nature machine intelligence

Article

https://doi.org/10.1038/s42256-024-00829-3

Maximum diffusion reinforcement learning

Received: 3 August 2023

Accepted: 19 March 2024

Published online: 02 May 2024

Check for updates

Thomas A. Berrueta, Allison Pinosky & Todd D. Murphey

Robots and animals both experience the world through their bodies and senses. Their embodiment constrains their experiences, ensuring that they unfold continuously in space and time. As a result, the experiences of embodied agents are intrinsically correlated. Correlations create fundamental challenges for machine learning, as most techniques rely on the assumption that data are independent and identically distributed. In reinforcement learning, where data are directly collected from an agent's sequential experiences, violations of this assumption are often unavoidable. Here we derive a method that overcomes this issue by exploiting the statistical mechanics of ergodic processes, which we term maximum diffusion reinforcement learning. By decorrelating agent experiences, our approach provably enables single-shot learning in continuous deployments over the course of individual task attempts. Moreover, we prove our approach generalizes well-known maximum entropy techniques and robustly exceeds state-of-the-art performance across popular benchmarks. Our results at the nexus of physics, learning and control form a foundation for transparent and reliable decision-making in embodied reinforcement learning agents.

Reinforcement learning (RL) is a flexible decision-making framework based on the experiences of artificial agents, whose potential for scalable real-world impact has been well-established through the power of deep learning architectures. From controlling nuclear fusion reactors[1] to besting curling champions[2], RL agents have achieved remarkable feats when they can exhaustively explore how their actions impact the state of their environment. Despite their impressive achievements, RL agents—especially deep RL agents—suffer from limitations preventing their widespread deployment in the real world: their performance varies across initializations, their sample inefficiency demands the use of simulators, and they struggle to learn outside of episodic problem structures[3–5]. At the heart of these shortcomings lies a violation of the assumption that data are independent and identically distributed (i.i.d.), which underlies most of machine learning. Learning typically requires i.i.d. data, but the experiences of RL agents are unavoidably sequential and correlated across points in time. It is no wonder, then, that many of deep RL's most impactful advances have sought to overcome this roadblock[6–8].

In recent decades, researchers have started to converge onto an understanding that destroying temporal correlations is essential to sample efficiency and agent performance, seeking to address them in two primary domains: during optimization and during sample generation. When we consider optimizing a policy from a database of sequential agent–environment interactions, sampling in random batches is known to reduce temporal correlations. For this reason, experience replay[9] and its many variants[10–12] have been successful in producing large performance and sample efficiency gains across tasks and algorithms[13–15]. This simple insight—merely sampling experiences out of order—was a key contributing factor to one of deep RL's landmark triumphs: achieving superhuman performance in Atari video game benchmarks[16].

Nonetheless, temporal correlations also arise during data generation, where their impact cannot be alleviated through sampling alone. In turn, temporal correlations must be mitigated during data acquisition as well, which requires techniques to sufficiently randomize the sample-generation process. In this regard, the maximum entropy (MaxEnt) RL framework has emerged as a key advance[17–25]. These methods seek to generate randomness during optimization and data acquisition by maximizing the entropy of an agent's policy, which decorrelates

Department of Mechanical Engineering, Northwestern University, Evanston, IL, USA. e-mail: tberrueta@u.northwestern.edu; t-murphey@northwestern.edu

its action sequences. In doing so, MaxEnt RL techniques have been able to achieve better exploration and more robust performance[26]. However, does maximizing the entropy of an agent's policy actually decorrelate its experiences?

Here we prove that this is generally not the case. To address this gap, we introduce maximum diffusion (MaxDiff) RL, a framework that provably decorrelates agent experiences during sample generation and realizes statistics indistinguishable from i.i.d. sampling by exploiting the statistical mechanics of ergodic processes. Our approach efficiently exceeds state-of-the-art performance by diversifying agent experiences and improving state exploration. By articulating the relationship between an agent's embodiment, diffusion and learning, we prove that MaxDiff RL agents are capable of single-shot learning regardless of how they are initialized. We additionally prove that MaxDiff RL agents are robust to random seeds and environmental stochasticity, which enables consistent and reliable performance with low variance across agent deployments and learning tasks. Our work sheds a light on foundational issues holding back the field, highlighting the impact that agent properties and data acquisition can play on downstream learning tasks and paving the way towards more transparent and reliable decision-making in embodied RL agents.

Results

Temporal correlations hinder performance

Whether temporal correlations and their impact can be avoided depends on the properties of the underlying agent–environment dynamics. Completely destroying correlations between agent experiences requires the ability to discontinuously jump from state to state without continuity of experience. For some RL agents, this poses no issue. In settings where agents are disembodied, there may be nothing preventing effective exploration through jumps between uncorrelated states. This is one of the reasons deep RL recommender systems have been successful in a wide range of applications, such as YouTube video suggestions[27–29]. However, continuity of experience is essential to many RL problem domains. For instance, the smoothness of Newton's laws makes correlations unavoidable in the motions of most physical systems, even in simulation. This suggests that for systems like robots and self-driving cars, overcoming the impact of temporal correlations on performance presents a major challenge.

To illustrate the impact this can have on learning performance, we devised a toy task to evaluate deep RL algorithms as a function of correlations intrinsic to the agent's state transitions. Our toy task and agent dynamics are shown in Fig. 1a, corresponding to a double integrator system with parametrized momentum anisotropy. The task requires learning reward, dynamics and policy representations from scratch to move a planar point mass from a fixed initial position to a goal location. The system's true linear dynamics are simple enough to explicitly write down, which allows us to rigorously study temporal correlations across state transitions by analysing its controllability. Controllability is a formal property of control systems that describes their ability to reach arbitrary states in an environment[30,31]. In linearizable systems, state transitions become degenerate and irreversibly correlated when they are uncontrollable. However, if the agent is controllable, the impact of correlations can be overcome, at least in principle. Although the relationship between controllability and temporal correlations has been studied for decades[32], it is only recently that researchers have begun to study its impact on learning processes[33–35].

Figure 1 parametrically explores the relationship between our toy system's controllability properties and the learning performance of state-of-the-art deep RL algorithms. The point-mass dynamics are parametrized by $\beta \in [0, 1]$, which determines the relative difficulty of translating along the x axis (Fig. 1a). When $\beta = 0$, the system is uncontrollable and can only translate along the y axis, which illustrates the sense in which state transitions become irreversibly correlated. Although the system is formally controllable for all non-zero β, as $\beta \to 0$, fewer lateral transitions become available for the same range of actions, introducing temporal correlations along the system's x coordinate (Supplementary Fig. 1). We evaluated the performance of state-of-the-art model-based and model-free deep RL algorithms on our task—model-predictive path integral control (NN-MPPI)[36] and soft actor-critic (SAC)[7], respectively—at varying values of β from 1 to 0.001. As expected, at $\beta = 1$, both NN-MPPI and SAC are able to accomplish the toy task (Fig. 1b). However, as $\beta \to 0$, the performance of NN-MPPI and SAC degrades parametrically (Fig. 1c) until the point that neither algorithm can solve the task, as shown in Fig. 1d. Hence, temporal correlations can completely hinder the learning performance of the state of the art in deep RL even in toy problem settings such as this one, where a globally optimal policy can be analytically computed in closed form.

Failure to mitigate temporal correlations between state transitions can prevent effective exploration, severely impacting the performance of deep RL agents. As Fig. 1d illustrates, neither NN-MPPI nor SAC agents are able to sufficiently explore in the x dimension of their state space as a result of their decreasing degree of controllability (Supplementary Note 2.1). This is the case despite the fact that NN-MPPI and SAC are both MaxEnt RL algorithms[7,37] designed specifically to achieve improved exploration outcomes by decorrelating their agent's action sequences. In contrast, our proposed approach—MaxDiff RL—is able to consistently succeed at the task and is guaranteed to realize effective exploration by focusing instead on decorrelating agent experiences, or, equivalently, agent state sequences (purple in Fig. 1b–d), as we discuss in the following section.

MaxDiff exploration and learning

Most RL methods presuppose that taking random actions produces effective exploration[38,39], and sophisticated techniques like MaxEnt RL are no different. However, as we previously illustrated, whether this is actually possible depends on the agent's controllability properties and the temporal correlations these spontaneously induce in its experiences (Fig. 2c and Supplementary Note 2.1). To overcome these limitations, we propose decorrelating agent experiences as opposed to action sequences, which forms the starting point to our derivation of the MaxDiff RL framework.

Before synthesizing policies or assessing their impact on learning outcomes, we require a formalization of agent experiences. Without considering policies, we see the agent–environment state-transition dynamics as an autonomous stochastic process, whose sample paths $x(t)$ take value in a state space $\mathcal{X} \subset \mathbb{R}^d$ at each point in time within an interval $\mathcal{T} = [t_0, t]$. Then, we see agent experiences as collections of random variables parametrized by time, whose realizations $x(t)$ are the sample paths of the underlying agent–environment process. When $\mathcal{T} = \{0, \cdots, T\}$ is discrete, we use $x_{0:T}$ instead of $x(t)$. In this context, the probability density function over state trajectories, $P[x(t)]$ (or $P[x_{0:T}]$), completely characterizes an agent's experiences and their properties (Supplementary Note 2.2). We may now begin our derivation by asking, what is the most decorrelated that agent experiences can be?

To answer this question, we draw from the statistical physics literature on maximum calibre[40–42], which is a generalization of the variational principle of MaxEnt[43]. The goal of a maximum calibre variational optimization is to find the trajectory distribution $P_{\max}[x(t)]$, which optimizes an entropy functional $S[P[x(t)]]$. The optimal distribution would then describe the paths of an agent with the least-correlated experiences, but its specific form and properties depend on how the variational optimization is constrained. Without constraints, agents could sample states discontinuously and uniformly in a way that is equivalent to i.i.d. sampling but is not consistent with the continuous experiences of embodied agents (Fig. 2a,b). Hence, we tailor our assumptions to agents with continuous experiences. Then, to ensure that our optimization produces a distribution over continuous paths, we constrain the volume of states accessible within any finite time interval by bounding their fluctuations (Supplementary Note 2.3).

https://doi.org/10.1038/s42256-024-00829-3

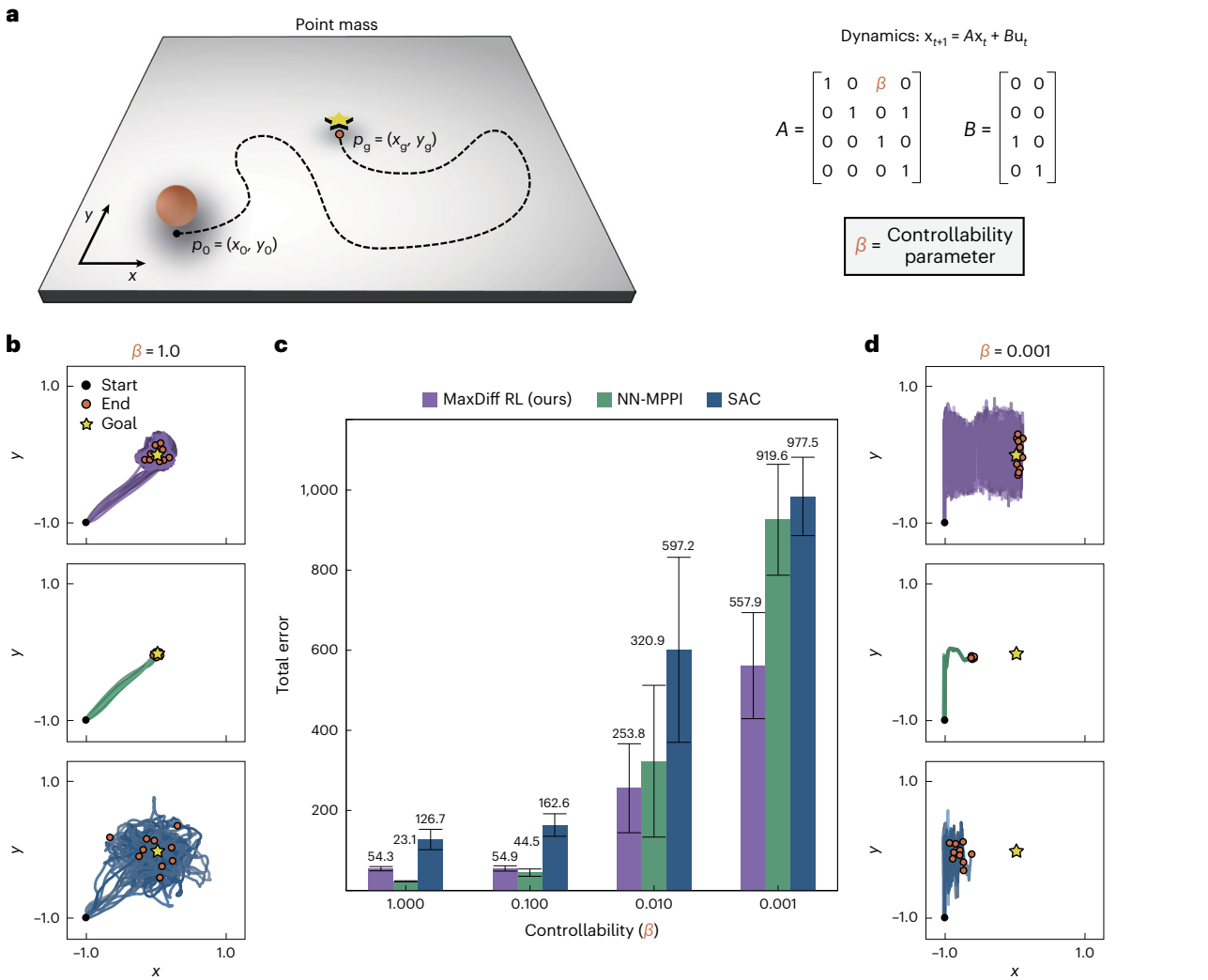


Fig. 1 | Temporal correlations break the state of the art in RL. For most systems, controllability properties determine temporal correlations between state transitions (Supplementary Note 2.1). **a**, Planar point mass with dynamics simple enough to explicitly write down and whose policy admits a globally optimal analytical solution. The system's four-dimensional state space comprises its planar positions and velocities. We parametrize its controllability through $\beta \in [0, 1]$, where $\beta = 0$ produces a formally uncontrollable system. The task is to translate the point mass from p_0 to p_g within a fixed number of steps at different values of β, and the reward is specified by the negative squared Euclidean distance between the agent's state and the goal. We compare state-of-the-art model-based and model-free algorithms, NN-MPPI and SAC, respectively, to our proposed MaxDiff RL framework (see Supplementary Note 4 for implementation details). **b**,**d**, Representative snapshots of MaxDiff RL, NN-MPPI and SAC agents (top to bottom) in well-conditioned ($\beta = 1$ (**b**)) and poorly conditioned ($\beta = 0.001$ (**d**)) controllability settings. **c**, Even in this simple system, poor controllability can break the performance of RL agents. As $\beta \to 0$, the system's ability to move in the x direction diminishes, hindering the performance of NN-MPPI and SAC, whereas MaxDiff RL remains task-capable. For all bar charts, data are presented as mean values above each error bar, where each error bar represents the standard deviation from the mean with $n = 1{,}000$ (100 evaluations over ten seeds for each condition). All differences between MaxDiff RL and comparisons within this figure are statistically significant with $P < 0.001$ using an unpaired two-sided Welch's t-test (Methods and Supplementary Table 2).

Surprisingly, this constrained variational optimization admits an analytical solution for the MaxEnt path distribution. The derived optimal path distribution is

$$P_{\max}[x(t)] = \frac{1}{Z} \exp\left[-\frac{1}{2}\int_{t_0}^{t} \dot{x}(\tau)^T C^{-1}[x(\tau)]\dot{x}(\tau)\mathrm{d}\tau\right], \quad (1)$$

where Z is a normalization constant. At every point in space $x^* \in \mathcal{X}$, the matrix $C[x^*]$ measures temporal correlations locally over an interval of duration Δt, such that

$$C[x^*] = \int_{t_i}^{t_i+\Delta t} K_{XX}(t_i, \tau)\mathrm{d}\tau, \quad (2)$$

where $K_{XX}(t_1, t_2)$ is the autocovariance of $x(t)$ at pairs of samples in time evaluated over a chosen interval, $[t_i, t_i + \Delta t] \subset \mathcal{T}$, with a given $x(t_i) = x^*$ (Supplementary Note 2.2). This distribution describes the trajectories of an optimal agent with minimally correlated paths, subject to the constraints imposed by continuity of experience. Moreover, equation (1) is equivalent to the path distribution of an anisotropic, spatially inhomogeneous diffusion process. Thus, minimizing correlations among agent trajectories leads to diffusion-like exploration whose properties can actually be analysed using statistical mechanics (Supplementary Fig. 3). This also means that the sample paths of the optimal agent are Markovian and ergodic (see Supplementary Notes 2.4 and 2.5 for associated theorems and corollaries and their proofs). Unlike alternative RL frameworks, our approach does not assume the Markov property but rather enforces it as a property intrinsic to the optimal agent's path distribution.

https://doi.org/10.1038/s42256-024-00829-3

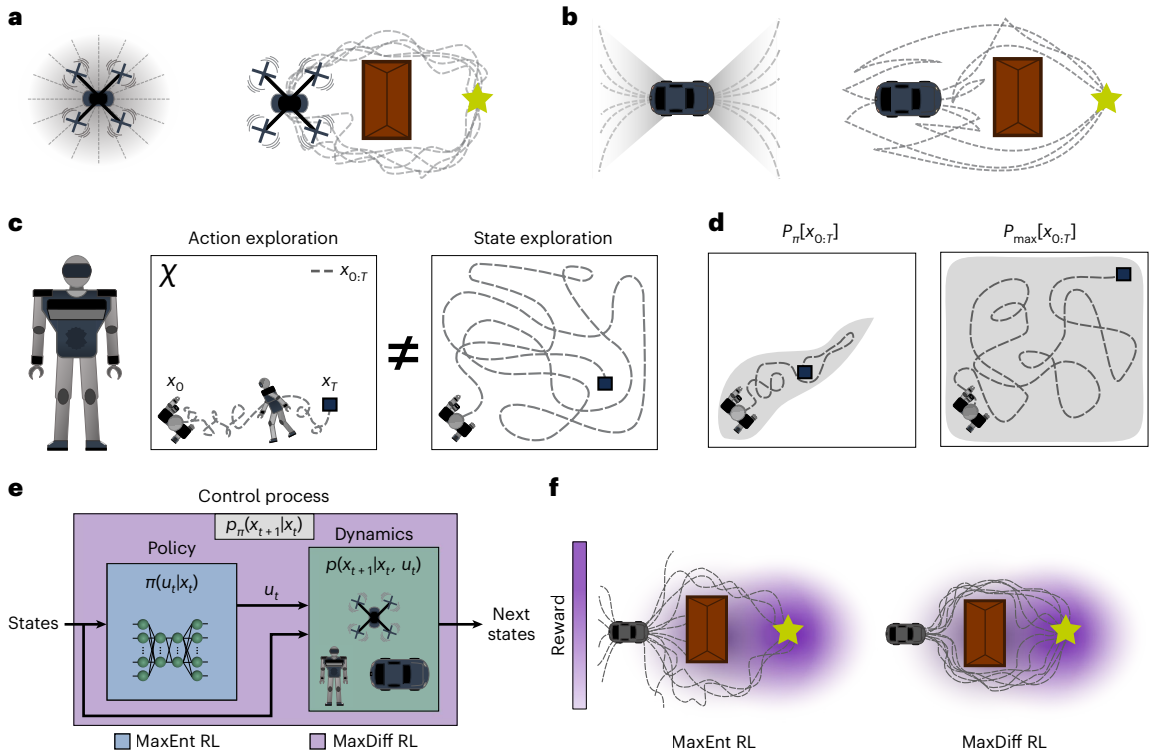


Fig. 2 | MaxDiff RL mitigates temporal correlations to achieve effective exploration. a,**b**, Systems with different planar controllability properties. Although some systems may have close to trivial controllability properties (**a**), others may be more complex (**b**). **c**, Whether action randomization leads to effective state exploration depends on the properties of the underlying state-transition dynamics (Supplementary Note 2.1), as in our illustration of a complex bipedal robot falling over and failing to explore. **d**, Although any given policy induces a path distribution (left), MaxDiff RL produces policies that maximize the path distribution's entropy (right). The projected support of the robot's path distribution is illustrated by the shaded grey region. We prove that maximizing the entropy of an agent's state transitions results in effective exploration (Supplementary Notes 2.4 and 3.6). **e**, Our approach generalizes the MaxEnt RL paradigm by considering agent trajectories. We prove that maximizing a policy's entropy does not generally maximize the entropy of an agent's state transitions (Supplementary Note 3.3). **f**, This approach leads to distinct learning outcomes because agents reason about the impact of their actions on state transitions, rather than their actions alone.

Satisfying ergodicity has profound implications for the properties of resulting agents. Ergodicity is a formal property of dynamical systems that guarantees the statistics of individual trajectories are asymptotically equivalent to those of a large ensemble of trajectories[44,45]. Put in RL terms, although the sequential nature of RL agent experiences can make i.i.d. sampling technically impossible, the global statistics of an ergodic RL agent are indistinguishable from those of an i.i.d. sampling process. In this sense, ergodic Markov sampling is the best possible alternative to i.i.d. sampling in sequential decision-making processes. Beyond resolving the issue of generating i.i.d. samples in RL, ergodicity forms the basis of many of MaxDiff RL's theoretical guarantees, as we show in the following sections.

When an agent's trajectories satisfy equation (1), we describe the agent as maximally diffusive. However, agents do not realize maximally diffusive trajectories spontaneously. Doing so requires finding a policy capable of satisfying maximally diffusive path statistics, which forms the core of what we term MaxDiff RL. Although any given policy induces a path distribution, finding policies that realize maximally diffusive trajectories requires optimization and learning (Fig. 2d). To this end, we define

$$\begin{aligned} P_\pi[x_{0:T}, u_{0:T}] &= \prod_{t=0}^{T-1} p(x_{t+1}|x_t, u_t)\pi(u_t|x_t) \\ P^r_{\max}[x_{0:T}, u_{0:T}] &= \prod_{t=0}^{T-1} p_{\max}(x_{t+1}|x_t)e^{r(x_t,u_t)}, \end{aligned} \tag{3}$$

where we discretized the distribution in equation (1) as $p_{\max}(x_{t+1}|x_t)$ and analytically rederived the optimal path distribution under the influence of a reward landscape, $r(x_t, u_t)$ (Supplementary Note 2.5). In doing so, we can account for the effect of action sequences, $u_{0:T}$, on the path distribution. Given the distributions in equation (3), the goal of MaxDiff RL can be framed as minimizing the Kullback–Leibler (KL) divergence between them, D_{KL}—that is, between the agent's current path distribution and the maximally diffusive one.

To draw connections between our framework and the broader MaxEnt RL literature, we recast the KL-divergence formulation of MaxDiff RL as an equivalent stochastic optimal control problem. In stochastic optimal control, the goal is to find a policy that maximizes the expected cumulative rewards of an agent in an environment. In this way, we can express the MaxDiff RL objective as

$$\pi^*_{\text{MaxDiff}} = \underset{\pi}{\text{argmax}}\ E_{(x_{0:T},u_{0:T})\sim P_\pi}\left[\sum_{t=0}^{T-1} \gamma^t \hat{r}(x_t, u_t)\right], \tag{4}$$

with $\gamma \in [0, 1)$ and modified rewards given by

$$\hat{r}(x_t, u_t) = r(x_t, u_t) - \alpha \log \frac{p(x_{t+1}|x_t, u_t)\pi(u_t|x_t)}{p_{\max}(x_{t+1}|x_t)}, \tag{5}$$

where $\alpha > 0$ is a temperature-like parameter that we introduce to balance diffusive exploration and reward exploitation, as we discuss in the following section. With these results in hand, we may now state one of our main theorems.

Theorem 1. (MaxDiff RL Generalizes MaxEnt RL.) *Let the state-transition dynamics due to a policy* π *be* $p_\pi(x_{t+1}|x_t) = E_{u_t\sim\pi}[p(x_{t+1}|x_t, u_t)]$.

If the state-transition dynamics are assumed to be decorrelated, then the optimum of equation (4) is reached when $D_{KL}(p_\pi||p_{max}) = 0$ and the MaxDiff RL objective reduces to the MaxEnt RL objective.

Proving this result is simple and only relies on the sense in which state transitions are decorrelated, which we discuss in detail in Supplementary Note 3.3.

Completely destroying temporal correlations generally requires discontinuous jumps between states, which can only be achieved by fully controllable agents[22]. When an agent is fully controllable, there always exists a policy that enables it to take any arbitrary path through state space. If this condition is met, then the optimum of equation (4) is attained when $p_{\pi^{\max}}(x_{t+1}|x_t) = p_{\max}(x_{t+1}|x_t)$ at each point in time, where actions are drawn from an optimized policy $\pi^{\max}$. In turn, this simplifies equation (5) and recovers the MaxEnt RL objective[7], as shown in Supplementary Note 3.3. This not only proves that MaxDiff RL is a generalization of the MaxEnt RL framework to agents with correlations in their state transitions but also makes clear that maximizing policy entropy cannot decorrelate agent experiences in general. In contrast, MaxDiff RL actively enforces path decorrelation at all points in time. We can think of this intuitively by noting that MaxDiff RL simultaneously accounts for the effect of the policy and of the temporal correlations induced by agent–environment dynamics in its optimization (Fig. 2e). As such, MaxDiff RL typically produces distinct learning outcomes from MaxEnt RL (Fig. 2f). Our result also implies that all theoretical robustness guarantees of MaxEnt RL (for example, ref. 26) should be interpreted as guarantees of MaxDiff RL when state transitions are decorrelated. Moreover, we suggest that many of the gaps between MaxEnt RL's theoretical results and their practical performance may be explained by the impact of temporal correlations, as we saw in Fig. 1.

Although these results seem to suggest that model-free implementations of MaxDiff RL are not feasible, we note that local estimates of the agent's path entropy can be learned from observations. This effectively reinterprets temporal correlations as a state-dependent property of the environment (Supplementary Note 3.5). Similar entropy estimates have been used in model-free RL[46] and more broadly in the autoencoder literature[47]. For the results in this Article, we derived a model-agnostic objective using an analytical expression for the local path entropy

$$\underset{\pi}{\operatorname{argmax}}\ E_{(x_{0:T},u_{0:T})\sim P_\pi}\left[\sum_{t=0}^{T-1} r(x_t,u_t) + \frac{\alpha}{2}\log\det C[x_t]\right], \qquad (6)$$

whose optimum realizes the same optimum as equation (4) and where we omitted γ. There are many ways to express the MaxDiff RL objective, each of which may have implementation-specific advantages (Fig. 3a and Supplementary Note 3.4). In this sense, MaxDiff RL is not a specific algorithm implementation but rather a general problem statement and solution framework, similar to MaxEnt RL. In this work, our MaxDiff RL implementation is exactly identical to NN-MPPI except for the path entropy term shown above. However, this simple modification can have a drastic effect on agent outcomes.

Robustness to initializations in ergodic agents

The introduction of an entropy term in equation (6) means that MaxDiff RL agents must balance between two aims: achieving the task and embodying diffusion (Fig. 3a). Although asymptotically there is no trade-off between maximally diffusive exploration and task exploitation, managing the relative balance between these two aims is important over finite time horizons, which we achieve with a temperature-like parameter, α. In practice, our entropy term plays a similar role as other exploration bonuses that reward agent curiosity or provide intrinsic motivation[48–50]. Unlike other bonuses, however, the role of path entropy can be interpreted through the lens of statistical mechanics. If α is set too high, the system's fluctuations can overpower the reward and break the agent's ergodicity in ways that resemble the physics of diffusion processes in potential fields[51]. Unfortunately, predicting where this critical α threshold lies is generally challenging due to its conceptual ties to the phenomenon of ergodicity-breaking in nonequilibrium processes[52].

Because ergodicity provides many of MaxDiff RL's desirable properties and guarantees, tuning the value of α is essential. In Fig. 3 and Supplementary Video 1, we explore the effect of tuning α on the learning performance of MaxDiff RL agents in MuJoCo's swimmer environment. The swimmer system comprises three rigid links of nominally equal mass, $m = 1$, with two degrees of actuation at the joints. The agent's objective is to swim as fast as possible within a fixed time interval while being subjected to viscous drag forces (Fig. 3a). In Fig. 3b, we vary α across multiple orders of magnitude and examine its impact on the terminal returns of MaxDiff RL swimmer agents. As we modulate the value of α from 1 to 100, we observe that diffusive exploration leads to greater returns. However, after $\alpha = 100$, we cross the critical threshold beyond which the strength of the system's diffusive exploration overpowers the reward (inset dashed line in Fig. 3b), thereby breaking the ergodicity of our agents with respect to the underlying potential and performing poorly at the task—just as predicted by our theoretical framework.

Given a constant temperature of $\alpha = 100$ that preserves the swimmer's ergodicity, we compared the performance of MaxDiff RL to NN-MPPI and SAC across ten seeds each. To ensure that the task was solvable by all agents, we lowered the mass of the swimmer's third link (that is, its tail) to $m_s = 0.1$. We find that whereas SAC struggles to succeed within a million environment interactions, NN-MPPI achieves good performance but with high variance across seeds. This is in stark contrast to MaxDiff RL, whose performance is near-identical and competitive across all random seeds (Fig. 3c and Supplementary Video 2). Hence, by decorrelating state transitions, our agent was able to exhibit robustness to seeds and environment randomization beyond what is typically possible in deep RL. Moreover, because our implementation of MaxDiff RL is identical to that of NN-MPPI, we can attribute any performance gains and added robustness to the properties of MaxDiff RL's theoretical framework.

Robustness to random seeds and environmental randomizations is a highly desirable feature of deep RL agents[4,53,54]. However, guaranteeing such robustness is challenging because it requires modelling the impact of neural representations on learning outcomes. Nonetheless, we can provide representation-agnostic guarantees through the probably approximately correct in Markov decision processes (PAC-MDP) learning framework[55,56]. In short, an algorithm is PAC-MDP if it is capable of generating policies that are at least ϵ-optimal at least $100 \times (1-\delta)\%$ of the time, for any $\epsilon > 0$ and $\delta \in (0, 1)$ (Methods). Under this framework, we can provide formal robustness guarantees.

Theorem 2. (MaxDiff RL Agents Are Robust To Random Seeds.) *If there exists a PAC-MDP algorithm $\mathcal{A}$ with policy π^{max} for the MaxDiff RL objective (equation (4)), then the Markov chain induced by π^{max} is ergodic, and $\mathcal{A}$ will be asymptotically ϵ-optimal regardless of initialization.*

We refer the reader to Supplementary Note 3.3 for details, but the proof follows from treating the condition for PAC-MDP learnability as an observable in Birkhoff's ergodic theorem[44]. Because maximally diffusive agents are ergodic, any two arbitrary initializations will asymptotically achieve identical learning outcomes, which implies robustness to random seeds and environmental stochasticity. Despite excluding neural representations from our analysis, Fig. 3c suggests that our guarantees hold empirically.

Zero-shot generalization across embodiments

When agents can find optimal policies, their dynamics become indistinguishable from an ergodic diffusion process. In doing so, the MaxDiff RL objective (equation (5)) reduces the influence of agent dynamics on performance. This suggests that successful MaxDiff RL policies may

https://doi.org/10.1038/s42256-024-00829-3

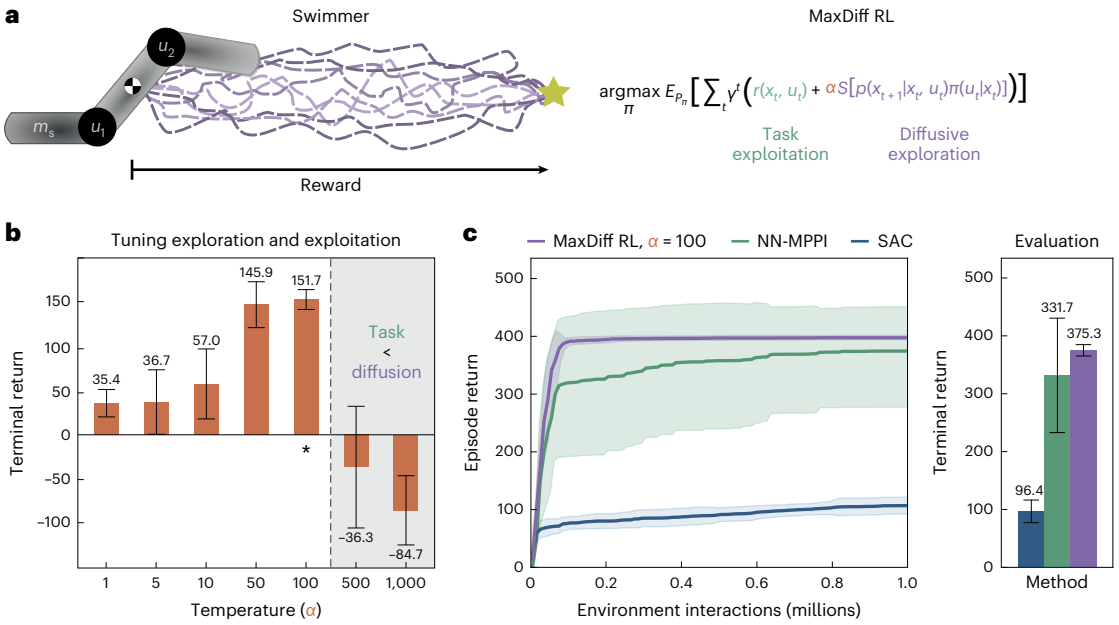


Fig. 3 | Maximally diffusive RL agents are robust to random seeds and initializations. a, Illustration of MuJoCo swimmer environment (left). The swimmer has two degrees of actuation, u_1 and u_2, that rotate its limbs at the joints, with tail mass m_s and $m = 1$ for other limbs. MaxDiff RL synthesizes robust agent behaviour by learning policies that balance task-capability and diffusive exploration (right). In practice, this balance is tuned by a temperature-like parameter, α. **b**, To explore the role that α plays in the performance of MaxDiff RL, we examine the terminal returns of swimmer agents (ten seeds each) across values of α with $m_s = 1$. Diffusive exploration leads to greater returns until a critical point (inset dashed line), after which the agent starts valuing diffusing more than accomplishing the task (see also Supplementary Video 1). **c**, Using $\alpha = 100$, we compared MaxDiff RL against SAC and NN-MPPI with $m_s = 0.1$. We observe that MaxDiff RL outperforms comparisons on average with near-zero variability across random seeds, which is a formal property of MaxDiff RL agents (see also Supplementary Video 2). For all reward curves, the shaded regions correspond to the standard deviation from the mean across ten seeds. For all bar charts, data are presented as mean values above each error bar, where each error bar represents the standard deviation from the mean with $n = 1{,}000$ (100 evaluations over ten seeds for each condition). All differences between MaxDiff RL and comparisons within this figure are statistically significant with $P < 0.001$ using an unpaired two-sided Welch's t-test (Methods and Supplementary Table 2).

exhibit favourable generalization properties across agent embodiments. To explore this possibility, as well as the robustness of MaxDiff RL agents to variations in their neural representations, we devised a transfer experiment in the MuJoCo swimmer environment. We designed two variants of the swimmer: one with a heavy, less controllable tail of $m_s = 1$ and another with a light, more controllable tail of $m_s = 0.1$ (Fig. 4a). We trained two sets of representations for each algorithm. One set was trained with the light-tailed swimmer, and another set was trained with the heavy-tailed swimmer. Then we deployed and evaluated each set of representations on both the swimmer variant observed during training and its counterpart. Our experiment's outcomes are shown in Fig. 4b,c, where the results are categorized as 'baseline' if the trained and deployed swimmer variants match or 'transfer' if they were swapped. The baseline experiments validate other results shown throughout the Article: all algorithms benefit from working with a more controllable system whose dynamics induce weaker temporal correlations (Fig. 4b and Supplementary Video 2). However, as MaxDiff RL is the only approach taking temporal correlations into account, it is the only method that remains task-capable with a heavy-tailed swimmer.

For the transfer experiments, all of the learned neural representations of the reward function, control policy and agent dynamics were deployed on the swimmer variant that was not seen during training (Fig. 4a). First, we note that for both NN-MPPI and SAC representations, transfer leads to degrading performance across the board. This is the case even when the swimmer variant they were deployed onto was more controllable, which is counterintuitive and undesirable behaviour. In contrast, our MaxDiff RL agents can actually benefit and improve their performance when deployed on the more controllable swimmer variant, as desired ('heavy-to-light' transfer in Fig. 4c and Supplementary Video 3). In other words, as the task becomes easier in this way, we can expect the performance of MaxDiff RL agents to improve.

A more surprising result is the performance increase in MaxDiff RL agents between the baseline heavy-tailed swimmer and the 'light-to-heavy' transfer swimmer (Fig. 4c and Supplementary Video 3). We found that training with a more controllable swimmer increased the performance of agents when deployed on a heavy-tailed swimmer, showing that system controllability during training matters more to overall performance than the particular embodiment of the deployed system. This kind of zero-shot generalization[57] from an easier task to a more challenging task is reminiscent of results seen in RL agents trained via curriculum learning[58] as well as of the incremental learning dynamics of biological systems during motor skill acquisition[59]. However, here it emerges spontaneously from the properties of MaxDiff RL agents. In part, this occurs because greater controllability leads to improved exploration, which increases the diversity of data observed during training.

Single-shot learning in ergodic agents

When agents are deployed in the real world, they face situations at test time that were never encountered during training. Because exhaustively accounting for every possible scenario is infeasible, agents capable of real-time adaptation and learning during individual deployments are desirable[5]. Most RL methods excel at episodic multi-shot learning over the course of several deployments (Fig. 5b), where randomized instantiations of a given task and environment passively provide a kind of variability that is essential to the learning process[60]. However, episodic problems of this kind are very rare in real-world applications. For this reason, there is a need for methods that allow agents to perform a task successfully within a single trial—or, in other words, for methods that enable single-shot learning.

https://doi.org/10.1038/s42256-024-00829-3

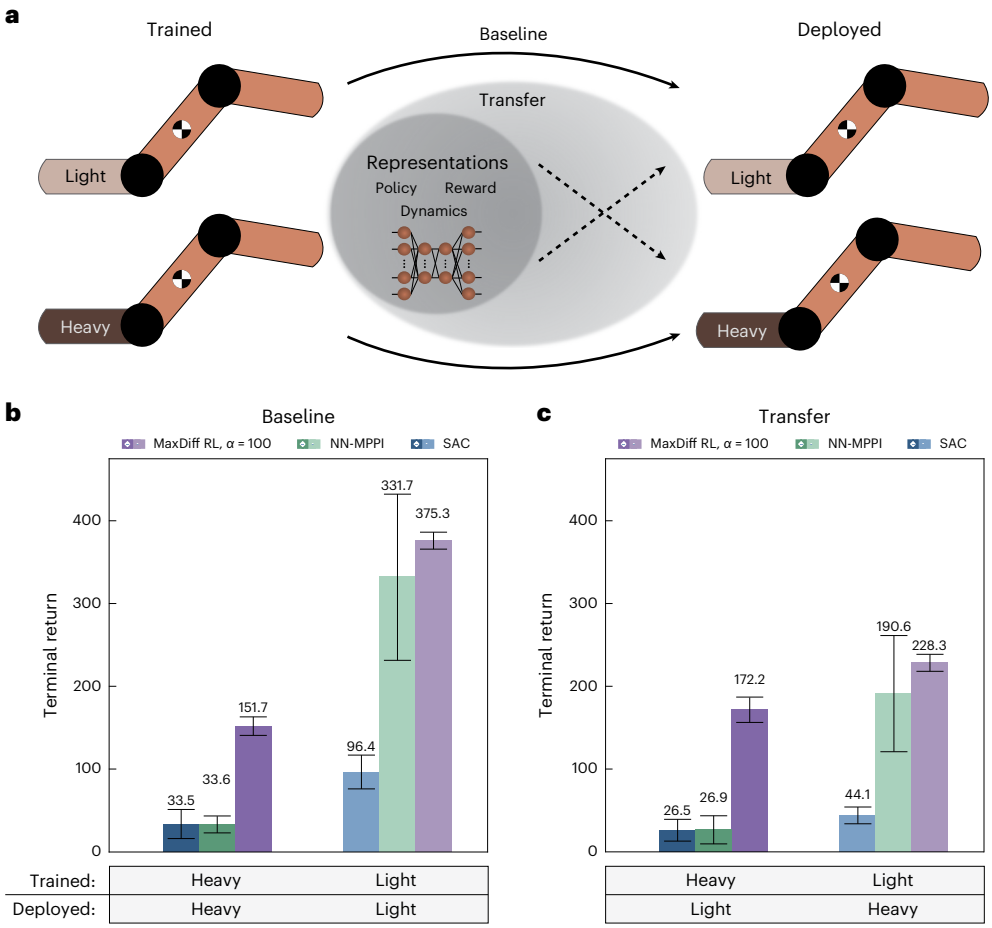


Fig. 4 | Trained system embodiment determines deployed system performance. a, Two variants of the MuJoCo swimmer environment: one with $m_s = 1$ and one with $m_s = 0.1$. As a baseline, we deploy learned representations on the same swimmer variant trained on. Then we carry out a transfer experiment where the trained and deployed swimmer variants are swapped. **b**, Baseline experiments confirm previous results: all algorithms benefit from a more controllable swimmer. Because MaxDiff RL optimizes system controllability, it is the only method capable of achieving the task with a heavy-tailed swimmer (see also Supplementary Video 2). **c**, Performance of both NN-MPPI and SAC degrades when deployed on a more controllable system than was trained on, which is undesirable. In contrast, MaxDiff RL benefits from the 'heavy-to-light' transfer because it learns policies that take advantage of a more capable system during deployment. We also observe that MaxDiff RL performance further increases in the 'light-to-heavy' transfer experiment, showing that system controllability during training is more important to overall performance than the particular embodiment of the system it is ultimately deployed on (see also Supplementary Video 3). For all bar charts, data are presented as mean values above each error bar, where each error bar represents the standard deviation from the mean with $n = 1{,}000$ (100 evaluations over ten seeds for each condition). All differences between MaxDiff RL and comparisons within this figure are statistically significant with $P < 0.001$ using an unpaired two-sided Welch's t-test (Methods and Supplementary Table 2).

Single-shot learning concerns learning in non-episodic environments over the course of a single task attempt, similar to the 'single-life' RL setting considered in ref. 61. Despite the challenges associated with studying the behaviour of agents based on neural network representations, the ergodic properties of MaxDiff RL enable one to provide representation-agnostic guarantees on the feasibility of single-shot learning through the PAC-MDP learning framework.

Theorem 3. (MaxDiff RL Agents Can Learn In Single-Shot Deployments.) *If there exists a PAC-MDP algorithm $\mathcal{A}$ with policy π^{max} for the MaxDiff RL objective (equation (4)), then the Markov chain induced by π^{max} is ergodic, and any individual initialization of $\mathcal{A}$ will asymptotically satisfy the same ϵ-optimality as an ensemble of initializations.*

Thus, any MaxDiff RL agent capable of solving a task in a multi-shot fashion (in the PAC-MDP sense) is capable of solving the same task in a single-shot fashion. This theorem also follows from Birkhoff's ergodic theorem and is closely related to Theorem 2. Because any two MaxDiff RL agents will asymptotically achieve identical learning outcomes, any individual MaxDiff RL agent will also achieve identical learning outcomes as an ensemble (see Supplementary Note 3.3 for details). Because ergodicity is central to this proof, we expect its guarantees to fail when ergodicity is broken by either the agent or the environment.

Figure 5 demonstrates the single-shot learning capabilities of MaxDiff RL agents and explores what happens when ergodicity is broken by the topological properties of the environment. Here, we examine both the MuJoCo swimmer and ant environments (Fig. 5a). The primary difference between these two environments is the existence of irreversible state transitions that can violate the ergodicity requirement of our single-shot learning guarantees topologically (Fig. 5c), which have been previously referred to as 'sink states' in the literature[60]. Unlike the swimmer, the ant is capable of transitioning into such states by flipping upside down, thereby breaking ergodicity. Irreversible state transitions are common in real-world applications because they can arise as a result of unsafe behaviour, such as a robot breaking or malfunctioning during learning. Although such transitions can be prevented in principle through the use of safety-preserving methods[62–64], we omit their implementation to illustrate our point. As expected, the MaxDiff RL single-shot swimmer is capable of learning in continuous

https://doi.org/10.1038/s42256-024-00829-3

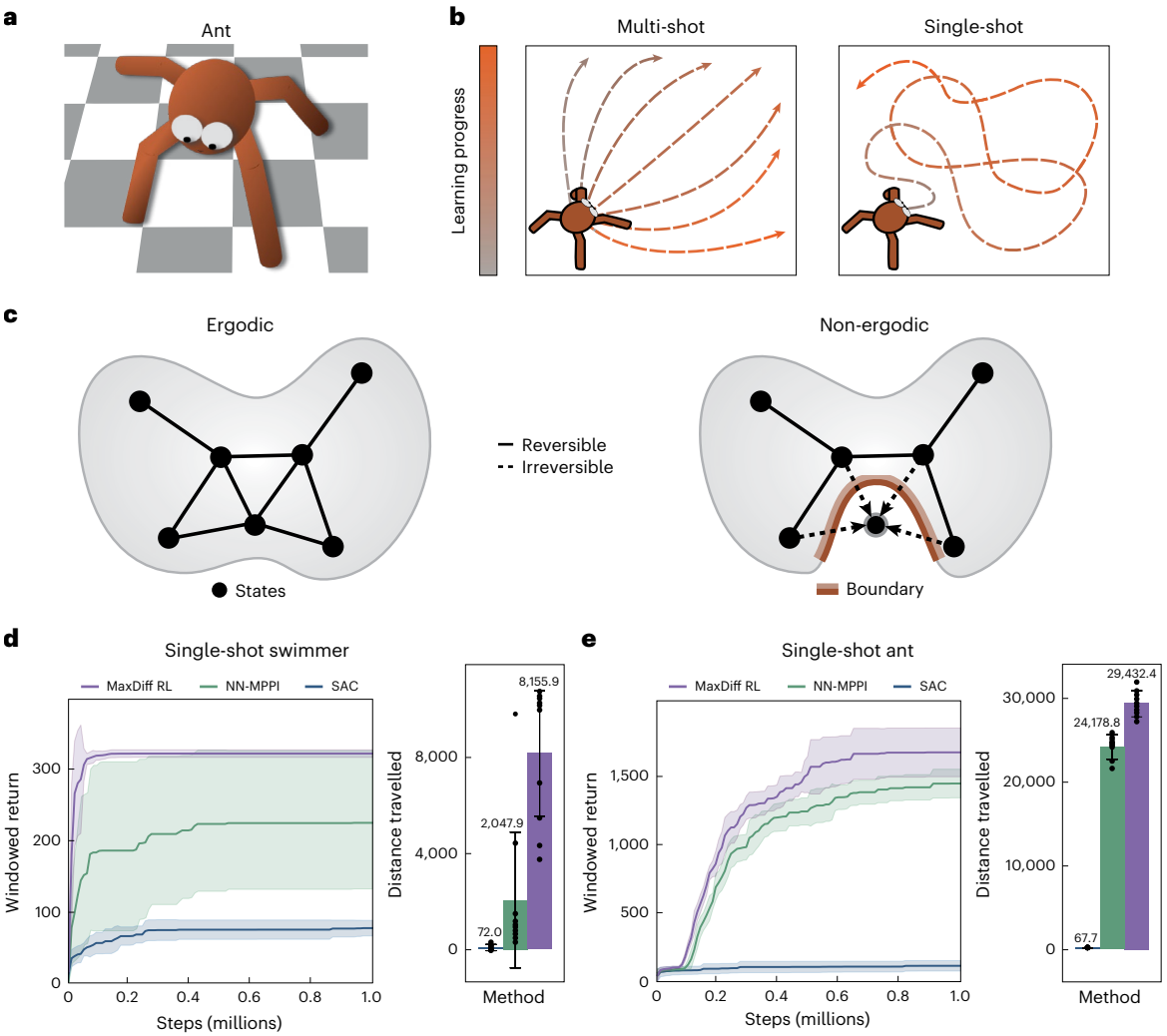


Fig. 5 | Maximally diffusive RL agents are capable of single-shot learning. **a**, Illustration of MuJoCo ant environment. **b**, Typical algorithms learn across many different initializations and deployments of an agent, which is known as multi-shot learning. In contrast, single-shot learning insists on a single task attempt, which requires learning through continuous deployments. Here, we prove that MaxDiff RL agents are equivalently capable of single-shot and multi-shot learning in a broad variety of settings. **c**, Single-shot learning depends on the ability to generate data samples ergodically, which MaxDiff RL guarantees when there are no irreversible state transitions in the environment. **d**, Single-shot learning in the swimmer MuJoCo environment. We find that MaxDiff RL achieves robust performance comparable to its multi-shot counterpart (see also Supplementary Video 4). **e**, In contrast to the swimmer, the MuJoCo ant environment contains irreversible state transitions (for example, flipping upside down) preventing ergodic trajectories. Nonetheless, MaxDiff RL remains state of the art in single-shot learning. Note that we report returns over a window of 1,000 steps in analogy to our multi-shot results, where episodes consist of 1,000 environment interactions. For all reward curves, the shaded regions correspond to the standard deviation from the mean across ten seeds. For all bar charts, data are presented as mean values above each error bar, where each error bar represents the standard deviation from the mean and the data distribution is plotted directly (n = ten seeds for each condition). All differences between MaxDiff RL and comparisons within this figure are statistically significant with $P < 0.001$ using an unpaired two-sided Welch's t-test (Methods and Supplementary Table 2).

deployments (Fig. 5d and Supplementary Video 4), retaining the same robustness of its multi-shot counterpart in Fig. 3c and achieving similar task performance. Despite ergodicity-breaking in the single-shot ant environment, MaxDiff RL still leads to improved outcomes over NN-MPPI and SAC, as in Fig. 5e, where we plot the final distance travelled to ensure that no reward hacking took place. However, the loss of ergodicity leads to an increase in the variance of single-shot MaxDiff RL agent performance, as well as equivalent performance to NN-MPPI in multi-shot (Supplementary Fig. 9), which we expect as a result of our robustness guarantees no longer holding.

Discussion

Throughout this work, we have highlighted the ways in which RL is fragile to temporal correlations intrinsic to many sequential decision-making processes. We introduced a framework based on the statistical mechanics of ergodic processes to overcome these limitations, which we term MaxDiff RL. Our framework offers a generalization of the current state of the art in RL and addresses many foundational issues holding back the field: the ergodicity of MaxDiff RL agents enables data acquisition that is indistinguishable from i.i.d. sampling, performance that is robust to seeds and single-shot learning. Through its roots in statistical physics, our work forms a starting point for a more scientific study of embodied RL—one in which falsifiable predictions can be made about agent properties and their performance.

However, much more work at the nexus of physics, learning and control remains to be done in pursuit of this goal. For one, approaches grounded in statistical physics for tuning or annealing temperature-like parameters during learning will be necessary to achieve effective exploration without sacrificing agent performance[65]. Additionally, control techniques capable of enforcing ergodicity in the face of environmental

irreversibility are needed to guarantee desirable agent properties like robustness to random seeds in complex problem settings[45]. Beyond RL, our work also has the potential to open new lines of interdisciplinary enquiry in areas such as biological learning and animal behaviour. For example, the importance of ergodicity to animal behaviours like foraging and tracking has been extensively studied[66]. As such, our work presents an avenue for studying these behaviours within an RL framework that is sensitive to physical embodiment. For biological motor learning, our findings also suggest that controllability may be a promising frame of reference for studying motor skill acquisition[67]. More broadly, our work is particularly well-suited to applications in soft matter systems where the impact of correlations may in fact be impossible to avoid entirely[68]. Taken together, our results present a major advance towards transparently understanding and reliably synthesizing complex behaviour in embodied decision-making agents, which will be crucial to the long-term viability of deep RL as a field.

Methods

RL preliminaries

RL problems are modelled as MDPs. MDPs are typically defined according to a 5-tuple, $(\mathcal{X}, \mathcal{U}, p, r, \gamma)$, where we take both the state space, $\mathcal{X}$, and the action space, $\mathcal{U}$, to be continuous. Note that in this work, we typically take $\mathcal{X}$ to be some subset of $\mathbb{R}^d$. Then $p : \mathcal{X} \times \mathcal{X} \times \mathcal{U} \to [0, \infty)$ represents the probability density of transitioning from state $x_t \in \mathcal{X}$ to state $x_{t+1} \in \mathcal{X}$ after taking action $u_t \in \mathcal{U}$. At every state and for each action taken, the environment emits a bounded reward $r : \mathcal{X} \times \mathcal{U} \to [r_{\min}, r_{\max}]$ discounted by a factor of $\gamma \in [0, 1)$. In general, the goal is to learn an optimized policy $\pi : \mathcal{U} \times \mathcal{X} \to [0, \infty)$ capable of producing actions that maximize an agent's expected cumulative rewards over the course of T discrete time stages, where $t \in \{0, \cdots, T\}$. In standard RL, this optimization takes place over the course of ensembles of episodes (that is, task attempts) of duration T, where the environment is reset after each episode[69]. This is what we refer to as the multi-shot learning setting. In contrast, non-episodic RL considers reset-free learning over the course of a single task attempt in the limit of $T \to \infty$ or until the task is done[60,61]. We refer to this as the single-shot learning setting.

PAC-MDP framework

Many properties of MaxDiff RL agents arise from the relationship between ergodicity and learning performance. To formalize how this is the case, we use the PAC-MDP learning framework[55,56].

Definition 1. An algorithm $\mathcal{A}$ is said to be PAC-MDP if, for any $\epsilon > 0$ and $\delta \in (0, 1)$, a policy π can be produced with $\text{poly}(|\mathcal{X}|, |\mathcal{U}|, 1/\epsilon, 1/\delta, 1/(1-\gamma))$ sample complexity that is at least ϵ-optimal with probability at least $1 - \delta$. In other words, if $\mathcal{A}$ satisfies

$$\Pr\left(\mathcal{V}_{\pi^*}(x_0) - \mathcal{V}_{\pi}(x_0) \leq \epsilon\right) \geq 1 - \delta$$

with polynomial sample complexity for all $x_0 \in \mathcal{X}$, where $\mathcal{V}_{\pi}(\cdot)$ is a value function due to policy π and $\mathcal{V}_{\pi^*}(\cdot)$ is the optimal value function, then $\mathcal{A}$ is PAC-MDP.

Thus, an algorithm is PAC-MDP if it is capable of producing a policy that is at least ϵ-optimal at least $100 \times (1 - \delta)\%$ of the time for any valid choice of ϵ and δ.

Statistical analysis of empirical benchmarks

Because all learning experiments were run across ten seeds, for each task there are ten policies per method (that is, MaxDiff RL, NN-MPPI and SAC). Due to differences between multi-shot and single-shot settings, we evaluated them differently. In multi-shot experiments, we took the ten final policies learned and evaluated their performance across 100 episodes with randomized initial conditions. For each algorithm, this results in a total of 1,000 sampled returns per task. Then, to assess statistical differences between the sampled 1,000 episodic returns per algorithm, we used an unpaired two-sided Welch's t-test as implemented in Python's scientific computing package[70]. An important note is that episodic return curves illustrate the policies' learning progress across each of the ten random seeds, rather than policy evaluation. Policy evaluation is depicted in bar plots instead (for example, Fig. 3c, right).

The non-episodic nature of single-shot learning means that there is no individual time stamp at which policies can be fairly evaluated. For this reason, in single-shot experiments we used a task-specific performance measure (that is, distance travelled) sampled across the ten runs of each task to perform statistical comparisons. In addition to our task-specific metrics, we also took the terminal windowed returns sampled during each of the ten seeds of the learning tasks. As before, we applied a Welch's t-test onto the ten sampled returns and ten sampled task-specific metrics per algorithm. For statistics, we refer readers to Supplementary Table 2.

Data availability

Data supporting the findings of this study are available via Zenodo at https://doi.org/10.5281/zenodo.10723320 (ref. 71).

Code availability

Code supporting the findings of this study is available via Zenodo at https://doi.org/10.5281/zenodo.10723320 (ref. 71).

References

1. Degrave, J. et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature* **602**, 414–419 (2022).
2. Won, D.-O., Müller, K.-R. & Lee, S.-W. An adaptive deep reinforcement learning framework enables curling robots with human-like performance in real-world conditions. *Sci. Robot.* **5**, eabb9764 (2020).
3. Irpan, A. Deep reinforcement learning doesn't work yet. *Sorta Insightful* www.alexirpan.com/2018/02/14/rl-hard.html (2018).
4. Henderson, P. et al. Deep reinforcement learning that matters. In *Proc. 32nd AAAI Conference on Artificial Intelligence* (eds McIlraith, S. & Weinberger, K.) 3207–3214 (AAAI, 2018).
5. Ibarz, J. et al. How to train your robot with deep reinforcement learning: lessons we have learned. *Int. J. Rob. Res.* **40**, 698–721 (2021).
6. Lillicrap, T. P. et al. *Proc. 4th International Conference on Learning Representations* (ICLR, 2016).
7. Haarnoja, T., Zhou, A., Abbeel, P. & Levine, S. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proc. 35th International Conference on Machine Learning* (eds Dy, J. & Krause, A.) 1861–1870 (PMLR, 2018).
8. Plappert, M. et al. *Proc. 6th International Conference on Learning Representations* (ICLR, 2018).
9. Lin, L.-J. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Mach. Learn.* **8**, 293–321 (1992).
10. Schaul, T., Quan, J., Antonoglou, I. & Silver, D. *Proc. 4th International Conference on Learning Representations* (ICLR, 2016).
11. Andrychowicz, M. et al. Hindsight experience replay. In *Proc. Advances in Neural Information Processing Systems 30* (eds Guyon, I. et al.) 5049–5059 (Curran Associates, 2017).
12. Zhang, S. & Sutton, R. S. A deeper look at experience replay. Preprint at https://arxiv.org/abs/1712.01275 (2017).
13. Wang, Z. et al. *Proc. 5th International Conference on Learning Representations* (ICLR, 2017).
14. Hessel, M. et al. Rainbow: combining improvements in deep reinforcement learning. In *Proc. 32nd AAAI Conference on Artificial Intelligence* (eds McIlraith, S. and Weinberger, K.) 3215–3222 (AAAI Press, 2018).
15. Fedus, W. et al. Revisiting fundamentals of experience replay. In *Proc. 37th International Conference on Machine Learning* (eds Daumé III, H. & Singh, A.) 3061–3071 (JMLR.org, 2020).

16. Mnih, V. et al. Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
17. Ziebart, B. D., Maas, A. L., Bagnell, J. A. & Dey, A. K. Maximum entropy inverse reinforcement learning. In *Proc. 23rd AAAI Conference on Artificial Intelligence* (ed. Cohn, A.) 1433–1438 (AAAI, 2008).
18. Ziebart, B. D., Bagnell, J. A. & Dey, A. K. Modeling interaction via the principle of maximum causal entropy. In *Proc. 27th International Conference on Machine Learning* (eds Fürnkranz, J. & Joachims, T.) 1255–1262 (Omnipress, 2010).
19. Ziebart, B. D. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy.* PhD thesis, Carnegie Mellon Univ. (2010).
20. Todorov, E. Efficient computation of optimal actions. *Proc. Natl Acad. Sci. USA* **106**, 11478–11483 (2009).
21. Toussaint, M. Robot trajectory optimization using approximate inference. In *Proc. 26th International Conference on Machine Learning* (eds Bottou, L. & Littman, M.) 1049–1056 (ACM, 2009).
22. Rawlik, K., Toussaint, M. & Vijayakumar, S. On stochastic optimal control and reinforcement learning by approximate inference. In *Proc. Robotics: Science and Systems VIII* (eds Roy, N. et al.) 353–361 (MIT, 2012).
23. Levine, S. & Koltun, V. Guided policy search. In *Proc. 30th International Conference on Machine Learning* (eds Dasgupta, S. & McAllester, D.) 1–9 (JMLR.org, 2013).
24. Haarnoja, T., Tang, H., Abbeel, P. & Levine, S. Reinforcement learning with deep energy-based policies. In *Proc. 34th International Conference on Machine Learning* (eds Precup, D. & Teh, Y. W.) 1352–1361 (JMLR.org, 2017).
25. Haarnoja, T. et al. Learning to walk via deep reinforcement learning. In *Proc. Robotics: Science and Systems XV* (eds Bicchi, A. et al.) (RSS, 2019).
26. Eysenbach, B. & Levine, S. *Proc. 10th International Conference on Learning Representations* (ICLR, 2022).
27. Chen, M. et al. Top-K off-policy correction for a REINFORCE recommender system. In *Proc. 12th ACM International Conference on Web Search and Data Mining* (eds Bennett, P. N. & Lerman, K.) 456–464 (ACM, 2019).
28. Afsar, M. M., Crump, T. & Far, B. Reinforcement learning based recommender systems: a survey. *ACM Comput. Surv.* **55**, 1–38 (2022).
29. Chen, X., Yao, L., McAuley, J., Zhou, G. & Wang, X. Deep reinforcement learning in recommender systems: a survey and new perspectives. *Knowl. Based Syst.* **264**, 110335 (2023).
30. Sontag, E. D. *Mathematical Control Theory: Deterministic Finite Dimensional Systems* (Springer, 2013).
31. Hespanha, J. P. *Linear Systems Theory* 2nd edn (Princeton Univ. Press, 2018).
32. Mitra, D. W-matrix and the geometry of model equivalence and reduction. *Proc. Inst. Electr. Eng.* **116**, 1101–1106 (1969).
33. Dean, S., Mania, H., Matni, N., Recht, B. & Tu, S. On the sample complexity of the linear quadratic regulator. *Found. Comput. Math.* **20**, 633–679 (2020).
34. Tsiamis, A. & Pappas, G. J. Linear systems can be hard to learn. In *Proc. 60th IEEE Conference on Decision and Control* (ed. Prandini, M.) 2903–2910 (IEEE, 2021).
35. Tsiamis, A., Ziemann, I. M., Morari, M., Matni, N. & Pappas, G. J. Learning to control linear systems can be hard. In *Proc. 35th Conference on Learning Theory* (eds Loh, P.-L. & Raginsky, M.) 3820–3857 (PMLR, 2022).
36. Williams, G. et al. Information theoretic MPC for model-based reinforcement learning. In *Proc. IEEE International Conference on Robotics and Automation* (ed. Nakamura, Y.) 1714–1721 (IEEE, 2017).
37. So, O., Wang, Z. & Theodorou, E. A. Maximum entropy differential dynamic programming. In *Proc. IEEE International Conference on Robotics and Automation* (ed. Kress-Gazit, H.) 3422–3428 (IEEE, 2022).
38. Thrun, S. B. *Efficient Exploration in Reinforcement Learning*. Technical report (Carnegie Mellon Univ., 1992).
39. Amin, S., Gomrokchi, M., Satija, H., van Hoof, H. & Precup, D. A survey of exploration methods in reinforcement learning. Preprint at https://arXiv.org/2109.00157 (2021).
40. Jaynes, E. T. Information theory and statistical mechanics. *Phys. Rev.* **106**, 620–630 (1957).
41. Dixit, P. D. et al. Perspective: maximum caliber is a general variational principle for dynamical systems. *J. Chem. Phys.* **148**, 010901 (2018).
42. Chvykov, P. et al. Low rattling: a predictive principle for self-organization in active collectives. *Science* **371**, 90–95 (2021).
43. Kapur, J. N. *Maximum Entropy Models in Science and Engineering* (Wiley, 1989).
44. Moore, C. C. Ergodic theorem, ergodic theory, and statistical mechanics. *Proc. Natl Acad. Sci. USA* **112**, 1907–1911 (2015).
45. Taylor, A. T., Berrueta, T. A. & Murphey, T. D. Active learning in robotics: a review of control principles. *Mechatronics* **77**, 102576 (2021).
46. Seo, Y. et al. State entropy maximization with random encoders for efficient exploration. In *Proc. 38th International Conference on Machine Learning, Virtual* (eds Meila, M. & Zhang, T.) 9443–9454 (ICML, 2021).
47. Prabhakar, A. & Murphey, T. Mechanical intelligence for learning embodied sensor-object relationships. *Nat. Commun.* **13**, 4108 (2022).
48. Chentanez, N., Barto, A. & Singh, S. Intrinsically motivated reinforcement learning. In *Proc. Advances in Neural Information Processing Systems 17* (eds Saul, L. et al.) 1281–1288 (MIT, 2004).
49. Pathak, D., Agrawal, P., Efros, A. A. & Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *Proc. 34th International Conference on Machine Learning* (eds Precup, D. & Teh, Y. W.) 2778–2787 (JLMR.org, 2017).
50. Taiga, A. A., Fedus, W., Machado, M. C., Courville, A. & Bellemare, M. G. *Proc. 8th International Conference on Learning Representations* (ICLR, 2020).
51. Wang, X., Deng, W. & Chen, Y. Ergodic properties of heterogeneous diffusion processes in a potential well. *J. Chem. Phys.* **150**, 164121 (2019).
52. Palmer, R. G. Broken ergodicity. *Adv. Phys.* **31**, 669–735 (1982).
53. Islam, R., Henderson, P., Gomrokchi, M. & Precup, D. Reproducibility of benchmarked deep reinforcement learning tasks for continuous control. Preprint at https://arXiv.org/1708.04133 (2017).
54. Moos, J. et al. Robust reinforcement learning: a review of foundations and recent advances. *Mach. Learn. Knowl. Extr.* **4**, 276–315 (2022).
55. Strehl, A. L., Li, L., Wiewiora, E., Langford, J. & Littman, M. L. PAC model-free reinforcement learning. In *Proc. 23rd International Conference on Machine Learning* (eds Cohen, W. W. & Moore, A.) 881–888 (ICML, 2006).
56. Strehl, A. L., Li, L. & Littman, M. L. Reinforcement learning in finite MDPs: PAC analysis. *J. Mach. Learn. Res.* **10**, 2413–2444 (2009).
57. Kirk, R., Zhang, A., Grefenstette, E. & Rocktäaschel, T. A survey of zero-shot generalisation in deep reinforcement learning. *J. Artif. Intell. Res.* **76**, 201–264 (2023).
58. Oh, J., Singh, S., Lee, H. & Kohli, P. Zero-shot task generalization with multi-task deep reinforcement learning. In *Proc. 34th International Conference on Machine Learning* (eds Precup, D. & Teh, Y. W.) 2661–2670 (JLMR.org, 2017).
59. Krakauer, J. W., Hadjiosif, A. M., Xu, J., Wong, A. L. & Haith, A. M. Motor learning. *Compr. Physiol.* **9**, 613–663 (2019).

60. Lu, K., Grover, A., Abbeel, P. & Mordatch, I. *Proc. 9th International Conference on Learning Representations* (ICLR, 2021).
61. Chen, A., Sharma, A., Levine, S. & Finn, C. You only live once: single-life reinforcement learning. In *Proc. Advances in Neural Information Processing Systems 35* (eds Koyejo, S. et al.) 14784–14797 (NeurIPS, 2022).
62. Ames, A., Grizzle, J. & Tabuada, P. Control barrier function based quadratic programs with application to adaptive cruise control. In *Proc. 53rd IEEE Conference on Decision and Control* 6271–6278 (IEEE, 2014).
63. Taylor, A., Singletary, A., Yue, Y. & Ames, A. Learning for safety-critical control with control barrier functions. In *Proc. 2nd Conference on Learning for Dynamics and Control* (eds Bayen, A. et al.) 708–717 (PLMR, 2020).
64. Xiao, W. et al. BarrierNet: differentiable control barrier functions for learning of safe robot control. *IEEE Trans. Robot.* **39**, 2289–2307 (2023).
65. Seung, H. S., Sompolinsky, H. & Tishby, N. Statistical mechanics of learning from examples. *Phys. Rev. A* **45**, 6056–6091 (1992).
66. Chen, C., Murphey, T. D. & MacIver, M. A. Tuning movement for sensing in an uncertain world. *eLife* **9**, e52371 (2020).
67. Song, S. et al. Deep reinforcement learning for modeling human locomotion control in neuromechanical simulation. *J. Neuroeng. Rehabil.* **18**, 126 (2021).
68. Berrueta, T. A., Murphey, T. D. & Truby, R. L. Materializing autonomy in soft robots across scales. *Adv. Intell. Syst.* **6**, 2300111 (2024).
69. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction* (MIT, 2018).
70. Virtanen, P. et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272 (2020).
71. Berrueta, T. A., Pinosky, A. & Murphey, T. D. Maximum diffusion reinforcement learning repository. *Zenodo* https://doi.org/10.5281/zenodo.10723320 (2024).

Acknowledgements

We thank A. T. Taylor, J. Weber and P. Chvykov for their comments on early drafts of this work. We acknowledge funding from the US Army Research Office MURI grant no. W911NF-19-1-0233 and the US Office of Naval Research grant no. N00014-21-1-2706. We also acknowledge hardware loans and technical support from Intel Corporation, and T.A.B. is partially supported by the Northwestern University Presidential Fellowship.

Author contributions

T.A.B. derived all theoretical results, performed supplementary data analyses and control experiments, supported RL experiments and wrote the manuscript. A.P. developed and tested RL algorithms, carried out all RL experiments and supported manuscript writing. T.D.M. secured funding and guided the research programme.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-024-00829-3.

Correspondence and requests for materials should be addressed to Thomas A. Berrueta or Todd D. Murphey.

Peer review information *Nature Machine Intelligence* thanks the anonymous reviewers for their contribution to the peer review of this work.

nature machine intelligence

Article

https://doi.org/10.1038/s42256-024-00829-3

Maximum diffusion reinforcement learning

In the format provided by the authors and unedited

Supplementary information

Contents

Correspondence: tberrueta@u.northwestern.edu, and t-murphey@northwestern.edu.

Supplementary notes

1 Introduction

In the following supplementary notes, we lay out the theoretical framework of maximum diffusion reinforcement learning (MaxDiff RL). MaxDiff RL is a generalization of maximum entropy (MaxEnt) RL in a similar sense as the principle of maximum caliber [1] is a generalization of the principle of maximum entropy [2]. This requires a deliberate shift in the way we interpret the underlying goal of RL algorithms: from reaching desirable states to realizing desirable trajectories. By assigning conceptual importance to the "state trajectory" as a mathematical abstraction, our approach has an explicit focus on the way that properties of the underlying agent-environment state transition dynamics impact the performance of RL algorithms. In particular, we consider the impact that temporal correlations in the trajectories of RL agents can have on their performance and design MaxDiff RL to overcome this impact.

The broad structure of the supplement is the following: first, in Supplementary Note 2, we derive a novel understanding of exploration through the lens of maximum caliber trajectory sampling. In doing so, we are able to derive analytical expressions that describe the trajectories of optimally exploring agents in settings where there is no goal or reward, as well as in settings where there is one. Then, in Supplementary Note 3, we provide a mathematical framework for synthesizing agent behavior that satisfies optimal exploration statistics, which we show to be formally equivalent to the usual stochastic optimal control formulation of RL problems. Finally, Supplementary Note 4 provides implementation details and statistical analyses of our empirical results. We will now provide a per-section summary of the results provided in Supplementary Notes 2 and 3.

Supplementary Note 2 establishes the theoretical foundations of our approach. First, Supplementary Note 2.1 motivates our primary conceptual point in a restricted class of systems—that temporal correlations in the state transition dynamics of embodied agents can have an impact on effective exploration and learning performance. Then, Supplementary Note 2.2 establishes some mathematical preliminaries, such as how to think of an agent's experiences or state trajectories as collections of random variables parametrized by a time-like variable, and how to measure temporal correlations. Supplementary Note 2.3 formalizes the problem of undirected state exploration through the lens of maximum caliber trajectory sampling. In doing so, we pay particular attention to realizing exploration with continuous trajectories. In Supplementary Note 2.4, we prove that optimal exploration with continuous trajectories is achieved by state-space diffusion (Theorem 2.1). Moreover, we prove that agents who satisfy optimal exploration statistics are Markovian (Corollary 2.1.1) and ergodic (Corollary 2.1.2). Notably, this is not something we assumed a priori. In Supplementary Note 2.5, we extend our results to directed exploration settings where there is a cost or reward function that assigns some notion of preference to particular states. While the Markov property holds in this setting automatically, we prove that optimal directed exploration is still ergodic (Theorem 2.2). Finally, Supplementary Note 2.6 provides additional motivation that illustrates the sense in which maximum caliber directed exploration leads to goal-directed behavior. In doing so, we analyze the maximum likelihood trajectories of optimally exploring path distribution and find that they have inertial dynamics resembling gradient descent.

Supplementary Note 3 establishes the computational foundations of our approach. First, in Supplementary Note 3.1 we define MaxDiff trajectory synthesis more broadly in terms of KL control. In short, we define the objective of MaxDiff trajectory synthesis as finding controllers or policies that minimize the distance between an agent's trajectory distribution and the optimal trajectory distributions (as derived in Supplementary Note 2). In Supplementary Note 3.2, we show that the KL control objective from the previous section can be written as an equivalent stochastic optimal control problem, which allows us to formally state the MaxDiff RL objective. Supplementary Note 3.3 explores the formal properties of MaxDiff RL: the sense in which it generalizes MaxEnt RL (Theorem 3.1 and Main Text Theorem 1), its single-shot learning capabilities (Theorem 3.2 and Main Text Theorem 3), and their robustness to seeds and initializations (Theorem 3.4 and Main Text Theorem 2). Supplementary Notes 3.4 and 3.5 introduce alternative formulations of the MaxDiff RL objective that are easier to compute, as well as more amenable to model-free RL implementations. Finally, Supplementary Note 3.6 provides some examples of MaxDiff trajectory synthesis outside of RL.

2 Theoretical framework for maximum diffusion

Throughout this section we analytically derive and establish the theoretical properties of maximally diffusive agents and their trajectories, as well as their relationship to *i.i.d.* data, temporal correlations, controllability, and exploration. We do not directly discuss reinforcement learning within this section beyond framing our results, but rather establish mathematical foundations that elucidate the relationship between an agent's properties and its ability to explore and learn. For our implementation of these principles within a reinforcement learning framework, refer to Supplementary Note 3.

2.1 The role of temporal correlations in exploration and learning

Exploration is a process by which agents become exposed to new experiences, which is of broad importance to their learning performance. While many learning systems can function as abstract processes insulated from the challenges and uncertainties associated with embodied operation [3], physical agents—simulated or otherwise—have no such luxury [4–7]. The laws of physics, material properties, and dynamics all impose fundamental constraints on what can be achieved by a learning system. The main conceptual point of this work is that the state transition dynamics of embodied learning agents can introduce temporal correlations that hinder their performance. In this section, we provide a formal mathematical argument in favor of this point in a particular class of systems. We do this in hopes of motivating how the controllability properties of the agent-environment state transition dynamics—and the temporal correlations these induce—may have an impact on the efficacy of action randomization as an exploration strategy more generally, and as a result on performance.

Drawing inspiration from the study of multi-armed bandits [8], the most common exploration strategy in reinforcement learning is randomized action exploration. The simplest of these methods merely requires that agents randomly sample actions from either uniform or Gaussian distributions to produce exploration. More sophisticated methods, such as maximum entropy reinforcement learning [9–11], elaborate on this basic idea by learning a distribution from which to sample random actions. For the purpose of our analysis, these more advanced methods are functionally equivalent to each other—they assume that taking random actions produces effective state exploration. However, from the perspective of control theory we know that this is not necessarily the case. For a system to be able to reach desired states arbitrarily, it must be controllable [12].

To illustrate how the controllability properties of the agent-environment state transition dynamics can determine the structure and magnitude of temporal correlations, we will briefly consider randomized action exploration in linear time-varying (LTV) control systems. This is a broad class of systems for which we can provide formal mathematical arguments in favor of our main point. LTV dynamics can be expressed in terms of continuous-time deterministic trajectories in the following way:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), \tag{1}$$

where $A(t)$ and $B(t)$ are appropriately dimensioned matrices with state and control vectors $x(t) \in \mathcal{X} \subset \mathbb{R}^d$ and $u(t) \in \mathcal{U} \subset \mathbb{R}^m$, and $x(t_0) = x^*$ for $\mathcal{T} = [t_0, t] \subset \mathbb{R}$. The general form of solutions to this system of linear differential equations is expressed in terms of a convolution with the system's state-transition matrix, $\Psi(t, t_0)$, in the following way:

$$x(t) = \Psi(t, t_0)x^* + \int_{t_0}^{t} \Psi(t, \tau)B(\tau)u(\tau)d\tau. \tag{2}$$

We consider these dynamics because by working with LTV dynamics we implicitly consider a very broad class of systems—all while retaining the simplicity of linear controllability analysis [13]. This is due to the fact that the dynamics of any nonlinear system that is locally linearizable along its trajectories can be effectively captured by LTV dynamics. Hence, any results applicable to the dynamics in Eq. 1 will apply to linearizable nonlinear systems. However, we note that our derivations in subsequent sections do *not* assume dynamics of this form. We only consider them to motivate our approach in this section.

To develop an understanding of the exploration capabilities of a given LTV system, we may ask what states are reachable by this system. After all, states that are not reachable cannot be explored or learned from. This is precisely what controllability characterizes:

Definition 2.1. *A system is said to be controllable over a time interval $[t_0, t] \subset \mathcal{T}$ if given any states $x^*, x_1 \in \mathcal{X}$, there exists a controller $u(t) : [t_0, t] \to \mathcal{U}$ that drives the system from state x^* at time t_0 to x_1 at time t.*

While this definition intuitively captures what is meant by controllability, it does not immediately seem like an easily verifiable property. To this end, different computable metrics have been developed that equivalently characterize the controllability properties of certain classes of systems (e.g., the Kalman controllability rank condition [14]). In particular, here we analyze the controllability Gramian of our system, as well as its rank and determinant as metrics on system controllability.

For our class of LTV systems, characterizing controllability with this method is simple:

$$W(t_0, t) = \int_{t_0}^{t} \Psi(t, \tau)B(\tau)B(\tau)^T\Psi(t, \tau)^T d\tau, \tag{3}$$

where the Gramian is a symmetric positive semidefinite matrix that depends on the state-control matrix $B(t)$ and the state-transition matrix $\Psi(t, t_0)$. The Gramian is a controllability metric that quantifies the amount of energy required to actuate the different degrees of freedom of the system [15, 16]. For any given finite time interval, the controllability Gramian also characterizes the set of states reachable by the system. Importantly, when the controllability Gramian is full-rank, the system is provably controllable in the sense of Definition 2.1 [12], and capable of fully exploring its environment. However, when the controllability Gramian is poorly conditioned, substantial temporal correlations are introduced into the agent's state transitions, which can prevent effective exploration and—as a direct consequence—learning, as we will show.

To draw the connection between naive random exploration, controllability, and temporal correlations explicitly, we will now revisit the dynamics in Eq. 1 under a slight modification. Let us design a controller that performs naive action randomization, i.e., let $u(t) = \xi$, where $\xi \sim \mathcal{N}(\mathbf{0}, \mathrm{Id})$ and Id is an identity matrix with diagonal of the same dimension as the control inputs, and $\mathbf{0}$ is the zero vector of the same dimension. Note that the system trajectories are now random variables—or rather, collections of random variables, which we define formally in the following section. Then, we have:

$$\dot{x}(t) = A(t)x(t) + B(t) \cdot \xi. \tag{4}$$

Here, we abuse notation slightly to minimize the difference between this equation and Eq. 1, but we can interpret the system as having linear Langevin dynamics [17]. With these modifications in mind, we are now interested in examining the mean and covariance trajectory statistics in hopes of characterizing the structure of temporal correlations induced by the agent dynamics. We begin by taking the expectation over system trajectories described by Eq. 2:

$$\begin{aligned} E[x(t)] &= E\Big[\Psi(t, t_0)x^* + \int_{t_0}^{t} \Psi(t, \tau)B(\tau) \cdot \xi d\tau\Big] \\ &= \Psi(t, t_0)x^* + E\Big[\int_{t_0}^{t} \Psi(t, \tau)B(\tau) \cdot \xi d\tau\Big] \\ &= \Psi(t, t_0)x^*. \end{aligned} \tag{5}$$

Hence, the expected sample paths of the dynamics will be centered around the autonomous paths of the system—that is, the paths the system takes in the absence of control inputs. We may now characterize the covariance of our system's sample paths. To do so, let $\mathbf{C}[x^*] = E\big[(x(t) - E[x(t)])(x(t) - E[x(t)])^T \big| x(t_0) = x^*\big]$ be the trajectory autocovariance. Although we will formalize this idea in the following section, for now we note that the trajectory-wise expectation is taken as the time-integration of point-wise autocovariances. With these preliminaries taken care of, we have:

$$\begin{aligned} \mathbf{C}[x^*] &= E\big[(x(t) - E[x(t)])(x(t) - E[x(t)])^T \big| x(t_0) = x^*\big] \\ &= E\Big[\Big(\Psi(t, t_0)x^* + \int_{t_0}^{t} \Psi(t, \tau)B(\tau) \cdot \xi d\tau - E[x(t)]\Big)\Big(\Psi(t, t_0)x^* + \int_{t_0}^{t} \Psi(t, \tau)B(\tau) \cdot \xi d\tau - E[x(t)]\Big)^T\Big] \\ &= E\Big[\Big(\int_{t_0}^{t} \Psi(t, \tau)B(\tau) \cdot \xi d\tau\Big)\Big(\int_{t_0}^{t} \Psi(t, \tau)B(\tau) \cdot \xi d\tau\Big)^T\Big] \\ &= E\Big[\int_{t_0}^{t} \Psi(t, \tau)B(\tau) \cdot (\xi\xi^T) \cdot B(\tau)^T\Psi(t, \tau)^T d\tau\Big] \\ &= \int_{t_0}^{t} \Psi(t, \tau)B(\tau)B(\tau)^T\Psi(t, \tau)^T d\tau. \end{aligned} \tag{6}$$

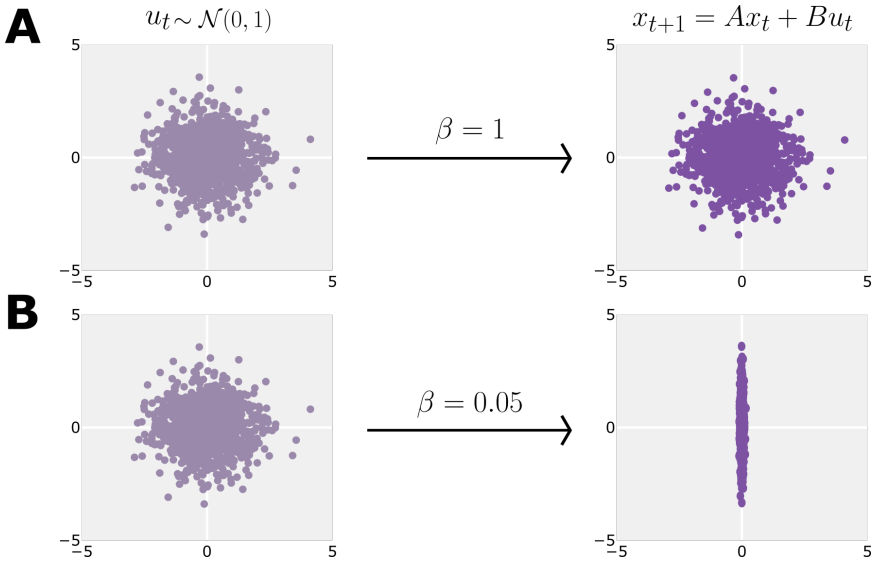
By inspection of the above expression and Eq. 3, we arrive at the following important connection:

$$\mathbf{C}[x^*] = W(t_0, t) \tag{7}$$

which tells us that for LTV dynamics (and by extension for linearizable nonlinear dynamics), a measure of temporal correlations—the trajectory autocovariance $\mathbf{C}[x^*]$ (see Supplementary Note 2.2)—is exactly equivalent to the controllability Gramian of the system. Thus, for a broad class of systems, an agent's controllability properties are given by a measure of temporal correlations along their state trajectories. Moreover, in LTV systems these are not state-dependent properties. In other words,

$$\nabla_x \mathbf{C}[x^*] = \nabla_x W(t_0, t) = \mathbf{0}, \tag{8}$$

where $\mathbf{0}$ is an appropriately dimensioned zero matrix. However, for linearizable nonlinear systems, as well as more general nonlinear systems, these properties will be state-dependent. While our controllability analysis has been restricted to the class of dynamics describable by linear differential equations with time-varying parameters, we note that the connections we observe between trajectory autocovariance and controllability Gramians have been shown to hold for even more general classes of



Supplementary Figure 1: **Effect of controllability on the distribution of reachable states**. **a,** For a linear system with dynamics like those in Figure 1 of the main text initialized with an x_t of all zeroes, we depict the effect of controllability on a naive random action exploration strategy. For a linear system with ideal controllabilty properties, isotropic distributions of actions map onto isotropic distributions of states. **b,** However, when the system is poorly conditioned the system dynamics distort the isotropy of the original input distribution, introducing temporal correlations induced by the controllability properties of the system, and fundamentally changing its properties as an exploration strategy.

nonlinear systems through more involved analyses [18]. Nonetheless, the results of our manuscript hold regardless of whether there is a formal and easily characterizable relationship between controllability and temporal correlations.

From Eq. 4 we can describe the system's reachable states by analyzing its state probability density function, which can be found analytically by solving its associated Fokker-Planck equation [19]. To do this, we only require the mean and covariance statistics of the process, in Eqs. 5 and 6. Hence, the system's time-dependent state distribution is

$$p(x,t,t_0) = \frac{1}{\sqrt{(2\pi)^d \det[W(t_0,t)]}} \exp\Big[-\frac{1}{2}\big(x - \Psi(t,t_0)x(t_0)\big)^T W^{-1}(t_0,t)\big(x - \Psi(t,t_0)x(t_0)\big)\Big] \tag{9}$$

for some choice of initial conditions at t_0, where we have substituted Eq. 7 to highlight the role of controllability in the probability density of states reachable by the system through naive random exploration. Thus, how easy or hard it is to explore in a given direction (as characterized by Eq. 9) is entirely determined by the controllability properties of the system—or, equivalently, by a measure of temporal correlations of its state trajectories. Supplementary Fig. 1 illustrates this concept for the toy dynamical system introduced in the main text. We observe that changes in β have an effect on the distribution of reachable states for the system that are consistent with Eq. 9, where we note we recentered the distribution mean.

On the basis of these results, which have been known for decades [20], we can clearly see that controllability and temporal correlations play a key role in exploration and data acquisition. We cannot assume that random inputs are capable of producing effective exploration of system states without an understanding of its controllability. For example, if $W(t_0,t)$ is not full-rank, then exploration would be restricted to a linear subspace of an agent's exploration domain. This amounts to a complete collapse of the *i.i.d.* assumption on the experiences of an agent, because its state transitions become deterministically correlated as a result of the degeneracy of Eq. 9. However, if we were to instead design $u(t)$ by exploiting knowledge about $\mathbf{C}[x^*]$, these limitations can be overcome. For example, if we let $u(t) = B(t)^T \mathbf{C}^{-1}[x^*]B(t) \cdot \xi$ instead, then better exploration can be achieved by can be reshaping $p(x,t,t_0)$ in a way that accounts for the system's temporal correlations. This is in fact the conceptual crux of our entire reinforcement learning framework, as we will show.

In more complex settings, where the input distribution is not Gaussian and the dynamics are strongly nonlinear, analyzing controllability may be more challenging. However, insofar as learning requires an embodied agent to either collect data or visit desirable states to optimize some objective, temporal correlations and controllability will continue to play an important role.

Remark 2.1. *Temporal correlations and controllability can determine whether it is possible and how challenging it is to learn.*

While one can construct proofs that illustrate this in a variety of simplified settings—as others have recently shown in the case of controllability [21, 22]—we leave the more general claim as a remark to frame the motivation behind our upcoming derivations. Hence, we should strive to develop exploration and learning strategies that reflect—and try to overcome—the effect of controllability and its induced temporal correlations, as we do in the following sections.

2.2 Exploration as trajectory sampling

In this section, we develop the mathematical formalism necessary for framing exploration in a controllability-aware manner that may allow us to overcome temporal correlations. While exploration with disembodied agents can be quite simple (e.g., sampling from a distribution, or performing a random walk), embodied agents must achieve exploration by changing the state of the environment through action. Our goal is to achieve state exploration in an embodied system, such as a robotic agent or otherwise, where their embodiment constrains the ways they can explore the states of an environment. While this motivation is most natural for physical systems, our framing is relevant to any setting in which the underlying agent-environment dynamics obey some notion of continuity of experience. To this end, we will need to define a formal notion of control system from which we can begin to model the experiences of agents.

First, we formally define stochastic processes by adapting the definition provided in [23] to our use case.

Definition 2.2. *A stochastic process is a family of random variables parametrized by a totally ordered indexing set $\mathcal{T}$,*

$$\{X_t\}_{t\in\mathcal{T}} \text{ when } \mathcal{T} \text{ is discrete, or } \{X(t)\}_{t\in\mathcal{T}} \text{ when } \mathcal{T} \text{ is continuous,}$$

defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. We take the sample space Ω to be measurable, $\mathcal{F}$ to be a Borel σ-algebra, and $\mathbb{P}$ to be a probability measure. We note that the random variables assume values in a compact state space $\mathcal{X} \subset \mathbb{R}^d$, and that each sample path takes value in a measurable space $\mathcal{X}^{\mathcal{T}}$ with Borel σ-algebra $\mathcal{B}(\mathcal{X}^{\mathcal{T}})$.

Thus, stochastic processes are families of random variables indexed according to some "time-like" set, $\mathcal{T}$. For each $\omega \in \Omega$, the sample paths of the stochastic process, $x_{\mathcal{T}}(\omega) = \{X(t,\omega)\}_{t\in\mathcal{T}}$, take value in $\mathcal{X}^{\mathcal{T}}$. We note that we often take $\mathcal{T}$ to be an interval, e.g., $[t_0, t]$ or a halfline. When $\mathcal{T}$ is discrete, e.g., $\{1, \cdots, N\}$, we have $x_{1:N}(\omega) = \{X_t(\omega)\}_{t\in\{1,\cdots,N\}}$ instead. Then, we can define the pushforward measure of $x_{\mathcal{T}} : \Omega \to \mathcal{X}^{\mathcal{T}}$ in the usual way. That is, $P_F : \mathcal{B}(\mathcal{X}^{\mathcal{T}}) \to [0,1]$ is given by $P_F[x_{\mathcal{T}} \in A] = \mathbb{P}(x_{\mathcal{T}}^{-1}(A))$ for some $A \subset \mathcal{X}^{\mathcal{T}}$. Finally, for a each ω, we use $x(t) = x_{\mathcal{T}}(\omega) \in \mathcal{X}^{\mathcal{T}}$ to denote individual realizations of the stochastic process, and refer to $x(t)$ as an agent's experiences, *state trajectories*, or *paths*. To describe the likelihoods of individual state trajectories, we assume that the probability density function associated with the pushforward measure exists and is given by $P : \mathcal{X}^{\mathcal{T}} \to [0, \infty)$, such that

$$P_F[x_{\mathcal{T}} \in A] = \mathbb{P}(x_{\mathcal{T}}^{-1}(A)) = \int_{x_{\mathcal{T}}^{-1}(A)} d\mathbb{P}(\omega) = \int_A P[x(t)]\mathcal{D}x(t) \tag{10}$$

where $\mathcal{D}x(t)$ denotes integration over sample paths, as in the Feynman path integral formalism [24]. Thus, we will refer to this density over paths as the *path* or *trajectory distribution*, and use $P[x(t)]$ to express the probability density of a given state trajectory of the stochastic process $x(t) \in \mathcal{X}^{\mathcal{T}}$. Alternatively, we use $P[x_{1:N}]$ when $\mathcal{T}$ is discrete.

To quantify correlations along sample paths or state trajectories, we evaluate a local measure of temporal correlations, $\mathbf{C}[x^*]$, over particular time intervals of a given stochastic process, $[t_i, t_i + \Delta t] \subset \mathcal{T}$. If $\{X(t)\}_{t\in\mathcal{T}}$ is a stochastic process defined according to Definition 2.2, then an autocovariance function, $K_{XX}(t_1, t_2)$, expresses the covariance of the process with itself at any two points in time $t_1, t_2 \in \mathcal{T}$, or

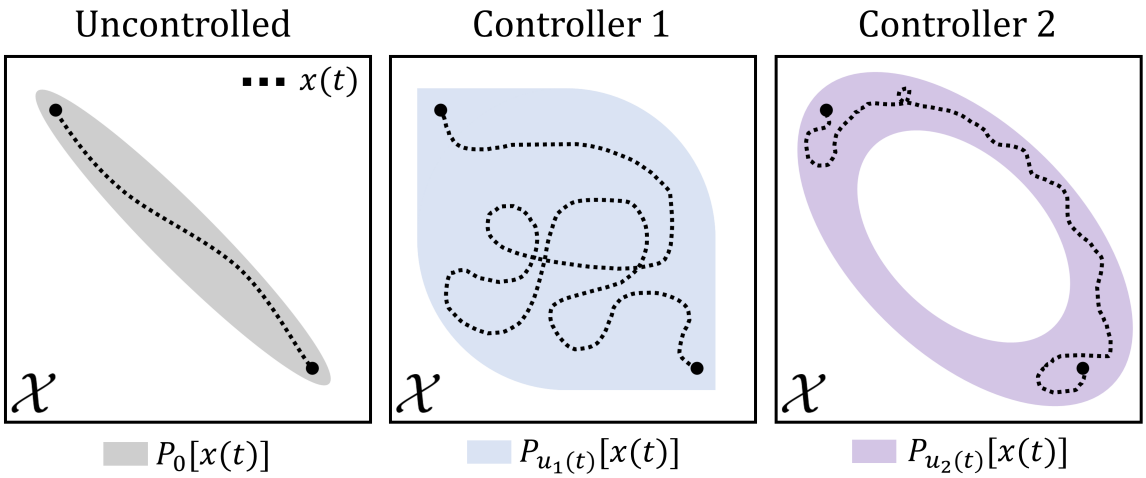
$$K_{XX}(t_1, t_2) = E\big[(X(t_1) - E[X(t_1)])(X(t_2) - E[X(t_2)])^T\big]. \tag{11}$$

With these preliminaries taken care of, we define our measure of temporal correlations with respect to an initial condition $X(t_i) = x^*$ for some $x^* \in \mathcal{X}$ in the following way:

$$\begin{aligned}\mathbf{C}[x^*] &= E\big[(X(t) - E[X(t)])(X(t) - E[X(t)])^T \big| X(t_i) = x^*\big] \\ &= \int_{t_i}^{t_i+\Delta t} K_{XX}(t_i, \tau)d\tau.\end{aligned} \tag{12}$$

Thus, our measure of temporal correlations $\mathbf{C}[x^*]$ could also be characterized as an integrated autocovariance function along the state trajectories of a stochastic process. This usage of the term "temporal correlations" is in line with its broad usage in statistical mechanics (see Ch. 10 of [25]), and we note that in practice one can divide Eq. 12 by Δt to prevent numerical estimates from strongly depending on the duration of the time-interval under consideration.

Lastly, it is important to note that the probability densities over state trajectories are strongly dependent on the dynamics that govern the agent-environment's time-evolution through state space. However, when the dynamics are nonautonomous, as is the case in control systems, this distribution will also depend on the choice of controller and the effect it has on the state transitions of the process. We define a controller as a function, $u(t) : \mathcal{T} \to \mathcal{U}$, that produces an input to the system dynamics at every point in the index set, where $\mathcal{U}$ is usually a subset of $\mathbb{R}^m$. At this point, we are not considering the system dynamics themselves, how controllers are synthesized, or how much influence either of these can have in shaping the sample paths of the underlying control system. All we care about is acknowledging the fact that a choice of controller induces a different probability density over sample paths. With these definitions we can now establish our notion of control system, or stochastic control process.



Supplementary Figure 2: **Effect of controllers on the sample path distribution of stochastic control processes**. (left) Sample path and support of the probability density over the paths of an autonomous stochastic process (i.e., with null controller "0"). (middle and right) Sample paths and distributions induced by two distinct controllers $u_1(t)$ and $u_2(t)$. Here, we illustrate that depending on the nature of the controller the distribution over sample paths can be nontrivial. Note that we do not illustrate the values of the probability densities, only their support. The reason for this is that so long as a regions of space have non-zero probability they will be sampled asymptotically.

Definition 2.3. *A stochastic control process is a stochastic process (Definition 2.2) on a probability space* $(\Omega, \mathcal{F}, \mathbb{P}_{u(t)})$*, with indexing set* $\mathcal{T}$*, where sample paths take value in a measurable space* $(\mathcal{X}^{\mathcal{T}}, \mathcal{B}(\mathcal{X}^{\mathcal{T}}))$*, and the resulting density* $P_{u(t)} : \mathcal{X}^{\mathcal{T}} \to [0, \infty)$ *is parametrized by a controller* $u(t) : \mathcal{T} \to \mathcal{U}$*.*

Thus, we think of control systems as stochastic processes that are parametrized by their controllers, or equivalently as a collection of distinct stochastic processes for each choice of controller.

In a stochastic control process the controller plays an important role in structuring the sample paths of a system—clearly, the sample path distribution of a robot with a controller that resists all movements is very different than one with a controller that encourages the robot to explore (see Supplementary Fig. 2 for an illustration). Hence, controllers determine which regions of the state space the system is capable of sampling from. With this in mind, we can express the problem of exploration in control systems: to design a controller that maximizes the regions of the exploration domain from which we can sample trajectories. In part, this requires the use of control actions in order to maximize the support of the agent's sample path distribution. The support of a probability distribution is the subset of all elements in its domain with greater than zero probability. However, merely maximizing the sample path distribution's support is not enough to realize effective exploration in most settings. For directed state exploration, we would ideally also like to control *how* probability mass is spread around the state space—if a given task demands that the agent's sample paths are biased towards a given goal, then our agent's path distribution should reflect this. In the following sections, we will work towards this goal of deriving path distributions for optimal undirected and directed exploration strategies.

2.3 Undirected exploration as variational optimization

One way of simultaneously controlling the spread of probability mass and the support of a probability distribution is to optimize its entropy [26]. For now, we consider the undirected exploration case, in which no task or objective biases the underlying agent's path distribution. As we will see in Supplementary Note 2.5, this approach will also enable us to control the spread of probability mass in a more fine-grained manner in order to realize directed exploration with respect to an objective or task—and eventually to do reinforcement learning.

Optimizing the entropy of an agent's path distribution through control synthesis can have a profound effect on the resulting behavior of the agent. This can be understood intuitively when there are no constraints on how we can increase the entropy of a sample path distribution. In this case, the maximum entropy distribution would be uniform over the entirety of the system's compact state space, leading to complete asymptotic exploration of the domain in a way that is equivalent to *i.i.d.* uniform sampling. However, a process realizing the statistics described by such a path distribution would require teleportation—that is, that points in space be visited uniformly at random at every moment in time. While this may pose no problems for disembodied agents with unconstrained dynamics, this creates issues for any agent whose experiences are constrained by their embodiment or otherwise. For example, in physical control systems subject to the laws of physics, this is infeasible

behavior. Hence, throughout the rest of this section we will take on the work of deriving the maximum entropy distribution for describing the state trajectories of agents with continuous experiences—a broad class of systems that includes all physical systems and many non-physical systems—as well as analyzing the formal properties of agents whose experiences satisfy such statistics. By maximizing trajectory entropy, this distribution will capture the statistics of an agent with minimally-correlated experiences. The analytical form of this distribution is crucial to the control and policy synthesis approach we derive in Supplementary Note 3. However, we note that our results will also apply for disembodied agents with discontinuous paths when we consider the uniform distribution as the optimal distribution instead of the one we derive in this section.

We proceed by identifying the analytical form of the maximum entropy path distribution with no consideration given to the problem of generating actions that achieve such statistics. Hence, we begin by framing our exploration problem in the maximum caliber formalism of statistical mechanics [1, 27, 28]. Maximum caliber is a generalization of the principle of maximum entropy to function spaces, such as distributions over trajectories or sample paths. In order to apply the principle of maximum caliber within our stochastic process formalism, we first note that we interpret path integrals in the following way. Consider some real-valued function $f(\cdot)$ of $x_{\mathcal{T}}$, then we define its expectation over sample paths as

$$E[f(x_{\mathcal{T}})] = \int_{\Omega} f(x_{\mathcal{T}}(\omega)) d\mathbb{P}(\omega) = \int_{\mathcal{X}^{\mathcal{T}}} P[x(t)] f(x(t)) \mathcal{D}x(t), \tag{13}$$

which is consistent with our definition of probability densities over state trajectories, $P[x(t)]$. Thus, path integrals integrate over the state trajectories of a stochastic process. Now, we are interested in finding a distribution which maximizes the entropy of sample paths, $S[P[x(t)]]$. Because we are looking for the unique analytical form of this distribution, we omit the controller-specific notation that was previously introduced—at least until we consider control and policy synthesis in the context of stochastic optimal control and reinforcement learning in Supplementary Note 3. The general form of the maximum caliber variational optimization is then the following:

$$\underset{P[x(t)]}{\operatorname{argmax}} \; -\int_{\mathcal{X}^{\mathcal{T}}} P[x(t)] \log P[x(t)] \mathcal{D}x(t) \tag{14}$$

However, as written the optimization is ill-posed and leads to a trivial solution. We can see this by taking the variation with respect to the sample path distribution, where we would find that the optimal sample path distribution is uniform, yet not a valid probability density as it is unnormalized. Thus, we need to constrain the optimization problem so that we only consider behavior realizable by the class of agents we are interested in modeling.

Since we are interested in framing our exploration problem for application domains like optimal control and reinforcement learning, we tailor our modeling assumptions to these settings. What sorts of principled constraints could be applied? No constraints based on conservation of energy are applicable because autonomous systems are inherently nonequilibrium systems. Nonetheless, the behavior of many autonomous systems (especially physically embodied ones) is constrained by other aspects of their morphology, such as actuation limits and continuity of movement. In particular, the rates at which agent experiences or states can vary—and *co-vary*—in time are typically bounded, which prevents them from discontinuously jumping between states by limiting their local rate of exploration. In fact, this is precisely what we found in Supplementary Note 2.1, where we saw that a system's ability to explore is closely tied to a measure of its temporal correlations, $\mathbf{C}[x^*]$, as defined in the previous section. Thus, we will choose to constrain the velocity fluctuations of our stochastic process so that they are finite and consistent with the integrated autocovariance statistics of the process, which may be determined empirically, and are related to a system's controllability properties in a broad class of systems. The use of an empirical (or learned) autocovariance estimate to quantify velocity fluctuations is important because different embodied agents have different limitations, which may additionally be spatially inhomogeneous and difficult to know a priori. Through this constraint, we can ensure that agent sample paths are continuous in time.

To formulate this path continuity constraint, we must first express the system's velocity fluctuations at each point in state space, $x^* \in \mathcal{X}$. We define the system's velocity fluctuations along sample paths $x(t)$ in the following way:

$$\langle \dot{x}(t) \dot{x}(t)^T \rangle_{x^*} = \int_{\mathcal{X}^{\mathcal{T}}} P[x(t)] \int_{\mathcal{T}} \dot{x}(\tau) \dot{x}(\tau)^T \delta(x(\tau) - x^*) d\tau \mathcal{D}x(t), \tag{15}$$

where $\delta(\cdot)$ denotes the Dirac delta function, and we note that the $\langle \cdot \rangle$ notation of statistical physics is equivalent to an expectation, i.e., $E[\cdot]$. We assume that the tensor described by Eq. 15 is full-rank so that the system's velocity fluctuations are not degenerate anywhere in the state space of the stochastic process. This assumption is crucial because it guarantees that our resulting path distribution is non-degenerate. If we had instead chosen to constrain the system by directly bounding the magnitude of its velocities, as opposed to its velocity fluctuations, we would not be able to guarantee the non-degeneracy of the resulting path distribution. Another important note is that the velocities of the trajectories of the stochastic process in this expression should be interpreted in the Langevin sense [17]. That is to say, not as expressions of the differentiability of

the sample paths of the underlying stochastic process, but as a shorthand for an integral representation of the stochastic differential equations describing the evolution of the sample paths of the system.

We can now express our constraint as,

$$\langle \dot{x}(t)\dot{x}(t)^T \rangle_{x^*} = \mathbf{C}[x^*], \quad \forall x^* \in \mathcal{X}. \tag{16}$$

Crucially, these statistics are bounded everywhere in the exploration domain, and we assume them to satisfy Lipschitz continuity so that their spatial variations are bounded. We note that linearizability of the underlying agent dynamics is a sufficient condition to satisfy this property. Hence, we now have equality constraints on the system's velocity fluctuations that can vary at each point in the exploration domain—as one would expect for a complex embodied system, such as a robot. As an additional constraint, we require that $P[x(t)]$ integrates to 1 so that it is a valid probability density over trajectories.

With expressions for each of our constraints, we may now express the complete variational optimization problem using Lagrange multipliers:

$$\underset{P[x(t)]}{\text{argmax}} \; -\int_{\mathcal{X}^\mathcal{T}} P[x(t)] \log P[x(t)] \mathcal{D}x(t) - \lambda_0 \Big(\int_{\mathcal{X}^\mathcal{T}} P[x(t)] \mathcal{D}x(t) - 1 \Big) - \int_{\mathcal{X}} Tr\Big(\Lambda(x^*)^T \big(\langle \dot{x}(t)\dot{x}(t)^T \rangle_{x^*} - \mathbf{C}[x^*] \big) \Big) dx^*. \tag{17}$$

Here, we express the constraints at all points x^* by taking an integral over all points in the domain. The λ_0 is a Lagrange multiplier enforcing our constraint that ensures valid probability densities, and $\Lambda(\cdot)$ is a matrix-valued Lagrange multiplier working to ensure that the rate of exploration constraints hold at every point in the domain. By solving this optimization we can obtain an expression for the maximum entropy distribution over sample paths. The solution to this problem will determine the distribution over sample paths with the greatest support, with the most uniformly spread probability mass, and with the least-correlated sample paths—thereby specifying the statistical properties of our optimal undirected exploration strategy, subject to a path continuity constraint.

2.4 Maximizing path entropy produces diffusion

In this section, we lay out the derivation of our solution to the variational optimization problem in Eq. 17. We begin by stating our main result in the following theorem.

Theorem 2.1. *The maximum caliber sample paths of a stochastic control process (Definition 2.3) with a maximum entropy exploration (in the sense of Eq. 17) are given by diffusion with spatially-varying coefficients.*

Proof. Letting $\mathcal{T} = [t_0, t]$, we begin by substituting Eq. 15 into Eq. 17, taking its variation with respect to the probability density $\delta S[P[x(t)]]/\delta P[x(t)]$, and setting it equal to 0:

$$\frac{\delta S}{\delta P[x(t)]} = -1 - \log P_{max}[x(t)] - \lambda_0 - \int_{\mathcal{X}} \int_{t_0}^{t} Tr\Big(\Lambda(x^*)^T (\dot{x}(\tau)\dot{x}(\tau)^T) \Big) \delta(x(\tau) - x^*) d\tau dx^* = 0.$$

Then, taking advantage of the following linear algebra identity, $a^T B a = Tr(B^T(aa^T))$, for any $a \in \mathbb{R}^m$ and $B \in \mathbb{R}^{m \times m}$; as well as the properties of the Dirac delta, we can simplify our expression to the following:

$$\frac{\delta S}{\delta P[x(t)]} = -1 - \log P_{max}[x(t)] - \lambda_0 - \int_{t_0}^{t} \dot{x}(\tau)^T \Lambda(x(\tau)) \dot{x}(\tau) d\tau = 0,$$

which allows us to solve for the maximum entropy probability distribution over the sample paths of our stochastic control process. The solution will then be of the form:

$$P_{max}[x(t)] = \frac{1}{Z} \exp\Big[-\int_{t_0}^{t} \dot{x}(\tau)^T \Lambda(x(\tau)) \dot{x}(\tau) d\tau \Big], \tag{18}$$

where we have subsumed the constant and Lagrange multiplier, λ_0, into a normalization factor, Z. We note that even without determining the form of our Lagrange multipliers, the maximum entropy probability density in Eq. 18 is already equivalent to the path probability of a diffusing particle with a (possibly anisotropic) spatially-inhomogeneous diffusion tensor (see [17], Ch. 9). While there is more work needed to characterize the diffusion tensor of this process, $\Lambda^{-1}(\cdot)$, this completes our proof. □

Thus, the least-correlated sample paths, which optimally sample from the exploration domain, are statistically equivalent to diffusion. This is to say that the distribution of paths with the greatest support over the state space describes the paths of a diffusion process. Hence, if the goal of some stochastic control process is to optimally explore and sample from its state space, the best strategy is to move randomly—that is, to decorrelate its sample paths. An additional benefit of our diffusive

exploration strategy is that we did not have to presuppose that our agent dynamics were Markovian or ergodic. Instead, we find that these properties emerge through our derivation as intrinsic properties of the optimal exploration strategy itself. The following corollaries of Theorem 2.1 follow from the connection to diffusion processes and Markov chains, and as such more general forms of these proofs may be found in textbooks on stochastic processes and ergodic theory. Here, we assume that the diffusion tensor in Eq. 18, $\Lambda^{-1}(\cdot)$, is full-rank and invertible everywhere in the state space. Additionally, for now we will assume that $\Lambda^{-1}(\cdot)$ is Lipschitz and bounded everywhere on $\mathcal{X}$. We will later find that these are not in fact different assumptions from those made in Eqs. 15 and 16.

Corollary 2.1.1. *The sample paths of a stochastic control process (Definition 2.3) with a maximum entropy exploration strategy (in the sense of Eq. 17) satisfy the Markov property.*

Proof. This follows trivially from the temporal discretization of our path distribution in Eq. 18, or alternatively from the properties of Langevin diffusion processes. Letting x_t be the initial condition, we can see that,

$$\begin{aligned} p_{max}(x_{t+\delta t}|x_t) &= \frac{1}{Z}\exp\Big[-\int_t^{t+\delta t}\dot{x}(\tau)^T\Lambda(x(\tau))\dot{x}(\tau)d\tau\Big] \\ &\approx \frac{1}{Z_d}\exp\Big[-|x_{t+\delta t}-x_t|^2_{\Lambda(x_t)}\Big], \end{aligned} \tag{19}$$

where we subsumed δt into a new normalization constant Z_d for convenience, and note that the support of $p_{max}(x_{t+\delta t}|x_t)$ is infinite. Importantly, our local Lagrange multiplier $\Lambda(x_t)$ enforces our velocity fluctuation constraint within a neighborhood of states reachable from x_t for a sufficiently small time interval δt, which is guaranteed by our Lipschitz continuity assumption. In what remains of this manuscript we use $\delta t = 1$ for notational convenience, but without loss of generality. Thus, our distribution in Eq. 19 depends only on the current state, which concludes our proof. □

Corollary 2.1.2. *A stochastic control process (Definition 2.3) in a compact and connected space $\mathcal{X} \subset \mathbb{R}^d$ with a maximum entropy exploration strategy (in the sense of Eq. 17) is ergodic.*

Proof. To prove the ergodicity of the process described by Eq. 18, we use Corollary 2.1.1 and the properties of $\mathcal{X}$. We begin by discretizing our optimal stochastic control process in time and space such that $P_{max}[x_{1:N}] = \prod_{t=1}^{N-1} p_{max}(x_{t+1}|x_t)$, which we can do without loss of generality as a result of Corollary 2.1.1 and because $\mathcal{X}$ is compact, resulting in a finite space. Importantly, since $p_{max}(x_{t+1}|x_t) > 0,\ \forall x_t, x_{t+1} \in \mathcal{X},\ \forall t \in \mathcal{T}$, and $\mathcal{X}$ is finite and connected, then all states in $\mathcal{X}$ communicate. Moreover, because for all $x^* \in \mathcal{X}$, $p_{max}(x^*|x^*) > 0$, the underlying Markov chain described by the transition kernel is aperiodic. Therefore, the Markov chain describing the stochastic control process is ergodic [29]. □

To finish our derivation and fully characterize the nature of our maximum entropy exploration strategy, we must return to Eq. 18 and determine the form of the matrix-valued Lagrange multiplier $\Lambda(\cdot)$. Hence, we will return to our expression for $\langle\dot{x}(t)\dot{x}(t)^T\rangle_{x^*}$ in Eq. 15 and discretize our continuous sample paths, which we can do without loss of generality due to Corollary 2.1.1. Since Eq. 15 represents a proportionality, we take out many constant factors throughout the derivation. Additionally, any constant factor of $\Lambda(\cdot)$ would be taken care of by the normalization constant Z in the final expression for Eq. 18. We proceed by discretizing Eq. 15, using i and j as time indices and $p_{max}(\cdot|\cdot)$ as the conditional probability density defined in Eq. 19. We do this by slicing the time interval $[t_0, t]$ into time indices $\{1, \cdots, N\}$. Our resulting expression is the following:

$$\langle\dot{x}(t)\dot{x}(t)^T\rangle_{x^*} = \prod_{i=1}^{N-1}\Big[\int_{\mathcal{X}} dx_{i+1}\ p_{max}(x_{i+1}|x_i)\Big]\sum_{j=1}^{N-1}(x_{j+1}-x_j)(x_{j+1}-x_j)^T\delta(x_j-x^*), \tag{20}$$

where the path integrals are discretized according to the Feynman formalism [24], using the same discretization as in our proof of Corollary 2.1.1.

From this expression in Eq. 20, we take the following two steps. First, we switch out the order of summation and product by applying the Fubini-Tonelli theorem. Then, we factor out two integrals from the product expression—one capturing the probability flow *into* x_j and one capturing the flow *out of* it:

$$= \sum_{j=1}^{N-1}\prod_{i\neq j,j-1}\Big[\int_{\mathcal{X}} dx_{i+1}\ p_{max}(x_{i+1}|x_i)\Big]\int_{\mathcal{X}} p_{max}(x_j|x_{j-1})\int_{\mathcal{X}} p_{max}(x_{j+1}|x_j)(x_{j+1}-x_j)(x_{j+1}-x_j)^T\delta(x_j-x^*)dx_{j+1}dx_j.$$

Then we can apply the Dirac delta function to simplify our expression and get:

$$= \sum_{j=1}^{N-1}\prod_{i\neq j,j-1}\Big[\int_{\mathcal{X}} dx_{i+1}\ p_{max}(x_{i+1}|x_i)\Big]p_{max}(x^*|x_{j-1})\int_{\mathcal{X}} p_{max}(x_{j+1}|x^*)(x_{j+1}-x^*)(x_{j+1}-x^*)^T dx_{j+1}. \tag{21}$$

To simplify further we will tackle the following integral as a separate quantity:

$$I = \int_{\mathcal{X}} p_{max}(x_{j+1}|x^*)(x_{j+1} - x^*)(x_{j+1} - x^*)^T dx_{j+1}. \tag{22}$$

where we can substitute Eq. 19 into Eq. 22 to get:

$$I = \int_{\mathcal{X}} \frac{1}{Z_d} e^{-(x_{j+1}-x^*)^T \Lambda(x^*)(x_{j+1}-x^*)}(x_{j+1} - x^*)(x_{j+1} - x^*)^T dx_{j+1}.$$

This integral can then be tackled using integration by parts and closed-form Gaussian integration. Thus far, we have not had any need to specify the domain in which exploration takes place. However, in order to evaluate this multi-dimensional integral-by-parts we require integration limits. To this end, we will assume that the domain of exploration is large enough so that the distance between x^* and x_{j+1} makes the exponential term approximately decay to 0 at the limits, which we shorthand by placing the limits at infinity:

$$I = \frac{1}{Z_d}\Lambda(x^*)^{-1}\Big[\sqrt{\det(2\pi\Lambda^{-1}(x^*))} - (x_{j+1} - x^*)^T \mathbf{1} e^{-(x_{j+1}-x^*)^T \Lambda(x^*)(x_{j+1}-x^*)}\Big|_{x_{j+1}=-\infty}^{x_{j+1}=\infty}\Big], \tag{23}$$

where $\mathbf{1}$ is the vector of all ones, and the exponential term vanishes when evaluated at the limits. Note that our assumption on the domain of integration implies that we do not consider boundary effects, and that the quantity within the brackets is a scalar that can commute with our Lagrange multiplier matrix.

We are now ready to put together our final results. By combining Eq. 23 and plugging it into Eq. 21 we have

$$\langle \dot{x}(t)\dot{x}(t)^T \rangle_{x^*} = \frac{1}{Z_d} \sum_{j=1}^{N-1} \prod_{i \neq j, j-1} \Big[\int_{\mathcal{X}} dx_{i+1}\ p_{max}(x_{i+1}|x_i) \Big] p_{max}(x^*|x_{j-1}) \sqrt{\det(2\pi\Lambda^{-1}(x^*))}\Lambda(x^*)^{-1}. \tag{24}$$

Since $\langle \dot{x}(t)\dot{x}(t)^T \rangle_{x^*}$ is everywhere full-rank, we can see that $\Lambda(x^*)^{-1}$ must be full-rank as well. Next, we recognize that $\sqrt{\det(2\pi\Lambda(x^*)^{-1})}$ cancels out with Z_d, and that we can re-expand $p_{max}(x^*|x_{j-1})$ as an integral over $\delta(x_j - x^*)$ and fold it back into the integral product. Rearranging terms we have:

$$\langle \dot{x}(t)\dot{x}(t)^T \rangle_{x^*} = \prod_{i=1}^{N-1} \Big[\int_{\mathcal{X}} dx_{i+1}\ p_{max}(x_{i+1}|x_i) \Big] \sum_{j=1}^{N-1} \delta(x_j - x^*)\Lambda(x^*)^{-1}. \tag{25}$$

At this point, we note that this expression merely computes the average of $\Lambda(x^*)^{-1}$ over all possible state trajectories that pass through x^*. However, because $\Lambda(x^*)^{-1}$ is a constant for any given x^*, this expression reduces down to $\langle \dot{x}(t)\dot{x}(t)^T \rangle_{x^*} = \Lambda(x^*)^{-1}$. Thus, using Eq. 16, we find that our Lagrange multiplier is given by:

$$\Lambda(x^*) = \mathbf{C}^{-1}[x^*]. \tag{26}$$

This result is significant because now we can relate a measure of temporal correlations to the sample path distribution of an optimally exploring agent. Taking this result and returning to Eq. 18, we now have the final form of the maximum entropy exploration sample path distribution in terms of our measure of temporal correlations:

$$P_{max}[x(t)] = \frac{1}{Z} \exp\Big[-\frac{1}{2} \int_{t_0}^{t} \dot{x}(\tau)^T \mathbf{C}^{-1}[x(\tau)]\dot{x}(\tau) d\tau \Big], \tag{27}$$

where we have added a factor of one half to precisely match the path probability of diffusive spatially-inhomogeneous dynamics. This final connection can be made rigorous by noting that $\mathbf{C}[x^*]$ is an estimator of a system's local diffusion tensor through the following relation: $\mathbf{C}[\cdot] = \frac{1}{2}\mathbf{D}[\cdot]\mathbf{D}[\cdot]^T$ for some diffusion tensor $\mathbf{D}[\cdot]$ [30, 31]. Lastly, we can discretize this distribution to arrive at the discrete-time maximum entropy sample path probability density:

$$p_{max}(x_{t+1}|x_t) = \frac{1}{Z_d} \exp\Big[-\frac{1}{2} |x_{t+1} - x_t|^2_{\mathbf{C}^{-1}[x_t]} \Big]. \tag{28}$$

Thus, when faced with path continuity constraints, the optimal exploration strategy is given by diffusion in state space, which concludes our derivation. In line with this, we describe systems that satisfy these statistics as *maximally diffusive*.

Throughout this derivation, we have assumed for convenience that the velocity fluctuations of the stochastic control process are full-rank everywhere. This is equivalent to saying that the control system is capable of generating variability along all dimensions of its degrees of freedom—or equivalently, as shown in Supplementary Note 2.1 for linearizable nonlinear systems,

that our system is controllable. However, this assumption is somewhat artificial because typically we are not interested in exploring directly on the full state space of our control system. Instead, we often consider some differentiable coordinate transformation $y(t) = \psi(x(t))$ that maps our states in $\mathcal{X}$ onto the desired exploration domain $\mathcal{Y}$. In this case, all results described thus far will still hold and we will have a valid expression for $P_{max}[y(t)]$ with diffusion tensor $\mathbf{C}[y^*]$, so long as $\mathbf{C}[y^*] = \mathbf{J}_\psi[x^*]\mathbf{C}[x^*]\mathbf{J}_\psi[x^*]^T$ is everywhere full-rank, where $\mathbf{J}_\psi[\cdot]$ is the Jacobian matrix corresponding to the coordinate transformation ψ. Hence, we only require that the new system coordinates are controllable. This is particularly useful when we are dealing with high-dimensional systems with which we are interested in exploring highly coarse-grained domains.

2.5 Directed exploration as variational optimization

In the previous section we derived the analytical form of our maximum entropy exploration strategy, which describes agents with maximally-decorrelated experiences and whose path probabilities are equivalent to those of an ergodic diffusion process. Thus far, we have only discussed exploration as an undirected (or passive) process. This is to say, as a process that is blind to any notion of importance or preference ascribed to regions of the state space or exploration domain [32]. However, under a simple reformulation of our exploration problem we will see that we can also achieve efficient directed exploration with theoretical guarantees on its asymptotic performance.

In many exploration problems, we have an a priori understanding of what regions of the exploration domain are important or informative. For example, in reinforcement learning this is encoded by the reward function [9], and in optimal control this is often encoded by a cost function or an expected information density [33, 34]. In such settings, one may want an agent to explore states while taking into account a measure of information or importance of that state, which is known as directed (or active) exploration. In order to realize directed exploration, we require a notion of the "importance" of states that is amenable to the statistical-mechanical construction of our approach. To this end, we reformulate our maximum entropy objective into a "free energy" minimization objective by introducing a bounded potential function $V[\cdot]$. Across fields, potential functions are used to ascribe (either a physical or virtual) cost to system states. A potential function is then able to encode tasks in control theory, learning objectives in artificial intelligence, desirable regions in spatial coverage problems, etc. Hence, we will extend the formalism presented in the previous sections to parsimoniously achieve goal-directed exploration by considering the effect of potential functions.

Since our maximum entropy functional is an expression over all possible trajectories, we need to adapt our definition of a potential to correctly express our notion of "free energy" over possible system realizations. To this end, we define our potential over $\mathcal{T} = [t_0, t]$ in the following way,

$$\langle V[x(t)]\rangle_P = \int_{\mathcal{X}^\mathcal{T}} P[x(t)] \int_{t_0}^{t} V[x(\tau)] d\tau \mathcal{D}x(t), \tag{29}$$

which captures the average cost over all possible system paths (integrated over each possible state and time for each possible path). Formally, we must assume that $\langle V[x(t)]\rangle_P$ is bounded, which in practice will be the case for policies and controllers derived from these principles. Our new free energy functional objective is

$$\underset{P[x(t)]}{\operatorname{argmin}} \ \langle V[x(t)]\rangle_P - S[P[x(t)]], \tag{30}$$

where we use $S[P[x(t)]]$ as a short-hand for the argument to Eq. 17. Thankfully, to find the optimal path distribution all of the work carried out in Supplementary Notes 2.3 and 2.4 remains unchanged. All that's needed is to take the variation of Eq. 29 with respect to $P[x(t)]$ and integrate it into the optimal path distribution. As this arithmetic is very similar to the derivation provided in the proof of Theorem 2.1, we omit it here. The resulting minimum free energy path distribution is then

$$P^V_{max}[x(t)] = \frac{1}{Z} \exp\Big[- \int_{t_0}^{t} \Big(V[x(\tau)] + \frac{1}{2} \dot{x}(\tau)^T \mathbf{C}^{-1}[x(\tau)] \dot{x}(\tau) \Big) d\tau \Big], \tag{31}$$

which corresponds to the path distribution of a diffusion process in a potential field. Hence, the optimal directed exploration strategy is to scale the strength of diffusion with respect to the desirability of the state. In this sense, the net effect of the potential is merely to bias the diffusion process. We refer to systems satisfying such statistics as *maximally diffusive with respect to the underlying potential*. As an aside, we note that,

$$P^V_{max}[x(t)] = P_{max}[x(t)] \cdot e^{-\int V[x(\tau)] d\tau} \tag{32}$$

from which we can recover $P_{max}[x(t)]$ in the absence of a potential (i.e., $V[\cdot] = 0$). We note that we can manipulate the above expression into a form amenable to Markov decision processes by letting $l(\cdot) = V[\cdot]$ be a standard cost function, which leads us to the following equivalent expression:

$$P^l_{max}[x_{1:N}] = \prod_{t=1}^{N-1} p_{max}(x_{t+1}|x_t) e^{-l(x_t)}, \tag{33}$$

where we have discretized agent state trajectories without loss of generality. Remarkably, this path distribution resembles the form of those used in the control-as-inference literature [35]. We will find that the form of this distribution we derived is crucial to the approach we take in trajectory synthesis and reinforcement learning, particularly once we introduce a dependence on agent actions into the cost function.

What are the properties of such an exploration strategy? Since we already know that the sample paths of agents applying our exploration strategy are Markovian, as long as the potential function and its interactions with our agent are memory-less the sample paths generated by Eq. 30 will continue to be as well. However, ergodicity is a more challenging property to ascertain as it depends on the properties of the underlying potential function and of our diffusion process. Nonetheless, in the following theorem we show that the trajectories of an agent successfully diffusing according to our exploration strategy in a non-singular potential will continue to be ergodic under some mild assumptions.

Theorem 2.2. *A stochastic control process (Definition 2.3) in a compact and connected space $\mathcal{X} \subset \mathbb{R}^d$ with a maximum entropy exploration strategy in a potential (in the sense of Eq. 31) is ergodic.*

Proof. The proof of this theorem can be easily arrived at by extending the proof of Corollary 2.1.2. As long as $V[\cdot]$ is bounded everywhere in the domain, we may discretize the stochastic control process in space and time everywhere in the domain, as in Corollary 2.1.2. Then, we can see that $p^V_{max}(x_{t+1}|x_t) = p_{max}(x_{t+1}|x_t)e^{-V[x_t]} > 0,\ \forall x_t, x_{t+\delta t} \in \mathcal{X},\ \forall t \in \mathcal{T}$. This is because we have already shown that $p_{max}(\cdot|\cdot) > 0$ in Corollary 2.1.2, and because of the properties of the potential. Thus, the underlying Markov chain described by the $p^V_{max}(x_{t+1}|x_t)$ transition kernel is aperiodic and all states communicate, which guarantees ergodicity and concludes our proof. □

Hence, the net effect of the potential is to reshuffle probability mass in the stationary distribution of the agent's underlying Markov chain. We note that these proofs can be carried out without discretizations by instead invoking the physics of diffusion processes, as in [36] where the authors proved that heterogeneous diffusion processes in a broad class of non-singular potentials are ergodic when the strength of the potential exceeds the strength of diffusion-driven fluctuations. However, here we limit ourselves to methods from the analysis of stochastic processes. In short, minimum free energy exploration leads to ergodic coverage of the exploration domain with respect to the potential. We note that this is an important result when it comes to the applicability of our results in robotics and reinforcement learning, as it is effectively an asymptotic guarantee on learning when the learning task is encoded by the choice of potential function—as we will illustrate in the following sections.

2.6 Minimizing path free energy produces diffusive gradient descent

To develop further intuition about the sense in which the statistics of Eq. 31 describe goal-directed exploratory behavior, we can examine the maximum likelihood trajectory of our minimum free energy path distribution under the assumption of path differentiability, which the rest of our analysis does not require. To do this, we begin by calculating the negative log-likelihood of $P^V_{max}[x(t)]$ neglecting the normalization factor:

$$\begin{aligned} -\log[P^V_{max}[x(t)]] &= \int_{t_0}^{t} V[x(\tau)] + \frac{1}{2}\dot{x}(\tau)^T\mathbf{C}^{-1}[x(\tau)]\dot{x}(\tau)d\tau \\ &= \int_{t_0}^{t} \mathcal{H}(\tau, x(\tau), \dot{x}(\tau))d\tau \\ &= \int_{t_0}^{t} \dot{x}(\tau)^T\mathbf{C}^{-1}[x(\tau)]\dot{x}(\tau) - \mathcal{L}(\tau, x(\tau), \dot{x}(\tau))d\tau, \end{aligned} \tag{34}$$

where we noted that the integral's argument is a Hamiltonian whose Legendre transform we can take, and arrive at an equivalent Lagrangian description of the system. Then, to derive an expression for the maximum likelihood trajectories of our path distribution we can extremize the Lagrangian's associated action functional:

$$\mathcal{A} = \int_{t_0}^{t} \mathcal{L}(\tau, x(\tau), \dot{x}(\tau))d\tau = \int_{t_0}^{t} V[x(\tau)] - \frac{1}{2}\dot{x}(\tau)^T\mathbf{C}^{-1}[x(\tau)]\dot{x}(\tau)d\tau. \tag{35}$$

Assuming that our potential is differentiable, we can find the dynamics of the maximum likelihood trajectory by using the Euler-Lagrange equations:

$$\begin{aligned} 0 &= \nabla_x\mathcal{L} - \frac{d}{dt}\big[\nabla_{\dot{x}}\mathcal{L}\big] \\ &= \nabla_x V[x(t)] - \frac{1}{2}\dot{x}(t)^T\nabla_x\mathbf{C}^{-1}[x(t)]\dot{x}(t) - \Big[-\ddot{x}(t)^T\mathbf{C}^{-1}[x(t)] - \dot{x}(t)^T\nabla_x\mathbf{C}^{-1}[x(t)]\dot{x}(t)\Big] \\ &= \nabla_x V[x(t)] + \ddot{x}(t)^T\mathbf{C}^{-1}[x(t)] + \frac{1}{2}\dot{x}(t)^T\nabla_x\mathbf{C}^{-1}[x(t)]\dot{x}(t) \end{aligned} \tag{36}$$

which we can rearrange into our final expression,

$$\ddot{x}(t) = -\mathbf{C}[x(t)]\Big[\nabla_x V[x(t)] + \frac{1}{2}\dot{x}(t)^T \nabla_x \mathbf{C}^{-1}[x(t)]\dot{x}(t)\Big] \tag{37}$$

This last expression represents the maximum likelihood dynamics of a system whose trajectories satisfy our minimum free energy path distribution. We note that $\nabla_x \mathbf{C}^{-1}[x(t)] = -\mathbf{C}^{-1}[x(t)]\nabla_x \mathbf{C}[x(t)]\mathbf{C}^{-1}[x(t)]$, which we omitted from Eq. 37 for notational simplicity. Our expression is comprised of two gradient-like terms. The first of these terms points in directions of descent for the potential, and the second in directions that increase the system's autocovariance statistics (or controllability, when applicable).

To simplify these dynamics further, we can make one of two assumptions: either that our measure of temporal correlations varies slowly over space (at least relative to $\nabla_x V$), or that our system dynamics are LTV. Taken together, our assumptions imply that $\nabla_x \mathbf{C} \approx \mathbf{0}$, which leads to a simplification of the final expression in Eq. 37. For the sake of making a connection to controllability, consider simplifying the maximum likelihood dynamics by assuming they are LTV. For this class of dynamics, Eq. 8 tells us that our system's controllability properties do not vary over state-space, as discussed in Supplementary Note 2.1. For systems with fixed or quasi-static morphologies this assumption holds well. Then, we have the following simplified dynamics:

$$\begin{aligned} \ddot{x}(t) &= -\mathbf{C}[x(t)]\nabla_x V[x(t)] \\ &= -W(t, t_0)\nabla_x V[x(t)]. \end{aligned} \tag{38}$$

By inspection we see that these second order dynamics resemble those of inertial gradient descent [37–40], with two key differences. First, the absence of a damping term in the expression, which can be artificially introduced and tuned to guarantee and optimize convergence. Alternatively, we can note that any physical system approximately satisfying maximally diffusive trajectory statistics will experience dissipation, which means there may be no need to introduce it artificially. Second, and more importantly, that our system's ability to produce descent directions that optimize the potential is affected by its controllability properties. Thus, our results show that controllable agents can minimize arbitrary potentials merely through diffusive state exploration, which forms the conceptual basis of our approach to optimization and learning in the following sections.

As a final note on the derivations carried out throughout all of Supplementary Note 2, we point out that most of the work we have done largely amounts to formulating an exploration problem and deriving the optimal trajectory distribution for a sufficiently broad class of agents—those with continuous paths. However, it is entirely possible for agents with discontinuous paths to have constraints on their state transitions as well. In other words, just because an agent may be capable of teleporting from one state to another (e.g., in a digital environment) it does not mean that it is equally easy to teleport to and from every state in the environment. Thus, as a final note we point out that every formal result we have proven throughout Supplementary Note 2 still holds when we remove the path continuity constraint (except for the analysis in this section). However, in this case the optimal distribution will be uniform over the state space, i.e., $p^U_{max}(x_{t+1}|x_t) = 1/|\mathcal{X}|$. In the presence of a potential our agent would also provably realize ergodic Markov exploration with respect to a cost or potential function. In this case, the optimal path distribution would take a similar form as Eq. 33, i.e., $p^{U,l}_{max}(x_{t+1}|x_t) = p^U_{max}(x_{t+1}|x_t)e^{-l(x_t)}$. However, realizing these path statistics is only possible when the underlying agent is fully controllable in the sense of Definition 3.1, as we discuss in the following section. We note that it is exclusively under these conditions that agents can completely overcome correlations between state transitions. Agents satisfying these statistics will achieve *i.i.d.* sequential sampling; however, the connection to the statistical mechanics of diffusion processes will no longer hold.

3 Synthesizing maximally diffusive trajectories

Throughout the previous section, we have been studying the properties of a theoretical agent whose experiences spontaneously satisfy the path statistics of a maximally diffusive stochastic control process. However, the autonomous dynamics of control systems will typically not satisfy these statistics on their own. Hence, we require an approach from which to synthesize controllers (and policies) that generate maximally diffusive trajectories. In this section, we provide a general formulation of such an approach as well as simplifications amenable to use in real-time optimal control synthesis and reinforcement learning. All results derived herein form part of what we refer to as *maximum diffusion (MaxDiff) trajectory synthesis*.

3.1 Maximally diffusive trajectories via KL control

In previous sections, we derived the maximally diffusive path distribution, $P^V_{max}[x(t)]$, and characterized the properties of sample paths drawn from it in the presence of a potential that ascribes a cost to system states, $V[\cdot]$. Now, we turn to the question of synthesizing policies and controllers that can actually satisfy these trajectory distributions. To this end, we recall that in Supplementary Note 2.2 we defined a path probability density for an arbitrary stochastic control process, $P_{u(t)}[x(t)]$. Equipped with this distribution, we are able to express the most general form of the MaxDiff trajectory synthesis objective. To synthesize maximally diffusive trajectories, it suffices to generate policies and controllers that minimize the Kullback-Leibler (KL) divergence between the analytical optimum we derived in Supplementary Note 2 and the system's current path distribution. Equivalently, we can express this as,

$$\underset{u(t)}{\operatorname{argmin}}\ D_{KL}(P_{u(t)}[x(t)]||P^V_{max}[x(t)]), \tag{39}$$

which we can reformulate into many alternative forms through simple manipulations, as we illustrate throughout the following sections. Here, we first manipulate the objective into a form that highlights the different roles of the terms comprising it. Importantly, we note that taking the KL divergence is a well-defined operation in this context because the support of $P^V_{max}[x(t)]$ is infinite, and we have assumed that $\mathcal{X}$ is a compact domain. Using the definition of the KL divergence over path distributions and taking $\mathcal{T} = [t_0, t]$, we can factor our objective in the following way:

$$\begin{aligned} D_{KL}(P_{u(t)}[x(t)]||P^V_{max}[x(t)]) &= \int_{\mathcal{X}^\tau} P_{u(t)}[x(t)] \log \frac{P_{u(t)}[x(t)]}{P^V_{max}[x(t)]} \mathcal{D}x(t) \\ &= \int_{\mathcal{X}^\tau} P_{u(t)}[x(t)] \Big[\log P_{u(t)}[x(t)] - \log P^V_{max}[x(t)] \Big] \mathcal{D}x(t) \\ &= \int_{\mathcal{X}^\tau} P_{u(t)}[x(t)] \Big[\log P_{u(t)}[x(t)] - \log P_{max}[x(t)] + \int_{t_0}^{t} V[x(\tau)] d\tau \Big] \mathcal{D}x(t) \\ &= \langle V[x(t)] \rangle_{P_{u(t)}} + D_{KL}(P_{u(t)}[x(t)]||P_{max}[x(t)]), \end{aligned} \tag{40}$$

where we used Eq. 32 to arrive at our final expression. Now, we can rewrite our control synthesis problem as the following

$$\underset{u(t)}{\operatorname{argmin}}\ \langle V[x(t)] \rangle_{P_{u(t)}} + D_{KL}(P_{u(t)}[x(t)]||P_{max}[x(t)]), \tag{41}$$

or equivalently

$$\underset{u(t)}{\operatorname{argmin}}\ E_{P_{u(t)}}\big[L[x(t), u(t)]\big] + D_{KL}(P_{u(t)}[x(t)]||P_{max}[x(t)]), \tag{42}$$

where we replace our potential with a cost function $L[x(t), u(t)] = \int_{\mathcal{T}} l(x(t), u(t)) dt$ in terms of the running cost $l(\cdot, \cdot)$. While potential functions are a natural way to ascribe thermodynamic costs to the states of physical systems, such as diffusion processes, there is no reason to restrict ourselves to that formalism now that we are focused on control synthesis. We also replaced our physics-based expected value notation, but note that they are formally equivalent (i.e., $\langle \cdot \rangle_p = E_p[\cdot]$). Finally, we note that we can introduce a temperature-like parameter $\alpha > 0$ to balance between the two terms in our objective: the first, which optimizes task performance; and the second, which optimizes the statistics of the system's state space diffusion. Thus, when the system is able to achieve maximally diffusive trajectory statistics, our approach reduces to solving the task with thorough exploration of the cost landscape.

An interesting property of this result is that in our theoretical approach there is no formal trade-off between exploration and exploitation—at least asymptotically. This is because when a system is capable of realizing maximally diffusive trajectories, the KL divergence term goes to zero. That being said, in practice this is not the case and the introduction of α will be of practical use in balancing between exploration and exploitation. Moreover, when maximally diffusive statistics are satisfied the expected value of the objective is taken with respect to the optimal maximum entropy trajectory distribution (i.e., $E_{P_{max}}\big[L[x(t), u(t)]\big]$), which is a bias-minimizing estimator of the cost function asymptotically equivalent to *i.i.d.* sampling of state-action costs (or rewards) as a result of the ergodic properties of $P_{max}[x(t)]$. This is particularly useful in applications like reinforcement learning where the cost (or reward) function is unknown.

3.2 Maximally diffusive trajectories via stochastic optimal control

We can formulate our KL control problem as an equivalent stochastic optimal control (SOC) problem by making use of their well-known connections [35]. SOC problems are typically framed as Markov decision processes (MDPs) where the objective is to find a policy that optimizes the expected cost of a given cost-per-stage function over some time-horizon. More formally, an MDP is a 5-tuple $(\mathcal{X},\mathcal{U},p,r,\gamma)$, with state space, $\mathcal{X}$, and action space, $\mathcal{U}$. Then, $p:\mathcal{X}\times\mathcal{X}\times\mathcal{U}\to[0,\infty)$ represents the probability density of transitioning from state $x_t\in\mathcal{X}$ to state $x_{t+1}\in\mathcal{X}$ after taking action $u_t\in\mathcal{U}$. At every point in time, for each state, and for each action taken, the environment emits a bounded loss $l:\mathcal{X}\times\mathcal{U}\to[l_{min},l_{max}]$ discounted at a rate $\gamma\in[0,1)$. Given this formalism, the standard discrete time formulation of the SOC objective is

$$\pi^*=\underset{\pi}{\operatorname{argmin}}\ E_{(x_{1:N},u_{1:N})\sim P_\pi}\left[\sum_{t=1}^{N}\gamma^t l(x_t,u_t)\right], \tag{43}$$

where $l(\cdot,\cdot)$ is a discretized running cost and the expectation is taken with respect to the trajectory distribution induced by the policy P_π, which we will now motivate and define.

To translate our KL control results from the previous section into an equivalent SOC problem, we will have to make some modifications to our approach. In particular, the introduction of a policy $\pi(\cdot|\cdot)$ that replaces our notion of a controller (as defined in Supplementary Note 2.2) requires careful treatment. Whereas our definition of a path distribution allowed us to express a distribution directly over the state trajectories of the agent-environment dynamical process, the introduction of a policy induces a distribution over actions as well. In other words, instead of $P_{u_{1:N}}[x_{1:N}]$, we will now have $P_\pi[x_{1:N},u_{1:N}]$. This creates a complication by making the KL divergence in Eq. 39 ill-posed—the agent's state-action path distribution and our maximally diffusive distribution are now defined over different domains. To solve this issue, we introduce the following distributions:

$$\begin{aligned}P_\pi[x_{1:N},u_{1:N}]&=\prod_{t=1}^{N}p(x_{t+1}|x_t,u_t)\pi(u_t|x_t)\\ P^l_{max}[x_{1:N},u_{1:N}]&=\prod_{t=1}^{N}p_{max}(x_{t+1}|x_t)e^{-l(x_t,u_t)},\end{aligned} \tag{44}$$

where $p_{max}(x_{t+1}|x_t)$ is the discretized maximally diffusive conditional density in Eq. 28. The second of these distributions was analytically derived in Eq. 33, and we can formally introduce an action dependence because the maximally diffusive path distribution is action-independent. Note that for the first time in our derivation we are making use of the Markov property to express our system's dynamics. However, since the analytically-derived optimal transition dynamics are Markovian, the synthesized controller will attempt to make the agent's true dynamics satisfy the Markov property as a result of the underlying optimization, which makes this a benign assumption under our framework. We note that the more general problem description in Eq. 39 does not require us to assume that our dynamics are Markovian because we are minimizing the KL divergence between the trajectory distributions directly.

Taken together, these modifications allow us to rewrite Eq. 39 as,

$$\underset{\pi}{\operatorname{argmin}}\ D_{KL}(P_\pi[x_{1:N},u_{1:N}]||P^l_{max}[x_{1:N},u_{1:N}]). \tag{45}$$

Then, working from the definition of the KL divergence we have

$$\begin{aligned}D_{KL}(P_\pi[x_{1:N},u_{1:N}]||P^l_{max}[x_{1:N},u_{1:N}])&=E_{P_\pi}\left[\log\frac{P_\pi[x_{1:N},u_{1:N}]}{P^l_{max}[x_{1:N},u_{1:N}]}\right]\\ &=E_{P_\pi}\left[\log\prod_{t=1}^{N}\frac{p(x_{t+1}|x_t,u_t)\pi(u_t|x_t)}{p_{max}(x_{t+1}|x_t)e^{-l(x_t,u_t)}}\right]\\ &=E_{P_\pi}\left[\sum_{t=1}^{N}\log\frac{p(x_{t+1}|x_t,u_t)\pi(u_t|x_t)}{p_{max}(x_{t+1}|x_t)e^{-l(x_t,u_t)}}\right]\\ &=E_{P_\pi}\left[\sum_{t=1}^{N}l(x_t,u_t)+\log\frac{p(x_{t+1}|x_t,u_t)\pi(u_t|x_t)}{p_{max}(x_{t+1}|x_t)}\right].\end{aligned}$$

At this point, we explicitly introduce a temperature-like parameter, $\alpha>0$, to balance between the terms of our objective, as mentioned in the previous section and in the main text. We note that this is a benign modification because equivalent to

scaling our costs or rewards by $1/\alpha$, and leads to the following result:

$$D_{KL}(P_\pi[x_{1:N}, u_{1:N}]||P^l_{max}[x_{1:N}, u_{1:N}]) = E_{P_\pi}\left[\sum_{t=1}^{N} l(x_t, u_t) + \alpha \log \frac{p(x_{t+1}|x_t, u_t)\pi(u_t|x_t)}{p_{max}(x_{t+1}|x_t)}\right]. \tag{46}$$

With this result we are now able to write our final expression for an equivalent SOC representation of the KL control problem in Eq. 39:

$$\pi^*_{\text{MaxDiff}} = \underset{\pi}{\text{argmin}}\ E_{(x_{1:N}, u_{1:N})\sim P_\pi}\left[\sum_{t=1}^{N} \gamma^t \hat{l}(x_t, u_t)\right], \tag{47}$$

where we introduced the discounting factor γ, and with

$$\hat{l}(x_t, u_t) = l(x_t, u_t) + \alpha \log \frac{p(x_{t+1}|x_t, u_t)\pi(u_t|x_t)}{p_{max}(x_{t+1}|x_t)}, \tag{48}$$

as our modified running cost function, which concludes our derivation of the formal equivalence between the KL control and SOC MaxDiff trajectory synthesis problems. When we modify the objective above by instead maximizing a reward function $\hat{r}(x_t, u_t)$ with $r(x_t, u_t) = -l(x_t, u_t)$, we refer to this objective as the *MaxDiff RL* objective as we have done in the main text, and whose properties we will explore in the following section.

3.3 Maximum diffusion reinforcement learning

The objective we derived in the previous section in Eq. 48 is the standard form of the MaxDiff RL objective, as discussed in the main text. In this section, we will explore the relationship between MaxDiff RL and MaxEnt RL, and prove some properties of MaxDiff RL agents. Central to these discussions is the way that temporal correlations and controllability play a role in our theoretical framework. For this reason, we first formalize and define a particular notion of controllability in the context of MDPs that was partially introduced in [41], implicit in the results of [42], and explicitly called out in [35].

Definition 3.1. *The state transition dynamics, $p(x_{t+1}|x_t, u_t)$, in an MDP, $(\mathcal{X}, \mathcal{U}, p, r, \gamma)$, are fully controllable when there exists a policy, $\pi : \mathcal{U} \times \mathcal{X} \to [0, \infty)$, such that:*

$$p_\pi(x_{t+1}|x_t) = E_{u_t\sim\pi(\cdot|x_t)}[p(x_{t+1}|x_t, u_t)] \tag{49}$$

and

$$D_{KL}\Big(p_\pi(x_{t+1}|x_t)\Big|\Big|\nu(x_{t+1}|x_t)\Big) = 0, \quad \forall t \in \mathbb{Z}^+ \tag{50}$$

for any arbitrary choice of state transition probabilities, $\nu : \mathcal{X} \times \mathcal{X} \to [0, \infty)$.

Thus, a system is *fully controllable* when it is simultaneously capable of reaching every state and controlling *how* each state is reached. In other words, a fully controllable agent can arbitrarily manipulate its state transition probabilities, $p_\pi(x_{t+1}|x_t)$, by using an optimized policy to match any desired transition probabilities, $\nu(x_{t+1}|x_t)$. Whether the underlying policy is deterministic or stochastic is irrelevant to Definition 3.1. However, our interpretation of $p_\pi(x_{t+1}|x_t)$ is different in either setting. When the policy is stochastic we interpret the agent's state transition dynamics due to a policy as

$$p_\pi(x_{t+1}|x_t) = \int_{\mathcal{U}} p(x_{t+1}|x_t, u_t)\pi(u_t|x_t)du_t, \tag{51}$$

where the integral over control actions arises from the expectation in Eq. 49. Alternatively, in the deterministic case the agent's state transition dynamics are given by

$$p_\pi(x_{t+1}|x_t) = \int_{\mathcal{U}} p(x_{t+1}|x_t, u_t)\delta(u_t - \tau_\pi(x_t))du_t = p(x_{t+1}|x_t, \tau_\pi(x_t)), \tag{52}$$

where action sequences are drawn from $\pi(u_t|x_t) = \delta(u_t - \tau_\pi(x_t))$, which is a Dirac delta where $u_t = \tau_\pi(x_t)$ is some deterministic function of the current state [35].

Equipped with our definition of full controllability, we may now shed a light on the relationship between our MaxDiff RL framework and the broader MaxEnt RL literature [9, 10, 43], and present one of our main theorems.

Theorem 3.1. *(Theorem 1 of Main Text) Let the state transition dynamics due to a policy π be $p_\pi(x_{t+1}|x_t)$. If the state transition dynamics are assumed to be decorrelated, then the optimum of Eq. 48 is reached when $D_{KL}(p_\pi||p_{max}) = 0$ and the MaxDiff RL objective reduces to the MaxEnt RL objective.*

Proof. Our goal in this proof will be to take the MaxDiff RL objective function in Eq. 47 and explore its relationship to the MaxEnt RL objective. Neglecting the discounting factor γ but without loss of generality, we begin our proof by algebraically manipulating the MaxDiff RL objective function in Eq. 47:

$$\begin{aligned}
E_{P_\pi}\left[\sum_{t=1}^{N}\hat{l}(x_t,u_t)\right] &= E_{P_\pi}\left[\sum_{t=1}^{N} l(x_t,u_t) + \alpha\log\frac{p(x_{t+1}|x_t,u_t)\pi(u_t|x_t)}{p_{max}(x_{t+1}|x_t)}\right] \\
&= E_{P_\pi}\left[\sum_{t=1}^{N} l(x_t,u_t)\right] + \sum_{t=1}^{N} E_{(x_t,u_t)\sim p,\pi}\left[\alpha\log\frac{p(x_{t+1}|x_t,u_t)\pi(u_t|x_t)}{p_{max}(x_{t+1}|x_t)}\right] \\
&= E_{P_\pi}\left[\sum_{t=1}^{N} l(x_t,u_t) + \alpha\log\pi(u_t|x_t)\right] + \sum_{t=1}^{N} E_{(x_t,u_t)\sim p,\pi}\left[\alpha\log\frac{p(x_{t+1}|x_t,u_t)}{p_{max}(x_{t+1}|x_t)}\right].
\end{aligned}$$

So far, we have merely rearranged the terms in the MaxDiff RL objective by taking advantage of the linearity of expectations and the definition of P_π in Eq. 44. Now, we proceed by applying Jensen's inequality to the last term of our expression above—bringing in the expectation over control actions into the logarithm, noting that $E_{u_t\sim\pi}[p_{max}(x_{t+1}|x_t)] = p_{max}(x_{t+1}|x_t)$, and doing more algebraic manipulations:

$$\begin{aligned}
E_{P_\pi}\left[\sum_{t=1}^{N}\hat{l}(x_t,u_t)\right] &\leq E_{P_\pi}\left[\sum_{t=1}^{N} l(x_t,u_t) + \alpha\log\pi(u_t|x_t)\right] + \sum_{t=1}^{N} E_{x_t\sim p}\left[\alpha\log\frac{E_{u_t\sim\pi}[p(x_{t+1}|x_t,u_t)]}{p_{max}(x_{t+1}|x_t)}\right] \\
&\leq E_{P_\pi}\left[\sum_{t=1}^{N} l(x_t,u_t) + \alpha\log\pi(u_t|x_t)\right] + \sum_{t=1}^{N} E_{x_t\sim p}\left[\alpha\log\frac{p_\pi(x_{t+1}|x_t)}{p_{max}(x_{t+1}|x_t)}\right] \\
&\leq E_{P_\pi}\left[\sum_{t=1}^{N} l(x_t,u_t) + \alpha\log\pi(u_t|x_t) + \alpha D_{KL}\big(p_\pi(x_{t+1}|x_t)\big|\big|p_{max}(x_{t+1}|x_t)\big)\right],
\end{aligned} \tag{53}$$

where we also used the definition of $p_\pi(x_{t+1}|x_t)$ from Eq. 49.

To conclude our proof, we must show that the MaxEnt RL objective emerges from the MaxDiff RL objective under the assumption that an agent's state transitions are decorrelated. We can formalize what decorrelation requires of an agent in one of two contexts—that of agents with continuous experiences, or in general. Our derivation throughout Supplementary Note 2 achieves this in the context of agents with continuous experiences. Therein, we proved that the least-correlated continuous agent paths uniquely satisfy maximally diffusive trajectory statistics, which requires that $D_{KL}(p_\pi||p_{max}) = 0$ when there exists an optimizing policy π. Alternatively, completely decorrelating the state transitions of an agent in general requires being able to generate arbitrary jumps between states—as discussed in the main text—which requires full controllability (see Definition 3.1). Given full controllability, the optimum of Eq. 53 is also reached when $D_{KL}(p_\pi||p_{max}) = 0$.

Applying the assumption of decorrelated state transitions in either of the two senses expressed above not only simplifies Eq. 53 by removing the KL divergence term but also by saturating Jensen's inequality, which recovers the equality between the left and right hand sides of our equations:

$$E_{P_\pi}\left[\sum_{t=1}^{N}\hat{l}_c(x_t,u_t)\right] = E_{P_\pi}\left[\sum_{t=1}^{N} l(x_t,u_t) + \alpha\log\pi(u_t|x_t)\right],$$

where we added the subscript c to indicate that this applies under the assumption of decorrelated state transitions—either in the context of agents with continuous paths (with maximum diffusivity as a necessary condition) or in general (with full controllability as a sufficient condition). Putting together our final results, we may now write down the simplified MaxDiff RL optimization objective with the added assumption of decorrelated state transitions:

$$\pi^* = \underset{\pi}{\operatorname{argmin}}\ E_{(x_{1:N},u_{1:N})\sim P_\pi}\left[\sum_{t=1}^{N}\gamma^t\hat{l}_c(x_t,u_t)\right], \tag{54}$$

where we reintroduced γ, and with

$$\hat{l}_c(x_t,u_t) = l(x_t,u_t) + \alpha\log\pi(u_t|x_t), \tag{55}$$

or equivalently, we can write Eq. 54 as a maximization by replacing the cost with a reward function:

$$\hat{r}_c(x_t,u_t) = r(x_t,u_t) + \alpha\mathcal{H}(\pi(u_t|x_t)), \tag{56}$$

where we briefly changed our entropy notation, using $\mathcal{H}(\pi(u_t|x_t)) = S[\pi(u_t|x_t)]$, to highlight similarities with other results in the literature. Crucially, we recognize this objective as the MaxEnt RL objective, which proves that MaxDiff RL is a strict generalization of MaxEnt RL to agents with temporally correlated experiences and concludes our proof. We note that this also proves that maximizing policy entropy does not decorrelate state transitions in general because maximizing policy entropy does not minimize $D_{KL}(p_\pi||p_{max})$. □

In contrast to MaxEnt RL, when the agent-environment state transition dynamics introduce temporal correlations, the MaxDiff RL objective continues to prioritize effective exploration by decorrelating state transitions and encouraging the system to realize maximally diffusive trajectories. As we have shown above, MaxEnt RL's strategy of decorrelating action sequences is only as effective as MaxDiff RL's strategy of decorrelating state sequences when the underlying system's dynamics do not introduce temporal correlations on their own.

Now, we turn to analyzing the formal properties of MaxDiff RL agents. In particular, we will analyze how the ergodic properties of maximally diffusive trajectories (i.e., Theorem 2.2) can have an impact on the learning performance of MaxDiff RL agents. Namely, on their single-shot learning capabilities and their robustness to seeds and initializations. Prior to proceeding formally, we must first provide a framework from which to asses the learning performance of RL agents. To this end, we will make use of the representation-agnostic probably approximately correct in Markov decision processes (PAC-MDP) framework [44, 45].

Definition 3.2. *An algorithm $\mathcal{A}$ is said to be PAC-MDP (Probably Approximately Correct in Markov Decision Processes) if, for any $\epsilon > 0$ and $\delta \in (0, 1)$, a policy π can be produced with poly$(|\mathcal{X}|, |\mathcal{U}|, 1/\epsilon, 1/\delta, 1/(1-\gamma))$ sample complexity that is at least ϵ-optimal with probability at least $1 - \delta$. In other words, if $\mathcal{A}$ satisfies*

$$\Pr\big(\mathcal{V}_{\pi^*}(x_0) - \mathcal{V}_\pi(x_0) \leq \epsilon\big) \geq 1 - \delta \tag{57}$$

with polynomial sample complexity for all $x_0 \in \mathcal{X}$, where

$$\mathcal{V}_\pi(x_t) = E_{p,\pi}\left[\sum_{n=0}^{\infty} \gamma^n r(x_{n+t}, u_{n+t}) \middle| x_t = x\right] \tag{58}$$

is the value function and $\mathcal{V}_{\pi^*}(x_t)$ is the optimal value function, then $\mathcal{A}$ is PAC-MDP.

Thus, this framework states that an algorithm $\mathcal{A}$ is PAC-MDP when it is capable of learning a policy with polynomial sample complexity that can get arbitrarily close to the optimal policy with arbitrarily high probability. We note that this framework is representation-agnostic in the sense that, regardless of whether $\mathcal{A}$ involves any kind of neural network representation, any algorithm that satisfies Definition 3.2 is guaranteed to be at least ϵ-optimal.

Given our definition of the PAC-MDP framework, we will now consider the implications of our results on single-shot learning. Most applications of deep RL take place in episodic environments where after each execution of a given task, the agent and environment are reset, and their initial conditions are randomized. This is the setting that we have referred to as "multi-shot" learning throughout our manuscript. However, learning outside of episodic environments is crucial to real-world applications of deep RL [46, 47]. In non-episodic tasks, agents are expected to learn within a single deployment without resetting the task or environment, which we have referred to as the "single-shot" learning setting throughout our manuscript. With this in mind, we are now able to state our next formal result in terms of the PAC-MDP framework.

Theorem 3.2. *(Theorem 3 of Main Text) If there exists a PAC-MDP algorithm $\mathcal{A}$ with policy π^{max} for the MaxDiff RL objective (Eq. 48), then the Markov chain induced by π^{max} is ergodic, and any individual initialization of $\mathcal{A}$ will asymptotically satisfy the same ϵ-optimality as an ensemble of initializations.*

Proof. This theorem follows directly from the ergodicity of maximally diffusive trajectories (Theorem 2.2), some basic facts about MDPs [29], and the application of Birkhoff's ergodic theorem [48] onto our definition of PAC-MDP (Definition 3.2). First, since $\mathcal{A}$ is capable of producing an ϵ-optimal policy, π^{max}, we take $D_{KL}(p_{\pi^{max}}||p_{max}) \approx 0$ for some choice of ϵ, given that $p_{\pi^{max}}(x_{t+1}|x_t) = \int_{\mathcal{U}} p(x_{t+1}|x_t, u_t)\pi^{max}(u_t|x_t)du_t$. Then, it is well-known that any given policy in an MDP gives rise to a Markov chain on the state-space of the MDP [29]. Naturally, the properties of the policy-induced Markov chain depend on the properties of the resulting state transition kernel (e.g., $p_\pi(x_{t+1}|x_t)$).

Now, let $\{x_t\}_{t\in\mathbb{N}}$ be a Markov chain with state transition properties determined by $p_{\pi^{max}}(x_{t+1}|x_t)$. Because we know that $D_{KL}(p_{\pi^{max}}||p_{max}) \approx 0$, the Markov chain described by $p_{\pi^{max}}(x_{t+1}|x_t)$ is ergodic (per Theorem 2.2) with invariant measure ρ. To proceed further, we will now state Birkhoff's well-known ergodic theorem [48, 49].

Theorem 3.3. *(Birkhoff's ergodic theorem) Let $\{x_t\}_{t\in\mathbb{N}}$ be an aperiodic and irreducible Markov process on a state space $\mathcal{X}$ with invariant measure ρ and let $f : \mathcal{X} \to \mathbb{R}$ be any measurable function with $E[|f(x)|] < \infty$. Then, one has*

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} f(x_t) = E_{x_0\sim\rho}[f(x_0)] \tag{59}$$

almost surely.

In other words, Birkhoff's ergodic theorem states the the time-average of any function of an ergodic Markov chain is equal to its ensemble average.

Now, we return to the definition of PAC-MDP to slightly manipulate the expression:

$$\Pr\big(\mathcal{V}_{\pi^*}(x_0) - \mathcal{V}_{\pi^{max}}(x_0) \leq \epsilon\big) \geq 1 - \delta$$
$$E_{x_0\sim\rho}\Big[\mathbf{1}\{\mathcal{V}_{\pi^*}(x_0) - \mathcal{V}_{\pi^{max}}(x_0) \leq \epsilon\}\Big] \geq 1 - \delta,$$

where $\mathbf{1}\{\cdot\}$ denotes an indicator function. In other words, to be PAC-MDP is equivalent to being at least ϵ-optimal on average at least $100 \times (1 - \delta)\%$ of episodes. To conclude our proof, let

$$f(x_t) = \mathbf{1}\{\mathcal{V}_{\pi^*}(x_t) - \mathcal{V}_{\pi^{max}}(x_t) \leq \epsilon\}$$

be an observable, which we note is bounded, and apply Birkhoff's theorem. Then, we will have

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \mathbf{1}\{\mathcal{V}_{\pi^*}(x_t) - \mathcal{V}_{\pi^{max}}(x_t) \leq \epsilon\} = E_{x_0\sim\rho}[\mathbf{1}\{\mathcal{V}_{\pi^*}(x_0) - \mathcal{V}_{\pi^{max}}(x_0) \leq \epsilon\}],$$

which proves that any individual initial condition will satisfy the ensemble average. In turn, we have

$$\Pr\big(\mathcal{V}_{\pi^*}(x_0) - \mathcal{V}_{\pi^{max}}(x_0) \leq \epsilon\big) \geq 1 - \delta \implies \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \mathbf{1}\{\mathcal{V}_{\pi^*}(x_t) - \mathcal{V}_{\pi^{max}}(x_t) \leq \epsilon\} \geq 1 - \delta$$

almost surely, which proves that an algorithm that is PAC-MDP during multi-shot (episodic) learning is guaranteed to be PAC-MDP during single-shot (non-episodic) learning if the underlying Markov chain induced by the policy is ergodic. This concludes our proof. □

An important note about this proof is that it clarifies why ergodic sampling along continuous Markovian trajectories is the best possible alternative to *i.i.d.* sampling (via Theorem 3.3). As a result of ergodicity, the sampling statistics of any individual realization of an ergodic process are indistinguishable from *i.i.d.* sampling asymptotically, almost surely.

Next, we will prove that any PAC-MDP algorithm applied onto the MaxDiff RL objective will be robust to initial conditions and random seeds, which is a highly desirable property of deep RL agents.

Theorem 3.4. *(Theorem 2 of Main Text) If there exists a PAC-MDP algorithm $\mathcal{A}$ with policy π^{max} for the MaxDiff RL objective (Eq. 48), then the Markov chain induced by π^{max} is ergodic, and $\mathcal{A}$ will be asymptotically ϵ-optimal regardless of initialization.*

Proof. The proof of this theorem is simple given the proof to Theorem 3.2. Once again, let

$$f(x_t) = \mathbf{1}\{\mathcal{V}_{\pi^*}(x_t) - \mathcal{V}_{\pi^{max}}(x_t) \leq \epsilon\}$$

be an observable. Now, let $\{x_t\}_{t\in\mathbb{N}}$ and $\{x'_t\}_{t\in\mathbb{N}}$ both be ergodic Markov chains with identical transition kernels given by $p_{\pi^{max}}$, but with different initial conditions $x_0, x'_0 \in \mathcal{X}$. Then, since Birkhoff's ergodic theorem guarantees that the time-averages of observables from $\{x_t\}_{t\in\mathbb{N}}$ and $\{x'_t\}_{t\in\mathbb{N}}$ will converge to the same unique ensemble average over the invariant measure ρ (Theorem 3.3), the following is true:

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} |f(x_t) - f(x'_t)| = 0$$

for any $x_0, x'_0 \in \mathcal{X}$ almost surely. This proves that any PAC-MDP algorithm is guaranteed to be robust to random seeds and environmental initializations if the underlying Markov chain induced by the policy is ergodic, which concludes our proof. □

Thus, we have now proven that MaxDiff RL generalizes MaxEnt RL, that MaxDiff RL agents are capable of single-shot learning, and that MaxDiff RL agents are robust to initial conditions and random seeds—all because of the deep connection between maximally diffusive trajectories and ergodicity.

An interesting aside is that the MaxDiff RL objective formally requires model-based techniques to optimize because of its dependence on the system's state transition dynamics. In this sense, MaxEnt RL is the best one can do in a model-free setting—yet, with model-based techniques better performance is attainable when the system dynamics introduce temporal correlations. However, if one has direct access to state transition entropy estimates, then by reformulating the objective function in Eq. 47, it is technically possible to extend our results to model-free algorithms, as we show in the following sections.

3.4 Alternative synthesis approach via path entropy maximization

In Supplementary Note 3.1, we derived a synthesis approach based on KL control that optimizes exploration and task performance by making agents realize maximally diffusive trajectories. Alternatively, we can use the fact that in Supplementary Note 2 we derived the unique trajectory distribution $P_{max}[x(t)]$ with maximum entropy $S[P_{max}[x(t)]]$ that satisfies our constraints—which merely amount to prohibiting teleportation between states. As a result of this, we know that $S[P_{max}[x(t)]] \geq S[P_{u(t)}[x(t)]]$ with equality if and only if $P_{max}[x(t)] = P_{u(t)}[x(t)]$. Thus, instead of minimizing the KL divergence, we can instead maximize $S[P_{u(t)}[x(t)]]$, leading to the following equivalent optimization problem,

$$\underset{u(t)}{\text{argmax}}\ S[P_{u(t)}[x(t)]], \tag{60}$$

whose optimum satisfies $S[P_{u^*(t)}[x(t)]] = S[P_{max}[x(t)]]$. Based on this specification, we can define several other equivalent MaxDiff trajectory synthesis problem specifications that may be more or less convenient depending on the details of the application domain:

$$\max_{u(t)} S[P_{u(t)}[x(t)]], \quad \max_{u_{1:N-1}} S\left[\prod_{t=1}^{N} p(x_{t+1}|x_t, u_t)\right], \quad \max_{\pi} S[P_\pi[x(t), u(t)]], \quad \max_{\pi} S\left[\prod_{t=1}^{N} p(x_{t+1}|x_t, u_t)\pi(u_t|x_t)\right], \tag{61}$$

where $P_\pi[x(t), u(t)]$ is a continuous-time distribution over states and control actions analogous to $P_\pi[x_{1:N}, u_{1:N}]$, and we can think of a controller as a policy given by a Dirac delta distribution centered at u_t. The equivalence between the KL control and SOC formulations of the problem, and the formulation we have produced in this section, leads to

$$\underset{u(t)}{\text{argmin}}\ E_{P_{u(t)}}\big[L[x(t), u(t)]\big] - \alpha S[P_{u(t)}[x(t)]], \quad \underset{\pi}{\text{argmin}}\ E_{P_\pi}\big[L[x(t), u(t)]\big] - \alpha S[P_\pi[x(t), u(t)]] \tag{62}$$

and

$$\underset{u_{1:N}}{\text{argmin}}\ E_{P_{u_{1:N}}}\left[\sum_{t=1}^{N} l(x_t, u_t) - \alpha S[p(x_{t+1}|x_t, u_t)]\right], \quad \underset{\pi}{\text{argmin}}\ E_{P_\pi}\left[\sum_{t=1}^{N} l(x_t, u_t) - \alpha S[p(x_{t+1}|x_t, u_t)\pi(u_t|x_t)]\right] \tag{63}$$

also being formally equivalent to Eq. 39, where we omit γ. While the different objectives listed in Eqs. 61-63 may seem redundant, some of these may prove to be more readily applicable in particular domains, or to a given practitioner's preferred policy synthesis approach. In the following section, we derive an additional objective that attains the same optimum as Eqs. 61-63, but is better suited to model-free optimizations.

3.5 Simplified synthesis via local entropy maximization

As currently written, the objectives specified thus far require access to $p(x_{t+1}|x_t, u_t)$. To avoid this, we can simplify the problem by assuming that our agent's path statistics are already within a *local* variational neighborhood of the optimal statistics. We formalize this optimistic assumption by asserting that our agent's path probability densities are of the following form,

$$P^L_{u(t)}[x(t)] = \frac{1}{Z} \exp\Big[-\frac{1}{2}\int_{t_0}^{t} \dot{x}(\tau)^T \mathbf{C}^{-1}_{u(t)}[x(\tau)]\dot{x}(\tau)d\tau\Big], \tag{64}$$

where it is still the case that $S[P_{max}[x(t)]] \geq S[P^L_{u(t)}[x(t)]]$, and that the optimum can only be reached if and only if $P_{max}[x(t)] = P^L_{u(t)}[x(t)]$. Hence, by optimizing $S[P^L_{u(t)}[x(t)]]$ we merely change the direction from which our system approaches the true variational optimum of Eq. 39.

We proceed by analytically deriving the functional form of $S[P_{max}[x(t)]]$, and then using it to formulate our optimization of $S[P^L_{u(t)}[x(t)]]$. We begin by considering the path entropy along a finite path, $S[P_{max}[x_{1:N}]]$, where we can apply the chain rule of conditional entropies. For the reader's convenience, we state the chain rule as it is commonly formulated below:

$$S[P[x_{1:N}]] = \sum_{t=1}^{N} S[p(x_{t+1}|x_{1:t})]. \tag{65}$$

Then, applying this property directly onto $P_{max}[x_{1:N}]$ we have,

$$S[P_{max}[x_{1:N}]] = \sum_{t=1}^{N} S[p_{max}(x_{t+1}|x_t)] \propto \frac{1}{2}\sum_{t=1}^{N} \log\det \mathbf{C}[x_t], \tag{66}$$

where we made use of the Markov property to simplify our sum over conditional entropies, and then the analytical form of the entropy of a Gaussian distribution (up to a constant offset) to reach our final expression. Thus, realizing maximally diffusive trajectories merely requires synthesizing a controller $u(t)$ or policy $\pi(\cdot|\cdot)$ that satisfies $\mathbf{C}_{u(t)}[x^*] = \mathbf{C}_{\pi}[x^*] = \mathbf{C}[x^*]$ for all $x^* \in \mathcal{X}$, which can be done through optimization.

In this way, we can arrive at the MaxDiff RL objective presented in the main text, which is expressed in terms of an instantaneous reward function, $r(x_t, u_t)$. Omitting γ, the implemented MaxDiff RL objective is the following,

$$\underset{\pi}{\text{argmax}}\ E_{(x_{1:N},u_{1:N})\sim P_\pi}\left[\sum_{t=1}^{N} r(x_t,u_t) + \frac{\alpha}{2}\log\det \mathbf{C}_\pi[x_t]\right]. \tag{67}$$

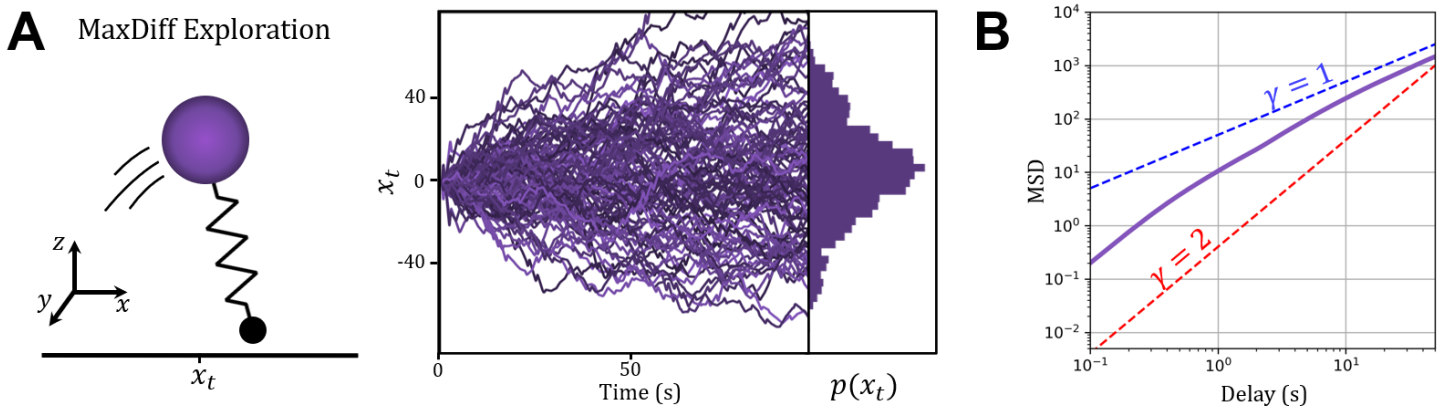
This objective is the one that we used to produce all empirical results in the main text. Here, we would like to make a few practical notes regarding its implementation. First, in practice we may not always have guarantees on the full-rankness of $\mathbf{C}[x^*]$, which can make its determinant evaluate to zero and create numerical stability issues. To remedy this, we may take advantage of another property of the log-determinant and instead optimize $\sum_{i=1}^{M}\log\lambda_i$, where the sum is taken over the leading M eigenvalues of $\mathbf{C}[x^*]$. However, it is important to note that this effectively restricts the exploration to an M-dimensional subspace of $\mathcal{X}$. Separately, we note that one can instead optimize the logarithm of the trace of $\mathbf{C}[x^*]$ as an approximation that drastically reduces the computational complexity of the determinant in high dimensional systems. However, this approximation only formally produces equivalent results to the log-determinant when system states vary independently from one another (i.e., when $\mathbf{C}[x^*]$ is diagonal), which is generally not the case. Nonetheless, it may be of help to a practitioner at the cost of some added distance to the assumptions underlying our formal guarantees.

Since the objective in Eq. 67 does not explicitly depend on $p(x_{t+1}|x_t, u_t)$, it provides us with a clear setting in which to frame model-free implementations of MaxDiff RL. Because our theoretical approach in Supplementary Note 2 assumes that there is a unique $\mathbf{C}[x^*]$ for all $x^* \in \mathcal{X}$, we can reinterpret $\frac{1}{2}\log\det\mathbf{C}[x^*]$ as an environmental property. In other words, because the function $S[x^*] = \frac{1}{2}\log\det\mathbf{C}[x^*]$ is a real-valued function of state that in principle does not require direct access to the state transition dynamics, we can think of $S[x^*]$ as a state-dependent property of the environment. By doing so, implementing model-free MaxDiff RL becomes very similar to existing state entropy maximization techniques in model-free RL [50–52]. The primary challenge is to learn a parametric estimator of $S[x^*]$—similar to those used in [50–52], or [53]—such that $\hat{S}_\theta[x^*] \approx S[x^*]$, where θ represents some vector of parameters (e.g., neural network weights). Then, given such an estimate model-free implementations are possible by augmenting the value function with $\hat{S}_\theta[x_t]$. Moreover, we note that our implementation of model-based MaxDiff RL makes use of data-driven estimates of $\mathbf{C}[x^*]$ during its optimization. While it may seem that evaluating $\mathbf{C}[x^*]$ still requires access to predictive system rollouts in a model-based fashion, $\mathbf{C}[x^*]$ can be empirically estimated from data moving backwards in time—in other words, $\mathbf{C}[x^*] = \int_{t_i-\Delta t}^{t_i} K_{XX}(\tau, t_i)d\tau$. However, we note that equality between the forwards and backwards data-driven estimates of $\mathbf{C}[x^*]$ is only guaranteed for stationary processes. Nonetheless, this shows that non-parametric model-free data-driven estimates of $\mathbf{C}[x^*]$ are also possible.

3.6 Example applications of MaxDiff trajectory synthesis

In this section, we implement MaxDiff trajectory synthesis across handful of applications outside of reinforcement learning that require both directed and undirected exploration. These should illustrate the sense in which our theoretical framework can extend beyond a particular algorithmic implementation, or even reinforcement learning as a problem setting. Moreover, here we will analyze the behavior of various dynamical systems made to follow maximally diffusive trajectories through the lens of statistical mechanics.

We begin by studying MaxDiff trajectory synthesis in the undirected exploration of a nontrivial control system—a spring-loaded inverted pendulum (SLIP) model. The SLIP model is a popular dynamic model of locomotion and encodes many important properties of human locomotion [55]. In particular, we will implement the SLIP model as in [56], where it is described as a 9-dimensional nonlinear nonsmooth control system. The SLIP model is shown in Supplementary Fig. 3(a) and consists of a "head" which carries its mass, and a "toe" which makes contact with the ground. Its state-space is defined by the 3D velocities and positions of its head and toe, or $x = [x_h, \dot{x}_h, y_h, \dot{y}_h, z_h, \dot{z}_h, x_t, y_t, q]^T$, where $q = \{c, a\}$ is a variable that



Supplementary Figure 3: **Maximally diffusive trajectories of a spring-loaded inverted pendulum (SLIP)**. **a**, The SLIP model (left panel) is a 9-dimensional nonlinear and nonsmooth second-order dynamical system, which is used as a popular model of human locomotion. (right panel) We choose this system because it is far from the ideal assumptions under which our theory is formulated, and yet its sample paths behave as we expect. The sample paths of the SLIP model with MaxDiff trajectories in the one dimensional space determined by its x-coordinate approximately match the statistics of pure Brownian motion in one dimension. **b**, Mean squared displacement (MSD) plots give the deviation of the position of an agent over time with respect to a reference position. We can distinguish between diffusion processes by comparing the growth of their MSD over time. In general, we expect them to follow a relationship described by $\text{MSD}(x) \propto t^{\gamma}$, where γ is an exponent that determines the different diffusion regimes (normal diffusion $\gamma = 1$, superdiffusion $1 < \gamma < 2$, ballistic motion $\gamma \geq 2$). As we can see, the behavior of the diffusing SLIP model is superdiffusive at short time-scales, but gradually becomes more like a standard diffusion process as we coarse-grain. Similar short-delay superdiffusion regimes have been observed in systems with nontrivial inertial properties [54], such as those of our macroscopic SLIP agent.

tracks whether the system is in contact with the ground or in the air. The SLIP dynamics are the following:

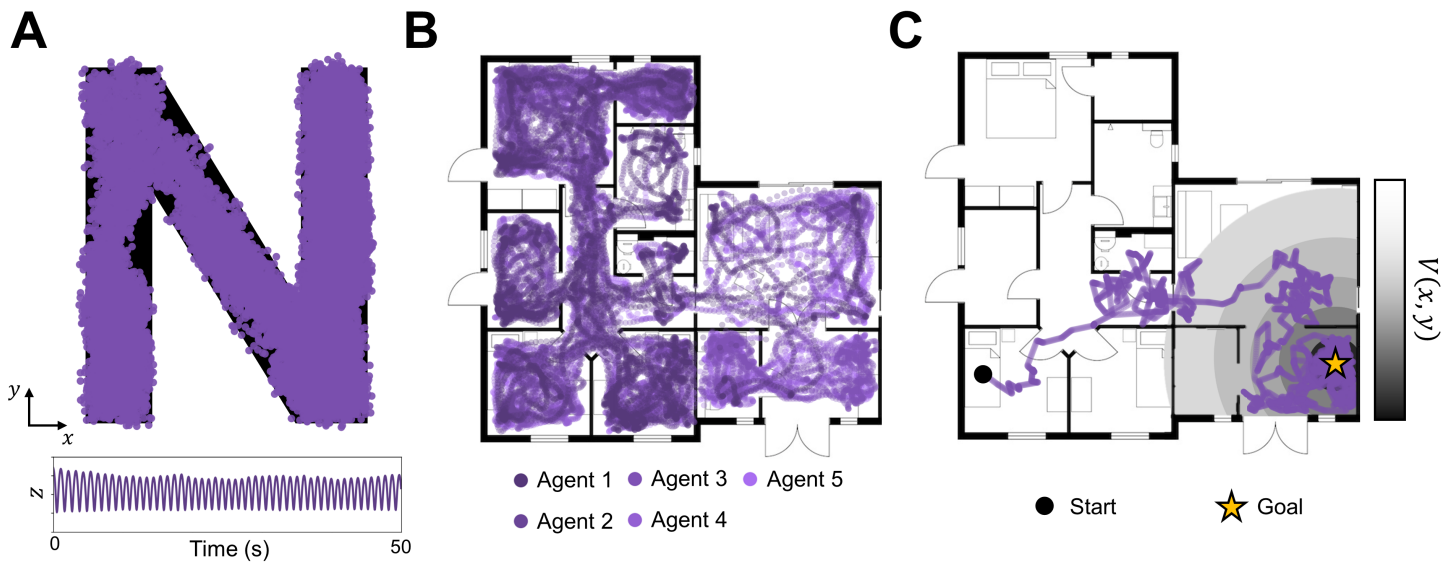
$$\dot{x} = f(x,u) = \begin{cases} f_c(x,u), & \text{if } l_c < l_0 \\ f_a(x,u), & \text{otherwise} \end{cases}, \; f_c(x,u) = \begin{bmatrix} \dot{x}_h \\ \frac{(k(l_0-l_s)+u_c)(x_h-x_t)}{ml_c} \\ \dot{y}_h \\ \frac{(k(l_0-l_c)+u_c)(y_h-y_t)}{ml_c} \\ \dot{z}_h \\ \frac{(k(l_0-l_c)+u_c)(z_h-z_t)}{ml_c} - g \\ 0 \\ 0 \end{bmatrix}, \; f_a(x,u) = \begin{bmatrix} \dot{x}_h \\ 0 \\ \dot{y}_h \\ 0 \\ \dot{z}_h \\ -g \\ \dot{x}_h + u_{t_x} \\ \dot{y}_h + u_{t_y} \end{bmatrix}, \tag{68}$$

where $f_c(x,u)$ captures the SLIP dynamics during contact with the ground, and $f_a(x,u)$ captures them while in the air. During contact the SLIP can only exert a force, u_c, by pushing along the axis of the spring, whose resting length is l_0 and its stiffness is k. During flight the SLIP is subject to gravity, g, and is capable of moving the x,y-position of its toe by applying u_{t_x} and u_{t_y}, respectively. To finish specifying the SLIP dynamics, and determine whether or not the spring is in contact with the ground, we define,

$$l_c = \sqrt{(x_h - x_t)^2 + (y_h - y_t)^2 + (z_h - z_G)^2},$$

which describes the distance along the length of the spring to the ground, and z_G is the ground height. Rather than explore diffusively in the entirety of the SLIP model's 9-dimensional state-space, we will first demand that it only explores a 1-dimensional space described by its x-coordinate, starting from an initial condition of $x(0) = 0$. We can think of this as a projection to a 1-dimensional subspace of the system, or equivalently as a coordinate transformation with a constant Jacobian matrix. We note that the system's nonsmoothness should break the path continuity constraint that our approach presumes to hold. However, since we use a coordinate transformation to formulate the exploration problem in terms of the system's x-coordinate we do not violate the assumptions of MaxDiff trajectory synthesis. This is because, while the system's velocities experience discontinuities, its position coordinates do not. In general, the use of coordinate transformations can extend the applicability of MaxDiff trajectory synthesis to even broader classes of systems than those claimed by our theoretical framework throughout Supplementary Note 2. However, this will require a formal analysis of the observability properties of maximally diffusive agents, which lies outside the scope of this work.

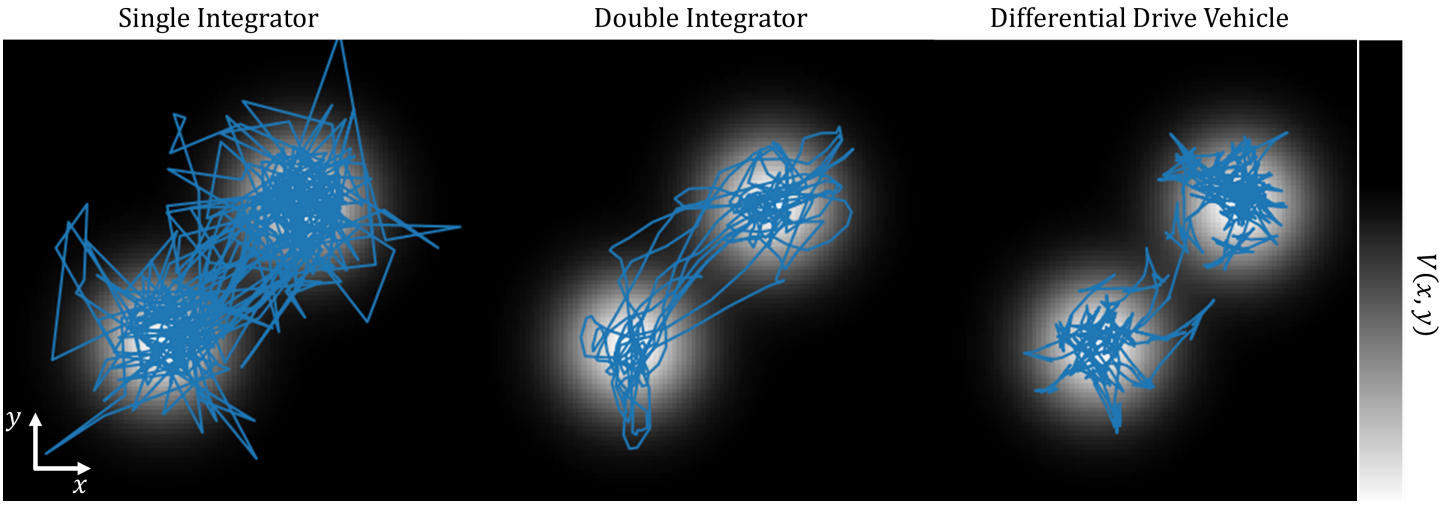
In order to realize maximally diffusive exploration, we make use of MPPI in conjunction with the MaxDiff trajectory synthesis objective in Eq. 63. In Supplementary Fig. 3 we illustrate the results of this process. Supplementary Fig. 3(a) depicts



Supplementary Figure 4: **SLIP maximally diffusive exploration in various settings**. **a**, Undirected maximally diffusive exploration in a constrained **N**-shaped environment. The boundaries of the environment, as well as safety constraints, are established through the use of control barrier functions, which enable safe and continuous maximally diffusive exploration without modifications to our approach. **b**, Undirected multiagent maximally diffusive exploration of more complex environment: a house's floor plan. Here, five agents with identical objectives perform maximally diffusive exploration. Because maximally diffusive exploration is ergodic, many tasks are inherently distributable between agents with linear scaling in complexity. **c**, Directed maximally diffusive exploration in a complex environment. Here, a single agent in a complex environment performs directed exploration in a potential that encodes a navigation goal.

the sample paths generated by the maximally diffusive exploration of the SLIP model's x-coordinate. The sample paths of the SLIP agent resemble the empirical statistics of Brownian particle paths despite the fact that the SLIP model is far from a non-inertial point mass. In Supplementary Fig. 3(b), we study the fluctuations of maximally diffusive exploration from the lens of statistical mechanics. Here, we analyze the mean squared displacement (MSD) statistics of undirected maximally diffusive exploration and compare to the statistics of standard and anomalous diffusion processes. MSD plots capture the deviations of a diffusing agent from some reference position over time. In standard diffusion processes, the relationship between MSD and time elapsed is linear on average. That is, we expect the squared deviation of a diffusing agent from its initial condition to grow linearly in proportion to the time elapsed (see blue line in Supplementary Fig. 3(b)). However, in general there exist other diffusion regimes characterized by the growth of MSD over time. These regimes are typically determined by fitting the exponent γ in $\text{MSD}(x) \propto t^{\gamma}$, where normal diffusion has $\gamma = 1$, superdiffusion has $1 < \gamma < 2$, and ballistic motion has $\gamma \geq 2$. The purple line in Supplementary Fig. 3(b) depicts the MSD statistics of the SLIP model. The diffusion generated by the SLIP model's maximally diffusive exploration has superdiffusive displacements over short-time scales owing to the the inertial properties of the system. However, as we consider longer time-scales, the behavior of the SLIP model becomes indistinguishable from standard diffusion processes with $\gamma = 1$. This difference in scaling exponents has been shown to be a general property of diffusion with inertial particles and should be expected in macroscopic systems [54].

Keeping with the SLIP dynamical system, in Supplementary Fig. 4 we study the behavior of MaxDiff trajectory synthesis across various standard robotics applications. In Supplementary Fig. 4(a), a single SLIP agent is performing undirected MaxDiff exploration within the bounds of an **N**-shaped environment. In this task, the agent must be able to explore its x-y plane by hopping along, without falling or exiting the bounds of the exploration domain. To ensure the SLIP model's safety, as well as establish the bounds of the environment, we made use of control barrier functions (CBFs) [57]—a standard technique in the field for guaranteeing safety. Then, to illustrate another application application of the ergodicity guarantees of our method, in Supplementary Fig. 4(b) we apply MaxDiff trajectory synthesis to multiagent exploration in a complex environment—a house floor plan—in conjunction with CBFs. Since maximally diffusive exploration is ergodic, the outcomes of a multiagent execution and a single agent execution are asymptotically identical. In this way, maximally diffusive exploration only incurs a linear scaling in computational complexity as a function of the number of agents. Finally, in Supplementary Fig. 4(c) we return to the single agent case to illustrate directed maximally diffusive exploration in the same complex environment as before. Here, a potential function encoding a goal destination is flat beyond a certain distance, which leads to undirected exploration initially. However, as the agent nears the goal, it can detect variations in the potential and follows its gradients



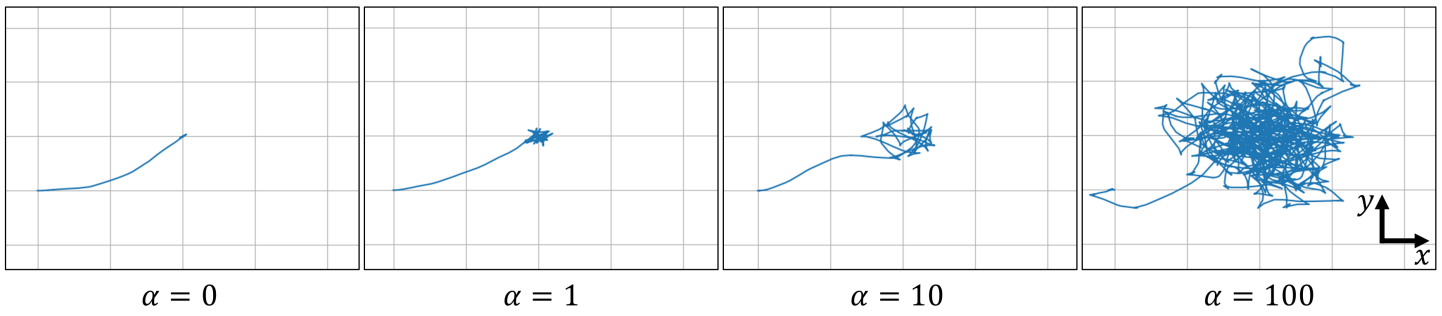
Supplementary Figure 5: **Directed maximally diffusive exploration of bimodal potential across systems**. (left panel) The single integrator is a linear system whose velocities are directly determined by the controller. Hence, its sample paths behave exactly as free Brownian particles in a potential. (middle panel) The double integrator is the second-order equivalent of the single integrator system. In this system, the controller inputs acceleration commands that the system then integrates subject to its inertial properties. Despite being an inertial system, its interactions with the potential approximately follow the behavior of a Brownian particle in a potential. (right panel) The differential drive vehicle is a car-like system with simple nonlinear and nonholonomic dynamics with more complex controllability properties. Nonetheless, when we subject the differential drive vehicle to directed maximally diffusive exploration it traverses the potential as desired.

diffusively towards the goal.

Now, we will highlight how the underlying properties of an agent's dynamics can affect the trajectories generated during maximally diffusive exploration. To this end, we consider a simple planar exploration task subject to a bimodal Gaussian potential ascribing a cost to system states far away from the distribution means. In Supplementary Fig. 5, we explore the planar domain with three different systems. First, exploration over the bimodal potential is shown with a single integrator system, which is a controllable first-order linear system. Since this system is effectively identical to a non-inertial point mass, its sample paths are formally the same as those of Brownian particles in a confining potential. In the middle panel of Supplementary Fig. 5, we consider a double integrator system, which is a controllable, linear, second-order system. However, for this system its diffusion tensor is degenerate because the noise only comes into the system as accelerations. Nonetheless, the system realizes ergodic coverage with respect to the underlying potential (in agreement with the theory of degenerate diffusion [58, 59]). Finally, we consider the differential drive vehicle, which is a simple first-order nonlinear dynamical system with nontrivial controllability properties. Yet, the differential drive vehicle realizes ergodic coverage in the plane, as predicted by the properties of maximally diffusive systems.

As a final look into the properties of directed maximally diffusive exploration, we examine the role that the temperature parameter α plays on the behavior of the agent in a simpler setting. To this end, we revisit the differential drive vehicle dynamics and make use of MPPI once again to optimize our objective. However, instead of a bimodal Gaussian potential, we consider a quadratic potential centered at the origin with the system initialized at $(x, y) = (-4, -2)$. Quadratic potentials such as these are routinely implemented as cost functions throughout robotics and control theory. In Supplementary Fig. 6, we depict the behavior of the system as a function of the temperature parameter. Initially, with the temperature set to zero the agent's paths are solely determined by the solution to the optimal control problem, smoothly driving towards the potential's minimum at the origin. Then, as we tune up α, we increase diffusivity of our agent's sample paths. While at $\alpha = 1$ the position of the system fluctuates very slightly at the bottom of the quadratic potential, at $\alpha = 100$ the agent diffuses around violently by overcoming its energetic tendency to stay at the bottom of the well. If we were to continue increasing α to larger and larger values, we would observe that directed maximally diffusive exploration would cease to be ergodic, as predicted by [36]. This occurs as a result of the strength of diffusive fluctuations (here set by our α parameter) dominating the magnitude of the drift induced by the potential's gradient. This is to say that for a given problem, system, and operator preferences, there should be a range of α values that best achieve the task.

Throughout this section we have illustrated how maximally diffusive exploration, as formulated in Eq. 63, satisfies the behaviors predicted by our theoretical framework. Moreover, we have motivated how MaxDiff trajectory synthesis can be applied in a variety of common robotic applications while simultaneously guaranteeing safety, ergodicity, and task distributability. Broadly speaking, incorporating maximally diffusive exploration into most optimal control or reinforcement



Supplementary Figure 6: **Varying the α parameter of directed MaxDiff exploration**. Here, we are making a differential drive vehicle explore a quadratic potential centered at the origin under varying choices of α modulating the strength of the diffusive exploration within the potential. As we increase α the strength of the diffusion increases as well, leading to greater exploration of the basin of attraction of the quadratic potential well.

learning frameworks should be simple—particularly in light of the effort we have put towards deriving optimization objectives realizable in a broad class of application domains.

4 Reinforcement learning implementation details

4.1 General

All simulated examples use the reward functions specified MuJoCo environments unless otherwise specified [60, 61]. Supplementary Table 1 provides a list of all hyperparameters used in all implementations of MaxDiff RL, NN-MPPI, and SAC, for each environment. All experiments were run for a total of 1 million environment steps with each epoch being comprised of 1000 steps. For multi-shot tests, the episode was reset upon satisfying a "done" condition or completing the number of steps in an epoch. For single-shot tests, the environment was never reset and each epoch only constituted a checkpoint for saving cumulative rewards during the duration of that epoch. All representations used ReLU activation functions, and 10 seeds were run for each configuration.

For all model-based examples (i.e., MaxDiff RL and NN-MPPI), the system dynamics are represented in the following form, $x_{t+1} = x_t + f(x_t, u_t)$, where the transition function $f(x_t, u_t)$ and reward function $r(x_t, u_t)$ are both modeled by fully-connected neural network representations. Both the reward function and transition function representations are optimized using Adam [62]. The network is regularized using the negative log-loss of a normal distribution where the variance, $\Sigma_{\text{model}} \in \mathbb{R}^{n \times n}$, is a hyperparameter that is simultaneously learned based on agent experience. The predicted reward utility is improved by the error between the predicted target and target reward equal to $\mathcal{L} = \|r_t + 0.95\ r(x_{t+1}, u_{t+1}) - r(x_t, u_t)\|^2$. The structure of this loss function is similar to those used in temporal-difference learning [63, 64]. The inclusion of the reward term from the next state and next action helps the algorithm learn in environments with rewards that do not strictly depend on the current state, as is the case with some MuJoCo examples.

For all model-free examples, we implement SAC to provide updates to our model-free policy. We use the hyperparameters for SAC specified by the parameters shared in [10], including the structure of the soft Q functions, but excluding the batch size parameter and the implemented policy's representation. Instead, we choose to match the batch size used during our model-based learning examples (i.e., with Maxdiff RL and NN-MPPI), and also introduce a simpler policy representation. As an alternative to the representation in [10], our policy is represented by a normal distribution parametrized by a mean function defined as a fully-connected neural network.

Reinforcement learning experiments were run on an Intel® Xeon(R) Platinum 8380 CPU @ 2.30GHz x 160 server running Ubuntu 18.04 and Python 3.6 (`pytorch` 1.7.0 and `mujoco_py` 2.0). This hardware was loaned by the Intel Corporation, whose technical support we acknowledge. Finally, we note that our code repository contains a Dockerfile to facilitate validation of our results across platforms without the need to worry about package versioning.

4.2 Point mass

The goal of the point mass environment is to learn to move to the origin of a 2D environment. This is a custom environment in which the point mass dynamics are simulated as a 2D double integrator with states $[x, y, \dot{x}, \dot{y}]$ and actions $[\ddot{x}, \ddot{y}]$. Each episode is initialized at state $[-1, -1, 0, 0] + \epsilon$ where $\epsilon \sim \mathcal{N}(0, 0.01)$. The reward function is specified in terms of location in the environment $r = -(x^2 + y^2)$. For multi-shot tests, the episode was terminated if the point mass exceeded a boundary defined as a square at $x, y = \pm 5$. The simulation uses RK-4 integration with a time step of 0.1.

4.3 Swimmer

The goal of the swimmer environment is to learn a gait to move forward in a 2D environment as quickly as possible. These tests use the "v3" variant of the OpenAI Gym MuJoCo Swimmer Environment, which includes all configuration states in the observation generated at each step. For the "heavy-tailed" tests, the default `xml` swimmer file is used, which includes a 3-link body with identical links. For the "light-tailed" tests, we modify the density of the "tail" link to be 10 times lighter than other two links. The default link density in the swimmer is 1000 and modified tail density is 100. There is no "done" condition for this environment.

4.4 Ant

The goal of the ant environment is to learn a gait to move forward in a 3D environment as quickly as possible. These tests use the "v3" variant of the OpenAI Gym MuJoCo Ant Environment, which includes all configuration states in the observation generated at each step and includes no contact states. The control cost, contact cost, and healthy reward weights are all set to zero, so the modified reward function only depends on the change in the x-position during the duration of the step (with positive reward for progress in the positive x-direction). We also modified the "done" condition to make it possible for the ant to recover from falling. The "done" condition is triggered if the ant has been upside down for 1 second, and the ant is considered "upside down" if the torso angle that is nominally perpendicular to the ground exceeds 2.7 radians.

4.5 Half-cheetah

The goal of the half-cheetah environment is to learn a gait to move forward by applying torques on the joints in a 2D vertical plane. These tests use the "v3" variant of the OpenAI Gym MuJoCo Half-Cheetah Environment, which includes all configuration states in the observation generated at each step. There is no "done" condition for this environment.

Supplementary tables

Algorithm	Hyperparameter		Toy Problem 2D Point mass	MuJoCo Gym (v3) Swimmer	Ant	Half-cheetah
All	State Dim		4	10	29	18
	Action Dim		2	2	8	6
	Learning Rate		0.0005	0.0003	0.0003	0.0003
	Batch Size		128	128	256	256
SAC	Policy Layers		128×128	256×256	$512 \times 512 \times 512$	256×256
	Discount (γ)		0.99	0.99	0.99	0.99
	Smoothing coefficient (τ)		0.01	0.005	0.005	0.005
	Reward Scale		0.25	100	5	5
NN-MPPI/ MaxDiff RL (Planning)	Model Layers		128×128	200×200	$512 \times 512 \times 512$	200×200
	Horizon		30	40	20	10
	Discount (γ)		0.95	0.95	0.95	0.95
	Multi	Samples	500	500	1000	500
		Lambda	0.5	0.5	0.5	0.5
	SS	Samples	NA	1000	1000	1000
		Lambda		0.1	0.5	0.5
MaxDiff RL (Exploration)	Multi	Alpha	5	1,5,10,50, 100,500,1000	15	5
		Dimensions	$\lfloor x, y, \dot{x}, \dot{y} \rfloor$	$\lfloor x, y, \dot{x}, \dot{y} \rfloor$	$\lfloor x, y, z \rfloor$	$\lfloor x, y, \dot{x}, \dot{y} \rfloor$
		Weights	$[1, 1, 0.01, 0.01]$	$[1, 1, 0.05, 0.05]$	$[1, 1, 0.005]$	$[1, 1, 0.05, 0.05]$
	SS	Alpha	NA	50	15	5
		Dimensions		$\lfloor x, y, \dot{x}, \dot{y} \rfloor$	$\lfloor x, y, \dot{x}, \dot{y} \rfloor$	$\lfloor x, y, \dot{x}, \dot{y} \rfloor$
		Weights		$[1, 1, 0.05, 0.05]$	$[1, 1, 0.05, 0.05]$	$[1, 1, 0.05, 0.05]$

Supplementary Table 1: **Simulation hyperparmeters for paper results.** "Multi" parameters only apply to multi-shot runs, and "SS" parameters only apply to single-shot runs. All weights are diagonal matrices with the values specified.

		Comparison			
		NN-MPPI		SAC	
Task		P-value	$P < 0.05$	P-value	$P < 0.05$
Point Mass	$\beta = 1$	< 0.001	True	< 0.001	True
	$\beta = 0.1$	< 0.001	True	< 0.001	True
	$\beta = 0.01$	< 0.001	True	< 0.001	True
	$\beta = 0.001$	< 0.001	True	< 0.001	True
Swimmer	Light Multi	< 0.001	True	< 0.001	True
	Light SS (Return)	0.0131	True	0.0034	True
	Light SS (Distance)	< 0.001	True	< 0.001	True
	Heavy Multi	< 0.001	True	< 0.001	True
	Light-to-Heavy Multi	< 0.001	True	< 0.001	True
	Heavy-to-Light Multi	< 0.001	True	< 0.001	True
Ant	Multi	0.8343	False	< 0.001	True
	SS (Return)	0.0154	True	< 0.001	True
	SS (Distance)	< 0.001	True	< 0.001	True
Half-cheetah	Multi	< 0.001	True	< 0.001	True
	SS (Return)	< 0.001	True	< 0.001	True
	SS (Distance)	< 0.001	True	< 0.001	True

Supplementary Table 2: **Results of statistical comparisons between MaxDiff RL and alternatives.** Across all learning experiments in the manuscript, differences between MaxDiff RL and comparisons are statistically significant ($P < 0.05$) according to an unpaired two-sided Welch's t-test implementation in SciPy [65], except for one. However, since the Ant environment breaks ergodicity, we do not expect improvements over MaxEnt RL in multi-shot settings. "Multi" indicates a multi-shot experiment and "SS" indicates a single-shot experiment. For Multi experiments, statistical significance was determined by evaluating each final policy across 100 episodes. For SS experiments, significance was evaluated in two ways: first according to the terminal windowed return, and second according to the terminal distance traveled in each spatial navigation task. For more details, we refer readers to the Methods section of the main text, where our statistical methodology is explained and justified in detail. We note that P-values below 0.001 are reported as < 0.001.

Supplementary movies

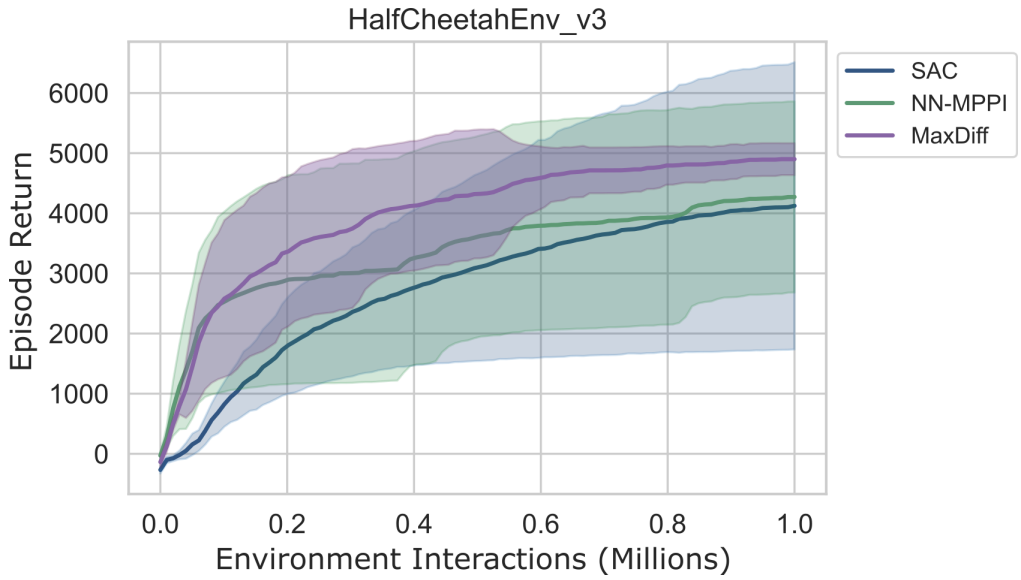
Movie 1: **Effect of temperature parameter on MaxDiff RL.** Here, we depict an application of MaxDiff RL to MuJoCo's swimmer environment. To explore the role of the parameter α on the performance of agents, we vary it across three orders of magnitude and observe its effect on system behavior (10 seeds each). Tuning α is crucial because it can determine whether or not the underlying agent is ergodic.

Movie 2: **Robustness of MaxDiff RL across random seeds.** Here, we depict an application of MaxDiff RL to MuJoCo's swimmer environment, comparing with alternative state-of-the-art MaxEnt RL algorithms, NN-MPPI and SAC. We observe that the performance of MaxDiff RL achieves state-of-the-art performance and does not vary across seeds, which is a formal property of our framework. We test across two different system conditions: one with a light-tailed and more controllable swimmer, and one with a heavy-tailed and less controllable swimmer (10 seeds each).

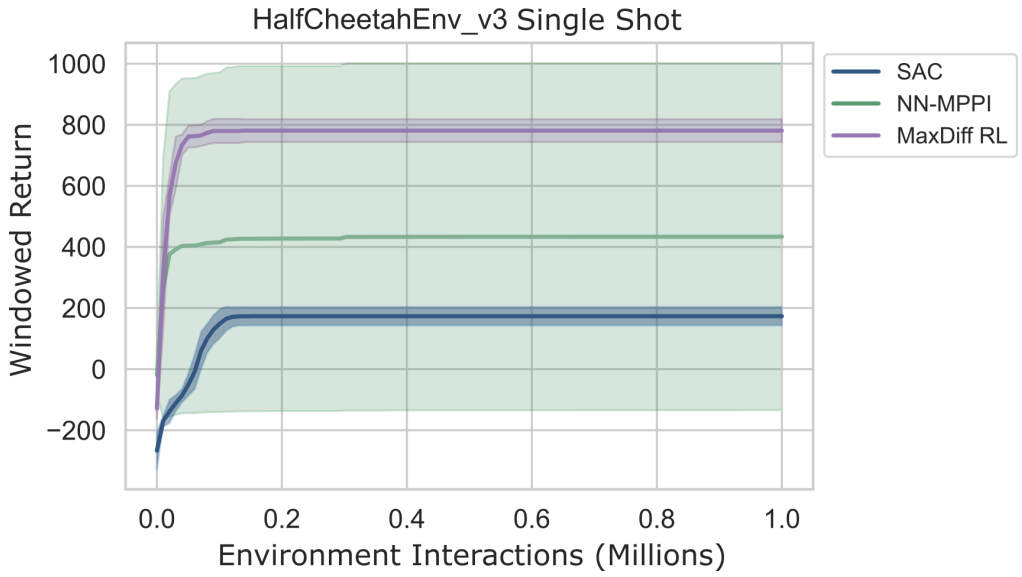
Movie 3: **Zero-shot generalization of MaxDiff RL across embodiments.** Here, we depict an application of MaxDiff RL to MuJoCo's swimmer environment. We implement a transfer learning experiment in which neural representations are learned on a system with a given set of physical properties, and then are deployed on a system with different physical properties. We find that unlike alternative approaches, MaxDiff RL remains task capable across agent embodiments.

Movie 4: **Single-shot learning in MaxDiff RL agents.** Here, we depict an application of MaxDiff RL to MuJoCo's swimmer environment under a significant modification. Agents are unable to reset their environment, which requires all algorithms to learn to solve the task in a single deployment. First, we show representative snapshots of agents using representations learned in single-shot deployments, and observe that MaxDiff RL still achieves state-of-the-art performance that is robust to seeds. Then, for MaxDiff RL we also show a complete playback of an individual single-shot learning trial. We stagger the playback such that the first swimmer covers environment steps 1-2000, the next one 2001-4000, and so on for a total of 20,000 environment steps. In doing so, we visualize the single-shot learning process in real time.

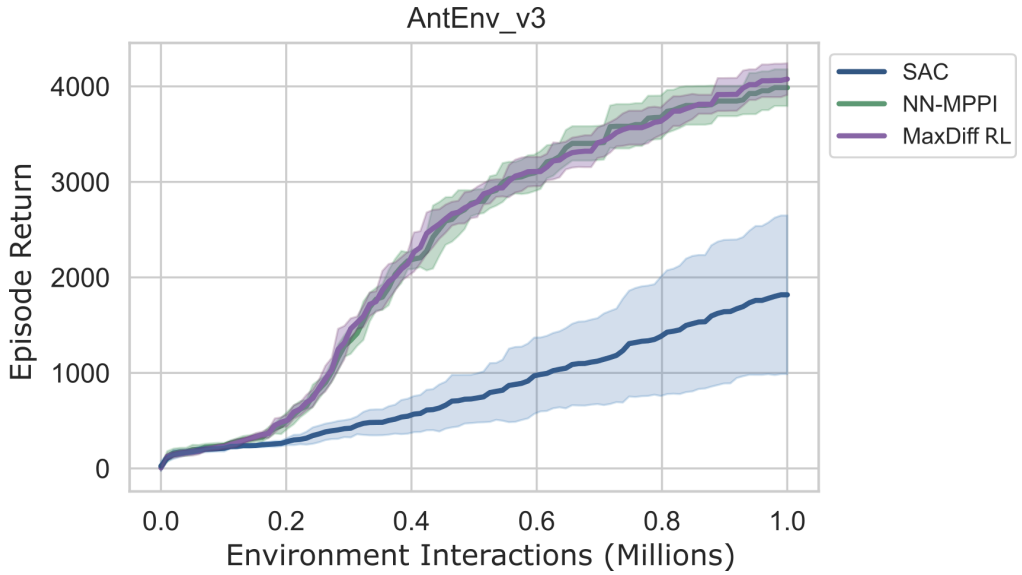
Supplementary figures



Supplementary Figure 7: **Results of the half-cheetah benchmark.** This figure compares the performance of MaxDiff RL to NN-MPPI and SAC on MuJoCo's HalfCheetahEnv v3 in multi-shot. Since the half-cheetah can fall into an irreversible state (i.e., flipping upside down) this environment breaks the assumptions of MaxDiff RL. Nonetheless, we still achieve state-of-the-art performance with substantially less variance than alternative algorithms. For all reward curves, the shaded regions correspond to the standard deviation from the mean across 10 seeds.



Supplementary Figure 8: **Results of the half-cheetah benchmark in single-shot.** This figure compares the performance of MaxDiff RL to NN-MPPI and SAC on MuJoCo's HalfCheetahEnv v3 in single-shot. Since the half-cheetah can fall into an irreversible state (i.e., flipping upside down) this environment breaks the assumptions of MaxDiff RL. Nonetheless, we still succeed at the task with substantially less variance than alternative algorithms. For all reward curves, the shaded regions correspond to the standard deviation from the mean across 10 seeds.



Supplementary Figure 9: **Results of the ant benchmark.** This figure compares the performance of MaxDiff RL to NN-MPPI and SAC on MuJoCo's AntEnv v3 in multi-shot. Just as with our main text single-shot example, the ant environment breaks ergodicity, which pushes MaxDiff RL outside of the domain of its assumptions. Nonetheless, MaxDiff RL remains state-of-the-art with comparable performance to NN-MPPI. This is to be expected because in the worst case scenario where MaxDiff's additional entropy term in the objective has no effect on agent outcomes, our implementation of MaxDiff RL is identical to NN-MPPI. For all reward curves, the shaded regions correspond to the standard deviation from the mean across 10 seeds.

Supplementary references

1. Dixit, P. D. *et al.* Perspective: Maximum caliber is a general variational principle for dynamical systems. *The Journal of Chemical Physics* **148,** 010901 (2018).
2. Kapur, J. N. *Maximum Entropy Models in Science and Engineering* ISBN: 9788122402162 (Wiley, 1989).
3. LeCun, Y., Bengio, Y. & Hinton, G. Deep Learning. *Nature* **521,** 436–444 (2015).
4. Taylor, A. T., Berrueta, T. A. & Murphey, T. D. Active learning in robotics: A review of control principles. *Mechatronics* **77,** 102576 (2021).
5. Ibarz, J. *et al.* How to train your robot with deep reinforcement learning: Lessons we have learned. *The International Journal of Robotics Research* **40,** 698–721 (2021).
6. Miki, T. *et al.* Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics* **7,** eabk2822 (2022).
7. Bloesch, M. *et al.* Towards Real Robot Learning in the Wild: A Case Study in Bipedal Locomotion. *Proceedings of the 5th Conference on Robot Learning. Proceedings of Machine Learning Research* **164,** 1502–1511 (2022).
8. Auer, P., Cesa-Bianchi, N. & Fischer, P. Finite-Time Analysis of the Multiarmed Bandit Problem. *Machine Learning* **47,** 235–256 (2002).
9. Haarnoja, T., Tang, H., Abbeel, P. & Levine, S. Reinforcement Learning with Deep Energy-Based Policies. *Proceedings of the International Conference on Machine Learning (ICML)* **70,** 1352–1361 (2017).
10. Haarnoja, T., Zhou, A., Abbeel, P. & Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *Proceedings of the International Conference on Machine Learning (ICML)* **80,** 1861–1870 (2018).
11. So, O., Wang, Z. & Theodorou, E. A. *Maximum Entropy Differential Dynamic Programming* in *2022 IEEE International Conference on Robotics and Automation (ICRA)* (2022), 3422–3428.
12. Sontag, E. D. *Mathematical Control Theory: Deterministic Finite Dimensional Systems* ISBN: 9781461205777 (Springer, 2013).
13. Hespanha, J. P. *Linear Systems Theory: Second Edition* ISBN: 9780691179575 (Princeton University Press, 2018).
14. Sontag, E. D. in *Mathematical System Theory: The Influence of R. E. Kalman* 453–462 (Springer, 1991). ISBN: 9783662085462.
15. Cortesi, F. L., Summers, T. H. & Lygeros, J. *Submodularity of energy related controllability metrics* in *2014 IEEE Conference on Decision and Control (CDC)* (2014), 2883–2888.
16. Summers, T. H., Cortesi, F. L. & Lygeros, J. On Submodularity and Controllability in Complex Dynamical Networks. *IEEE Transactions on Control of Network Systems* **3,** 91–101 (2016).
17. Kardar, M. *Statistical Physics of Fields* (Cambridge University Press, 2007).
18. Kashima, K. Noise Response Data Reveal Novel Controllability Gramian for Nonlinear Network Dynamics. *Scientific Reports* **6,** 27300 (2016).
19. Risken, H. in *The Fokker-Planck Equation* 63–95 (Springer, 1996). ISBN: 978-3-642-96807-5.
20. Mitra, D. *W* matrix and the geometry of model equivalence and reduction. *Proceedings of the Institution of Electrical Engineers* **116,** 1101–1106 (1969).
21. Tsiamis, A. & Pappas, G. J. Linear Systems can be Hard to Learn. *2021 60th IEEE Conference on Decision and Control (CDC),* 2903–2910 (2021).
22. Tsiamis, A., Ziemann, I. M., Morari, M., Matni, N. & Pappas, G. J. *Learning to Control Linear Systems can be Hard* in *Proceedings of 35th Conference on Learning Theory (COLT)* **178** (2022), 3820–3857.
23. Øksendal, B. *Stochastic Differential Equations: An Introduction with Applications* ISBN: 9783642143946 (Springer Berlin Heidelberg, 2010).
24. Feynman, R. P., Hibbs, A. R. & Styer, D. F. *Quantum Mechanics and Path Integrals* (Dover Publications, 2010).
25. Sethna, J. P. *Statistical Mechanics: Entropy, Order Parameters, and Complexity* ISBN: 9780192634535 (OUP Oxford, 2021).
26. Grendar, M. Entropy and Effective Support Size. *Entropy* **8,** 169–174 (2006).
27. Jaynes, E. T. Information Theory and Statistical Mechanics. *Phys. Rev.* **106,** 620–630 (1957).

28. Chvykov, P. *et al.* Low rattling: A predictive principle for self-organization in active collectives. *Science* **371,** 90–95 (2021).
29. Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming* ISBN: 9781118625873 (Wiley, 2014).
30. Michalet, X. & Berglund, A. J. Optimal diffusion coefficient estimation in single-particle tracking. *Physical Review E* **85,** 061916 (2012).
31. Boyer, D., Dean, D. S., Mejía-Monasterio, C. & Oshanin, G. Optimal estimates of the diffusion coefficient of a single Brownian trajectory. *Physical Review E* **85,** 031136 (2012).
32. Thrun, S. B. *Efficient Exploration in Reinforcement Learning* tech. rep. (Carnegie Mellon University, 1992).
33. Miller, L. M., Silverman, Y., MacIver, M. A. & Murphey, T. D. Ergodic Exploration of Distributed Information. *IEEE Transactions on Robotics* **32,** 36–52 (2016).
34. Mavrommati, A., Tzorakoleftherakis, E., Abraham, I. & Murphey, T. D. Real-Time Area Coverage and Target Localization Using Receding-Horizon Ergodic Exploration. *IEEE Transactions on Robotics* **34,** 62–80 (2018).
35. Rawlik, K., Toussaint, M. & Vijayakumar, S. On Stochastic Optimal Control and Reinforcement Learning by Approximate Inference. *Proceedings of Robotics: Science and Systems (RSS),* 353–361 (2012).
36. Wang, X., Deng, W. & Chen, Y. Ergodic properties of heterogeneous diffusion processes in a potential well. *The Journal of Chemical Physics* **150,** 164121 (2019).
37. Nesterov, Y. E. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Dokl. Akad. Nauk SSSR* **269,** 543–547 (1983).
38. Qian, N. On the momentum term in gradient descent learning algorithms. *Neural Networks* **12,** 145–151 (1999).
39. Attouch, H. & Cabot, A. Asymptotic stabilization of inertial gradient dynamics with time-dependent viscosity. *Journal of Differential Equations* **263,** 5412–5458 (2017).
40. Attouch, H., Chbaniand, Z. & Riahi, H. Rate of convergence of the Nesterov accelerated gradient method in the subcritical case $\alpha \leq 3$. *ESAIM: COCV* **25,** 2 (2019).
41. Todorov, E. *Linearly-solvable Markov decision problems* in *Advances in Neural Information Processing Systems (NeurIPS)* **19** (MIT Press, 2007), 1369–1376.
42. Todorov, E. Efficient computation of optimal actions. *Proceedings of the National Academy of Sciences* **106,** 11478–11483 (2009).
43. Eysenbach, B. & Levine, S. Maximum Entropy RL (Provably) Solves Some Robust RL Problems. *Proceedings of the International Conference on Learning Representations (ICLR)* (2022).
44. Strehl, A. L., Li, L., Wiewiora, E., Langford, J. & Littman, M. L. PAC model-free reinforcement learning. *Proceedings of the International Conference on Machine learning (ICML),* 881–888 (2006).
45. Strehl, A. L., Li, L. & Littman, M. L. Reinforcement Learning in Finite MDPs: PAC Analysis. *Journal of Machine Learning Research* **10** (2009).
46. Chen, A., Sharma, A., Levine, S. & Finn, C. *You Only Live Once: Single-Life Reinforcement Learning* in *Advances in Neural Information Processing Systems (NeurIPS)* **35** (2022), 14784–14797.
47. Lu, K., Grover, A., Abbeel, P. & Mordatch, I. *Reset-Free Lifelong Learning with Skill-Space Planning* in *Proceedings of the International Conference on Learning Representations (ICLR)* (2021).
48. Hairer, M. *Lecture notes on Ergodic Properties of Markov Chains* July 2018.
49. Moore, C. C. Ergodic theorem, ergodic theory, and statistical mechanics. *Proceedings of the National Academy of Sciences* **112,** 1907–1911 (2015).
50. Lee, L. *et al. Efficient Exploration via State Marginal Matching* in *Workshop on Task-Agnostic Reinforcement Learning at the International Conference on Machine Learning (ICLR)* (2020).
51. Seo, Y. *et al. State entropy maximization with random encoders for efficient exploration* in *Proceedings of the 38th International Conference on Machine Learning (ICML)* (2021), 9443–9454.
52. Mutti, M., Pratissoli, L. & Restelli, M. *Task-agnostic exploration via policy gradient of a non-parametric state entropy estimate* in *Proceedings of the AAAI Conference on Artificial Intelligence* **35** (2021), 9028–9036.
53. Prabhakar, A. & Murphey, T. Mechanical intelligence for learning embodied sensor-object relationships. *Nature Communications* **13,** 4108 (2022).

54. Scholz, C., Jahanshahi, S., Ldov, A. & Löwen, H. Inertial delay of self-propelled particles. *Nature Communications* **9,** 5156 (2018).

55. Srinivasan, M. & Ruina, A. Computer optimization of a minimal biped model discovers walking and running. *Nature* **439,** 72–75 (2006).

56. Ansari, A. R. & Murphey, T. D. Sequential Action Control: Closed-Form Optimal Control for Nonlinear and Nonsmooth Systems. *IEEE Transactions on Robotics* **32,** 1196–1214 (2016).

57. Ames, A., Grizzle, J. & Tabuada, P. *Control Barrier Function based Quadratic Programs with Application to Adaptive Cruise Control* in *2014 IEEE Conference on Decision and Control (CDC)* (2014).

58. Kliemann, W. Recurrence and invariant measures for degenerate diffusions. *Annals of Probability* **15,** 690–707 (1987).

59. Bou-Rabee, N. & Owhadi, H. Ergodicity of Langevin Processes with Degenerate Diffusion in Momentums. *International Journal of Pure and Applied Mathematics* **45,** 475–490 (2008).

60. Brockman, G. *et al.* OpenAI Gym. *arXiv preprint arXiv:1606.01540* (2016).

61. Todorov, E., Erez, T. & Tassa, Y. *MuJoCo: A physics engine for model-based control* in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2012), 5026–5033.

62. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

63. Boyan, J. A. *Least-squares temporal difference learning* in *Proceedings of the International Conference on Machine Learning (ICML)* (1999), 49–56.

64. Precup, D., Sutton, R. S. & Dasgupta, S. *Off-policy temporal-difference learning with function approximation* in *Proceedings of the International Conference on Machine Learning (ICML)* (2001), 417–424.

65. Virtanen, P. *et al.* SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **17,** 261–272 (2020).