Journal of Physics: Complexity

OPEN ACCESS

RECEIVED
12 February 2025

REVISED
4 July 2025

ACCEPTED FOR PUBLICATION
22 July 2025

PUBLISHED
26 August 2025

Original Content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence.

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

PAPER

Process empowerment for robust intrinsic motivation

Stas Tiomkin[1,*], Christoph Salge[2] and Daniel Polani[2]

[1] Computer Science Department, Whitacre College of Engineering, Texas Tech University, Lubbock, TX, United States of America
[2] Adaptive Systems Research Group, Department of Computer Science, University of Hertfordshire, Hertfordshire, United Kingdom
[*] Author to whom any correspondence should be addressed.

E-mail: stas.tiomkin@ttu.edu, c.salge@herts.ac.uk and d.polani@herts.ac.uk

Keywords: information processing in dynamical control systems, robust intrinsic motivation, feedback-aware empowerment, information capacity, computation in complex systems, feedback complexity

Abstract

Information processing in dynamical control systems influences the properties of the perception-action loop in natural and artificial agents. The ability to causally affect environment by agent's actions is crucial for learning meaningful behavior and survival. Empowerment is an information-theoretic approach to intrinsically discover this causality between actions and observations without externally provided domain expertise such as a reward function. This form of artificial intrinsic motivation has been successfully demonstrated to lead to the emergence of meaningful behavior in various domains ranging from robotics to transportation. The original formulation of the empowerment principle is based on the information flow from open-loop actions to future observations. This is not robust to randomness and unpredictable perturbations in environments with structures that require careful maneuvering. In this work we define a feedback-aware empowerment variant, called *process empowerment* and derive a solution given by self-consistent equations which can be used for its numerical evaluation. Process empowerment proves to be a robust intrinsic motivation in a paradigmatic proof-of-concept example ('Windy Bridge'), and in scenarios with obstacles and noisy perturbation ('Hallway') and with occasional adversarial action by an oracle agent ('Race'). It demonstrates superior robustness in dealing with noisy environments in delicate situations, and allows transferring solutions for deterministic problems into a noisy, disruptive and occasionally adversarial variant of the problem, through 'empowerment cushioning'.

1. Introduction

Recent successes in artificial intelligence can not disguise substantial gaps in the understanding of how intelligent behavior can emerge in complex operation spaces. This is particularly striking when one considers the discrepancy between the effort in terms of massive data and energy use [5, 14] and, in contrast to that, the parsimony in terms of energy and data acquisition by organisms [4, 22]. To understand how the latter accomplish their tasks, one has to consider that an organism interacting with its environment is not passively exposed to a flow of data, but is instead situated in a context and environment with which it constantly interacts.

As a consequence, when it operates in that environment, it is not merely a unidirectional computational device that transforms an input data stream into an output data stream (and be this transformation ever so complex), but rather inherently interacts actively with that environment, impinging on specific degrees of freedom and eliciting particular responses. In other words, a substantial part of such an agent's information processing does not just consist in trying to construct a 'digital twin' of the environmental dynamics for its purposes, but rather to elicit the specific reactions relevant to it and, by effectively becoming part of that environment's dynamics, co-opt the latter for its information-processing purposes. Stated differently, part of the success of organismic agents emerges from causing the environment to contribute to their information processing and being aware of that [7].

© 2025 The Author(s). Published by IOP Publishing Ltd

This, in turn, has become a key insight in the development of artificial agents that mimic the principles found in biology. If one wishes to mimic biological flexibility and information parsimony, so the assumption, it is essential to consider the agent as situated, and specifically, embodied in its environment.

As such, environmental dynamics is increasingly made part of methodologies supporting the decision-making of agents. Among these, in the last decades, a particular class of methods, namely *intrinsic motivations* have taken a particularly central role.

Such intrinsic motivations are becoming increasingly important to define incentives for agents when reward structures are not available, too expensive to determine for an agent with a large state space or unnatural to postulate *a priori*. Among these, the subclass of information theory-based intrinsic motivations is of particular interest due to its universality. In the present paper, we will concentrate on a specific such incentive, *empowerment*.

Empowerment measures the maximum *potential* effect that an agent can possibly have on the environment via its actions, when starting in a given state. Formally, empowerment is given by the channel capacity of the external part of the action–perception loop, concretely, the channel between actions (or finite sequences of actions) that can be chosen in the present state and the effects these actions happen to cause in the world. This measure quantifies the influence that the agent has the potential to exert on its sensorimotor niche in the near future; and it only considers that part of the influence that the agent can itself sense. In a way, it measures the size of the agent's 'bubble of autonomy'. Another way to interpret empowerment is that it measures how much freedom of choice the agent has in controlledly selecting its future (see also [2]).

Empowerment, denoted by $\mathcal{E}(s')$, is a scalar quantity, measured in bits, which gives a value for each state s' the agent finds itself in. When using it as an intrinsic motivation to generate behavior, it substitutes for a utility function in the specific state s'[3]. In a concrete state s of a discrete world the agent will take the action that moves it to the successor state s' with the highest empowerment value; in a continuous world, it will take the action that moves it along the largest empowerment gradient.

In summary, there are two types of actions at play here: 1. the action that the agent *actually* takes, which is given, at each step, by the action that locally and greedily maximizes the immediate empowerment gain or gradient; and 2. the probing action sequences to determine the empowerment values themselves, and which consist of the *potential* future actions that the agent *could* take in the given state s' under consideration. We emphasize that these potential probing action sequences are never actually carried out by the agent: they only indicate the *potential* to change the world when starting in state s'. When using empowerment to direct behavior, they are wrapped up as part of the calculation of the empowerment value $\mathcal{E}(s')$ of the given successor states under consideration.

Substantial past work on empowerment as intrinsic motivation considers the *open-loop* version of empowerment [6, 10, 13, 15, 16, 19, 20, 24]. That is, when evaluating what the agent could do, they consider the repertoire of the available actions or a selection of possible sequences of actions, for instance, up to a certain maximum length, but only as as predetermined 'action scripts': once started in s', each such action sequence is sequentially executed in fixed succession until it terminates. In open-loop empowerment, the potential to change the world is measured with respect to how strongly the agent can affect the world by the choice of such different fixed action sequences. Since these action sequences are fixed at the beginning of the probing, they are 'reeled off' as prespecified without taking into account what may happen during the probing. Basically, the action sequences can be seen as equivalent to atomic 'super-action' added to the basic action repertoire. The term 'open-loop' refers to the fact that the probing action sequences are run 'single-mindedly', without taking into account potential changes in the world. Despite this apparent limitation, open-loop empowerment has been shown to work effectively as an intrinsic motivation in a wide variety of cases without any special provisions.

While successful in many scenarios, in a highly dynamic environment (e.g. one with substantial noise or short-term reactivity such as from other agents), 'naive' open-endedness will give a misleading reflection of how much the agent indeed could impact its environment in a controllable fashion in the near future. The limitations of open-loop empowerment have been discussed before [18, 21]. To address these limitations the authors in [8, 11] established bounds and estimates for a particular formulation of closed loop-type empowerment via the notion of implicit and explicit options. These important studies demonstrate the need for closing the loop in the empowerment computation; they show the usefulness of doing so by deriving bounds and estimates for the closed-loop calculation in a number of scenarios. However, in this work, feedback is not an inherent part of an objective and/or its bounds, but rather acts as the means for a

[3] Strictly spoken, empowerment is only a 'pseudo-utility', as it is not certainty-equivalent, see also [23]; nonetheless, in practice it is almost always used directly in lieu of a given utility function to produce *intrinsically motivated* behavior, without noticeable detriment.

derivation of policy in the general framework of reinforcement (RL) learning. In [17], the choice of controllers inducing variance in outcomes is suggested as another form of feedback-aware empowerment.

In the present paper, we aim at a principled generalization of the original open-loop empowerment quantity towards closed-loop (or as we will sometimes also say), *process* empowerment, i.e. one that permits the probing to consider feedback while choosing the actions. We still aim to capture the original spirit of measuring the potential influence of the agent on the world in analogy to the open-loop formulation, but now under the relaxed condition of allowing the agent to react to the world. In particular, we seek to obtain an interpretable and precise value (not just a bound), and also to provide suitable methods to compute this quantity.

We reiterate an earlier remark that empowerment consists of the information-theoretic channel capacity of the external action–perception channel of the agent. Now, even while closing the loop introduces feedback, note that the problem does not simply reduce to the feedback capacity of the action–perception loop in the sense of traditional information theory: rather, it needs to be treated more carefully. The present paper addresses this question in detail and studies several remarkable consequences and insights of generalizing empowerment to the closed-loop (process) case.

The contribution of our current work is to rigorously define an exact objective for closed-loop/process empowerment and to explore its properties with regard to robustness and noise compensation in the probing actions as compared to open-loop empowerment. This rigorous approach will allow us to study the essential properties of process empowerment rather them its bounds or estimates. Our approach provides new insights about robust intrinsic motivation, allows to validate our formulation by designing experiments addressing the difference between closed-loop and open-loop strategies. Furthermore, our solution admits a decomposition into 'future process empowerment' and 'past-to-future predictability', which provides a new understanding of intrinsic motivation in stochastic dynamics with unpredictable perturbations. The scope of the current paper focuses on the formal definition of process empowerment and on the demonstration of its essential properties for robust intrinsic motivation. Here, we limit ourselves to discrete environments and will extend the approach to continuous environments in future work.

In the paper, we will begin in section 2 with a review of open-loop empowerment which we will formulate in a way suitable for its later generalization. We will there also briefly highlight its limitation, though its consequences will then be discussed in detail in the Experiments and Discussion sections (sections 4 and 5).

In section 3 we introduce the generalization of traditional empowerment to *process empowerment*, where now the reaction of the environment is incorporated in the selection of the probing policies; we there develop the intuition for the quantity and discuss its properties. Importantly, we will show, as a 'sanity check', that in the special case of deterministic dynamics, where the actions chosen are sufficient to reconstruct the agent's trajectory perfectly, process empowerment precisely coincides with the traditional open-loop version, as one would expect.

In section 4, we compare the properties of process empowerment to those of open-loop empowerment in experiments which are specifically designed to highlight their differences, but also representative of a typical class of problems which could not be satisfactorily treated before. Here, we will show how the characteristic properties of process empowerment mitigate certain shortcomings of traditional open-loop empowerment. A final concluding discussion of the approach, together with future research directions is presented in section 5.

Notations

Random variables are denoted with uppercase letters, while their specific realizations with corresponding lowercase letters, for example $s_t \in \mathcal{S}_t$ is a particular state, s_t, sampled from the random variable S_t representing a state at time, t. Sequences of variables are denoted with lower and upper indexes; for example, A_0^T, is a sequence of random variables representing agent's actions from time, t, until and including time, T, and s_1^T, is a particular realization of a state sequence, S_1^T, starting at time 1 and ending at time T. Notation-wise, we may use s_0, s_1^T interchangeably with s_0^T. The special case of s_1^0 is the empty sequence and s_0^0 denotes the singleton s_0.

$\mathcal{I}[Y;X|Z]$ is the conditional mutual information computed for the conditional joint probability, $p(Y,X|Z)$. $\mathcal{E}[Y;X|Z]$ denotes the information capacity of the conditional probability, $p(Y|X,Z)$, (condition-dependent information channel), where the channel itself goes from X to Y, while conditioned on Z. We will use uppercase vs. lowercase letters to distinguish between the complete probability distribution $p(X)$ and a concrete probability value $p(x)$ for a specific realization x of the random variable X.

2. Open-loop empowerment

Open-loop empowerment quantifies the maximally diverse sensory response that could be predictably caused in the immediate future by an agent's actions. To do so, one maximizes the mutual information

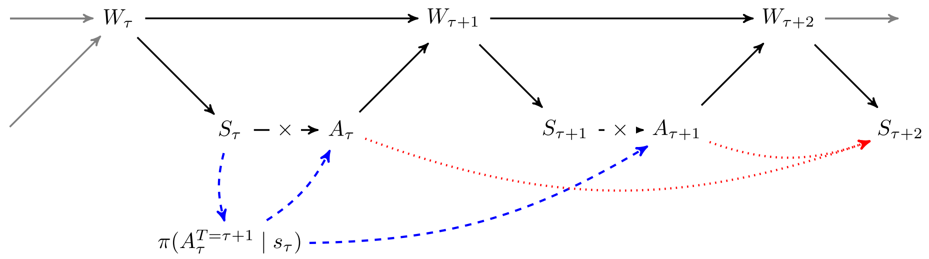

Figure 1. Illustration of open-loop empowerment in the perception-action loop of the agent. We visualize the interaction of the agent with the world as a time-unrolled causal Bayesian network (black solid lines). Here, S denotes the random variables representing the sensor values throughout time, A, the actuator values, and W, the external world states, with the index τ denoting the time step. A world state is observed via the sensor at that time, and the agent then selects the action; the new world state will depend on that action and the previous world state. We consider now a minimal example for how empowerment is computed in a simple agent for a state s_τ at some given time τ. We first disconnect the sensors from their corresponding actuators (indicated by the crosses). Then, a distribution over action sequences, denoted by $\pi(A_t^T \mid s_\tau)$, is 'freely' chosen, beginning at the initial time τ and ending at time T (blue dashed lines). This sequence overrides ('intervenes in', in the language of causal modeling) the respective actuator variables $A_\tau, \ldots, A_T$, without taking into account the intermediate sensor states $S_{\tau+1}, \ldots, S_T$ in those time steps (here only S_T). This is what we refer to as the *open-loop* character of the action sequence. Empowerment is then defined by the maximally achievable mutual information $\max_{\pi(A_\tau^T|s_\tau)} \mathcal{I}[S_{T+1}; A_\tau^T \mid s_\tau]$, indicating the potential causal flow from these actions to the final observation S_{T+1}, here indicated by dotted red lines. In the concrete example here, we have $T = \tau + 1$. As we consider the effect of *two* probing action steps A_τ and $A_{\tau+1}$, we speak of an empowerment time horizon of 2 steps ($T - \tau + 1 = 2$). Note that in most of the remaining text, we will assume, without loss of generality, the initial timestep to be $\tau = 0$.

between the potential action sequences in the coming steps A_0^T and the subsequently observed future sensation, S_{T+1}, given that we start in the current state. Consider figure 1.

The dependency between A_0^T and S_{T+1} is described by the influence that the sequence of actions has on the observation of the state at the end of the action sequence [10, 16, 17, 19, 23], and measured by the mutual information between the actions taken via the probing policy π, going forward from the starting state s_0:

$$\mathcal{I}\left[S_{T+1}; A_0^T \mid s_0\right] = \sum_{s_{T+1}, a_0^T} p\left(s_{T+1}, a_0^T | s_0\right) \log\left(\frac{p\left(s_{T+1}|a_0^T, s_0\right)}{p\left(s_{T+1}|s_0\right)}\right) \tag{1}$$

$$= \sum_{s_{T+1}, a_0^T, s_1^T} \prod_{t=0}^{T} \left[p\left(s_{t+1}|a_t, a_0^{t-1}, s_1^t, s_0\right) \pi\left(a_t|a_0^{t-1}, s_0\right)\right] \log\left(\frac{p\left(s_{T+1}|a_0^T, s_0\right)}{p\left(s_{T+1}|s_0\right)}\right). \tag{2}$$

Here we split the action/sensor probability sequence into a product for each time step according to the Bayesian network in figure 1. We furthermore separate the sensor sequence s_0^t into the initial sensor state s_0 and the sequence of the following ones s_1^t, since we have to consider the whole sensor past for the transitions due to the system being only partially observed (and, similarly, the actuator past). Finally $\pi(a_t|a_0^{t-1}, s_0)$ denotes the probing probability for each action, given the previous actions. Note that the action probabilities depend only on the starting state and the sequence of previously executed actions. Thus, the actions can be combined into one action sequence or 'super-action' a_0^T. These super-actions can be formally treated exactly as if they were elementary or 'atomic' actions. We obtain:

$$\mathcal{I}\left[S_{T+1}; A_0^T \mid s_0\right] = \sum_{s_{T+1}, a_0^T} p\left(s_{t+1}|a_0^T, s_0\right) \pi\left(a_0^T|s_0\right) \log\left(\frac{p\left(s_{T+1}|a_0^T, s_0\right)}{p\left(s_{T+1}|s_0\right)}\right). \tag{3}$$

where it proves convenient for the later optimization to instead replace the term in the logarithm by the reverse channel $q\left(A_0^T \mid S_{T+1}, s_0\right)$

$$= \sum_{s_{T+1}, a_0^T} p\left(s_{T+1} \mid a_0^T, s_0\right) \pi\left(a_0^T \mid s_0\right) \log\left(\frac{q\left(a_0^T \mid s_{T+1}, s_0\right)}{\pi\left(a_0^T \mid s_0\right)}\right). \tag{4}$$

In short, in open-loop empowerment, we treat $p(S_{T+1}|A_0^T, s_0)$ as a communication channel between A_0^T and S_{T+1}, the *information channel* between potential action sequences and the following observation, which depends on the starting state s_0. Here, the channel is fixed *a priori* by selecting the starting state s_0 (this will be extended to 'adaptable channels' in the proposed definition of 'process empowerment' in section 3).

Empowerment for the open-loop is finally defined as the maximum value that can be achieved in equation (1) by suitable choice of the probing policy. This measures the maximal influence that actions can take on the end result. Formally, this is equivalent to the information-theoretic capacity of the information channel, namely:

$$\mathcal{E}\left(s_0\right) = \underset{\pi\left(A_0^T \mid s_0\right)}{\text{maximum}}\, \mathcal{I}\left[S_{T+1}; A_0^T \mid s_0\right], \tag{5}$$

where the maximization is done with regard to the probability distributions over action sequences, $\pi(A_0^T \mid s_0)$, which, depends on the channel and, therefore on the state[4] (and only on the state) s_0.

In discrete environments an optimal solution can be derived by the iterative Blahut–Arimoto algorithm [3, 10]. In continuous environments, an optimal solution can be derived by calculating the Gaussian channel capacity [16, 17].

We reiterate that, once s_0 is given, the channel is fixed and the capacity computation does not take into account anything that happens during the action sequence until the final outcome s_{T+1} is observed. Contrasting to this, in the following we proceed to formulate the process empowerment which now takes feedback into account during the execution of the potential actions.

3. Process empowerment

Most of the literature on empowerment as intrinsic motivation focuses on open-loop empowerment. Its effectiveness has been now shown in a wide variety of scenarios [10, 13, 16, 17, 19, 24]. However, there are a number of scenarios where the open-loop nature of the probing action sequences fails to reflect the true control that the agent can exert on the environment. Typical such cases are where noise can divert the agent into irreversible (and possibly destructive) outcomes. In these, empowerment based on open-loop probing actions is likely to underestimate the actual control that the agent could exert if only it would be able to respond to these perturbations during the probing. Such examples will be discussed below in Experiments (section 4). Other such typical situations where the full possibilities of the agent under consideration could be underestimated is where one is contending with an environment that may respond adversarially (or cooperatively) to the agent. Studying full multi agent scenarios is beyond the scope of the present paper and will be covered in future work. Nonetheless, here we study a reduced version of such a scenario where an oracular environment counteracts the agent's actions, with the consequence that more reactivity by the agent's probing is required to elicit the full picture of how much it has control over its environment.

But first, we need to make precise how we intend to measure the causal control that the agent can exert on the environment when the agent no longer selects among blindly running open-loop probing action sequences (i.e. fixed scripts being automatically carried out), but instead these sequences become reactive to the environment, i.e. closed-loop. In particular, the causal control is now no longer measured with respect to the choice of action sequences, but of whole policies.

3.1. Problem definition and objective

We now extend the definition of open-loop empowerment towards process empowerment which takes feedback into account. In open-loop empowerment, we had action sequences which, for one, depend on the state in which one started (in our case, the sensor state s_0), and on the past actions in the probing sequence, without regard to external events; this is equivalent to running an automatic action script without external input in that one keeps only track of the current timestep in a given sequence.

In closed-loop empowerment, however, this nonreactive action sequence is instead replaced by a sequence of feedback policies which now, in addition to just depending on the time step and the past actions (i.e. de facto being a fixed action sequence), also depend on the past sensor states observed during probing (figure 2).

[4] Note that in above consideration we take the state s_0 to be the subjective sensor state of the agent. It is possible to instead compute empowerment with respect to the objective world state w_0 and in many applications, no distinction is made between w and s, i.e. the sensor has full access to the relevant world state. These distinctions are conceptually irrelevant for the present discussion and we just mention them for consistency with the literature.

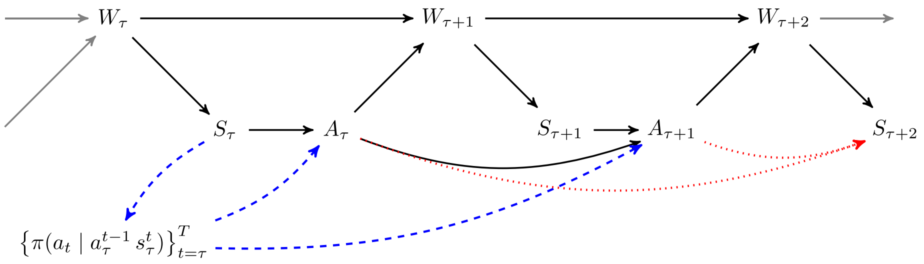

Figure 2. Illustration of process empowerment in the perception-action loop of the agent. We visualize the interaction of the agent with the world as a time-unrolled causal Bayesian network (black solid lines). As in figure 1, S denotes the sensor (random) variables throughout time, A, the actuator values, W, the external world states, and τ the time step; a world state is observed via the sensor, which affects the action which, in turn, produces a new world state from the old one. The figure illustrates the computation of process empowerment for the state s_τ and a horizon of 2 steps into the future. Unlike figure 1, for process empowerment, we consider a whole set of feedback policies $\{\pi(a_t \mid a_\tau^{t-1}, s_\tau^t)\}_{t=\tau}^T$, one per horizon step (with the usual convention that an end index earlier than the starting index indicates an empty sequence). Compared to index τ which denotes the time step for which the empowerment is calculated, t is the running time index of the potential future policies (the probing policies) over which the computation takes place. The blue dashed lines indicate the sequence of 'freely chosen' policies over the times $t = \tau \dots T$. Since we are in the closed loop case, at each $t = \tau, \tau+1, \dots, T$, the policies $\left\{\pi(a_t \mid a_\tau^{t-1}, s_\tau^t)\right\}_{t=\tau}^T$, take into account in each choice both the current sensor value S and previous A, this dependence on the past is indicated by the solid black arrows (while we intervene in the actions as in figure 1, acting and sensing interact here, therefore, unlike there, here we do *not* cross out the black arrows of the actions). Selecting $T = \tau + 1$, the present diagram illustrates the calculation of $T - \tau + 1 =$ 2-step process empowerment. The process empowerment value is the maximal mutual information $\mathcal{I}[S_{T+1}; S_{\tau+1}^T, A_\tau^T \mid s_\tau]$ achievable by suitably choosing the set of policies. The effect of these potential future interventions on the final outcome is illustrated here with the red dotted lines from the two action steps towards the final outcome. As in figure 1, without loss of generality, the main text assumes $\tau = 0$ throughout.

$$\mathcal{I}\left[S_{T+1}; S_1^T, A_0^T \mid s_0\right] = \sum_{\substack{s_1^{T+1}\\ a_0^T}} p\left(s_{T+1}, s_1^T, a_0^T \mid s_0\right) \log\left(\frac{p\left(s_{T+1}, s_1^T, a_0^T \mid s_0\right)}{p\left(s_1^T, a_0^T \mid s_0\right) p\left(s_{T+1} \mid s_0\right)}\right) \tag{6}$$

$$= \sum_{\substack{s_1^{T+1}\\ a_0^T}} \prod_{t=0}^{T} \left[p\left(s_{t+1} | a_t, a_0^{t-1}, s_1^t, s_0\right) \pi\left(a_t | s_1^t, a_0^{t-1}, s_0\right)\right] \log\left(\frac{p\left(s_{T+1} \mid a_0^T, s_1^T, s_0\right)}{p\left(s_{T+1} \mid s_0\right)}\right) \tag{7}$$

$$= \sum_{\substack{s_1^{T+1}\\ a_0^T}} p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) p\left(s_1^T, a_0^T \mid s_0\right) \log\left(\frac{q\left(s_1^T, a_0^T \mid s_0, s_{T+1}\right)}{p\left(s_1^T, a_0^T \mid s_0\right)}\right) \tag{8}$$

analogously to the reasoning of equations (1)–(4). Similar to there, we consider $p(S_{T+1} \mid S_1^T, A_0^T, s_0)$, $p(S_1^T, A_0^T \mid s_0)$, and $q(S_1^T, A_0^T \mid S_{T+1}, s_0)$, to be the information channel, the source, and the reverse channel respectively. The main difference to equation (1) is that now the channels also refer to observed states encountered during the probing . Thus, one can now no longer consider a probing policy that operates as a policy just over super-actions. Instead one now has a sequence of probing policies, one for each possible sensorimotor history t steps into the probing $\left\{\pi(a_t \mid a_0^{t-1}, s_0^t)\right\}_{t=0}^T$, each depending not only on the past actions, but on the past states, s_0^t, and actions, a_0^{t-1}, as per figure 2. It is now through these trajectory histories that the actions affect the future state, S_{T+1}.

In analogy to equation (5), the process empowerment is defined by the maximum value that equation (6) can achieve over all probing policy sequences:

$$\mathcal{E}^{\mathrm{PR}}\left(s_0\right) = \underset{\left\{\pi\left(a_t | a_0^{t-1}, s_0^t\right)\right\}_{t=0}^T}{\text{maximum}} \mathcal{I}\left[S_{T+1}; S_1^T, A_0^T \mid s_0\right] \tag{9}$$

3.2. Interpretation

To correctly interpret the resulting value, we note that, in the computation, the logarithm term contains the observed sensor sequence s_1^T (see e.g. equation (7)). This means that the influence not only of the actions, but also of the ensuing observed states is included in process empowerment. This specifically also incorporates the influence of process noise (which arises by the probing policy interacting with the environment). If one wished to separate the effect due to the agent's actions and ignore those produced by the process itself, one would drop the s_1^T term in the logarithm in equation (7), but not in the averaging. It is that averaging that, in this alternative formulation, takes into account the fact that one uses policies with feedback. However, the currently introduced formulation of the process empowerment objective has the advantage of being suitable to be treated via a variant of the Blahut–Arimoto algorithm and thus will be the one that we will exclusively discuss in the following.

3.3. Computation of process empowerment through self-consistent equations

In this section we show a solution to the optimization problem in equation (9) (note in particular the formulation in equation (8) for the following) and discuss its properties. As with traditional open-loop empowerment, the formalism itself carries over fully to partial observability since the algorithm to do so is formulated for arbitrary control processes, both Markovian and non-Markovian.

Theorem 1. *The optimization problem in equation (9) has a solution, formally given by the set of self-consistent equations:*

$$\forall t' \ : \pi\left(a_{t'} \mid a_0^{t'-1}, s_0^{t'}\right) = \frac{\exp\left(\mathcal{E}^{\mathrm{FPR}}\left[S_{T+1}; A_{t'+1}^T, S_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\right] + \mathcal{E}^{\mathrm{PPR}}\left[S_{T+1}; s_1^{t'}, a_0^{t'} \mid s_0\right]\right)}{\sum_{a_{t'}} \exp\left(\mathcal{E}^{\mathrm{FPR}}\left[S_{T+1}; A_{t'+1}^T, S_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\right] + \mathcal{E}^{\mathrm{PPR}}\left[S_{T+1}; s_1^{t'}, a_0^{t'} \mid s_0\right]\right)} \tag{10}$$

$$q\left(s_1^T, a_0^T \mid s_0, s_{T+1}\right) = \frac{p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) p\left(s_1^T, a_0^T \mid s_0\right)}{\sum_{s_1^T, a_0^T} p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) p\left(s_1^T, a_0^T \mid s_0\right)}, \tag{11}$$

where $\mathcal{E}^{\mathrm{FPR}}[S_{T+1}; A_{t'+1}^T, S_{t'+1}^T \mid a_0^{t'}, s_0^{t'}]$ and $\mathcal{E}^{\mathrm{PPR}}[S_{T+1}; s_1^{t'}, a_0^{t'} \mid s_0]$ are 'future process empowerment', conditioned on the past state (with respect to time t') and action trajectories, $s_1^{t'}, a_0^{t'}$, and 'past process empowerment' between the past state/action trajectories and the future state, S_{T+1}, respectively, given by:

$$\mathcal{E}^{\mathrm{FPR}}\left[S_{T+1}; A_{t'+1}^T, S_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\right] = \sum_{\substack{s_{t'+1}^{T+1} \\ a_{t'+1}^T}} p\left(s_{T+1}, a_{t'+1}^T, s_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\right) \log\left(\frac{q\left(s_{t'+1}^T, a_{t'+1}^T \mid a_0^{t'}, s_0, s_1^{t'}, s_{T+1}\right)}{p\left(a_{t'+1}^T, s_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\right)}\right)$$

$$\mathcal{E}^{\mathrm{PPR}}\left[S_{T+1}; s_1^{t'}, a_0^{t'} \mid s_0\right] = \sum_{\substack{s_{t'+1}^{T+1} \\ a_{t'+1}^T}} p\left(s_{T+1}, a_{t'+1}^T, s_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\right) \log\left(q\left(s_1^{t'}, a_0^{t'} \mid s_0, s_{T+1}\right)\right).$$

Proof. See appendix. □

The solution has an interesting structure. It decomposes into (i) 'future feedback empowerment', $\mathcal{E}^{\mathrm{FPR}}$, which is the information capacity between the final states, S_{T+1}, and the future state-action trajectories, $\{A_{t'+1}^T, S_{t'+1}^T\}$, conditioned on the particular past trajectory $\{a_0^{t'}, s_0^{t'}\}$ so far, and (ii) 'past feedback empowerment', $\mathcal{E}^{\mathrm{PPR}}$, which is the information capacity contribution between the final state and the actual past $\{a_0^{t'}, s_0^{t'}\}$, conditioned on the initial state, s_0. Such past-future decomposition means that for high process empowerment an agent needs to follow trajectories $\{a_0^{t'}, s_0^{t'}\}$ that allow for both good controllability in the future, $\mathcal{E}^{\mathrm{FPR}}$ and good predictability of S_{T+1} from the past, $\mathcal{E}^{\mathrm{PPR}}$.

The solution in equations (10) and (11) is unique and can be found by 'alternating maximization' [12]. That follows from the properties of the mutual information in the objective Equation (6), which is convex with regard to the product of the feedback policies $\left\{\pi(a_t \mid a^{t-1}, s^t)\right\}_{t=0}^T$. Consequently, it is convex in each of the policies.

3.4. ABA—alternating Blahut–Arimoto algorithm

We propose a practical algorithm, named 'ABA' for solving the set of self-consistent equations in theorem 1, which extends the classical Blahut–Arimoto algorithm to the problem of process empowerment with a set of input probability distributions, $\left\{\pi(a_t \mid a_0^{t-1}, s_0^t)\right\}_{t=0}^T$.

Alternating Blahut–Arimoto (ABA) 1.

```
1: Init: Randomly Initialize, $\{\pi(a_t \mid a_0^{t-1}, s_0^t)\}_{t=0}^T$
2: Iterate:
3:     $q(s_1^T, a_0^T \mid s_0, s_{T+1}) = \frac{p(s_{T+1}|s_1^T,a_0^T,s_0)p(s_1^T,a_0^T|s_0)}{\sum_{s_1^T,a_0^T} p(s_{T+1}|s_1^T,a_0^T,s_0)p(s_1^T,a_0^T|s_0)}$    ▷depends on $\{\pi(a_t \mid a_0^{t-1}, s_0^t)\}_{t=0}^T$
4: $\forall t' \ : \ \pi(a_{t'} \mid a_0^{t'-1}, s_0^{t'}) = \frac{\exp\left(\mathcal{E}^{FPR}[S_{T+1};A_{t'+1}^T,S_{t'+1}^T|a_0^{t'},s_0^{t'}]+\mathcal{E}^{PPR}[S_{T+1};s_1^{t'},a_0^{t'}|s_0]\right)}{\sum_{a_{t'}}\exp\left(\mathcal{E}^{FPR}[S_{T+1};A_{t'+1}^T,S_{t'+1}^T|a_0^{t'},s_0^{t'}]+\mathcal{E}^{PPR}[S_{T+1};s_1^{t'},a_0^{t'}|s_0]\right)}$    ▷in an arbitrary order
5: Until Convergence
```

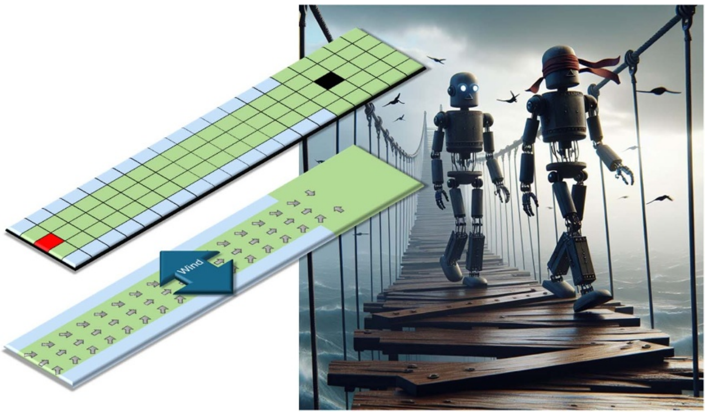

Figure 3. Illustration of the example of a robot crossing a windy bridge. Empowerment aims to capture the ability of an agent to affect the world and bring about certain sensory states. Open-loop empowerment though only considers probing action sequences chosen ahead of time at the starting point. The probing for open-loop empowerment cannot take into account and compensate noise which arises during the execution of the predetermined n-step probing sequence, in contrast to feedback-aware empowerment, such as process empowerment. Consider two types of robots trying to cross a windy bridge. After each step the wind might have blown them slightly off course. The robot who is actively using its sensor might be able to correct for these deviations, and therefore an empowerment variant using feedback-aware closed-loop probing might be a more complete evaluation of the robots capabilities, rather than assuming that the robot would 'blindly' attempt its probing without reacting to displacements, which therefore underestimates the robot's actual potentialities. (Illustration produced in part by Bing Copilot).

The update of the policy set in line 4 of algorithm 1 is linear in time (the size of the policy set). The solution can be found by fixed-point iteration starting from an arbitrary initialization of the policy set $\left\{\pi(a_t \mid a_0^{t-1}, s_0^t)\right\}_{t=0}^{T}$. Practically, each of the policies in line 4 can be updated with policies from the previous iteration in a separate thread/process, which further reduces this linear factor. We made our code repository publicly available for the reproduction of our results and further research.

In summary, we introduced process empowerment as a generalization of empowerment. In process empowerment, the agent, when probing its ability to choose the future, now has the additional freedom to use policies that react to feedback from changes in the environment as the probing progresses, as opposed to traditional open-loop empowerment which only uses fixed action sequence without feedback. We offer a self-consistent algorithm to compute process empowerment. The remarkable structure of the algorithm is a generalized Blahut–Arimoto which iterates the computation of the reverse channel vs. the probing policies, of which there is one per probed time step. For each such time step, the policy update derives from the future expected channel capacity with respect to the end state, together with the contribution of the current past to that end state. This iteration runs until convergence. In the appendix we prove the algorithm.

4. Experiments

To demonstrate the properties and advantages of process empowerment in comparison to open-loop empowerment, we consider the following experiment settings: (a) stochastic dynamics with no velocity nor oracle. The latter prescribes actions that constitute a solution to some given task under unperturbed dynamics and without delayed memory effects that would arise by including a velocity term; (b) stochastic dynamics, now with velocity, and a varying balance between agent actions and oracle actions which denote the optimal behavior in the deterministic setting. In both settings we compare between the robustness of process and open-loop empowerment with regard to the noise level and the trade-off balance between oracle action and empowerment-induced behavior. All experiments are formulated in in the discrete setting. The implementation of the ABA algorithm 1 for further research and development, and code for the experiments is available at [1].

4.1. Experiment: narrow 'Windy Bridge'

The *Windy Bridge* is a paradigmatic demonstration of the effect of noise on empowerment. It is a stochastic environment with trap states (the agent's 'death' states). In it, empowerment shows a drastically different profile, depending on whether the probing actions are run in an open-loop fashion or incorporate feedback. In the first case, once the probing starts, they are 'reeled off' automatically and thus subject to the vagaries of the noise encountered. In the second case, however, or they are permitted to incorporate feedback to compensate said noise. Thus, the second more realistically models the true capacity of the agent to control its futures.

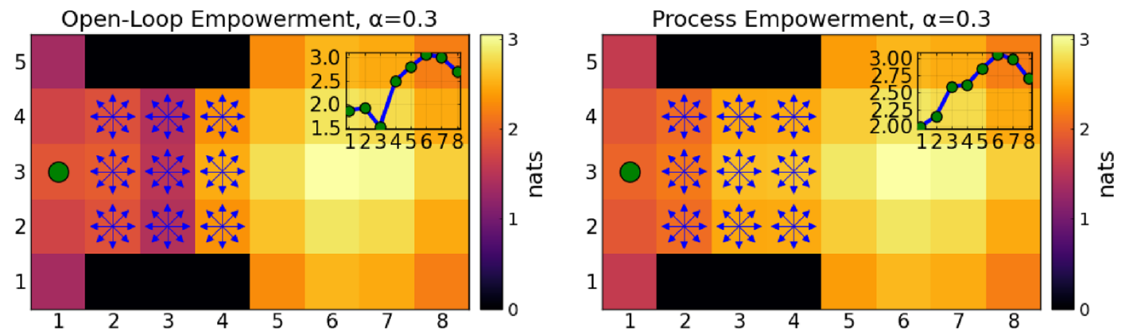

Figure 4. 'Windy Bridge' with a narrow corridor in the first column, connected by a three-step bridge in the columns 2–4 between the black blocks denoting water (termination/death states for the agent), to a free space with many potential future states to the right of the bridge, which is considered a desirable region from an empowerment point of view. The green point denotes the agents before crossing the bridge. The blue arrows denote the dynamics of noise on the bridge. The brightness of the different fields indicates the empowerment values: bright colors denote high values of empowerment in 'nats', while dark ones denote low values. The insets shows the empowerment values attained when in column 1…8 in the middle path. The right figure shows 'process empowerment' calculated by algorithm 1 with a gradually increasing empowerment through the bridge towards the free space with many potential states. The left figure shows that 'open-loop empowerment' is unable to see the potential in the future due to the noise 'fog' over the bridge and the lack of the compensation by feedback. This paradigmatic example shows the importance of feedback for intrinsic motivation, and, more broadly, for judging the truly causal power that an agent's actions can have on the environment.

As an intuitive motivation of the idea, consider figure 3. Traditional open-loop empowerment measures the potentially controllable future states, but, only based on action sequences fixed at the beginning of the probing. In the figure, this corresponds to the blindfolded robot who executes the respective probing action sequence according to a predefined plan. On a windy bridge, this will be outperformed by a probing policy which allows the agent to react to deviations.

The 'Windy Bridge' is implemented as a simple environment with static state-to-state movement, fully controlled by the agent. In particular, it does not yet include velocity or prescribed oracle actions, as the model from section 4.2. Here, we have a gridworld with a single agent (green circle), where there is a relatively narrow passage from the left side of the world to the right side of the world. This passage is squeezed between *death states*, i.e. states in which the agent can no longer affect anything, and in which empowerment drops to 0). This scenario models a bridge connecting the colored region on the left with that on the right in figure 4; this bridge is squeezed between the black regions which indicate the death states the agent might fall off into if taking a mis-step.

Importantly, the scenario includes stochasticity in dynamics ('wind'), schematically indicated by the blue arrows. The colors in figure 4 show the value of empowerment for the different states in the world, with the death states showing vanishing empowerment in black.

When probed by open-loop empowerment, the bridge now appears as an obstacle in the following sense: the empowerment values in the left region are moderately high, but drop when moving towards the middle of the bridge, before climbing again when that empowerment 'valley' is crossed. In other words, open-loop empowerment cannot 'see' through the 'fog' created by the noisy wind that there is a high-empowerment region with lots of optionality on the right side of the bridge until the bridge has been crossed halfway. This dip in open-loop empowerment is visible in the inset of figure 4, left. Since the probing is carried out in an open-loop fashion, all the probing sequences are predefined at the beginning, and cannot react to the perturbation by the noise, which leads to a drop in detected control over the future. Only once the middle of the bridge is crossed does open-loop empowerment pick up the availability of a rich diversity of controllable outcomes.

Figure 4, right, shows the same data, but for process empowerment. Now, we have probing policies instead of fixed sequences and these are allowed to respond to noise, thus generally reducing the probability that the agent will fall to its death in the probing phase.

In summary, traditional open-loop empowerment is not able to guide an agent from the left to the right side of the bridge, despite its much larger state space. The wind noise causes a large portion of probing sequences to fall off the bridge and creates a dip in empowerment which keeps the agent trapped on the left side, if empowerment gradients are exclusively used as drive. Once we consider process empowerment which allows the inclusion of feedback during probing, the noise can be compensated, and the agent is able to 'see' already on the left side that it has an improved choice of controllable outcomes when moving towards the right side of the bridge.

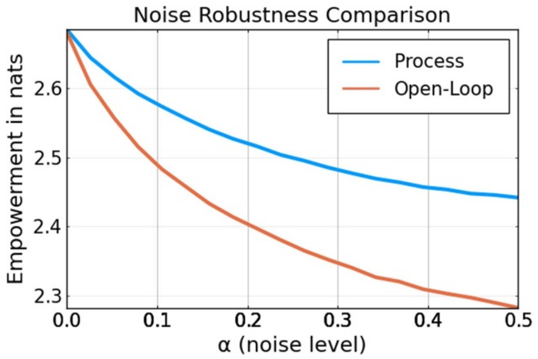

Figure 5. Effectiveness of process empowerment versus open-loop in the 'Windy Bridge' scenario. Each point on the curves represents the average empowerment value in the 'Windy Bridge' environment over all states. For example, the average of the process empowerment landscape of figure 4 appears on the blue curve with coordinates (0.3, 2.49). The top-left corner shows that the values of process and open-loop empowerment are identical in deterministic environments and thus consistent with expectation. With increasing noise level, they both drop, with the gap between them increasing.

4.1.1. State representation and dynamics

For completeness, we now list the details of the model. In this experiment, the state is represented as $s_t = (x_t, y_t)$, where x_t and y_t are the x and the y coordinates of the agent in the grid at time t. The gridworld is bounded. The agent has the same action set in every state, $a_t \in \mathcal{A} = (\rightarrow, \downarrow, \uparrow, \leftarrow)$, and the transition probability, $P(s_{t+1} \mid s_t, a_t)$ is given by:

$$P\left(s_{t+1} \mid s_t, a_t\right) = \begin{cases} s_{t+1} = f(s_t, a_t) \;\text{ with probability } 1-\alpha \\ s_{t+1} = \text{random Manhattan neighbor to } s_t \text{ with probability } \alpha \end{cases} \tag{12}$$

where $s_{t+1} = f(s_t, a_t) = s_t + a_t$ is the deterministic transition function in the environment with the component-wise addition, $'+'$, between $s = (x, y)$ and the actions $\uparrow = (0, 1)$, $\rightarrow = (1, 0)$, $\downarrow = (0, -1)$, and $\leftarrow = (-1, 0)$.

Figure 4 compares the values of open-loop and process empowerment in the Windy Bridge scenario, calculated by the classical Blahut–Arimoto algorithm [3] and the 'ABA' algorithm 1, respectively, both of which we run for the same number of iterations.

We begin with a first sanity check to establish the consistency of process empowerment with traditional open-loop empowerment: for this, note that, in deterministic environments with noise level $\alpha = 0$, one would expect process empowerment to be precisely equal to open-loop empowerment: given the starting point, the future state is fully determined by the action sequences, and thus feedback will not improve upon that, as the subsequent states can be fully predicted from the actions only. Figure 5 shows that this is indeed the case.

In stochastic environments, however, feedback is able to compensate for noise introduced into the probed trajectories. With feedback, the agent can detect that it has better control of the future than when probing the future with nonreactive open-loop trajectories. Thus, even while noise reduces controllability in general, now process empowerment becomes larger than open-loop empowerment. Also, the gap between the compensated (closed-loop) and uncompensated (open-loop) empowerment increases with growing noise, as shown at figure 5.

4.2. 'Empowerment cushioning' in environments with unpredictable perturbation and (Adversarial) action

We now investigate environments which show how process empowerment can help to make traditional deterministic solutions more robust when the original conditions are modified or made more noisy. More concretely, we will start out with deterministic policies solving basic scenarios (which we will model by oracle actions). The scenarios will then be perturbed, rendering the original solutions unsuitable. We will then enhance them with help of the different empowerment variants and show how this help overcome the perturbations.

We specifically consider environments where the agent is limited in its abilities to affect the entire state. Specifically, the state is comprised of position and velocity, and the agent's actions can only accelerate, i.e. only change the velocity components. Furthermore, there is an additional parameter which determines the balance between the agent choosing the intrinsically motivated action vs. the predefined oracle action (which solves a simpler, noiseless problem). This setting allows to examine the robustness of intrinsically motivated agents with regard to both the noise level and the balance coefficient. Here, we found process empowerment to be significantly more robust in comparison to open-loop empowerment. This suggests a new approach for the design of robust strategies for stochastic dynamics: namely, consider first solutions for a deterministic version of the problem which are usually simpler to derive; after that, endow them with 'empowerment cushioning' to deal with the stochastic variant of the problem.

We now proceed by defining the scenarios in detail. We will consider a hallway with obstacles scenario and a circular racetrack scenario. Importantly, both will use the same underlying movement dynamics which we will therefore define first.

4.2.1. State representation and dynamics

We now provide the definitions for the state representation and the transition functions used in the following examples. These incorporate the following: 1. acceleration control only (only velocity can be controlled); 2. environmental noise; 3. oracle actions solving the basic problem; 4. empowerment-induced actions climbing the empowerment gradient. The formal definitions follow.

In the following experiments we consider environments with a four-dimensional state given by:

$$s_t = (x_t, y_t, v_t, u_t) \tag{13}$$

determining $x_t, y_t, v_t$and u_t with a two-dimensional position (x_t, y_t) and the respective velocity (v_t, u_t). The deterministic transition function for the states $s_{t+1} = f(s_t, a_t)$ is given by

$$\left(x_{t+1}, y_{t+1}, v_{t+1}, u_{t+1}\right) = \left(x_t + v_t, y_t + u_t, v_t + a^x_t, u_t + a^y_t\right) \tag{14}$$

with $a_t = (a^x_t, a^y_t) \in \mathcal{A} = (\rightarrow, \downarrow, \uparrow, \leftarrow)$, where the actions are given by $\rightarrow = (0, 1)$ and the three other cardinal directions, effecting a velocity change ('acceleration'). However, note that, while the accelerations are confined to the cardinal directions, the movement itself can be diagonal, due to the lingering velocity.

In the next step, we enhance the model by assuming that there is a solution, $\mathcal{D}$, to an unperturbed version of the problem in question: the solution to that problem may, for example, have been derived by standard methods in deterministic optimal control (OC) or Deterministic RL policy, or simply assumed to be given by an oracle. The latter prescribes an action at every state, named the *oracle action*, $d(s_t) \in \mathcal{D}$. Here, $d(s_t)$ denotes a function, that at every state, s_t, returns a two-dimensional oracle vector representing a velocity change $d(s_t) = (d_x, d_y)$; this acts as a deterministic policy/controller which could have been computed e.g. via OC/RL or provided via a hand-crafted or otherwise predefined solution. Note that oracle actions are not limited to accelerations in cardinal directions and could also be diagonal.

Also, in principle, $d(s_t)$ could be replaced by a stochastic policy/controller as well, but for the purpose of demonstrating how empowerment can support the oracle policy, we will confine ourselves to deterministic policies.

Consider now above environment dynamics and how it combines velocity, noise, empowerment action and prescribed 'oracle-action'. First, noise is applied with a given probability α. If no noise applies, an action is chosen, the empowerment-induced action with probability η, and the oracle action otherwise. Formally, the transition probability, $P(s_{t+1} \mid s_t, a_t; \alpha, \zeta)$, where α and ζ are the noise and mixing parameters, respectively, is given by:

$$P\left(s_{t+1} \mid s_t, a_t; \alpha, \zeta\right) = \begin{cases} \text{with probability } 1-\alpha : \begin{cases} s_{t+1} = f(s_t, a_t) \;\; \text{with probability } \zeta \\ \begin{pmatrix} x_{t+1} \\ y_{t+1} \\ v_{t+1} \\ u_{t+1} \end{pmatrix} = \begin{pmatrix} x_{t+1} \\ y_{t+1} \\ d_x \\ d_y \end{pmatrix} \text{with probability } 1-\zeta \end{cases} \\ \text{with probability } \alpha : s_{t+1} = \text{random Moore neighbour of } s_t \, . \end{cases} \tag{15}$$

This means that, with probability α, the agent moves to a random neighboring state; if that does not happen, with probability ζ, the agent carries out the empowerment-induced action (i.e. picking the action that maximizes the empowerment gradient achieved by this action); and, when that is not the case, it executes the oracle action $d(s_t)$ for that state s_t.

In the following experiments we consider two scenarios, *Hallway* and *Race*. The agent's default empowerment-induced policy consists of ascending the gradient of empowerment, i.e. picking that action which maximizes the empowerment gain in that step. This corresponds to the standard strategy of all empowerment-based intrinsic motivation behavior models in various scenarios [17, 19, 24]. However, here we mix it with goal-directed behavior derived from a deterministic task which is represented as the oracle

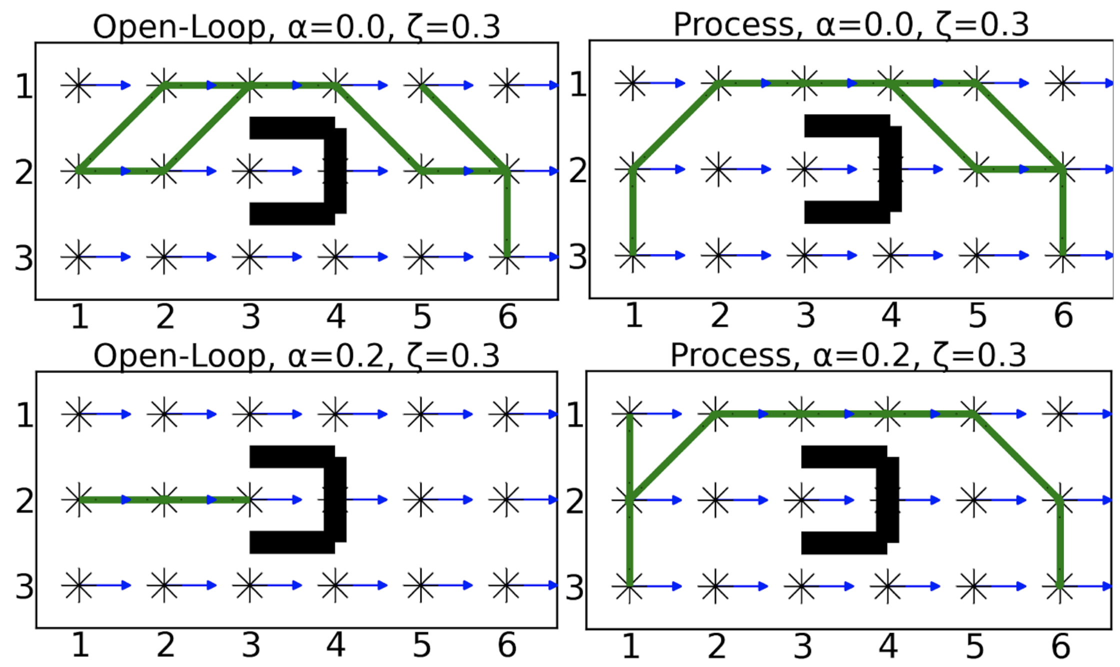

Figure 6. Hallway: robustification of a solution derived in an unperturbed environment. The environment consists of a rectangular region with a horseshoe obstacle in its middle. The arrows indicate the oracle action. The possible noise is indicated by the rosettes at each location, indicating the possible eight directions the noise can accelerate the agent towards. The oracle action would naively move the agent towards the right edge, if no noise or obstacle were present. For the deterministic case, the pure oracle policy will get stuck in the obstacle in the middle row; with noise, it may sometimes avoid the trap for the middle row, but now can sometimes additionally trap the upper and lower row case. Consider the green paths, derived by following the gradient of empowerment (here, we choose horizon $T = 3$, in all cases). In the left column, the green path for 'open-loop' empowerment is effective in guiding the agent around the obstacle in the case in the absence of noise ($\alpha = 0$); however, it ceases to become effective for noise probability of $\alpha = 0.2$ and larger. In contrast to that, the right column shows that the process empowerment gradient enables the agent to navigate to the right without getting stuck in the obstacle even in the presence of noise. This can be interpreted as empowerment 'cushioning' the agent with a 'local soft-controllability envelope', creating a robust route around obstacles and absorbing noise. This effect is significantly more effective with process empowerment.

action. With this setup, we demonstrate how process empowerment can be useful for the robustification of policies obtained from a deterministic basic scenario when newly confronted with stochastic dynamics and unpredictable obstacles. The purpose of this is to study how empowerment can 'widen' and robustify policy solutions obtained for a basic scenario to more general settings. This opens up various applications such as transfer learning, domain adaptation, or sim-to-real transition; in short, wherever a solution in one environment is needed to be robustly updated to another environment with potentially unpredictable perturbations.

4.2.2. Experiment: hallway with unpredictable perturbation

In this setting we examine the robustness of process and open-loop empowerment for different levels of noise, α, and different values of empowerment mixing coefficient, ζ.

Figure 6 compares the performance of the agent with open-loop vs process empowerment. The preselected oracle actions prescribe a move from left to right for all states and are shown by the blue arrows. They correspond to the task of moving down the hallway towards the right under the assumption of a deterministic dynamics and the absence of the obstacle in the middle.

Obviously, with the obstacle present, and a deterministic dynamics, an agent appearing on the left side of the hallway will get stuck in the obstacle a third of the time. Now, when empowerment is admixed with proportion ζ, as long as there is no noise, both open-loop and process empowerment manage to navigate the agent around the obstacle trap (green lines): the empowerment values reflect the loss of freedom in the cul-de-sac and the empowerment gradient pushes them out of it and towards the free lines of the trajectories induced by the oracle.

This situation changes when, additionally, noise (its possible presence indicated by the black rosettes at the center of each field) is added to the movement dynamics. In this case, open-loop empowerment is no longer able to contend with the noise—however, process empowerment with the same planning horizon, $T = 3$, is able to compensate the noise and helps the oracle effectively to navigate around the obstacle even in the presence of noise. In effect, empowerment, and especially process empowerment, 'cushions' the agent's

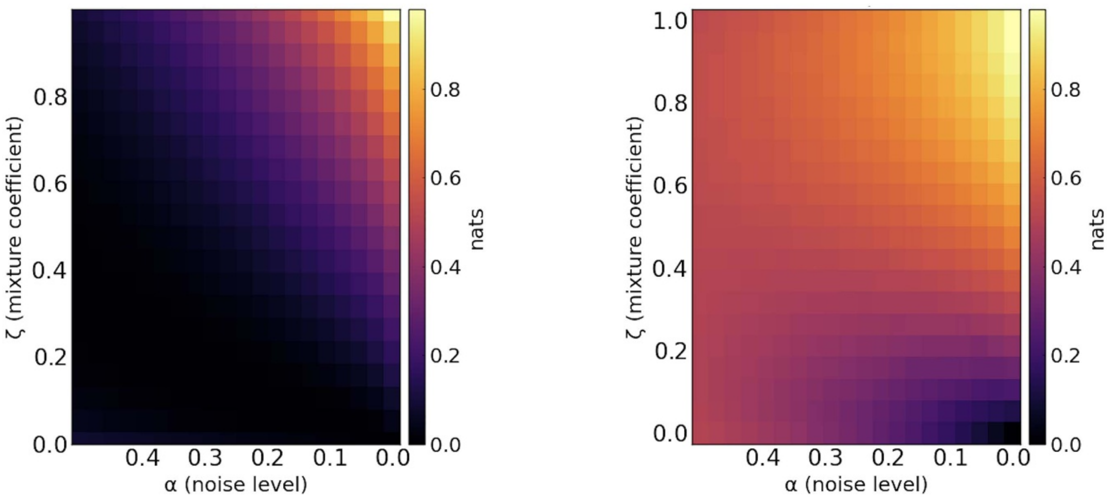

Figure 7. Open-loop and process empowerment landscapes for the 'Hallway environment' from figure 6 are shown in the left and right figures, respectively. Each tile shows the average value of empowerment over the four-dimensional (section 4.2.1) state space of the 'Hallway environment' for the given parameters. Note that the noise level (horizontal axis) drops from high to low from the left to the right. Process empowerment is effective in maintaining a higher empowerment value for a significantly larger parameter set of α and ζ in the transition dynamics, equation (4.2.1).

dynamics by a 'local soft-controllability envelope'. It incentivizes the agent to maintain a local environment or niche which remains under de facto control of the agent and thus allows the oracle actions to remain executable, rather than getting trapped. Notably, in our proof-of-concept scenario, process empowerment is significantly more effective at that.

We finally show the overall effect of noise intensity α and empowerment contribution ζ on open-loop and process empowerment in figure 7. The values of open-loop (left) and process (right) empowerment are computed using the standard Blahut–Arimoto algorithm [3] and algorithm 1, respectively.

4.2.3. Experiment: race with adversarial oracle-action

As a final proof-of-concept scenario, we investigate a relatively intricate scenario. We consider the task of going in a circle similar to the classic racetrack scenario. The state representation and dynamics in this environment is similar to that in 'Hallway' given in section 4.2.1 The oracle-action prescribes a motion in a circle around a discretized racetrack, simulating a racing car scenario. We note that, due to the acceleration-based dynamics model equation (15), this is a problem that requires some planning and is a typical example for a problem normally solved by RL learning-type algorithms.

Now, in our scenario, figure 8, the oracle takes on the role of a defective (occasionally adversarial) circular racetrack planner. It normally pushes the agent around the track, but, at some specific states, pushing the agent off the track into death states with zero empowerment simulating a crash. This is similar to falling off the 'Windy Bridge', figure 4 from section 4.1. This happens when the agent hits the black blocks due to the noise or if it falls of the track when the agent leaves the square while following the adversarial oracle action that appears at some states, such as $(x=3, y=1)$, $(x=4, y=1)$, or $(x=6, y=8)$.

Remarkably, even in this case, process empowerment provides a robust local 'controllability envelope' for both the noiseless case and for the case with relatively strong noise $\alpha = 0.25$ (figure 8, third and fourth from the left). Open-loop empowerment, on the other hand, almost completely fails for that same noise level (cf the second plot of figure 8).

The fact that process empowerment has a significantly enhanced robustness to perturbations and even adversarial actions makes it a promising candidate for as a robust intrinsic viability measure. Similar to traditional open-loop empowerment, process empowerment is also a *local* quantity. That is, it does not need to be computed throughout the state space, but only along the probed trajectory and no further than its time horizon. However, the present results demonstrate that process empowerment considerably exceeds that of open-loop empowerment in terms of robustness. This extends its applicability into further domains.

5. Discussion

Empowerment in its various guises (including some variants, see e.g. [2]) has been studied in its open-loop form for some time now. It has proven to be an effective local intrinsic motivation which can replace or enhance given task-driven behaviors of agents. Its information-theoretic formulation makes it universally

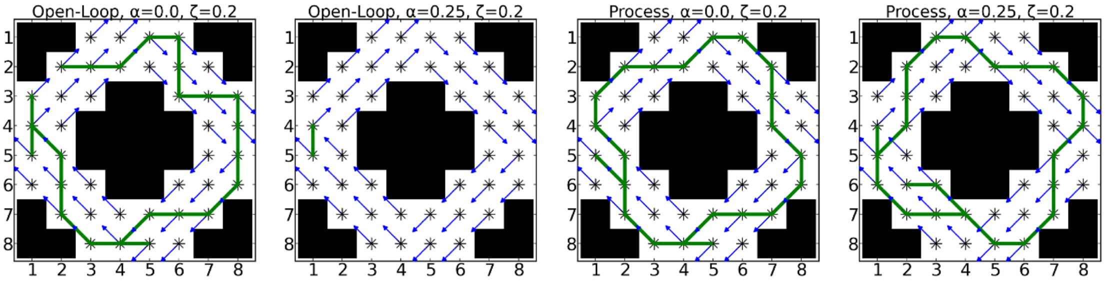

Figure 8. Race scenario with (sometimes) adversarial oracle actions. The agent dies if it leaves the square (due to adversarial oracle actions) or hits the black fields (green lines with an open end such as in (1, 5) indicate a local optimum in which the empowerment gradient does not lead further).

applicable across many scenarios involving sensorimotor loops. Beyond that, it has close links to quantities from dynamical systems [19]. In all these scenarios, its sometimes unexpected effectiveness is achieved using open-loop action probing. In this probing, empowerment evaluates to which degree a ‘cloud’ of states can be controllably reached by the agent in the near future of the present state. An empowerment-driven agent then moves along the gradient of this empowerment value towards states which afford it more such control.

Despite its success, there are situations, such as the scenarios discussed in the present paper, where the de facto reachable ‘cloud’ of states is underestimated by open-loop empowerment. Open-loop probing commits to action sequences ahead of the probing only. For this reason, it will not detect higher-empowerment states further ahead in time if one would need to compensate for perturbations in order for the probing to reach these states.

Despite a wide array of scenarios having been studied in past work, these scenarios had the property in common to have been ‘forgiving’ in one aspect: as noise gradually increases the effect of perturbation, the probing is gradually disrupted, but does not undergo substantial or qualitative shifts. With increasing noise, the effect on empowerment changes only gradually.

The scenarios discussed here open up a new class of environments for empowerment. Traditional open-loop empowerment will not ‘see’ higher-empowerment states further down its probing horizon if they are to be reached through narrow, perilous passages. The reason is that they can not be reached through pre-planned action scripts, but need to be carefully navigated with an active probing policy that is able react to a perturbation.

The formulation of a quantity that achieves this requires some care, and we formulated it in the form of process empowerment. Here, the probing policies are no longer confined to the distribution of action sequences predetermined at the starting state for which empowerment is computed; instead, actions are adapted to the particular state in which the agent finds itself during the probing. Thus, it can compensate to some extent for perturbations induced by noise also in treacherous passages. It can push through disruption in these passages to discover potential states with high controllability beyond that passage. This was demonstrated in the bridge example from section 4.1 which is a typical representative of this type of scenarios.

However, process empowerment has additional advantages over traditional open-loop empowerment, as the examples from section 4.2 demonstrate. In these examples, an oracle action simulated the existence of a solution to an underlying implicit deterministic OC problem. The problem was then disrupted, by obstacles, by a narrow racetrack and by noise; in the race-track, the oracle action was not only imperfect, but in some states even adversarial. However, when the oracle action was admixed with empowerment, the agent could again cope with the obstacles or racetracks well in both variants, that of open-loop as well as of process empowerment.

The introduction of moderate noise does not substantially change the effect of process empowerment, but open-loop empowerment ceases to be useful and can collapse. Empowerment, in essence, ‘cushions’ the agent’s dynamics by a ‘local soft-controllability envelope’, but it does so more effectively for process empowerment than for open-loop empowerment. Note that, despite empowerment being a local quantity, in many examples studied in previous work, aiming for higher empowerment values is consistent with finding desirable solutions on larger or problem-global scale [9, 19]. While the reason for this interaction between local and global scale has not been well understood beyond heuristical arguments, it is, so far, consistently observed in examples. Process empowerment, with its additional resilience to noise in sensitive scenarios, substantially extends the domain of scenarios in which empowerment is applicable.

The experiments with the adversarial racetrack indicate that process empowerment may be suitable if not necessary to satisfactorily deal with scenarios where one needs to compute empowerment for two or more closely interacting agents. Open-loop empowerment is neglecting substantial aspects of agent-agent interaction, since the probing sequences of two interacting agents should not consist of pre-determined action sequences being rattled off, but react directly to each other, even during the probing. In particular, as they interact with the world, they will likely interfere with each other if uncoordinated, and thus we can expect that a reactive probing such as that of process empowerment is necessary to fully capture the agents' causal capacity to affect their environment. For this reason, process empowerment promises to be a far more relevant estimation of realistic 'bubbles of autonomy' than open-loop empowerment, whenever multiple agents are involved.

Despite the clear-cut advantages with respect to traditional open-loop empowerment, there are still a number of issues with process empowerment. We first discuss a conceptional issue. As already mentioned in section 3.2, process empowerment includes the intermediate sensory observations when quantifying the influence on the final observation. This is due to the particular construction of the objective. For a more puristic version of feedback-aware empowerment, one would drop the intermediate sensor observations and focus on quantifying purely the effect that the actions have, even while these continue to depend on the intermediate sensor observations. Concretely, in this modified objective, the intermediate observations inside the logarithmic term would be marginalized over. This new objective does, however, not immediately offer a ready Blahut Arimoto-style convergent iteration algorithm as we employed here to compute process empowerment. Whether it is possible to conceive an analogously elegant algorithm for this modified objective will be explored in the future.

Secondly, the computation expands that of open-loop empowerment in discrete worlds. The complexity of latter empowerment computation, in its basic form, is exponential in the time horizon of the action sequences, as all of these are probed and their number grows exponentially with each decision inside the time horizon. While there are various ways to prune and probe these sequences, the difficulty remains, unless one moves to the continuum. There, however, substantial progress had been recently made. In [19], under the assumption of locally linear approximations around zero actions, the complexity is now linear in the number of discretization steps of the given time horizon. Since the convergence to the empowerment value obtained for infinite discretization is very benign, the complexity is much reduced in this continuous approximation.

Process empowerment is more involved as it involves the computation of policy rather than action sequences. However, we note that the essence of the computation is analogous to the open-loop case. Furthermore, the policies at different time steps can be updated in parallel and independently of each other, which means that the computation can be significantly sped up on a parallel computer. Additionally, we envisage that the idea could be transferable to the continuum in analogy to open-loop empowerment. In this case, we expect that not only will there be insightful connections to other fields of control and dynamical systems, but also a substantially faster algorithm for its computation.

Summarizing, process empowerment is a conceptually grounded, algorithmically accessible (at least in principle) extension of open-loop empowerment which addresses scenarios for which the latter was not equipped to deal with. This particularly includes scenarios which require compensatory behavior during probing, especially navigation/manipulation in noisy, but delicate and possibly adversarial environments.

Data availability statement

The data used in the experiments can be reconstructed using our publicly available repository [1]. All data that support the findings of this study are included within the article (and any supplementary files).

Acknowledgments

ST was supported in part by NSF Award (2513350) and by Pazy Grant (195-2020). DP was supported in part by Pazy Grant (195-2020). DP would like to thank Alexander Klyubin for discussions as to how to appropriately incorporate trajectory histories into the original empowerment definition. ST and DP would like to thank Professor Ilya Nemenman from the Emory University for insightful discussions on intrinsic motivation in complex multi-agent systems.

Appendix

Proof. (To theorem 1) For the present proof the control dynamics is considered Markovian, i.e. the following state depends only on the current state and the action taken,

$$\mathcal{E}^{\mathrm{PR}}(s_0) = \underset{\{\pi(a_t|a^{t-1},s^t)\}_{t=0}^{T}}{\text{maximum}} \; I\left[S_{T+1}; S_1^T, A_0^T \mid s_0\right] \tag{16}$$

$$\text{subject to: } \forall t \;:\; \sum_{a_t} \pi\left(a_t \mid a^{t-1}, s^t\right) = 1 \tag{17}$$

with the corresponding Lagrangian with T multipliers:

$$\begin{aligned} &L\left(\left\{\pi\left(a_t \mid a^{t-1}, s^t\right)\right\}_{t=0}^{T}, \left\{\lambda\left(a^{t-1}, s^t\right)\right\}_{t=1}^{T}\right) \\ &\quad = I\left[S_{T+1}; S_1^T, A_0^T \mid s_0\right] + \sum_{\substack{a^{t-1} \\ s^t}} \lambda\left(a^{t-1}, s^t\right)\left(\sum_{a_t} \pi\left(a_t \mid a^{t-1}, s^t\right) - 1\right), \end{aligned} \tag{18}$$

where the mutual information of the joint probability, $p(s_{T+1}, s_1^T, a_0^T \mid s_0)$, is given by:

$$I\left[S_{T+1}; S_1^T, A_0^T \mid s_0\right] = \sum_{\substack{s_1^{T+1} \\ a_0^T}} p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) p\left(s_1^T, a_0^T \mid s_0\right) \log\left(\frac{q\left(s_1^T, a_0^T \mid s_0, s_{T+1}\right)}{p\left(s_1^T, a_0^T \mid s_0\right)}\right) \tag{19}$$

with $p(s_{T+1} \mid s_1^T, a_0^T, s_0)$, $p(s_1^T, a_0^T \mid s_0)$, and $q(s_1^T, a_0^T \mid s_0, s_{T+1})$, the channel, the source, and the reverse channel respectively. The source and the reverse channel at particular time t' factorize as follows:

$$p\left(s_1^T, a_0^T \mid s_0\right) = p\left(s_{t'+1}^T, a_{t'+1}^T \mid s_0^{t'}, a_0^{t'}\right) p\left(s_{t'}, a_{t'} \mid a_0^{t'-1}, s_0^{t'-1}\right) p\left(a_1^{t'-1}, s_1^{t'-1} \mid a_0, s_0\right) \pi\left(a_0 \mid s_0\right) \tag{20}$$

$$q\left(s_1^T, a_0^T \mid s_0, s_{T+1}\right) = q\left(s_{t'+1}^T, a_{t'+1}^T \mid s_0^{t'}, a_0^{t'}, s_{T+1}\right) q\left(s_{t'}, a_{t'} \mid s_0^{t'-1}, a_0^{t'-1}, s_{T+1}\right) q\left(s_1^{t'-1}, a_0^{t'-1} \mid s_0, s_{T+1}\right). \tag{21}$$

The probability distributions in equations (20) and (21) are factorized as following:

$$p\left(s_1^T, a_0^T \mid s_0\right) = \prod_{t=1}^{T}\left(\pi\left(a_t \mid a_0^{t-1}, s_0^t\right) p\left(s_t \mid s_0^{t-1}, a_0^{t-1}\right)\right) \pi\left(a_0 \mid s_0\right) \tag{22}$$

$$= \prod_{t=t'+1}^{T} \pi\left(a_t \mid a_0^{t-1}, s_0^t\right) p\left(s_t \mid s_0^{t-1}, a_0^{t-1}\right) \pi\left(a_{t'} \mid a_0^{t'-1}, s_0^{t'}\right) p\left(s_{t'} \mid s_0^{t'-1}, a_0^{t'-1}\right) \tag{23}$$

$$\times \prod_{t=1}^{t'-1} \pi\left(a_t \mid a_0^{t-1}, s_0^t\right) p\left(s_t \mid s_0^{t-1}, a_0^{t-1}\right) \pi\left(a_0 \mid s_0\right) \tag{24}$$

$$\begin{aligned} &= p\left(s_{t'+1}^T, a_{t'+1}^T \mid s_0^{t'}, a_0^{t'}\right) \pi\left(a_{t'} \mid a_0^{t'-1}, s_0^{t'}\right) p\left(s_{t'} \mid s_0^{t'-1}, a_0^{t'-1}\right) \\ &\quad \times p\left(a_1^{t'-1}, s_1^{t'-1} \mid a_0, s_0\right) \pi\left(a_0 \mid s_0\right) \end{aligned} \tag{25}$$

$$q\left(s_1^T, a_0^T \mid s_0, s_{T+1}\right) = \prod_{t=1}^{T}\left(q\left(s_t, a_t \mid s_0^{t-1}, a_0^{t-1}, s_{T+1}\right)\right) q\left(a_0 \mid s_0, s_{T+1}\right) \tag{26}$$

$$= q\left(s_{t'+1}^T, a_{t'+1}^T \mid s_0^{t'}, a_0^{t'}, s_{T+1}\right) q\left(s_{t'}, a_{t'} \mid s_0^{t'-1}, a_0^{t'-1}, s_{T+1}\right) q\left(s_1^{t'-1}, a_0^{t'-1} \mid s_0, s_{T+1}\right) \tag{27}$$

$$\frac{\delta p\left(s_1^T, a_0^T \mid s_0\right)}{\delta \pi\left(a_{t'} \mid a_0^{t'-1}, s_0^{t'}\right)} = \begin{cases} \prod_{\substack{t=1 \\ t \neq t'}}^{T}\left(\pi\left(a_t \mid a_0^{t-1}, s_0^t\right) p\left(s_t \mid s_0^{t-1}, a_0^{t-1}\right)\right) \pi\left(a_0 \mid s_0\right) & \forall t' \in [1, \ldots, T] \\ \prod_{t=1}^{T}\left(\pi\left(a_t \mid a_0^{t-1}, s_0^t\right) p\left(s_t \mid s_0^{t-1}, a_0^{t-1}\right)\right) & t' = 0. \end{cases} \tag{28}$$

Furthermore, for any t':

$$
\begin{aligned}
I\left[S_{T+1}; S_1^T, A_0^T \mid s_0\right] = & \sum_{\substack{s_1^{T+1} \\ a_0^T}} p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) p\left(s_1^T, a_0^T \mid s_0\right) \log\left(\frac{q\left(s_{t'+1}^T, a_{t'+1}^T \mid s_0^{t'}, a_0^{t'}, s_{T+1}\right)}{p\left(s_{t'+1}^T, a_{t'+1}^T \mid s_0^{t'}, a_0^{t'}\right)}\right) \\
& + \sum_{\substack{s_1^{T+1} \\ a_0^T}} p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) p\left(s_1^T, a_0^T \mid s_0\right) \log\left(\frac{q\left(s_{t'}, a_{t'} \mid s_0^{t'-1}, a_0^{t'-1}, s_{T+1}\right)}{\pi\left(a_{t'} \mid a_0^{t'-1}, s_0^{t'}\right) p\left(s_{t'} \mid s_0^{t'-1}, a_0^{t'-1}\right)}\right) \\
& + \sum_{\substack{s_1^{T+1} \\ a_0^T}} p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) p\left(s_1^T, a_0^T \mid s_0\right) \log\left(\frac{q\left(s_1^{t'-1}, a_0^{t'-1} \mid s_0, s_{T+1}\right)}{p\left(a_1^{t'-1}, s_1^{t'-1} \mid a_0, s_0\right)}\right).
\end{aligned} \tag{29}
$$

The functional derivative of the Lagrangian in equation (29) with regard to policy, $\frac{\delta L}{\delta \pi(a_{t'} | a^{t'-1}, s^{t'})}$, is

$$
\sum_{\substack{s_{t'+1}^{T+1} \\ a_{t'+1}^T}} p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) \frac{\delta p\left(s_1^T, a_0^T \mid s_0\right)}{\delta \pi\left(a_{t'} \mid a_0^{t'-1}, s_0^{t'}\right)} \log\left(\frac{q\left(s_{t'+1}^T, a_{t'+1}^T \mid s_0^{t'}, a_0^{t'}, s_{T+1}\right)}{p\left(s_{t'+1}^T, a_{t'+1}^T \mid a_0^{t'}, a_0^{t'}\right)}\right) \tag{30}
$$

$$
+ \sum_{\substack{s_{t'+1}^{T+1} \\ a_{t'+1}^T}} p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) \frac{\delta p\left(s_1^T, a_0^T \mid s_0\right)}{\delta \pi\left(a_{t'} \mid a_0^{t'-1}, s_0^{t'}\right)} \log\left(\frac{q\left(s_{t'}, a_{t'} \mid s_0^{t'-1}, a_0^{t'-1}, s_{T+1}\right)}{\pi\left(a_{t'} \mid a_0^{t'-1}, s_0^{t'}\right) p\left(s_{t'} \mid s_0^{t'-1}, a_0^{t'-1}\right)}\right) \tag{31}
$$

$$
- \sum_{\substack{s_{t'+1}^{T+1} \\ a_{t'+1}^T}} p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) p\left(s_1^T, a_0^T \mid s_0\right) \frac{1}{\pi\left(a_{t'} \mid a_0^{t'-1}, s_0^{t'}\right)} \tag{32}
$$

$$
+ \sum_{\substack{s_{t'+1}^{T+1} \\ a_{t'+1}^T}} p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) \frac{\delta p\left(s_1^T, a_0^T \mid s_0\right)}{\delta \pi\left(a_{t'} \mid a_0^{t'-1}, s_0^{t'}\right)} \log\left(\frac{q\left(s_1^{t'-1}, a_0^{t'-1} \mid s_0, s_{T+1}\right)}{p\left(a_1^{t'-1}, s_1^{t'-1} \mid a_0, s_0\right)}\right) + \lambda\left(a_0^{t'-1}, s_0^{t'}\right) = 0 \tag{33}
$$

where the functional derivative $\frac{\delta p(s_1^T, a_0^T | s_0)}{\delta \pi(a_{t'} | a_0^{t'-1}, s_0^{t'})}$ is given by

$$
\frac{\delta p\left(s_1^T, a_0^T \mid s_0\right)}{\delta \pi\left(a_{t'} \mid a_0^{t'-1}, s_0^{t'}\right)} = \prod_{t=t'+1}^{T} \left(\pi\left(a_t \mid a_0^{t-1}, s_0^t\right) p\left(s_t \mid s_0^{t-1}, a_0^{t-1}\right)\right) \prod_{t=1}^{t'-1} \left(\pi\left(a_t \mid a_0^{t-1}, s_0^t\right) p\left(s_t \mid s_0^{t-1}, a_0^{t-1}\right)\right) \pi\left(a_0 \mid s_0\right) \tag{34}
$$

$$
= p\left(a_{t'+1}^T, s_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\right) p\left(a_0^{t'-1}, s_1^{t'-1} \mid s_0\right). \tag{35}
$$

Then,

$$
\sum_{\substack{s_{t'+1}^{T+1} \\ a_{t'+1}^T}} p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) p\left(a_{t'+1}^T, s_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\right) p\left(a_0^{t'-1}, s_1^{t'-1} \mid s_0\right) \log\left(\frac{q\left(s_{t'+1}^T, a_{t'+1}^T \mid s_0^{t'}, a_0^{t'}, s_{T+1}\right)}{p\left(a_{t'+1}^T, a_{t'+1}^T \mid a_0^{t'}, a_0^{t'}\right)}\right) \tag{36}
$$

$$
\begin{aligned}
& + \sum_{\substack{s_{t'+1}^{T+1} \\ a_{t'+1}^T}} p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) p\left(a_{t'+1}^T, s_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\right) p\left(a_0^{t'-1}, s_1^{t'-1} \mid s_0\right) \\
& \times \log\left(\frac{q\left(s_{t'}, a_{t'} \mid s_0^{t'-1}, a_0^{t'-1}, s_{T+1}\right)}{\pi\left(a_{t'} \mid a_0^{t'-1}, s_0^{t'}\right) p\left(s_{t'} \mid s_0^{t'-1}, a_0^{t'-1}\right)}\right)
\end{aligned} \tag{37}
$$

$$-\sum_{\substack{s_{t'+1}^{T+1}\\ a_{t'+1}^{T}}} p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) p\left(a_{t'+1}^T, s_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\right) p\left(a_0^{t'-1}, s_1^{t'-1} \mid s_0\right) \tag{38}$$

$$+\sum_{\substack{s_{t'+1}^{T+1}\\ a_{t'+1}^{T}}} p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) p\left(a_{t'+1}^T, s_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\right) p\left(a_0^{t'-1}, s_1^{t'-1} \mid s_0\right)$$

$$\times \log\left(\frac{q\left(s_1^{t'-1}, a_0^{t'-1} \mid s_0, s_{T+1}\right) q\left(a_0 \mid s_0, s_{T+1}\right)}{p\left(a_1^{t'-1}, s_1^{t'-1} \mid a_0, s_0\right) \pi\left(a_0 \mid s_0\right)}\right) \tag{39}$$

$$+\lambda\left(a_0^{t'-1}, s_0^{t'}\right) = 0. \tag{40}$$

Summing over $(s_{t'+1}^{T+1}, a_{t'+1}^T)$ and noting the normalized probabilities, equation (38) reduces to the term $p(a_0^{t'-1}, s_1^{t'-1} \mid s_0)$, which is absorbed in $\lambda(a_0^{t'-1}, s_0^{t'})$, giving:

$$\sum_{\substack{s_{t'+1}^{T+1}\\ a_{t'+1}^{T}}} p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) p\left(a_{t'+1}^T, s_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\right) \log\left(\frac{q\left(s_{t'+1}^T, a_{t'+1}^T \mid s_0^{t'}, a_0^{t'}, s_{T+1}\right)}{p\left(s_{t'+1}^T, a_{t'+1}^T \mid a_0^{t'}, a_0^{t'}\right)}\right) \tag{41}$$

$$+\sum_{\substack{s_{t'+1}^{T+1}\\ a_{t'+1}^{T}}} p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) p\left(a_{t'+1}^T, s_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\right) \log\left(\frac{q\left(s_{t'}, a_{t'} \mid s_0^{t'-1}, a_0^{t'-1}, s_{T+1}\right)}{\pi\left(a_{t'} \mid a_0^{t'-1}, s_0^{t'}\right) p\left(s_{t'} \mid s_0^{t'-1}, a_0^{t'-1}\right)}\right) \tag{42}$$

$$+\sum_{\substack{s_{t'+1}^{T+1}\\ a_{t'+1}^{T}}} p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) p\left(a_{t'+1}^T, s_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\right) \log\left(\frac{q\left(s_1^{t'-1}, a_0^{t'-1} \mid s_0, s_{T+1}\right) q\left(a_0 \mid s_0, s_{T+1}\right)}{p\left(a_1^{t'-1}, s_1^{t'-1} \mid a_0, s_0\right) \pi\left(a_0 \mid s_0\right)}\right) \tag{43}$$

$$+\lambda\left(a_0^{t'-1}, s_0^{t'}\right) = 0. \tag{44}$$

Equation (43) does not depend on the past policies $p(a_1^{t'-1}, s_1^{t'-1} \mid a_0, s_0)\pi(a_0 \mid s_0)$ because:

$$\sum_{\substack{s_{t'+1}^{T+1}\\ a_{t'+1}^{T}}} p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) p\left(a_{t'+1}^T, s_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\right) \log\left(p\left(a_1^{t'-1}, s_1^{t'-1} \mid a_0, s_0\right) \pi\left(a_0 \mid s_0\right)\right) \tag{45}$$

$$= \log\left(p\left(a_1^{t'-1}, s_1^{t'-1} \mid a_0, s_0\right) \pi\left(a_0 \mid s_0\right)\right) \times \sum_{\substack{s_{t'+1}^{T+1}\\ a_{t'+1}^{T}}} p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) p\left(a_{t'+1}^T, s_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\right) \tag{46}$$

$$= \log\left(p\left(a_1^{t'-1}, s_1^{t'-1} \mid a_0, s_0\right) \pi\left(a_0 \mid s_0\right)\right) \times 1, \tag{47}$$

which is absorbed in $\lambda(a_0^{t'-1}, s_0^{t'})$. Then combining all the q terms over all times

$$\sum_{\substack{s_{t'+1}^{T+1}\\ a_{t'+1}^{T}}} p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) p\left(a_{t'+1}^T, s_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\right) \log\left(q\left(s_1^T, a_0^T \mid s_0, s_{T+1}\right)\right) \tag{48}$$

$$-\sum_{\substack{s_{t'+1}^{T}\\ a_{t'+1}^{T}}} p\left(a_{t'+1}^T, s_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\right) \log\left(p\left(a_{t'+1}^T, s_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\right)\right) - \log\left(\pi\left(a_{t'} \mid a_0^{t'-1}, s_0^{t'}\right)\right) + \lambda\left(a_0^{t'-1}, s_0^{t'}\right) = 0 \tag{49}$$

where only the denominators from t' onwards survive. It follows:

$$\sum_{\substack{s_{t'+1}^{T+1}\\ a_{t'+1}^{T}}} p\left(s_{T+1} \mid s_1^T, a_0^T, s_0\right) p\left(a_{t'+1}^T, s_{t'+1}^T \mid a_0^{t'}, s_0^{t'}\right) \log\left(q\left(s_{t'+1}^T, a_{t'+1}^T \mid s_1^{t'}, a_0^{t'}, s_0, s_{T+1}\right)\right) \tag{50}$$

splitting the reverse channel into a part up to and following t'

$$+\sum_{\substack{s_{t'+1}^{T+1}\\ a_{t'+1}^{T}}} p\left(s_{T+1}\mid s_1^T,a_0^T,s_0\right) p\left(a_{t'+1}^T,s_{t'+1}^T\mid a_0^{t'},s_0^{t'}\right)\log\left(q\left(s_1^{t'},a_0^{t'}\mid s_0,s_{T+1}\right)\right) \tag{51}$$

$$-\sum_{\substack{s_{t'+1}^{T}\\ a_{t'+1}^{T}}} p\left(a_{t'+1}^T,s_{t'+1}^T\mid a_0^{t'},s_0^{t'}\right)\log\left(p\left(a_{t'+1}^T,s_{t'+1}^T\mid a_0^{t'},s_0^{t'}\right)\right) \tag{52}$$

$$-\log\left(\pi\left(a_{t'}\mid a_0^{t'-1},s_0^{t'}\right)\right)+\lambda\left(a_0^{t'-1},s_0^{t'}\right)=0. \tag{53}$$

Combining equation (50) with equation (52) for the optimal future policies gives 'future process empowerment' and 'past process empowerment' along the past process, respectively:

$$\mathcal{E}^{\mathrm{FPR}}\left[S_{T+1};A_{t'+1}^T,S_{t'+1}^T\mid a_0^{t'},s_0^{t'}\right]=\sum_{\substack{s_{t'+1}^{T+1}\\ a_{t'+1}^{T}}} p\left(s_{T+1}\mid s_1^T,a_0^T,s_0\right) p\left(a_{t'+1}^T,s_{t'+1}^T\mid a_0^{t'},s_0^{t'}\right) \tag{54}$$

$$\times\log\left(q\left(s_{t'+1}^T,a_{t'+1}^T\mid s_1^{t'},a_0^{t'},s_0,s_{T+1}\right)\right) \tag{55}$$

$$-\sum_{\substack{s_{t'+1}^{T}\\ a_{t'+1}^{T}}} p\left(a_{t'+1}^T,s_{t'+1}^T\mid a_0^{t'},s_0^{t'}\right)\log\left(p\left(a_{t'+1}^T,s_{t'+1}^T\mid a_0^{t'},s_0^{t'}\right)\right)$$

$$\mathcal{E}^{\mathrm{PPR}}\left[S_{T+1};s_1^{t'},a_0^{t'}\mid s_0\right]=\sum_{\substack{s_{t'+1}^{T+1}\\ a_{t'+1}^{T}}} p\left(s_{T+1}\mid s_1^T,a_0^T,s_0\right) p\left(a_{t'+1}^T,s_{t'+1}^T\mid a_0^{t'},s_0^{t'}\right)\log\left(q\left(s_1^{t'},a_0^{t'}\mid s_0,s_{T+1}\right)\right). \tag{56}$$

Then, the equations (50)–(53) appear as:

$$\mathcal{E}^{\mathrm{PR}}\left[S_{T+1};A_{t'+1}^T.S_{t'+1}^T\mid a_0^{t'},s_0^{t'}\right]+\mathcal{E}^{\mathrm{PR}}\left[S_{T+1};s_1^{t'},a_0^{t'}\mid s_0\right]-\log\left(\pi\left(a_{t'}\mid a_0^{t'-1},s_0^{t'}\right)\right)+\lambda\left(a_0^{t'-1},s_0^{t'}\right)=0 \tag{57}$$

and, the optimal policy and the optimal reverse channel normalized by the means of $\lambda(a_0^{t'-1},s_0^{t'})$ are given by:

$$\pi\left(a_{t'}\mid a_0^{t'-1},s_0^{t'}\right)=\frac{\exp\left(\mathcal{E}^{\mathrm{PR}}\left[S_{T+1};A_{t'+1}^T.S_{t'+1}^T\mid a_0^{t'},s_0^{t'}\right]+\mathcal{E}^{\mathrm{PR}}\left[S_{T+1};s_1^{t'},a_0^{t'}\mid s_0\right]\right)}{\sum_{a_{t'}}\exp\left(\mathcal{E}^{\mathrm{PR}}\left[S_{T+1};A_{t'+1}^T.S_{t'+1}^T\mid a_0^{t'},s_0^{t'}\right]+\mathcal{E}^{\mathrm{PR}}\left[S_{T+1};s_1^{t'},a_0^{t'}\mid s_0\right]\right)} \tag{58}$$

$$q\left(s_1^T,a_0^T\mid s_0,s_{T+1}\right)=\frac{p\left(s_{T+1}\mid s_1^T,a_0^T,s_0\right)p\left(s_1^T,a_0^T\mid s_0\right)}{\sum\limits_{s_1^T,a_0^T} p\left(s_{T+1}\mid s_1^T,a_0^T,s_0\right)p\left(s_1^T,a_0^T\mid s_0\right)}. \tag{59}$$

□

ORCID iDs

Stas Tiomkin 0000-0003-3677-6874
Christoph Salge 0000-0001-5520-8755
Daniel Polani 0000-0002-3233-5847

References

[1] Tiomkin S *et al* 2025 Code repository for 'process empowerment for robust intrinsic motivation' (available at: https://github.com/stastio/pe.git)

[2] Anthony T, Polani D and Nehaniv C L 2014 General self-motivation and strategy identification: case studies based on Sokoban and Pac-Man *IEEE Trans. Comput. Intellig. AI Games* **6** 1–17

[3] Arimoto S 1972 An algorithm for computing the capacity of arbitrary discrete memoryless channels *IEEE Trans. Inf. Theory* **18** 14–20

[4] Bale R, Hao M, Bhalla A P S and Patankar N A 2014 Energy efficiency and allometry of movement of swimming and flying animals *Proc. Natl Acad. Sci.* **111** 7517–21

[5] Cottier B, Rahman R, Fattorini L, Maslej N and Owen D 2024 The rising costs of training frontier ai models (arXiv:2405.21015)

[6] Dai S, Wei X, Hofmann A and Williams B 2020 An empowerment-based solution to robotic manipulation tasks with sparse rewards (arXiv:2010.07986)

[7] Du Y, Kosoy E, Dayan A, Rufova M, Abbeel P and Gopnik A 2023 What can AI learn from human exploration? Intrinsically-motivated humans and agents in open-world exploration *NeurIPS 2023 Workshop: Information-Theoretic Principles in Cognitive Systems*

[8] Gregor K, Rezende D J and Wierstra D 2017 Variational intrinsic control *Proc. Int. Conf. on Learning Representations (ICLR) 2017, Workshop Track*

[9] Jung T, Polani D and Stone P 2011 Empowerment for continuous agent-environment systems *Adapt. Behav.* **19** 16–39

[10] Klyubin A S, Polani D and Nehaniv C L 2005 Empowerment: a universal agent-centric measure of control *2005 IEEE Congress on Evolutionary Computation* vol 1 (IEEE) pp 128–35

[11] Kwon T 2021 Variational intrinsic control revisited *Int. Conf. on Learning Representations ICLR*

[12] Naiss I and Permuter H H 2012 Extension of the Blahut–Arimoto algorithm for maximizing directed information *IEEE Trans. Inf. Theory* **59** 204–22

[13] Papala H, Polani D and Tiomkin S 2024 Decentralized traffic flow optimization through intrinsic motivation *27th IEEE Int. Conf. on Intelligent Transportation Systems ITSC (Edmonton, Canada)* IEEE)

[14] Price W N and Cohen I J 2019 The price of artificial intelligence *J. Law, Med. Ethics* **47** 513–20

[15] Rayyes R 2023 Intrinsic motivation learning for real robot applications *Front. Robot. AI* **10** 1102438

[16] Salge C, Glackin C and Polani D 2013 Empowerment–an introduction *Guided Self-Organization: Inception* (Springer) pp 67–114

[17] Salge C and Polani D 2016 Dealing with noise and other agents: the empowerment of controller selection *Proc. Workshop on Guided Self-Organization (GSO) at Alife*

[18] Salge C and Polani D 2017 Empowerment as replacement for the three laws of robotics *Front. Robot. AI* **4** 06

[19] Tiomkin S, Nemenman I, Polani D and Tishby N 2024 Intrinsic motivation in dynamical control systems *PRX Life* **2** 033009

[20] van der Heiden T, Mirus F and van Hoof H 2020 Social navigation with human empowerment-driven deep reinforcement learning (arXiv:2003.08158)

[21] van der Heiden T, van Hoof H, Gavves E and Salge C 2022 Reliably re-acting to partner's actions with the social intrinsic motivation of transfer empowerment *Proc. 2022 Conf. on Artificial Life (ALIFE 2022)* (MIT Press)

[22] van Rossum M C W 2023 Competitive plasticity to reduce the energetic costs of learning (arXiv:2304.02594)

[23] Volpi N C and Polani D 2013 Goal-directed empowerment: combining intrinsic motivation and task-oriented behavior *IEEE Trans. Cogn. Dev. Syst.* **15** 361–72

[24] Zhao R, Kevin L, Abbeel P and Tiomkin S 2021 Efficient empowerment estimation for unsupervised stabilization *Int. Conf. on Learning Representations, ICLR 2021*