify relevance and importance for this work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The source code for the methods, experimentation and plot production in this work are in an anonymised repository which will be added for publication. For review process, the code along with instructions to reproduce all results can be found in the supplementary material. Proofs to proposition 1 and theorem 2 are already available in the main text. More detail on the practical computations needed for ABA execution can be found in A. Details on the experimental setup can be found in B.

REFERENCES

S. Arimoto. An algorithm for computing the capacity of arbitrary discrete memoryless channels. *IEEE Transactions on Information Theory*, 18(1):14–20, January 1972. ISSN 1557-9654. doi: 10.1109/TIT.1972.1054753. URL https://ieeexplore.ieee.org/document/1054753.

R. Blahut. Computation of channel capacity and rate-distortion functions. *IEEE Transactions on Information Theory*, 18(4):460–473, July 1972. ISSN 1557-9654. doi: 10.1109/TIT.1972.1054855. URL https://ieeexplore.ieee.org/document/1054855.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo de Lazcano, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & Miniworld: Modular & Customizable Reinforcement Learning Environments for Goal-Oriented Tasks, June 2023. URL http://arxiv.org/abs/2306.13831. arXiv:2306.13831 [cs].

Michail Fasoulakis, Konstantinos Varsos, and Apostolos Traganitis. Revisit the Arimoto-Blahut algorithm: New Analysis with Approximation, July 2025. URL http://arxiv.org/abs/2407.06013. arXiv:2407.06013 [cs.IT].

Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational Intrinsic Control. February 2017. URL https://openreview.net/forum?id=Skc-Fo4Yg.

Maximilian Karl, Philip Becker-Ehmck, Maximilian Soelch, Djalel Benbouzid, Patrick van der Smagt, and Justin Bayer. Unsupervised Real-Time Control Through Variational Empowerment. In Tamim Asfour, Eiichi Yoshida, Jaeheung Park, Henrik Christensen, and Oussama Khatib (eds.), *Robotics Research*, pp. 158–173, Cham, 2022. Springer International Publishing. ISBN 978-3-030-95459-8. doi: 10.1007/978-3-030-95459-8_10.

M. G. Kendall. A NEW MEASURE OF RANK CORRELATION. *Biometrika*, 30(1-2):81–93, June 1938. ISSN 0006-3444. doi: 10.1093/biomet/30.1-2.81. URL https://doi.org/10.1093/biomet/30.1-2.81.

A.S. Klyubin, D. Polani, and C.L. Nehaniv. Empowerment: A Universal Agent-Centric Measure of Control. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pp. 128–135, Edinburgh, Scotland, UK, 2005. IEEE. ISBN 978-0-7803-9363-9. doi: 10.1109/CEC.2005.1554676. URL http://ieeexplore.ieee.org/document/1554676/.

William H. Kruskal. Ordinal Measures of Association. *Journal of the American Statistical Association*, 53(284):814–861, December 1958. ISSN 0162-1459. doi: 10.1080/01621459.1958.10501481. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1958.10501481. _eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1958.10501481.

Taehwan Kwon. Variational intrinsic control revisited. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=P0p33rgyoE.

Cassidy Laidlaw. Bridge, June 2024. URL https://doi.org/10.5281/zenodo.12556298.

Cassidy Laidlaw, Stuart Russell, and Anca Dragan. Bridging RL Theory and Practice with the Effective Horizon. In *Advances in Neural Information Processing Systems (NeurIPS)*, November 2023. URL https://openreview.net/forum?id=Lr2swAfwff.

Cassidy Laidlaw, Banghua Zhu, Stuart Russell, and Anca Dragan. The effective horizon explains deep rl performance in stochastic environments. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 14370–14399, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/3e18f7152b5e1696e73722dc5121b6f2-Paper-Conference.pdf.

Shakir Mohamed and Danilo Jimenez Rezende. Variational Information Maximisation for Intrinsically Motivated Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/hash/e00406144c1e7e35240afed70f34166a-Abstract.html.

Stuart Jonathan Russell, Peter Norvig, John F Canny, Jitendra M Malik, and Douglas D Edwards. *Artificial intelligence: a modern approach*, volume 2. Prentice hall Englewood Cliffs, NJ, 1995.

Christoph Salge, Cornelius Glackin, and Daniel Polani. *Empowerment–An Introduction*, pp. 67–114. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. ISBN 978-3-642-53734-9. doi: 10.1007/978-3-642-53734-9_4. URL https://doi.org/10.1007/978-3-642-53734-9_4.

Joar Skalse, Lucy Farnik, Sumeet Ramesh Motwani, Erik Jenner, Adam Gleave, and Alessandro Abate. STARC: A General Framework For Quantifying Differences Between Reward Functions, December 2024. URL http://arxiv.org/abs/2309.15257. arXiv:2309.15257 [cs.LG].

Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=B1x62TNtDS.

Jinyeop Song, Jeff Gore, and Max Kleiman-Weiner. Estimating the empowerment of language model agents. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=r8qbhgGHnC.

Stas Tiomkin, Ilya Nemenman, Daniel Polani, and Naftali Tishby. Intrinsic motivation in dynamical control systems. *PRX Life*, 2:033009, Aug 2024. doi: 10.1103/PRXLife.2.033009. URL https://link.aps.org/doi/10.1103/PRXLife.2.033009.

Stas Tiomkin, Christoph Salge, and Daniel Polani. Process empowerment for robust intrinsic motivation. *J. Phys. Complex.*, 6(3):035011, August 2025. ISSN 2632-072X. doi: 10.1088/2632-072X/adf2ec. URL https://doi.org/10.1088/2632-072X/adf2ec.

Nicola Catenacci Volpi and Daniel Polani. Goal-directed empowerment: Combining intrinsic motivation and task-oriented behavior. *IEEE Transactions on Cognitive and Developmental Systems*, 15(2):361–372, 2023. doi: 10.1109/TCDS.2020.3042938.

Ruihan Zhao, Kevin Lu, Pieter Abbeel, and Stas Tiomkin. Efficient empowerment estimation for unsupervised stabilization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=u2YNJPcQlwq.

A Practical computation of Markovian ABA

Writing $Q(s_{t'}, a_{t'} \mid s_0, s_{T+1})$ and the ratio inside $\mathcal{E}^{\mathrm{FPR}}$ with Bayes' rule, the terms in $\log(P(s_{T+1} \mid s_{t'}, a_{t'}))$ cancel, and

$$\mathcal{E}^{\mathrm{FPR}}(a_{t'}, s_{t'}) + \mathcal{E}^{\mathrm{PPR}}(a_{t'}, s_{t'} \mid s_0) = \log(P(s_{t'} \mid s_0)) + \log(\pi(a_{t'} \mid s_{t'})) + G(a_{t'}, s_{t'})\,,$$

where the first term cancels in the normalization and G follows the policy-evaluation recursion

$$\begin{aligned} G(a_T, s_T) &= D_{\mathrm{KL}}(P(S_{T+1} \mid s_T, a_T) \parallel P(S_{T+1} \mid s_0))\,, \\ G(a_t, s_t) &= \sum_{\substack{s_{t+1} \\ a_{t+1}}} P(s_{t+1} \mid s_t, a_t)\, \pi(a_{t+1} \mid s_{t+1})\, G(a_{t+1}, s_{t+1})\,, \quad t < T. \end{aligned}$$

One forward pass for $P(s_t \mid s_0)$ and one backward pass for G each read every transition probability once per timestep, which gives the cost stated in Section 3.1.

The cost is established with respect to the number of reachable states in at most $T{+}1$ steps $\mathcal{S}_{T+1}(s_0)$. This is a product of the locality property of empowerment (Klyubin et al., 2005; Salge et al., 2014) which crucially allows computing it much more efficiently and with only local knowledge of the transition function.

Table 3: Environments, from BRIDGE. $|S|$: states of the deterministic MDP; $|S_{\text{sticky}}|$: states once extended with the previous action for sticky actions ($p = 0.25$); $|A|$: actions; Term.: absorbing terminal states; No pred.: states without a predecessor, which WS+ES computes from a cold start (–: no warm start was run); h: horizons computed.

Environment	$\lvert S\rvert$	$\lvert S_{\text{sticky}}\rvert$	$\lvert A\rvert$	Term.	No pred.	h
MultiRoom-N2-S4	60	335	6	1	1	2, 4, 6, 8
SimpleCrossingS9N2	132	770	6	1	1	2, 4, 6, 8
KeyCorridorS3R1	168	962	6	1	0	2, 4, 6, 8
DoorKey-5x5	264	1491	6	1	0	2, 4, 6, 8
FourRooms	1036	6093	6	1	0	2, 4, 6, 8
Unlock	1084	6117	6	1	0	2, 4, 6, 8

B Experimental details

B.1 Environments

We use six MiniGrid environments (Chevalier-Boisvert et al., 2023), taken from BRIDGE (Laidlaw, 2024; Laidlaw et al., 2023; 2024), a dataset that stores common reinforcement learning benchmarks as tabular MDPs: for every state and action, it gives the next state, or marks that the episode ends. Table 3 lists the six environments and their sizes.

Sticky actions. The BRIDGE environments are deterministic. To make them stochastic, we use sticky actions as done in Laidlaw et al. (2024): with probability $p = 0.25$, the environment ignores the chosen action and repeats the previous one. The next state then depends on the previous action too, so we add it to the state: a state of the expanded MDP is a pair (s, a_-) of an original state and the last action taken. From (s, a_-), choosing a leads to $(T(s, a), a)$ with probability $1 - p$ and to $(T(s, a_-), a_-)$ with probability p, where T is the original transition table. We keep only the pairs reachable from the start state, with any previous action. This gives the $|S_{\text{sticky}}|$ column of Table 3.

Terminal states. BRIDGE marks where an episode ends but does not store the state it ends in. We send every episode-ending transition to a single absorbing state, counted as one extra state (column Term. of Table 3). Its empowerment is zero: once there, no action changes what happens next.

B.2 Empowerment computation

For each environment and horizon $h \in \{2, 4, 6, 8\}$, we compute the empowerment of every state of the expanded MDP, in bits, with the Blahut–Arimoto iteration described in the main text. Each state starts from the uniform probing policy and stops when its empowerment changes by less than $\text{tol} = 5 \cdot 10^{-8}$ between two consecutive iterations, or after 2000 iterations. These values form the reference landscape E^*, against which every other landscape is compared. Table 4 lists all settings.

Error bounds. An error bound ε is a threshold on the change between iterations: a state *reaches* ε at the first iteration k with $|E^{(k)} - E^{(k-1)}| < \varepsilon$, and the landscape at ε takes each state's value at that iteration. We use 37 bounds: the 50 values evenly spaced on a logarithmic scale between 10^{-10} and 1, which also set the warm-start depth (below), keeping those of at least tol, plus tol itself. The cost of a bound is the number of iterations, or the time, a state needs to reach it, averaged over all states.

Early stopping. ε^* is the loosest error bound whose landscape is within STARC distance 0.025 of E^*. Early stopping (ES) stops every state at ε^*.

Warm start. Each state is assigned one predecessor, a state with a transition into it, drawn uniformly at random (seed 0). Blahut–Arimoto runs at the predecessor, from the uniform policy, until the cold run's ε^*, and the resulting probing policy becomes the initial policy of the state. Where that policy has no entry for a state at some step, we reuse the entry of the same state at the nearest

Table 4: Settings shared by all experiments.

Setting	Value
Sticky-action probability p	0.25
Horizons h	$2, 4, 6, 8$
Empowerment units	bits ($\log_2$)
Convergence tolerance tol	$5 \cdot 10^{-8}$
Iteration cap per state	2000
Error bounds ε	37, log-spaced from 1 down to tol
STARC bound defining ε^*	0.025
STARC discount γ	0.99
Kendall rounding δ	10^{-6} bits
Warm-start predecessor seed	0
Warm-start smoothing: threshold, λ	10^{-3}, 0.01

other step, or the uniform policy if there is none. Blahut–Arimoto can never raise a probability that starts at zero, so each action distribution with an entry below 10^{-3} is smoothed by adding $\lambda = 0.01$ to every entry and renormalising. States without a predecessor start from the uniform policy. The warm-started state then stops at its own ε^* (WS+ES). The predecessor's iterations are not counted, since computing the whole landscape computes the predecessor anyway. When the cold landscape is already within the STARC bound at every error bound, no warm start is run; this happens at $h = 2$ in four of the six environments.

Quality measures. We compare each landscape with E^* in three ways. The STARC distance (**?**) uses discount $\gamma = 0.99$, the empowerment of the next state as the reward of a transition, and the uniform policy as reference policy. Kendall's τ_b over states is computed after rounding both landscapes to multiples of $\delta = 10^{-6}$ bits, so that states whose values differ only by numerical noise count as ties. The mean absolute error is $\frac{1}{|S|} \sum_s |E_\varepsilon(s) - E^*(s)|$, in bits.

Hardware and software. All runs used a single thread on one core of a laptop with an AMD Ryzen 5 5500U processor and 8 GB of memory, running Ubuntu 24.04, with Julia 1.12.6 and Python 3.10. States are computed one after another and timed individually; two untimed warm-up passes keep Julia's compilation out of the timings. The complete set of runs took 15.7 hours, 13.7 of them at $h = 8$. This long experimentation time is due to the need to compute the whole empowerment landscape in order to use metrics STARC and τ.

B.3 Additional results

Table 5 counts, for each environment and horizon, the states for which the warm start needed more iterations than the cold start, each stopped at its own ε^*. Figures 2, 3 and 4 show how the mean absolute error, the STARC distance and Kendall's τ of the landscape change with the mean number of iterations per state, for every environment and horizon.

Table 5: Number of states (in parentheses, share of all states) for which the warm start with early stopping (WS+ES) needed more iterations than early stopping alone (ES), each counted at its own ε^*. ES (early stop): each state stops at ε^*, the loosest error bound whose landscape is within STARC distance 0.025 of the converged landscape. WS+ES (warm start + early stop): each state starts from the probing policy of a predecessor drawn uniformly at random (seed 0), computed to the ES bound, with boundary smoothing $\lambda = 0.01$ and floor 0.001, then stops early as in ES; the predecessor's own computation is not counted. STARC uses discount $\gamma = 0.99$ with empowerment as the reward of the next state, against the cold-start landscape converged to tol. –: no warm start was run, because the cold landscape is already within STARC distance 0.025 at every error bound, leaving nothing to warm start from.

	Horizon h			
Environment	2	4	6	8
MultiRoom-N2-S4	–	0 (0.0%)	0 (0.0%)	0 (0.0%)
SimpleCrossingS9N2	–	0 (0.0%)	0 (0.0%)	0 (0.0%)
KeyCorridorS3R1	–	0 (0.0%)	0 (0.0%)	0 (0.0%)
DoorKey-5x5	–	0 (0.0%)	0 (0.0%)	0 (0.0%)
FourRooms	6085 (99.9%)	0 (0.0%)	1267 (20.8%)	0 (0.0%)
Unlock	5134 (83.9%)	0 (0.0%)	0 (0.0%)	0 (0.0%)

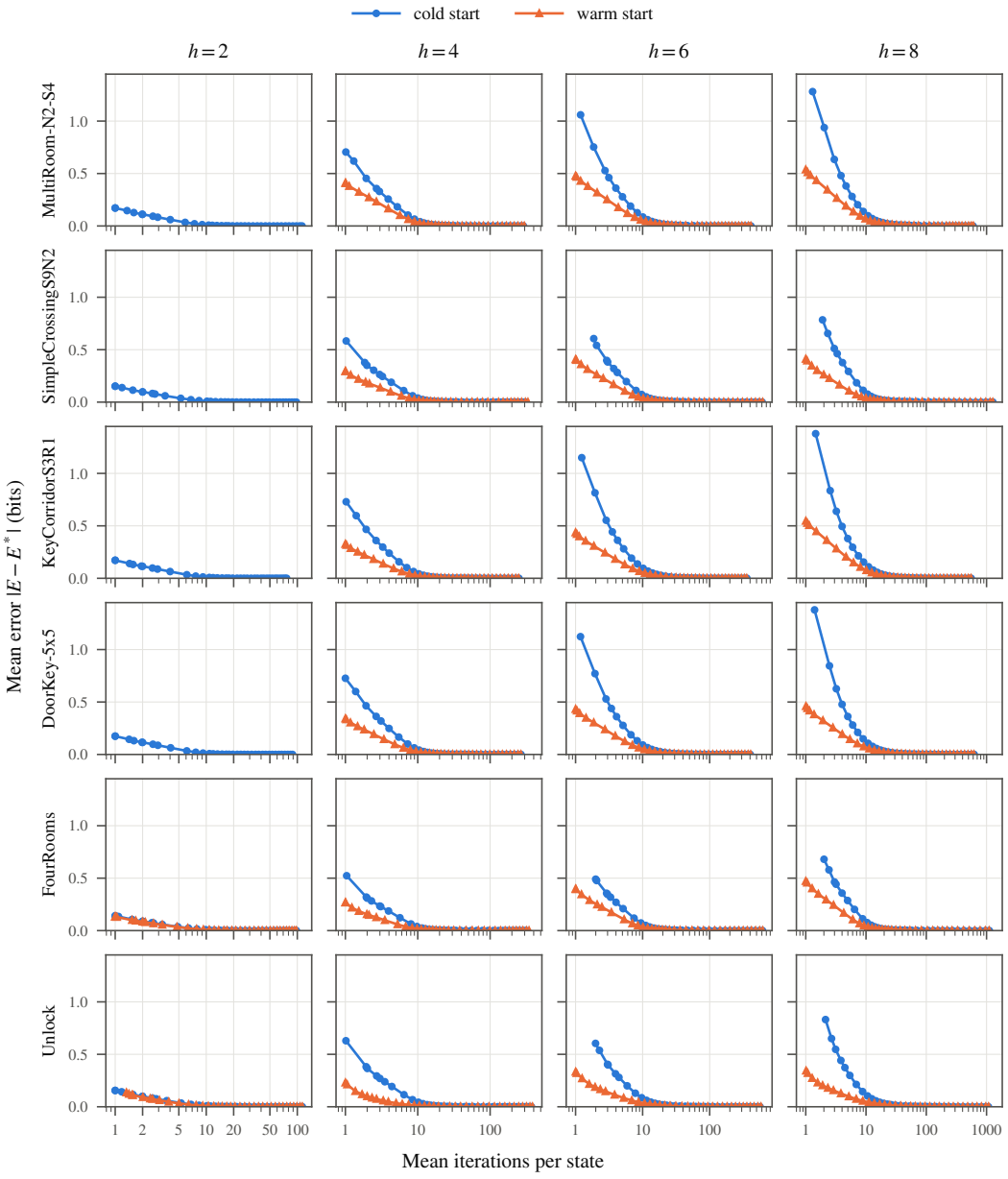

Figure 2: Mean absolute error $\frac{1}{|S|}\sum_s |E_\varepsilon(s) - E^*(s)|$, in bits, of the landscape at each error bound ε with respect to the converged landscape E^* (cold start run to tol), against the mean number of iterations a state needs to reach it, for every environment (rows) and horizon (columns); cold start (blue circles) and warm start (orange triangles). The horizontal axes are logarithmic. Error bounds: 37 values of ε from the warm start's own grid, restricted to $[\mathrm{tol}, 1]$; a state reaches ε at its first iteration with $|E^{(k)} - E^{(k-1)}| < \varepsilon$. The vertical axes are linear and start at 0. The warm start initialises each state with the probing policy of a predecessor drawn uniformly at random (seed 0), computed to the ES bound, with boundary smoothing $\lambda = 0.01$ and floor 0.001; the predecessor's own computation is not counted. Panels with a cold curve only: no warm start was run, the cold landscape being within the STARC bound at every error bound.

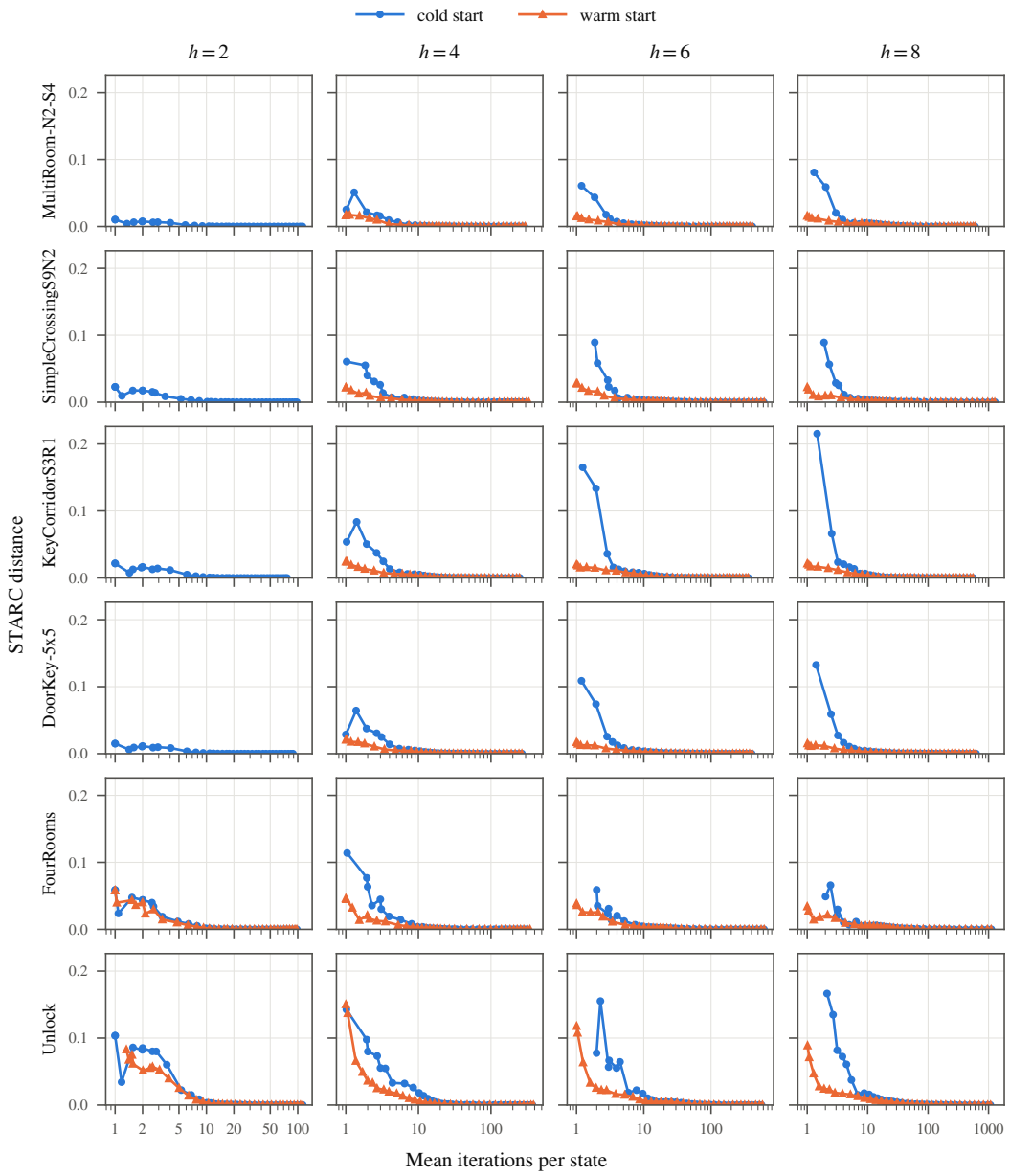

Figure 3: STARC distance between the landscape at each error bound ε and the converged landscape, against the mean number of iterations a state needs to reach it, for every environment (rows) and horizon (columns); cold start (blue circles) and warm start (orange triangles). The horizontal axes are logarithmic. Error bounds: 37 values of ε from the warm start's own grid, restricted to $[\mathrm{tol}, 1]$; a state reaches ε at its first iteration with $|E^{(k)} - E^{(k-1)}| < \varepsilon$. The distance axes are linear and start at 0. STARC uses discount $\gamma = 0.99$ with empowerment as the reward of the next state, against the cold-start landscape converged to tol. The warm start initialises each state with the probing policy of a predecessor drawn uniformly at random (seed 0), computed to the ES bound, with boundary smoothing $\lambda = 0.01$ and floor 0.001; the predecessor's own computation is not counted. Panels with a cold curve only: no warm start was run, the cold landscape being within the STARC bound at every error bound.

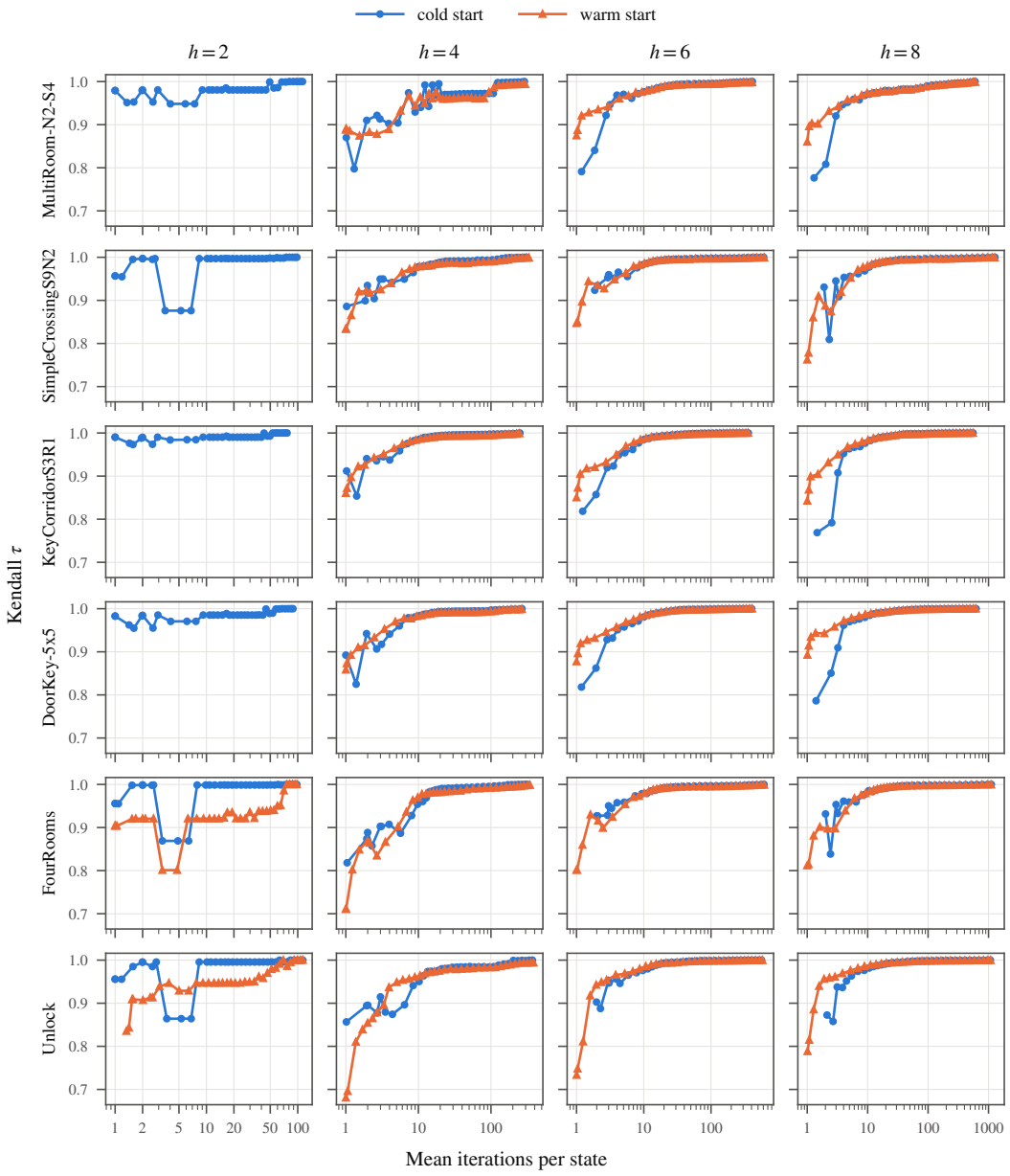

Figure 4: Kendall τ between the landscape at each error bound ε and the converged landscape, against the mean number of iterations a state needs to reach it, for every environment (rows) and horizon (columns); cold start (blue circles) and warm start (orange triangles). The horizontal axes are logarithmic. Error bounds: 37 values of ε from the warm start's own grid, restricted to $[\mathrm{tol}, 1]$; a state reaches ε at its first iteration with $|E^{(k)} - E^{(k-1)}| < \varepsilon$. Kendall τ is computed after rounding both landscapes to multiples of $\delta = 10^{-6}$ bits, so states closer than δ count as tied. The vertical axes are linear. The warm start initialises each state with the probing policy of a predecessor drawn uniformly at random (seed 0), computed to the ES bound, with boundary smoothing $\lambda = 0.01$ and floor 0.001; the predecessor's own computation is not counted. Panels with a cold curve only: no warm start was run, the cold landscape being within the STARC bound at every error bound.