Maximum Diffusion Reinforcement Learning

Thomas A. Berrueta* **Allison Pinosky** **Todd D. Murphey**
Center for Robotics and Biosystems
Northwestern University
Evanston, IL 60208
{tberrueta, apinosky}@u.northwestern.edu t-murphey@northwestern.edu

Abstract

The assumption that data are independent and identically distributed underpins all machine learning. When data are collected sequentially from agent experiences this assumption does not generally hold, as in reinforcement learning. Here, we derive a method that overcomes these limitations by exploiting the statistical mechanics of ergodic processes, which we term maximum diffusion reinforcement learning. By decorrelating agent experiences, our approach provably enables agents to learn continually in single-shot deployments regardless of how they are initialized. Moreover, we prove our approach generalizes well-known maximum entropy techniques, and show that it robustly exceeds state-of-the-art performance across popular benchmarks. Our results at the nexus of physics, learning, and control pave the way towards more transparent and reliable decision-making in reinforcement learning agents, such as locomoting robots and self-driving cars.

1 Introduction

Deep reinforcement learning (RL) is a powerful and flexible decision-making framework based on the experiences of artificial agents. From controlling nuclear fusion reactors [1] to besting Olympic curling champions [2] and StarCraft grandmasters [3], deep RL agents have achieved remarkable feats when they are able to exhaustively explore how their actions impact the state of their environment. Despite its impressive achievements, deep RL suffers from limitations preventing its widespread deployment in the real world: its performance varies across initial conditions, its sample inefficiency demands the use of simulators, and its agents struggle to learn outside of episodic problem structures [4–6]. At the heart of these shortcomings lies a violation of the assumption that data are independent and identically distributed (*i.i.d.*), which underlies all of deep learning. While deep learning requires *i.i.d.* data, the experiences of RL agents are unavoidably sequential and correlated. It is no wonder, then, that many of deep RL's most impactful advances have sought to overcome precisely this roadblock [7–10].

Over the past decade, researchers have started to converge onto an understanding that destroying temporal correlations is essential to agent performance. In offline RL, where learning occurs by sampling from a fixed database of agent experiences, the development of experience replay was a major breakthrough [11]. Experience replay and its many variants [12–14] found that sampling agent experiences in random batches can reduce temporal correlations, resulting in large performance gains across tasks and algorithms [15–17]. This simple insight—merely sampling agent experiences out of order—led to one of deep RL's landmark triumphs, achieving superhuman performance in Atari video game benchmarks [7]. Nevertheless, overcoming the effect of strong temporal correlations cannot be accomplished with sampling alone. Correlations must be destroyed during data acquisition as well,

*Corresponding author. Code and data are available at github.com/MurpheyLab/MaxDiffRL. Supplementary movies are available in the following YouTube Playlist.

Preprint. Under review.

as online RL techniques have attempted to do. In this regard, maximum entropy (MaxEnt) RL has emerged as a key advance [18–26]. These methods seek to destroy correlations by maximizing the entropy of an agent's policy. In doing so, MaxEnt RL techniques have been able to achieve better exploration and more robust performance [27]. However, does maximizing the entropy of an agent's policy actually decorrelate their 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 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 properties, diffusion, and learning, we prove that MaxDiff RL agents learn in single-shot deployments regardless of how they are initialized. We additionally prove that MaxDiff RL agents exhibit seed-invariance, which enables robust 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 deep RL agents.

2 Results

2.1 Temporal correlations hinder performance

Whether temporal correlations can be avoided depends on the properties of the underlying agent being controlled. Completely destroying correlations between an agent's state transitions requires the ability to discontinuously jump from state to state without continuity of experience. For some RL agents, this poses no issue. Particularly in settings where agents are disembodied, there may be nothing preventing effective exploration through jumps between uncorrelated states. This is one of the reasons why deep RL recommender systems have been successful in a broad range of applications, such as YouTube video suggestions [28–30]. However, continuity of experience is an essential element of 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 or self-driving cars overcoming the impact of temporal correlations presents a major challenge [6].

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. 1(a), corresponding to a double integrator system with parametrized momentum anisotropy. The task requires learning reward, dynamics, and policy models from scratch in order to move a planar point mass from a fixed initial position to a goal location. The true linear dynamics are simple enough to explicitly write down, which allows us to rigorously study temporal correlations in the agent's state transitions through the lens of controllability. Controllability is a formal property of control systems that describes their ability to reach arbitrary states in an environment [31, 32]. In linearizable systems, state transitions become pathologically correlated when they are uncontrollable. However, when the agent is controllable these correlations can be overcome, at least in principle. While the relationship between controllability and temporal correlations has been studied for decades [33], it is only recently that researchers have begun to study its impact on learning processes [34–36].

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 horizontally on the x-axis (Fig. 1(a)). When $\beta = 0$ the system is uncontrollable and can only translate vertically along the y-axis, which illustrates the sense in which our agent's state transitions become pathologically correlated. While the system is formally controllable for all non-zero β, its reachable states can only satisfy the exploration statistics specified by its action distribution when it is equal to 1 (see Supplementary Figure 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) [37] and soft actor-critic (SAC) [9], 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. 1(b)). However, as $\beta \to 0$ the performance of NN-MPPI and SAC degrades parametrically (Fig. 1(c)), up until the point that neither

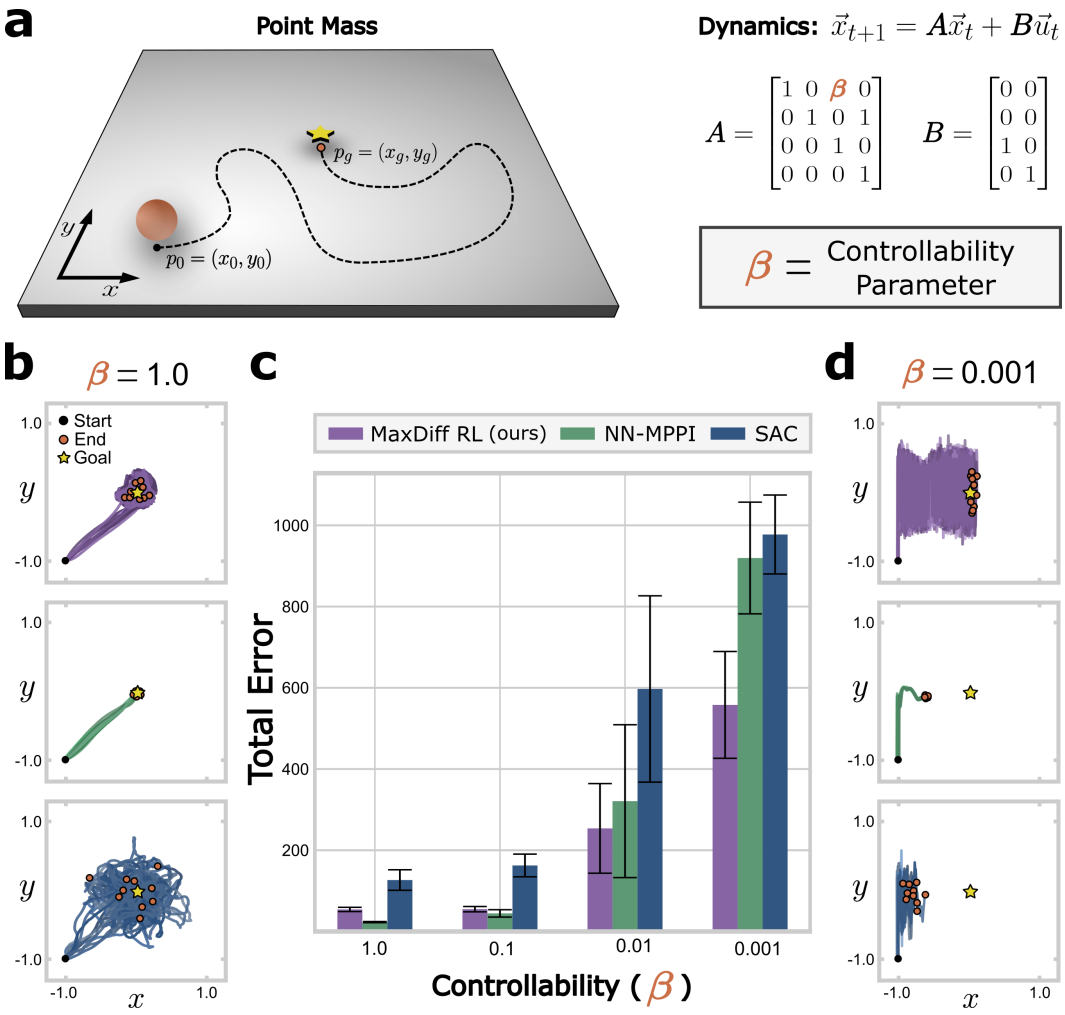


Figure 1: **Temporal correlations break the state-of-the-art in RL.** Controllability is a property of control systems that can determine how correlated state transitions are in linearizable systems (Supplementary Note 1.1). **a**, planar point mass system whose dynamics are simple enough to write down explicitly and whose policy admits a globally optimal analytical solution. The system's 4-dimensional state space is comprised of 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 maximum diffusion (MaxDiff) RL framework (see Supplementary Note 3 for implementation details). **b, d**, Representative snapshots of MaxDiff RL, NN-MPPI, and SAC agents (top to bottom) in well-conditioned ($\beta = 1$) and poorly-conditioned ($\beta = 0.001$) 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, while MaxDiff RL remains task-capable (10 seeds each).

algorithm can solve the task, as shown in Fig. 1(d). 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 overcome correlations between state transitions can prevent effective exploration, severely impacting the performance of deep RL agents. As Fig. 1(d) 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 (see Supplementary Note 1.1). This is the case despite the fact that NN-MPPI and SAC are both MaxEnt RL algorithms [9, 38], 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, i.e., their state sequences (see purple curves in Fig. 1(b-d)), as we discuss in the following section.

2.2 Maximum diffusion exploration and learning

Due to its history in the study of multi-armed bandits, most methods in the field of RL presuppose that taking random actions produces effective exploration [39, 40]. Even sophisticated techniques like MaxEnt RL implicitly rely on this assumption. Rather than sampling actions from a fixed uniform or Gaussian distribution, MaxEnt RL algorithms seek to maximize the entropy of a learned action distribution (i.e., a policy) in hopes of decorrelating agent experiences and improving exploration outcomes. However, as we have illustrated in the previous section, whether this is actually possible depends on the agent's controllability properties and the temporal correlations these spontaneously induce in their experiences (see Fig. 2(c) and Supplementary Note 1.1). To overcome these limitations, in this work we decorrelate agent experiences as opposed to their action sequences, which forms the starting point to our derivation of the MaxDiff RL framework.

Prior to synthesizing policies that try to decorrelate agent experiences, we start by asking what is the most decorrelated that agent experiences can get to begin with? To answer this question we draw from the statistical physics literature on maximum caliber [41–43], which generalizes the variational principle of maximum entropy [44] to distributions over trajectories or paths of agent states or experiences, $x(t)$, which we take to be continuous in time for the purposes of our derivation. Using this framework, we may derive a probability distribution over agent paths, $P[x(t)]$, by optimizing an entropy functional, $S[P[x(t)]]$. The optimal distribution, $P_{max}[x(t)]$, would describe the statistics of the least correlated agent paths, but its specific form and properties depend on how the variational optimization is constrained. In the absence of trajectory constraints, agents can 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 in the real world or in simulation (Fig. 2(a,b)). Hence, to ensure our optimization produces a distribution over continuous paths, we constrain the volume of states reachable within any finite time interval by accounting for the system's controllability properties (see Methods).

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

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

where $\mathbf{C}[x^*] = Cov[x(t)]_{x(t_0)=x^*}$ captures the local magnitude of temporal correlations induced by the agent's controllability properties, and Z is a normalization constant (see Methods). This distribution describes the statistics of an agent with minimally correlated continuous paths, subject to the constraints imposed by their controllability. Moreover, Eq. 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 analyzed through the lens of statistical mechanics (see Supplementary Figure 3). This also means that the sample paths of the optimal agent are Markovian and ergodic (see Supplementary Note 1.4 for associated theorems, 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 statistics.

Satisfying ergodicity has profound implications for the behavior of resulting agents. Ergodicity is a formal property of dynamical systems that guarantees that the statistics of individual trajectories are asymptotically equivalent to those of a large ensemble of trajectories [45, 46]. Put in terms of our problem setting, while the sequential nature of RL agent experience 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 satisfies the statistics of Eq. 1, we describe the agent as maximally diffusive. However, agents do not satisfy maximally diffusive statistics spontaneously. Matching these statistics requires

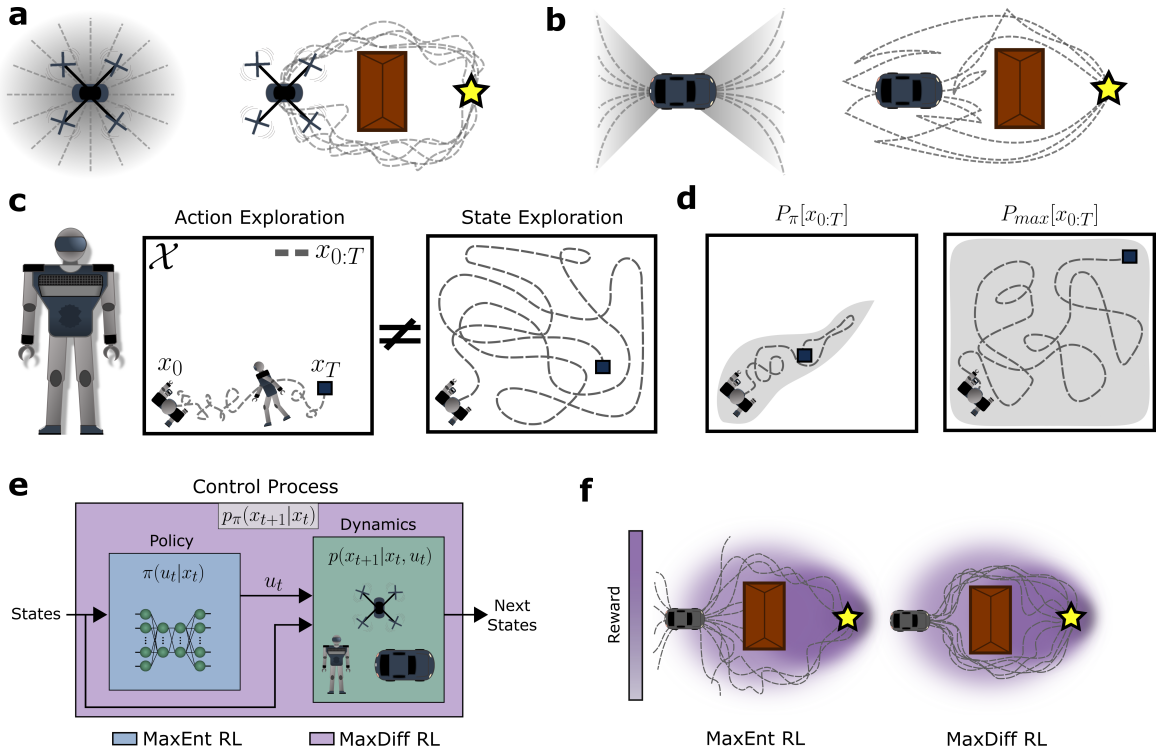


Figure 2: **Maximum diffusion RL exploits controllability to achieve effective exploration**. **a,b**, Systems with different planar controllability properties leading to different possible trajectories despite both systems being formally controllable. **c**, Whether action randomization leads to effective state exploration depends on an agent's controllability (see Supplementary Note 1.1), as in our illustration of a complex bipedal robot falling over and failing to explore. **d**, While 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 gray region. We prove that maximizing the entropy of an agent's state transitions results in effective exploration (see Supplementary Notes 1.4 and 2.5). **e**, Our approach generalizes the MaxEnt RL paradigm by considering agent dynamics in addition to their policy. We prove that maximizing a policy's entropy does not generally to maximize the entropy of an agent's state transitions (see Supplementary Note 2.2). **f**, This approach leads to distinct learning outcomes because agents can reason about the impact of their actions on state transitions, rather than their actions in isolation.

finding a policy capable of realizing them, which forms the core of what we term MaxDiff RL. While any given policy induces a path distribution, finding policies that realize maximally diffusive statistics requires optimization and learning (Fig. 2(d)). To satisfy the requirements of RL as a problem setting, we define:

$$\begin{aligned} P_\pi[x_{0:T}, u_{0:T}] &= \prod_{t=0}^{T-1} p(x_{t+1}|x_t, u_t)\pi(u_t|x_t) \\ P^r_{max}[x_{0:T}, u_{0:T}] &= \prod_{t=0}^{T-1} p_{max}(x_{t+1}|x_t)e^{r(x_t,u_t)}, \end{aligned} \tag{2}$$

where we discretized the distribution in Eq. 1 as $p_{max}(x_{t+1}|x_t)$, and analytically rederived the optimal path distribution under the influence of a reward landscape, $r(x_t, u_t)$ (see Methods). Given the distributions in Eq. 2, the goal of MaxDiff RL can be framed as minimizing the Kullback-Leibler (KL) divergence between them—that is, between the agent's current path distribution and the maximally diffusive one—as in the KL-control literature.

To draw connections between our framework and the broader MaxEnt RL literature, we recast the KL-control formulation of MaxDiff RL as an equivalent stochastic optimal control (SOC) problem. In SOC, 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

$$\pi^*_{\text{MaxDiff}} = \underset{\pi}{\text{argmax}}\, E_{(x_{0:T},u_{0:T})\sim P_\pi}\left[\sum_{t=0}^{T-1}\hat{r}(x_t,u_t)\right], \tag{3}$$

with modified rewards given by

$$\hat{r}(x_t,u_t) = r(x_t,u_t) - \alpha\log\frac{p(x_{t+1}|x_t,u_t)\pi(u_t|x_t)}{p_{max}(x_{t+1}|x_t)}, \tag{4}$$

where α is a positive temperature-like parameter 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. *MaxEnt RL is a special case of MaxDiff RL with the added assumption that state transitions are decorrelated.*

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 2.2.

Completely destroying correlations generally requires discontinuous jumps between states, which can only be achieved by fully controllable agents [23]. When an agent is fully controllable, there always exists a policy that enables it to reach every state and specify the statistics of how each state is reached. If this condition is met, then the optimum of Eq. 3 is attained when $p(x_{t+1}|x_t,u_t^*) = p_{\pi^*}(x_{t+1}|x_t) = p_{max}(x_{t+1}|x_t)$, where u_t^* are actions drawn from an optimized policy π^*. In turn, this simplifies Eq. 4 and recovers the MaxEnt RL objective [9], as shown in Supplementary Note 2.2. This proves not only 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 the decorrelation of state transitions 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 dynamics in its optimization (Fig. 2(e)). As such, MaxDiff RL typically produces distinct learning outcomes from MaxEnt RL (Fig. 2(f)). Our result also implies that all theoretical robustness guarantees of MaxEnt RL (e.g., [27]) 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 controllability properties of the underlying agent, as we saw in Fig. 1.

Finally, while our results seem to suggest that model-free implementations of MaxDiff RL are not feasible, we note that it is possible to indirectly account for the agent's controllability properties by learning local estimates of the agent's entropy generation from observations. Similar entropy estimates have been used in model-free RL [47] and more broadly in the autoencoder literature [48]. For the results presented in this manuscript, we derived a model-agnostic objective that uses an analytical expression for the state transition entropy,

$$\underset{\pi}{\text{argmax}}\, E_{(x_{0:T},u_{0:T})\sim P_\pi}\left[\sum_{t=0}^{T-1} r(x_t,u_t) + \frac{\alpha}{2}\log\det\mathbf{C}[x_t]\right], \tag{5}$$

whose optimum realizes the same maximally diffusive statistics as Eq. 3. We note that there are many ways to formulate the MaxDiff RL objective, each of which may have implementation-specific advantages (see Fig. 3(a) and Supplementary Note 2.3). 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 additional entropy term shown above. However, as we will demonstrate, this simple modification can have a drastic effect on agent behavior.

2.3 Seed-invariance in ergodic agents

The introduction of an entropy term in Eq. 5 means that MaxDiff RL agents must balance between two aims: achieving the task and embodying diffusion (Fig. 3(a)). While 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

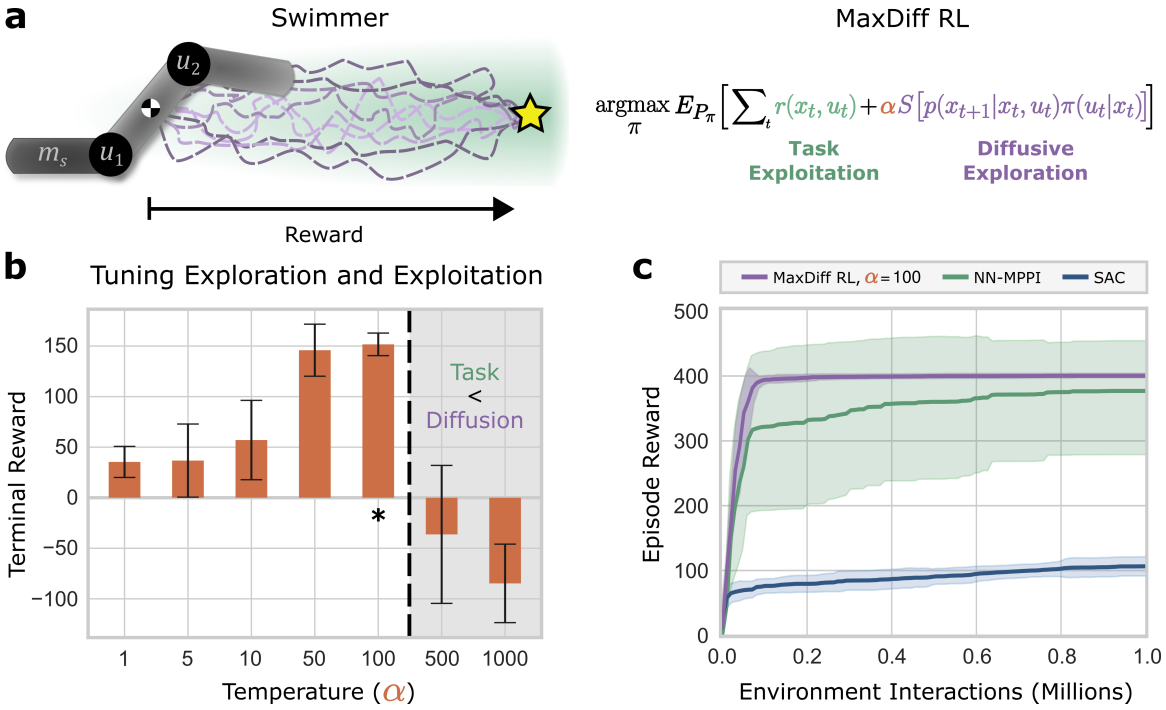


Figure 3: **Diffusive exploration produces robust seed-invariant performance**. **a**, Illustration of MuJoCo swimmer environment (left panel). The swimmer has 2 degrees of actuation, u_1 and u_2, that rotate its limbs at the joints, with tail mass m_s and $m = 1$ for other limbs. MaxDiff RL synthesizes robust agent behavior by learning policies that balance task-capability and diffusive exploration (right panel). 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 rewards of swimmer agents (10 seeds each) across values of α with $m_s = 1$. Diffusive exploration leads to greater rewards until a critical point (inset dotted line), after which the agent starts valuing diffusing more than accomplishing the task (see also Supplementary Movie 1). **c**, Using $\alpha = 100$, we compared MaxDiff RL against SAC and NN-MPPI with $m_s = 0.1$. We observe that SAC consistently achieves suboptimal performance, whereas NN-MPPI can achieve competitive performance but not reliably as there is substantial variation across seeds (see shaded area, 10 seeds each). Our approach performs robustly across seeds, since seed-invariance is a formal property of MaxDiff RL agents (see also Supplementary Movie 2).

temperature-like parameter, α. Unlike similar parameters in other RL frameworks, the role of α in MaxDiff RL can often be understood without the need for analogy through the lens of statistical mechanics. For simple MaxDiff RL agents with fixed controllability properties in a reward landscape, α sets the temperature of a heat bath induced by the policy (see Supplementary Figure 6). Hence, in line with the statistical mechanics of diffusion, the value of α can play a role in establishing the ergodicity of MaxDiff RL agents. If α is set too high, then the system's fluctuations can overpower the influence of the reward and break ergodicity, which has been shown in the context of diffusion processes in potential fields [49].

Since ergodicity provides many of MaxDiff RL's desirable properties and guarantees, tuning the value of α is essential. In Fig. 3 and Supplementary Movie 1, we explore the effect of tuning α on the learning performance of MaxDiff RL agents in MuJoCo's swimmer environment. The swimmer system is comprised of three rigid links of nominally equal mass, $m = 1$, with two degrees of actuation at the joints. The agent's objective is to swim in the goal direction as fast as possible within a fixed time interval, while in the presence of viscous drag forces (Fig. 3(a)). In Fig. 3(b), we vary α across multiple orders of magnitude and examine its impact on the terminal rewards of MaxDiff RL swimmer agents. As we modulate the value of α from 1 to 100, we observe that diffusive exploration leads to greater task rewards. However, after $\alpha = 100$ we cross a critical threshold, beyond which the strength of the system's diffusive exploration overpowers the reward (see inset dotted line in Fig 3(b)), 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 10 seeds each. To ensure that the task was solvable by all agents, we lowered the mass of the swimmer's third link (i.e., its tail) to $m_s = 0.1$. We find that while SAC struggles to succeed at this task 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 (see Fig. 3(c) and Supplementary Movie 2). Hence, merely by decorrelating state transitions, our agent was able to exhibit robustness to model and environment randomization beyond what is typically possible in deep RL. Moreover, since our implementation of MaxDiff RL is identical to that of NN-MPPI, we can completely attribute any performance gains and variance reduction to the properties of MaxDiff RL's theoretical framework.

This robustness to model and environmental randomizations is referred to as seed-invariance, and is a highly desirable feature of deep RL agents. However, guaranteeing seed-invariance is generally challenging because it requires modeling the impact of neural representation variability on learning outcomes. Nonetheless, we can still provide model-independent guarantees through the probably approximately correct (PAC) learning framework—one of the most successful and widely applied mathematical formalizations of learning [50]. The PAC framework assesses an agent's ability to learn function classes from data with a given likelihood and margin of error. Under this framework, we are able to provide formal seed-invariance guarantees.

Theorem 2. *MaxDiff PAC learners are seed-invariant.*

We refer the reader to Supplementary Note 1.5 for details, but the proof follows from treating PAC generalization risk as an observable in Birkhoff's ergodic theorem [45]. Since maximally diffusive agents are ergodic, any system initialization will eventually realize identical learning outcomes, leading to seed-invariance. Despite excluding neural representations from our analysis, Fig. 3(c) suggests that our guarantees hold empirically. Beyond PAC learning, we note that maximally diffusive agents are still provably robust to environmental randomization (see Supplementary Note 1.6).

2.4 Generalization across agent embodiments

As we saw in a previous section, when agents are capable of finding optimal policies, the MaxDiff RL objective in Eq. 4 becomes independent of the underlying agent's state transition statistics. This suggests that successful MaxDiff RL models and policies may exhibit favorable generalization properties across agent embodiments. To explore this question, as well as the robustness of MaxDiff RL agents to variations in their neural network models, we devised a transfer experiment in the MuJoCo swimmer environment. We designed two variants of the swimmer agent: 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. 4(a)). We trained two sets of models for each algorithm included in the comparison. 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 models on both the swimmer variant that they observed during training, as well as its counterpart. Our experiment's outcomes are shown in Fig. 4(b,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 manuscript: all algorithms benefit from working with a more controllable system (see Fig. 4(b) and Supplementary Movie 2). However, as MaxDiff RL is the only approach that takes into account system controllability, 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. 4(a)). First, we note that for both NN-MPPI and SAC model 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 behavior. In contrast, our MaxDiff RL agents can actually benefit and improve their performance when deployed on the more controllable swimmer variant, as desired (see "Heavy-to-Light" transfer in Fig. 4(c) and Supplementary Movie 3). In other words, as the task becomes easier in this way, we can expect the performance of MaxDiff RL agents to improve. Another crucial MaxDiff RL transfer result is the performance increase between the baseline heavy-tailed swimmer and the "Light-to-Heavy" transfer swimmer (Fig. 4(c) and Supplementary Movie 3). We found that training with a more controllable swimmer increased the performance of agents when deployed on the heavy-tailed swimmer, showing

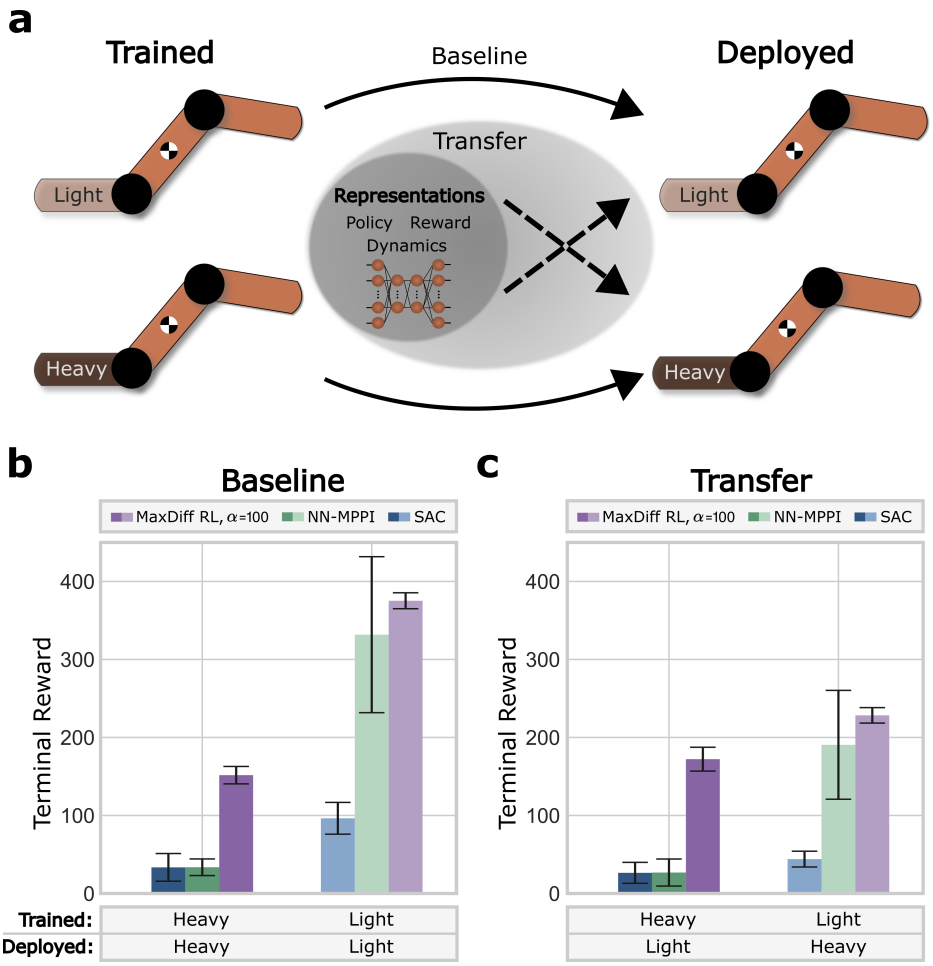


Figure 4: **Trained system controllability dominates deployed system performance**. **a**, Two variants of the MuJoCo swimmer environment: one with $m_s = 1$ and another with $m_s = 0.1$. As a baseline, we deploy learned models on the same swimmer variant trained on. Then, we carry out a transfer learning experiment where the trained and deployed swimmer variants are swapped. **b**, Baseline experiments confirm our previous results—all algorithms benefit from a more controllable swimmer. Since MaxDiff RL optimizes system controllability, it is the only method capable of achieving the task with a heavy-tailed swimmer (see also Supplementary Movie 2). **c**, Both NN-MPPI and SAC performance degrades when deployed on a more controllable system than was trained on, which is not desirable behavior. 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 Movie 3).

that system controllability during training matters more to overall performance that the particular embodiment of the deployed system. In part, this occurs because greater controllability leads to improved exploration, which increases the diversity of data observed during training. While formalizing this result is challenging, we note that MaxDiff RL encourages generalizable policies by focusing on agent outcomes instead of input actions. By forcing agent path statistics to match those of an ergodic diffusion process, MaxDiff RL implicitly minimizes the effect of agent dynamics on performance.

2.5 Single shot learning in ergodic agents

A longstanding challenge in the field of RL is the development of methods capable of supporting learning in single-shot agent deployments. Most methods in deep RL are designed to work in multi-shot settings (Fig. 5(b)), where randomized instantiations of tasks and environments, reset

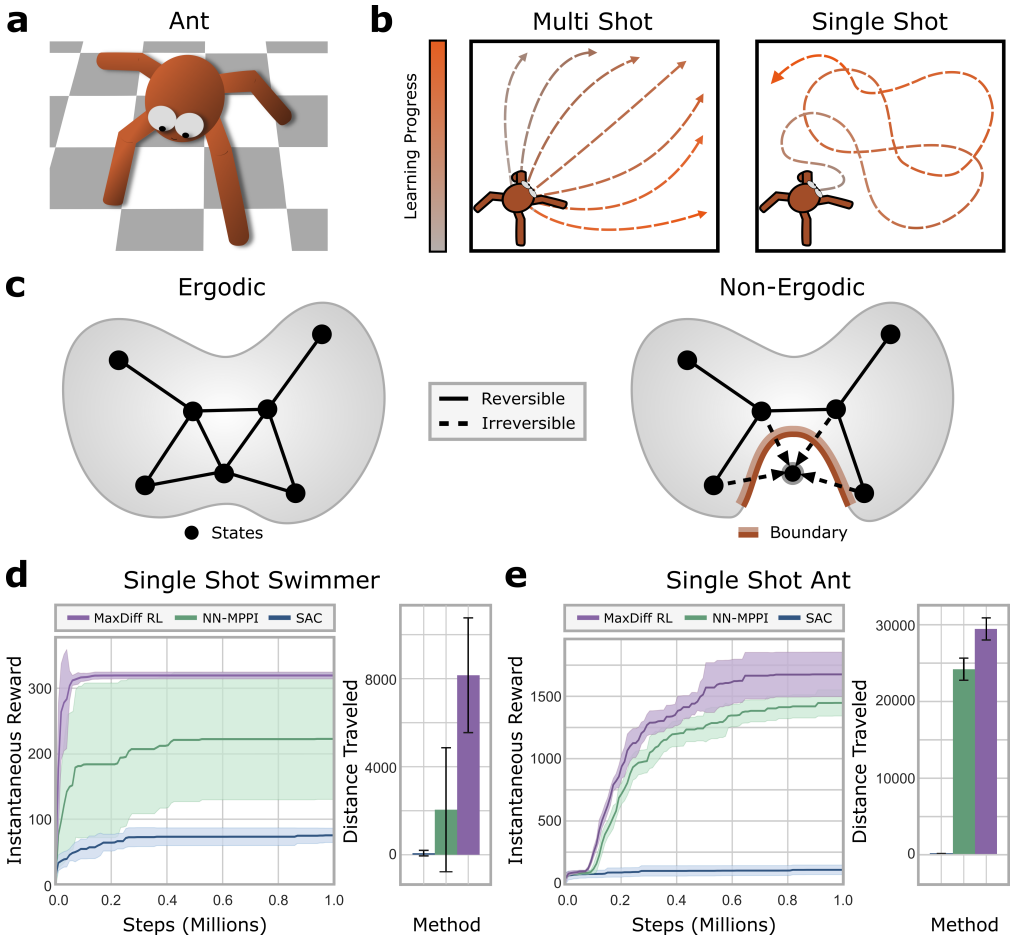


Figure 5: **Maximally diffusive RL agents are capable of single-shot learning. a**, Illustration of MuJoCo ant environment. **b**, Typical algorithms learn across many different initializations and deployments of an agent, which is known as multi-shot learning. In contrast, single-shot learning insists on a single agent deployment, which requires learning through contiguous experiences. 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 robustly seed-invariant performance comparable to its multi-shot counterpart (see also Supplementary Movie 4). **e**, In contrast to the swimmer, the MuJoCo ant environment contains irreversible state transitions (e.g., flipping upside down) preventing ergodic trajectories. Nonetheless, MaxDiff RL remains state-of-the-art in single-shot learning.

across distinct agent deployments, provide a kind of passive variability that is essential to learning processes. However, episodic problem structures of this kind are very rare in real-world applications. As a result, most multi-shot learning methods need simulations to work in practice, requiring the aid of sim-to-real techniques to bridge performance gaps in real-world deployment [6]. Thus, techniques capable of enabling continual learning from scratch, without resetting the agent or environment, are crucial to future applications of deep RL.

Despite the challenges associated with studying the behavior of agents based on neural network models, the ergodic properties of MaxDiff RL enables one to provide model-independent guarantees on the feasibility of single-shot learning through the PAC learning framework.

Theorem 3. *MaxDiff multi-shot PAC learners are also single-shot PAC learners.*

This theorem follows directly from the seed-invariance of maximally diffusive agents. Since ergodicity guarantees that the learning performance of any given PAC learner is indistinguishable from that of an ensemble of learner initializations, it also necessarily implies that the learning outcomes of single-shot

and multi-shot PAC learners are identical, asymptotically (see Supplementary Note 1.5). Because the ergodicity of maximally diffusive agents is central to this proof, we expect this equivalence 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. 5(a)). 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. 5(c)). 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 behavior, such as a robot breaking or malfunctioning during learning. While such transitions can be prevented in principle through the use of safety-preserving methods [51–53], we omit their implementation to illustrate our point. As expected, the MaxDiff RL single-shot swimmer is capable of learning in continuous deployments (see Fig. 5(d) and Supplementary Movie 4), retaining the same seed-invariance of its multi-shot counterpart in Fig. 3(c), 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. 5(e), where we plot the final distance traveled to ensure that no reward hacking took place. However, the loss of ergodicity leads to an increase in the variance of MaxDiff RL agent performance, which we expect as a result of seed-invariance no longer holding.

3 Discussion

Throughout this work, we have highlighted the ways in which deep RL is fragile to 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 deep 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 deep RL—one in which falsifiable predictions can be made about the properties of deep RL agents and their performance. However, much more interdisciplinary 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. Additionally, work in computational learning theory will be crucial to certifying the performance of agents subject to real-world conditions, such as distribution shift. And control techniques capable of enforcing ergodicity in the face of environmental irreversibility are needed to guarantee desirable agent properties like seed-invariance in complex problem settings. Taken together, our work paves the way towards more transparent and reliable decision-making with deep learning agents, which will be crucial to the long-term viability of deep RL as a field.

Methods

Reinforcement learning preliminaries

In this work, we make use of various notational conventions across multiple fields. Here, we summarize some basic notational norms of RL as a decision-making framework. RL problems are modeled as Markov decision processes (MDPs). MDPs are typically defined according to a 4-tuple, $(\mathcal{X}, \mathcal{U}, p, r)$, where we take both the state space, $\mathcal{X}$, and the action space, $\mathcal{U}$, to be continuous. 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 $r : \mathcal{X} \times \mathcal{U} \to [r_{min}, r_{max}]$. In general, the goal is to learn a 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 $t \in \{0, \cdots, T-1\}$.

Maximum caliber variational optimization

Here, we restate the objective function of our maximum caliber functional optimization and some key analytical results. For a complete motivation and derivation of the results that follow, we refer the reader to Supplementary Note 1. We begin by defining a stochastic control process with continuous state trajectories $x(t)$ over a compact measurable probability space $(\mathcal{X}, \mathcal{F}, P)$, where $\mathcal{X}$ is the agent's state space, $\mathcal{F}$ is the space of all possible paths through said state space, and P is a probability measure (see Supplementary Note 1.2 for more details). The objective consists of three components: first, a path integral entropy functional; then, a normalization term to ensure valid probability distributions; and finally a constraint on the local magnitude of the agent's velocity fluctuations. We define the local magnitude of these fluctuations up to a proportionality constant in the following way,

$$\langle \dot{x}(t)\dot{x}(t)^T \rangle_{x^*} \propto \int_{\mathcal{F}} P[x(t)] \int_{-\infty}^{\infty} \dot{x}(t)\dot{x}(t)^T \delta(x(t) - x^*) dt \mathcal{D}x(t),$$

where $\mathcal{D}x(t)$ denotes path integration, $\delta(\cdot)$ is a Dirac delta function, and x^* is a particular point in $\mathcal{X}$. We note that $\dot{x}(t)$ should not be interpreted as a statement on the differentiability of agent paths, but rather as a shorthand for the integral representation of the agent's evolution, as is standard in the Langevin process literature. This allows us to write the objective of our variational optimization as,

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

where λ_0 is a Lagrange multiplier, $\Lambda(x^*)$ is a matrix-valued Lagrange multiplier at all points $x^* \in \mathcal{X}$, and $\mathbf{C}[x^*] = Cov[x(t)]_{x(t_0)=x^*}$ empirically captures the local controllability properties of agents (see Supplementary Note 1.1). Our constraint on the system's velocity fluctuations locally bounds their magnitude to the strength of temporal correlations induced by the system's controllability properties, as measured by $\mathbf{C}[x^*]$. This effectively limits the volume of states reachable by the system within a finite time interval. Crucially, we note that it would not be sufficient to merely constrain the velocities' magnitudes because such a constraint would admit degenerate path distributions, with support on lower-dimensional manifolds. We also note that we omit boundary conditions in our problem statement for simplicity. The solution to this variational optimization is the following distribution describing the statistics of agent paths,

$$P_{max}[x(t)] = \frac{1}{Z} \exp \Big[-\frac{1}{2} \int_{-\infty}^{\infty} \dot{x}(t)^T \mathbf{C}^{-1}[x(t)] \dot{x}(t) dt \Big].$$

As discussed in the main text, this expression describes the path statistics of an anisotropic, spatially-inhomogenous, Markov, diffusion process (see [54], Ch. 9), the ergodicity of which we prove in Supplementary Note 1.4.

When agents are influenced by a potential field or a reward function, the maximum caliber objective above can be easily adapted to account for this influence under mild assumptions, as discussed in Supplementary Note 1.5. The new objective is a variational free energy minimization,

$$\underset{P[x(t)]}{\text{argmin}} \; \langle V[x(t)] \rangle_P - S[P[x(t)]],$$

where $S[P[x(t)]]$ is the original maximum caliber objective shown previously in this section, and the potential is defined in the following way:

$$\langle V[x(t)] \rangle_P = \int_{\mathcal{F}} P[x(t)] \int_{-\infty}^{\infty} V[x(t)] dt \mathcal{D}x(t).$$

The derivation of its solution is similar to the one for the original objective, attaining the following optimal path distribution up to normalization:

$$P^r_{max}[x(t)] = P_{max}[x(t)] \cdot e^{\int r[x(t)] dt},$$

where $r[x(t)] = -V[x(t)]$ is a bounded and Lipschitz instantaneous reward function defined over continuous agent paths. Notably, the resulting distribution continues to satisfy the Markov property

and ergodicity (see Supplementary Note 1.5 for proofs), and the influence of the reward is factorable from the optimal path statistics. Discretizing this distribution and accounting for control actions (see Supplementary Note 2.2), we have the following analytically-derived distribution as well,

$$P^r_{max}[x_{0:T}, u_{0:T}] = \prod_{t=0}^{T-1} p_{max}(x_{t+1}|x_t)e^{r(x_t,u_t)},$$

which we make use of to adapt our approach to maximally diffusive trajectory synthesis to RL and MDP problem settings, as in Supplementary Note 2.2.

Changes of coordinates and partial observability

Throughout this work, we have intentionally equated agent states and their experiences. In most problems in RL, the state space of the underlying agent and the space where learning and exploration takes place are different. While we have omitted this complication from the main text to retain notational simplicity, this does not necessarily pose issues to our framework. For example, when we are interested in exploring or learning in some space, $\mathcal{Y}$, defined by a coordinate transformation of our state variables, $y_t = \psi(x_t)$, the guarantees of MaxDiff RL can still hold depending on the coordinate transformation's properties. If our transformation is linearizable, then as long as $\mathbf{C}[y_t] = \mathbf{J}_\psi \mathbf{C}[x_t]\mathbf{J}_\psi^T$ is full rank everywhere in $\mathcal{X}$, where $\mathbf{J}_\psi$ is the Jacobian of the transformation, our results will hold. For all results presented in the manuscript, we made use of linear projections of our state variables to select the domain of maximally diffusive exploration (e.g., $[x, y, \dot{x}, \dot{y}]$ for the results in Fig. 3). All exploration variables used in our results are listed in Supplementary Table 1. More broadly, the formal observability properties of the underlying system should be characterized and accounted for to rigorously study the role of partial observability under our framework, which can be done with a Gramian approach similar to the way we have characterized controllability, but such an investigation is outside of the scope of this work [32].

References

[1] Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de las Casas, Craig Donner, Leslie Fritz, Cristian Galperti, Andrea Huber, James Keeling, Maria Tsimpoukelli, Jackie Kay, Antoine Merle, Jean-Marc Moret, Seb Noury, Federico Pesamosca, David Pfau, Olivier Sauter, Cristian Sommariva, Stefano Coda, Basil Duval, Ambrogio Fasoli, Pushmeet Kohli, Koray Kavukcuoglu, Demis Hassabis, and Martin Riedmiller. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022. doi: 10.1038/s41586-021-04301-9.

[2] Dong-Ok Won, Klaus-Robert Müller, and Seong-Whan Lee. An adaptive deep reinforcement learning framework enables curling robots with human-like performance in real-world conditions. *Science Robotics*, 5(46), 2020. doi: 10.1126/scirobotics.abb9764.

[3] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019. doi: 10.1038/s41586-019-1724-z.

[4] Alex Irpan. Deep reinforcement learning doesn't work yet. `https://www.alexirpan.com/2018/02/14/rl-hard.html`, 2018.

[5] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 32(392), 2018.

[6] Julian Ibarz, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research*, 40(4):698–721, 2021. doi: 10.1177/0278364920987859.

[7] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. doi: 10.1038/nature14236.

[8] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[9] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 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.

[10] Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y. Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[11] Long-Ji Lin. *Reinforcement learning for robots using neural networks*. Carnegie Mellon University, 1992.

[12] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[13] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[14] Shangtong Zhang and Richard S. Sutton. A deeper look at experience replay. *NeurIPS Deep Reinforcement Learning Symposium*, 2017.

[15] Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample efficient actor-critic with experience replay. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[16] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 32(1), 2018.

[17] William Fedus, Prajit Ramachandran, Rishabh Agarwal, Yoshua Bengio, Hugo Larochelle, Mark Rowland, and Will Dabney. Revisiting fundamentals of experience replay. *Proceedings of the International Conference on Machine Learning (ICML)*, pages 3061–3071, 2020.

[18] Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 8:1433–1438, 2008.

[19] Brian D. Ziebart, J. Andrew Bagnell, and Anind K. Dey. Modeling interaction via the principle of maximum causal entropy. *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 1255—-1262, 2010.

[20] Brian D. Ziebart. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. Carnegie Mellon University, 2010.

[21] Emanuel Todorov. Efficient computation of optimal actions. *Proceedings of the National Academy of Sciences*, 106(28):11478–11483, 2009. doi: 10.1073/pnas.0710743106.

[22] Marc Toussaint. Robot trajectory optimization using approximate inference. *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pages 1049—1056, 2009. doi: 10.1145/1553374.1553508.

[23] Konrad Rawlik, Marc Toussaint, and Sethu Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference. *Proceedings of Robotics: Science and Systems (RSS)*, pages 353–361, 2012.

[24] Sergey Levine and Vladlen Koltun. Guided policy search. *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 28(3):1–9, 2013.

[25] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. *Proceedings of the International Conference on Machine Learning (ICML)*, 70:1352–1361, 2017.

[26] Tuomas Haarnoja, Sehoon Ha, Aurick Zhou, Jie Tan, George Tucker, and Sergey Levine. Learning to walk via deep reinforcement learning. *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

[27] Benjamin Eysenbach and Sergey Levine. Maximum entropy RL (provably) solves some robust RL problems. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

[28] Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H. Chi. Top-k off-policy correction for a REINFORCE recommender system. *Proceedings of the 12th ACM International Conference on Web Search and Data Mining (WSDM)*, pages 456—-464, 2019. doi: 10.1145/3289600.3290999.

[29] M. Mehdi Afsar, Trafford Crump, and Behrouz Far. Reinforcement learning based recommender systems: A survey. *ACM Computing Surveys*, 55(7), 2022. doi: 10.1145/3543846.

[30] Xiaocong Chen, Lina Yao, Julian McAuley, Guanglin Zhou, and Xianzhi Wang. Deep reinforcement learning in recommender systems: A survey and new perspectives. *Knowledge-Based Systems*, 264:110335, 2023. doi: 10.1016/j.knosys.2023.110335.

[31] Eduardo D. Sontag. *Mathematical Control Theory: Deterministic Finite Dimensional Systems*, volume 6. Springer, 2013. ISBN 9781461205777.

[32] J. P. Hespanha. *Linear Systems Theory: Second Edition*. Princeton University Press, 2018. ISBN 9780691179575.

[33] D. Mitra. *W* matrix and the geometry of model equivalence and reduction. *Proceedings of the Institution of Electrical Engineers*, 116:1101–1106, 1969.

[34] Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. On the sample complexity of the linear quadratic regulator. *Foundations of Computational Mathematics*, 20(4): 633–679, 2020. doi: 10.1007/s10208-019-09426-y.

[35] Anastasios Tsiamis and George J. Pappas. Linear systems can be hard to learn. *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 2903–2910, 2021. doi: 10.1109/CDC45484.2021.9682778.

[36] Anastasios Tsiamis, Ingvar M Ziemann, Manfred Morari, Nikolai Matni, and George J. Pappas. Learning to control linear systems can be hard. In *Proceedings of 35th Conference on Learning Theory (COLT)*, volume 178, pages 3820–3857, 2022.

[37] Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M Rehg, Byron Boots, and Evangelos A. Theodorou. Information theoretic MPC for model-based reinforcement learning. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1714–1721, 2017.

[38] Oswin So, Ziyi Wang, and Evangelos A. Theodorou. Maximum entropy differential dynamic programming. In *2022 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3422–3428, 2022. doi: 10.1109/ICRA46639.2022.9812228.

[39] Susan Amin, Maziar Gomrokchi, Harsh Satija, Herke van Hoof, and Doina Precup. A survey of exploration methods in reinforcement learning. *arXiv preprint arXiv:2109.00157*, 2021.

[40] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[41] E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106:620–630, 1957.

[42] Purushottam D. Dixit, Jason Wagoner, Corey Weistuch, Steve Pressé, Kingshuk Ghosh, and Ken A. Dill. Perspective: Maximum caliber is a general variational principle for dynamical systems. *The Journal of Chemical Physics*, 148(1):010901, 2018.

[43] Pavel Chvykov, Thomas A. Berrueta, Akash Vardhan, William Savoie, Alexander Samland, Todd D. Murphey, Kurt Wiesenfeld, Daniel I. Goldman, and Jeremy L. England. Low rattling: A predictive principle for self-organization in active collectives. *Science*, 371(6524):90–95, 2021.

[44] J.N. Kapur. *Maximum Entropy Models in Science and Engineering*. Wiley, 1989. ISBN 9788122402162.

[45] Calvin C. Moore. Ergodic theorem, ergodic theory, and statistical mechanics. *Proceedings of the National Academy of Sciences*, 112(7):1907–1911, 2015. doi: 10.1073/pnas.1421798112.

[46] Annalisa T. Taylor, Thomas A. Berrueta, and Todd D. Murphey. Active learning in robotics: A review of control principles. *Mechatronics*, 77:102576, 2021. doi: 10.1016/j.mechatronics.2021.102576.

[47] Younggyo Seo, Lili Chen, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. State entropy maximization with random encoders for efficient exploration. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 9443–9454, 2021.

[48] Ahalya Prabhakar and Todd Murphey. Mechanical intelligence for learning embodied sensor-object relationships. *Nature Communications*, 13(1):4108, 2022. doi: 10.1038/s41467-022-31795-2.

[49] Xudong Wang, Weihua Deng, and Yao Chen. Ergodic properties of heterogeneous diffusion processes in a potential well. *The Journal of Chemical Physics*, 150(16):164121, 2019. doi: 10.1063/1.5090594.

[50] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning, (2nd Edition)*. Adaptive Computation and Machine Learning series. MIT Press, 2018. ISBN 9780262351362.

[51] A. Ames, J. Grizzle, and P. Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *2014 IEEE Conference on Decision and Control (CDC)*, 2014.

[52] Andrew Taylor, Andrew Singletary, Yisong Yue, and Aaron Ames. Learning for safety-critical control with control barrier functions. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control (L4DC)*, volume 120, pages 708–717, 2020.

[53] Wei Xiao, Tsun-Hsuan Wang, Ramin Hasani, Makram Chahine, Alexander Amini, Xiao Li, and Daniela Rus. Barriernet: Differentiable control barrier functions for learning of safe robot control. *IEEE Transactions on Robotics*, pages 1–19, 2023. doi: 10.1109/TRO.2023.3249564.

[54] Mehran Kardar. *Statistical Physics of Fields*. Cambridge University Press, 2007. ISBN 9780511815881.

Data availability

Data supporting the findings of this study are available in the following repository: github.com/MurpheyLab/MaxDiffRL.

Code availability

Code supporting the findings of this study is available in the following repository: github.com/MurpheyLab/MaxDiffRL.

Acknowledgements

We thank Annalisa T. Taylor, Jamison Weber, and Pavel Chvykov for their comments on early drafts of this work. We acknowledge funding from the US Army Research Office MURI grant #W911NF-19-1-0233, and the US Office of Naval Research grant #N00014-21-1-2706. We also acknowledge hardware loans and technical support from the 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 reinforcement learning experiments, and wrote the manuscript. A.P. developed and tested the reinforcement learning algorithm, carried out all reinforcement learning experiments, and supported manuscript writing. T.D.M. secured funding and guided the research program.

Maximum Diffusion Reinforcement Learning
Supplementary Information

Thomas A. Berrueta* **Allison Pinosky** **Todd D. Murphey**
Center for Robotics and Biosystems, Northwestern University, Evanston, IL, USA.

Contents

*Corresponding author: tberrueta@u.northwestern.edu

Supplementary notes

1 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 2.

1.1 The role of controllability 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 [1], physical agents—simulated or otherwise—have no such luxury [2–5]. The laws of physics, an agent's material properties, and dynamics all impose fundamental constraints on what can be achieved by a learning system. In particular, as discussed throughout the main text, the state transition dynamics of learning agents often introduce temporal correlations that can hinder agent performance. To illustrate this point broadly, here we consider the effect of an agent's controllability properties on the efficacy of a widely used exploration strategy in reinforcement learning: taking actions at random and observing what happens.

Drawing inspiration from the study of multi-armed bandits [6], 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 [7–9], elaborate on this basic idea by learning the distribution from which to sample random actions. For the purposes of our analysis, these more advanced methods are functionally equivalent to each other—they assume that taking random actions produces effective exploration of outcomes. However, from the perspective of control theory we know that this is not always the case. For an agent to be able to arbitrarily reach desired states, it must be *controllable* [10].

To illustrate the role controllability plays in the temporal correlations induced by an agent's controllability properties, we will briefly consider randomized action exploration in a linear time-varying (LTV) control system:

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

where $A(t)$ and $B(t)$ are appropriately dimensioned matrices with state and control vectors $x(t) \in \mathcal{X}$ and $u(t) \in \mathcal{U}$, and $x(t_0) = x^*$ for $[t_0, t] \subset T$. For now, we omit technical specification of T, $\mathcal{X}$, and $\mathcal{U}$. 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:

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

The state-transition matrix itself is the solution to an algebraic Riccati equation. 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 [11]. 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 1.1. *A system is said to be controllable over a time interval $[t_0, t] \subset T$ 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 [12]). In particular, here we will analyze the controllability Gramian of our system, as well as its rank and determinant as metrics on the controllability of our system.

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

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

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 [13, 14]. 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 1.1 [10], 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 have shown in the main text and will show in the following sections.

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. We will replace the controller $u(t)$ with a noise vector $\xi \sim \mathcal{N}(\mathbf{0}, \mathrm{Id})$ taken in the Itô sense, where 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:

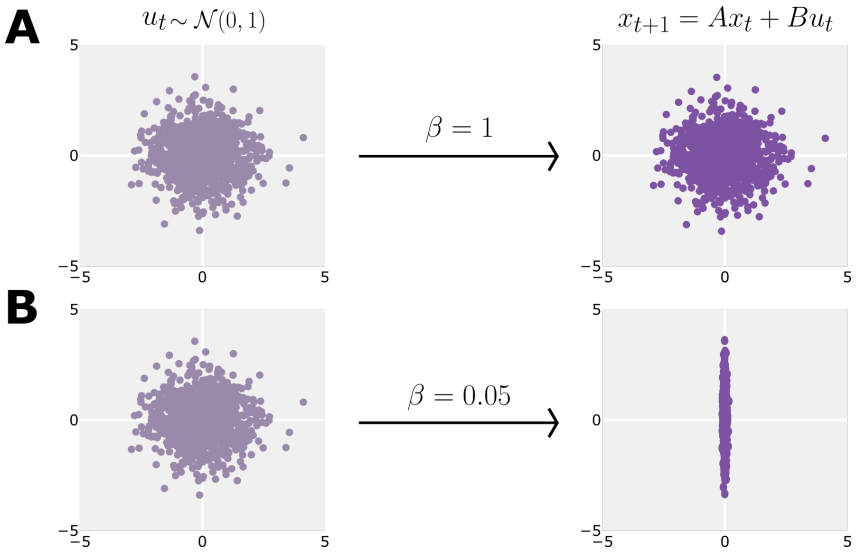
$$\dot{x}(t) = A(t)x(t) + B(t) \cdot \xi. \tag{4}$$

Here, we abuse notation slightly to minimize the difference between this equation and Eq. 1. The substitution of noise in place of control inputs is precisely what some simple exploration strategies in reinforcement learning do: using random actions to drive the system into previously unobserved states crucial to a given learning task. With these modifications in mind, we are now interested in examining the system's mean trajectory and covariance 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:

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

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 also characterize the system's temporal correlations through its covariance statistics, $\mathbf{C}[x^*] = Cov[x(t)]_{x(t_0)=x^*}$,

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



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.

where the $\times$ operator merely is there to indicate a product with the expression in the line above. By inspection of the above expression and Eq. 3, we arrive at the following important connection:

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

which tells us that for LTV dynamics (and by extension for linearizable nonlinear dynamics), the state covariance matrix is exactly equivalent to the controllability Gramian of the system. Thus, for a broad class of systems, an agent's controllability properties completely determine the structure of temporal correlations induced by their dynamics.

Moreover, we can see that our controllability is not a state-dependent property of LTV systems and linearizable nonlinear systems (at least within a neighborhood of their linearization),

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

where $\mathbf{0}$ is an appropriately dimensioned zero matrix. 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 state covariance and controllability Gramians have been shown to hold for even more general classes of nonlinear systems through more involved analyses [15]. We note that 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 [16]. To do this, we only require the mean and covariance statistics of the process, in Eqs. 5 and 6. Hence, the system's state distribution is

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

for some choice of initial conditions at t_0, where d is the dimension of x, and we have substituted Eq. 7 to highlight the role of controllability in the density of states reachable by the system through random exploration. Thus, how easy or hard it is to explore in a given direction—as characterized by the distribution of reachable states in Eq. 9—is entirely determined by the controllability properties

of the system as encoded by $W(t, t_0)$, and the temporal correlations these induce in the agent's 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.

On the basis of these results, which have been known for decades [17], 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 $W(t, t_0)$ 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 pathologically correlated as a result of the degeneracy of Eq. 9. As we show in future sections, preventing this degeneracy will be crucial for achieving effective exploration. 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, it will depend on the controllability properties of said agent.

Remark 1.1. *Controllability can determine whether or not it is possible, and how challenging it is, to learn.*

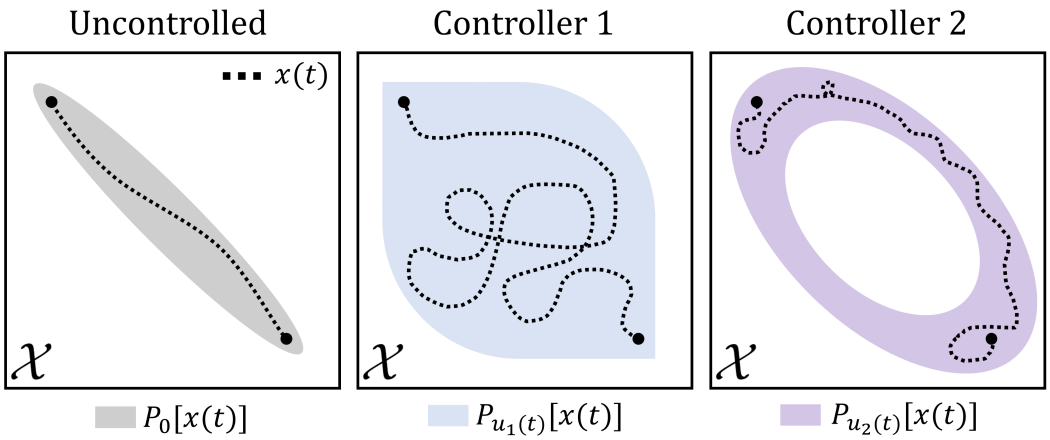
While one can construct proofs that illustrate this in a variety of simplified settings—as others have recently shown [18, 19]—we leave the more general claim as a remark to frame the motivation behind our upcoming derivations. Hence, we should strive to develop exploration and learning strategies that reflect—and try to overcome—the effect of controllability and its induced temporal correlations, as we do in the following sections.

1.2 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 their physical state or configuration through action. Our goal is to achieve exploration by means of a control system, such as a robotic agent or otherwise, where their properties constrain the ways they can explore. While this motivation is most natural for physically-embodied systems, our framing is relevant to any setting in which the underlying agent's dynamics cannot be arbitrarily chosen and obey some notion of continuity of experience. To this end, we need to first define a formal notion of control system from which we can begin to model the behavior of agents.

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. Typically, a stochastic process is a collection of random variables $\{x(t) : t \in T\}$ indexed according to some set, T. This collection of random variables will come to define the states and experiences of an underlying agent. We assume that the indexing set T is continuous and totally ordered so that it is time-like, and throughout this manuscript we will often use "time" to refer to this indexing set. Hence, we can think of the trajectories of an agent with autonomous dynamics as a realization of a continuum of time-indexed random variables taking place on a common measurable probability space, $(\mathcal{X}, \mathcal{F}, P)$ [20]. For the purpose of framing our problem, we choose the sample space $\mathcal{X}$ to be a compact, simply connected, metric space. We make this choice to deliberately circumvent the effects of environment topology and boundary conditions on the framework derived herein—as it was alluded to in the main text, these factors do play a role in agent outcomes but we leave a detailed investigation of their impact as future work. Importantly, we choose the sample space to be the same as the space in which the random variables take value. As with the main text, we also define the agent's experiences to take place directly on its state space for convenience. Then, $\mathcal{F}$ is a Borel σ-algebra of the sample space consisting of all sample paths of the process. Finally, P is a probability measure describing the likelihood of any sample path of the stochastic process.

Clearly, the likelihood of any sample path of the system $P[x(t)]$ is strongly dependent on the dynamics that govern the time-evolution of the stochastic process. However, when the dynamics are nonautonomous, as is the case in control systems, the probability measure will also depend on the choice of controller and the effect it has on the dynamics of the process. We define a controller as a



Supplementary Figure 2: **Effect of controllers on the sample path distribution of stochastic control processes**. (left) Sample path and support of the probability measure 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 measures, only their support. The reason for this is that so long as a regions of space are non-zero measure they will be sampled asymptotically.

function, $u(t) : T \to \mathcal{U}$, that produces an input to the system dynamics at every point in the index set. 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 measure over sample paths. With these definitions we can now establish our notion of control system, or stochastic control process. For convenience, we assume that $x(\cdot)$ and $u(\cdot)$ are vector-valued and of dimensions d and m respectively.

Definition 1.2. *A stochastic control process is a collection of random variables $\{x(t) : t \in T\}$ with index set T, defined on a probability space $(\mathcal{X}, \mathcal{F}, P_{u(t)})$, where the probability measure $P_{u(t)}$ is parametrized by a controller $u(t) : T \to \mathcal{U}$.*

In a stochastic control process the controller plays an important role in the resulting behavior observed in the sample paths of the 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 can affect which regions of the exploration domain the control 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 measure is the subset of all elements in the Borel σ-algebra $\mathcal{F}$ with non-zero measure. However, merely maximizing the path distribution's support is not enough. Ideally, we would also like to control how probability mass is spread—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. At this point it is helpful to note the basic way in which this approach differs from naive random exploration. Rather than letting $u(t)$ be substituted by some noise distribution and hoping to see exploration in $x(t)$, we are interested in deliberately designing $u(t)$ to maximize our exploration of $x(t)$. In the following sections, we formalize our exploration problem statement and illustrate the role that an agent's controllability properties and temporal correlations play in enabling—or hindering—effective exploration and learning.

1.3 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 [21]. 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 1.5, this approach is also able to control the spread of probability mass in a more fine-grained manner that will allow us to achieve 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 agent's compact sample 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 dynamics 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 trajectories of agents with continuous paths—a broad class of systems that includes all physical agents and many non-physical agents—as well as analyzing the formal properties of systems that satisfy such statistics. We will see that by maximizing trajectory entropy this distribution also captures the statistics of an agent with minimally-correlated paths. The analytical form of this distribution is crucial to the control and policy synthesis approach we derive in Supplementary Note 2. However, we note that our results can 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 sample 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 [22–24]. Maximum caliber is a generalization of the principle of maximum entropy to probability measures over function spaces, such as distributions over trajectories or sample paths. In this arena, we are interested in finding the distribution which maximizes the entropy of the sample path distribution of our system $S[P[x(t)]]$. Because in this section we are looking for the unique analytical form of this distribution, we omit the controller-specific notation that was introduced in the previous section. The general form of the maximum entropy (or caliber) functional variational optimization problem is the following:

$$\underset{P[x(t)]}{\operatorname{argmax}} \; -\int_{\mathcal{F}} P[x(t)] \log P[x(t)] \mathcal{D}x(t), \tag{10}$$

where $\mathcal{D}x(t)$ denotes path integration over all sample paths $\mathcal{F}$ of our stochastic control process. 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 measure 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 agents are inherently nonequilibrium systems. Nonetheless, the behavior of many autonomous agents (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 velocities can vary—and *co-vary*—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 showed in Eq. 3 of Supplementary Note 1.1, where we found that an agent's controllability properties are closely tied to its trajectory fluctuations, as well as its ability to locally explore space. Thus, we choose to constrain the velocity fluctuations of our stochastic control process so that they are finite and consistent with the local covariance 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) covariance estimate to quantify local exploration rates is important because different agents have different limitations, which may additionally be spatially inhomogeneous and difficult to know a priori. Through this constraint, we both ensure that agents have a bounded local rate of exploration and that their sample paths are continuous in time.

To formulate this constraint on the system's local rate of exploration, we must first express the system's velocity fluctuations at each point in the sample space, $x^* \in \mathcal{X}$. We define the system's local exploration rate in terms of its velocity fluctuations in the following way:

$$\langle \dot{x}(t)\dot{x}(t)^T \rangle_{x^*} \propto \int_{\mathcal{F}} P[x(t)] \int_{-\infty}^{\infty} \dot{x}(t)\dot{x}(t)^T \delta(x(t) - x^*) dt \mathcal{D}x(t), \tag{11}$$

where we only care about proportionality since normalization takes care of constants, and $\delta(\cdot)$ denotes the Dirac delta function. We assume that the local exploration rate is not degenerate in the chosen coordinates of the stochastic control process, and hence that the tensor described by Eq. 11 is full-rank. This assumption is crucial because it guarantees that our resulting path distribution is non-degenerate and capable of realizing effective exploration. However, our results would continue to hold for a linear subspace of the original domain if this condition is not met. If we had instead chosen to constrain the system's local exploration rates 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. Nonetheless, assumptions on the degeneracy of the system's fluctuations in state space are irrelevant to the more general problem where the sample space of the stochastic control process is not the same as the state space its random variables take value in. Another important note is that the velocities of the trajectories of the stochastic control process in this expression should be interpreted in the Langevin sense [25]. That is to say, not as expressions of the differentiability of the sample paths of our stochastic control 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 on the local rate of exploration as,

$$\langle \dot{x}(t)\dot{x}(t)^T \rangle_{x^*} = \mathbf{C}[x^*], \quad \forall x^* \in \mathcal{X}, \tag{12}$$

where $\mathbf{C}[x^*] = Cov[x(t)]_{x(t_0)=x^*}$ denotes the empirical covariance statistics associated with the local exploration of space over sample paths initialized at x^*. 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 local exploration rates 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 measure over paths.

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

$$\begin{aligned} \underset{P[x(t)]}{\text{argmax}} \; & - \int_{\mathcal{F}} P[x(t)] \log P[x(t)] \mathcal{D}x(t) - \lambda_0 \Big(\int_{\mathcal{F}} P[x(t)] \mathcal{D}x(t) - 1 \Big) \\ & - \int_{\mathcal{X}} Tr\Big(\Lambda(x^*)^T \big(\langle \dot{x}(t)\dot{x}(t)^T \rangle_{x^*} - \mathbf{C}[x^*] \big) \Big) dx^*. \end{aligned} \tag{13}$$

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 measures, 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.

1.4 Maximizing path entropy produces diffusion

In this section, we lay out the derivation of our solution to the variational optimization problem in Eq. 13. We begin by stating our main result in the following theorem.

Theorem 1.1. *The maximum entropy sample paths of a stochastic control process (Definition 1.2) engaging in maximum entropy exploration (in the sense of Eq. 13) are given by pure diffusion with spatially-varying coefficients.*

Proof. We begin by substituting Eq. 11 into Eq. 13, taking its variation with respect to the probability measure $\delta S[P[x(t)]]/\delta P[x(t)]$, and setting it equal to 0:

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

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:

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

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:

$$P_{max}[x(t)] = \frac{1}{Z} \exp\Big[- \int_{-\infty}^{\infty} \dot{x}(t)^T \Lambda(x(t)) \dot{x}(t) dt \Big], \tag{14}$$

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 measure in Eq. 14 is already equivalent to the path probability of a diffusing particle with a (possibly anisotropic) spatially-inhomogeneous diffusion tensor (see [25], Ch. 9). In other words, Eq. 14 describes the probability of a continuous random walk with increments determined by a Gaussian distribution with space-dependent variance [20]. We also note that the measure in Eq. 14 has infinite support. While there is more work needed to characterize the diffusion tensor of this process, $\Lambda^{-1}(\cdot)$, this completes our proof. □

So what does this result tell us? 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 sample space describes the paths of a diffusion process. Hence, if the goal of our stochastic control process is to optimally sample from its sample space, the best strategy is to move randomly—that is, to decorrelate its sample paths. As an exercise, let's consider how such exploration may relate to a learning problem. If the goal of our agent is to learn something about its dynamics or its environment, then this result suggests that the best strategy is to try to move randomly according to a diffusion process. At first glance, this seems to validate the strategy of taking random actions that many reinforcement learning algorithms use for exploration; however, this is not the case. Our result requires that we choose a controller $u(\cdot)$ such that the sample paths of our agent are random, which is not the same as choosing a controller that is random, as discussed in Supplementary Note 1.1. An additional benefit of our diffusive exploration strategy is that we did not have to presuppose that our agent dynamics were Markov or ergodic, or any of the typical assumptions that reinforcement learning algorithms make. Instead, we find that these properties emerge through our derivation as intrinsic properties of the optimal exploration strategy itself.

The following corollaries of Theorem 1.1 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. 14, $\Lambda^{-1}(\cdot)$, is full-rank and invertible everywhere in the exploration domain. Otherwise, these results would still hold but only for a linear subspace of the exploration domain. 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 about the local exploration rate in Eqs. 11 and 12.

Corollary 1.1.1. *The sample paths of a stochastic control process (Definition 1.2) with a maximum entropy exploration strategy (in the sense of Eq. 13) satisfy the Markov property.*

Proof. This follows trivially from the temporal discretization of our path distribution in Eq. 14, or alternatively from the properties of Langevin diffusion processes. We can see that,

$$\begin{aligned} p_{max}(x_{t+\delta t}|x_t) &= \frac{1}{Z} \exp\Big[- \int_t^{t+\delta t} \dot{x}(\tau)^T \Lambda(x(\tau)) \dot{x}(\tau) d\tau \Big] \\ &\approx \frac{1}{Z_d} \exp\Big[- \frac{1}{2} ||x_{t+\delta t} - x_t||^2_{\Lambda(x_t)} \Big], \end{aligned} \tag{15}$$

where we subsumed δt into a new normalization constant Z_d for convenience, and note that the support of $p_{max}(x_{t+\delta t}|x_t)$ remains 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 will assume that $\delta t = 1$ for notational convenience, but without loss of generality. In summary, our distribution over future states in Eq. 15 depends only on the current state, which concludes our proof. □

Corollary 1.1.2. *A stochastic control process (Definition 1.2) in a bounded, simply connected, exploration domain with a maximum entropy exploration strategy (in the sense of Eq. 13) is ergodic.*

Proof. To prove the ergodicity of the process described by the path distribution in Eq. 14, we take advantage of Corollary 1.1.1 and the properties of our exploration domain $\mathcal{X}$. We begin by discretizing our optimal stochastic control process such that $P_{max}[x_{1:N}] = \prod_{t=1}^{N} p_{max}(x_{t+1}|x_t)$, which we can do without loss of generality as a result of Corollary 1.1.1, using the conditional measure defined therein. Importantly, since $p_{max}(x_{t+1}|x_t) > 0,\ \forall x_t, x_{t+1} \in \mathcal{X},\ \forall t \in T$, and $\mathcal{X}$ is compact and simply connected, the transition operator induced by this Markov chain, M_{max}, is irreducible and aperiodic. Finally, making use of the well-known Perron-Frobenius theorem (see, e.g. [20], Ch. 4), we see that our stochastic control process admits an invariant measure with respect to which our process' sample paths are ergodic, which concludes our proof. □

In the context of optimal control and reinforcement learning, Corollary 1.1.2 is particularly important. For one, ergodicity guarantees that as time goes on the stochastic control process will sample from every non-zero measure set of the exploration domain. Depending on the details of the learning task, this can serve as an asymptotic guarantee on the learning process. Lastly, because Corollary 1.1.2 implies that our stochastic control process also satisfies Birkhoff's pointwise ergodic theorem as well as other ergodic theorems [26], the time-averaged behavior of the process' sample paths and their ensemble-average behavior are asymptotically the same. This is to say that the outcome of a single long rollout and the outcome of many rollouts should be equivalent in the limit to a reinforcement learning agent engaging in our exploration strategy, as we will prove in the following sections. Moreover, satisfying Birkhoff's theorem also guarantees that the statistics of data generated by such an agent will be asymptotically equivalent to those generated by *i.i.d.* sampling from the underlying data distribution.

To finish our derivation and fully characterize the nature of our maximum entropy exploration strategy, we must return to Eq. 14 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. 11 and discretize our continuous sample paths, which we can do without loss of generality due to Corollary 1.1.1. Since Eq. 11 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. 14. We proceed by discretizing Eq. 11, using i and j as time indices and $p_{max}(\cdot|\cdot)$ as the conditional probability measure defined in Eq. 15. Our resulting expression is the following

$$\langle \dot{x}(t)\dot{x}(t)^T \rangle_{x^*} \propto \prod_{i=-\infty}^{\infty} \Big[\int_{\mathcal{X}} dx_{i+1}\, p_{max}(x_{i+1}|x_i) \Big] \sum_{j=-\infty}^{\infty} (x_{j+1} - x_j)(x_{j+1} - x_j)^T \delta(x_j - x^*), \tag{16}$$

where the path integrals are discretized according to the Feynman formalism [27], using the same discretization as in our proof of Corollary 1.1.1 for convenience.

From this expression in Eq. 16, we take the following two steps. First, we switch out the order of summation and product, which we can do since there are no mutual dependencies between their arguments. 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:

$$\begin{aligned} &= \sum_{j=-\infty}^{\infty} \prod_{i \neq j, j-1} \Big[\int_{\mathcal{X}} dx_{i+1}\, p_{max}(x_{i+1}|x_i) \Big] \\ &\quad \times \int_{\mathcal{X}} p_{max}(x_j|x_{j-1}) \int_{\mathcal{X}} p_{max}(x_{j+1}|x_j)(x_{j+1} - x_j)(x_{j+1} - x_j)^T \delta(x_j - x^*) dx_{j+1} dx_j, \end{aligned} \tag{17}$$

where the $\times$ operator indicates a product with the expression in the line above. Then we can apply the Dirac delta function to simplify our expression and get:

$$= \sum_{j=-\infty}^{\infty} \prod_{i\neq j,j-1} \Big[\int_{\mathcal{X}} dx_{i+1}\, p_{max}(x_{i+1}|x_i) \Big] \times p_{max}(x^*|x_{j-1}) \int_{\mathcal{X}} p_{max}(x_{j+1}|x^*)(x_{j+1}-x^*)(x_{j+1}-x^*)^T dx_{j+1}. \quad (18)$$

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

$$I = \int_{\mathcal{X}} p_{max}(x_{j+1}|x^*)(x_{j+1}-x^*)(x_{j+1}-x^*)^T dx_{j+1}. \quad (19)$$

where we can substitute Eq. 15 into Eq. 19 up to proportionality to get:

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

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:

$$I = \frac{1}{2}\Lambda(x^*)^{-1}\Big[\sqrt{\det(2\pi\Lambda^{-1}(x^*))} - (x_{j+1}-x^*)^T \mathbf{1} e^{-(x_{j+1}-x^*)^T\Lambda(x^*)(x_{j+1}-x^*)}\Big|_{x_{j+1}=-\infty}^{x_{j+1}=\infty}\Big], \quad (20)$$

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. 20 and plugging it into Eq. 18 we have

$$\langle \dot{x}(t)\dot{x}(t)^T\rangle_{x^*} \propto \frac{1}{2}\sum_{j=-\infty}^{\infty} \prod_{i\neq j,j-1} \Big[\int_{\mathcal{X}} dx_{i+1}\, p_{max}(x_{i+1}|x_i) \Big] \times p_{max}(x^*|x_{j-1})\sqrt{\det(2\pi\Lambda^{-1}(x^*))}\Lambda(x^*)^{-1}. \quad (21)$$

Since $\langle \dot{x}(t)\dot{x}(t)^T\rangle_{x^*}$ is everywhere full-rank and $p_{max}(\cdot|\cdot)$ has infinite support, neither $\det(2\pi\Lambda(x^*)^{-1})$ nor $p_{max}(x^*|x_{j-1})$ can evaluate to 0. Thus, the full-rankness of our exploration implies the full-rankness of $\Lambda(x^*)^{-1}$. As the rest of this expression consists of constants independent of x^*, this means that we may consolidate all scalars in Eq. 21 and subsume them into the normalization constant Z in our final expression. Now, we can determine the form of our Lagrange multiplier by making use of the constraint in Eq. 12, leading us to

$$\Lambda(x^*) = \mathbf{C}^{-1}[x^*]. \quad (22)$$

This result is significant because for a broad class of systems it allows us to make a direct connection between an agent's ability to explore, its controllability properties, and the temporal correlations these induce, as discussed in Supplementary Note 1.1. We now have the final form of the maximum entropy exploration sample path distribution in terms of the covariance matrix:

$$P_{max}[x(t)] = \frac{1}{Z}\exp\Big[-\frac{1}{2}\int_{-\infty}^{\infty} \dot{x}(t)^T\mathbf{C}^{-1}[x(t)]\dot{x}(t)dt\Big], \quad (23)$$

where we have added a factor of one half to precisely match the path probability of purely diffusive spatially-inhomogeneous dynamics. This final connection can be made rigorous by thinking of the covariance matrix as an estimator of the diffusion tensor through the following relation:

$\mathbf{C}[\cdot] = \frac{1}{2}\mathbf{D}[\cdot]\mathbf{D}[\cdot]^T$ for some diffusion tensor $\mathbf{D}[\cdot]$ [28, 29]. Hence, when faced with path continuity constraints the optimal exploration strategy is given by diffusion, 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 local exploration rate of the stochastic control process is everywhere full-rank. 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 1.1 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. For example, if we have a differential drive vehicle whose state space is its position and orientation, exploration in some planar environment usually only requires that we can fully vary its position. The orientation, while key to describing the microscopic dynamics of the process, may not matter to the broader exploration task. Instead, we may 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 $\mathbf{C}[y(t)]$, so long as $\mathbf{C}[y(t)] = \mathbf{J}_\psi[x(t)]\mathbf{C}[x(t)]\mathbf{J}_\psi[x(t)]^T$ 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.

1.5 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 trajectories and whose path statistics are equivalent to those of controllability-dependent ergodic diffusion. 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 exploration domain [30]. 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 [7], and in optimal control this is often encoded by a cost function or an expected information density [31, 32]. In such settings, one may want an agent to explore states while taking into account the 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 thermodynamic 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 in the following way,

$$\langle V[x(t)]\rangle_P = \int_{\mathcal{F}} P[x(t)] \int_{-\infty}^{\infty} V[x(t)]dt\mathcal{D}x(t), \tag{24}$$

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

$$\underset{P[x(t)]}{\operatorname{argmin}} \langle V[x(t)]\rangle_P - S[P[x(t)]], \tag{25}$$

where we use $S[P[x(t)]]$ as a short-hand for the argument to Eq. 13. Thankfully, to find the optimal path distribution all of the work carried out in Supplementary Notes 1.3 and 1.4 remains unchanged. All that's needed is to take the variation of Eq. 24 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 1.1, we omit it here. The resulting minimum free energy path distribution is then

$$P^V_{max}[x(t)] \propto \exp\Big[-\int_{-\infty}^{\infty}\Big(V[x(t)] + \frac{1}{2}\dot{x}(t)^T\mathbf{C}^{-1}[x(t)]\dot{x}(t)\Big)dt\Big], \tag{26}$$

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,

$$P^V_{max}[x(t)] = P_{max}[x(t)] \cdot e^{-\int V[x(t)]dt} \tag{27}$$

up to normalization, 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 (MDPs) by letting $l(\cdot) = V[\cdot]$ be a standard cost function, which leads us to the following equivalent expression:

$$P^l_{max}[x_{1:N}] = \prod_{t=1}^{N} p_{max}(x_{t+1}|x_t)e^{-l(x_t)}, \tag{28}$$

where we have discretized agent trajectories without loss of generality to match the formal requirements of MDPs. Remarkably, this path distribution resembles the form of those used in the control-as-inference literature [33]. 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. 25 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 1.2. *A stochastic control process (Definition 1.2) achieving maximum diffusion exploration in a potential (in the sense of Eq. 26) is ergodic with respect to the measure induced by the potential.*

Proof. The proof of this theorem can be easily arrived at by extending the proof of Corollary 1.1.2. So long as $V[\cdot]$ is non-singular and has bounded spatial derivatives, we may discretize Eq. 27 in the same way we discretized Eq. 14. This ensures that the potential is integrable and that it does overpower the system's diffusion by diverging. To this end, we assume $V[\cdot]$ is Lipschitz and, as before, that $V[x_t]$ applies to a neighborhood of x_t containing all states reachable between t and $t + \delta t$ for a sufficiently small δt. From these assumptions we can see that $p^V_{max}(x_{t+\delta t}|x_t) = p_{max}(x_{t+\delta t}|x_t)e^{-\int V[x_t]dt} > 0,\ \forall x_t, x_{t+\delta t} \in \mathcal{X},\ \forall t \in T$ with sufficiently small δt. This is because we have already shown that $p_{max}(\cdot|\cdot) > 0$ in Corollary 1.1.2, and because of the properties of our potential. As before, the transition operator induced by Markov chain and the potential, M^V_{max}, is aperiodic and irreducible, which allows us to establish the process' ergodicity through the Perron-Frobenius theorem and concludes our proof. Thus, the net effect of the potential is to reshuffle probability mass in the stationary distribution of our agent's Markov chain. We note that these proofs can be carried out without discretizations by instead invoking the physics of diffusion processes, as in [34] 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. □

Thus, minimum free energy exploration leads to ergodic coverage of the exploration domain in proportion to the measure induced by the potential function. 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 section. Another important note is that ergodicity guarantees that the outcomes of single-shot and multi-shot learning processes can be formally the same in a broad class of learning processes. A longstanding challenge in the field of reinforcement learning has been the development of frameworks and algorithms that can—both formally and in practice—learn in single-shot settings without the need for resetting the environment. However, if ergodicity guarantees that the statistical properties of an agent's sample paths are asymptotically equivalent in single-rollout and multi-rollout learning, then single-shot learning must be possible in some settings.

To briefly demonstrate the equivalence of single-shot vs. multi-shot learning in ergodic processes, we illustrate its effect in the Probably Approximately Correct (PAC) learning framework (see [35], Ch. 2).

Theorem 1.3. *A stochastic control process (Definition 1.2) achieving maximum diffusion exploration (in the sense of Eq. 23) and capable of PAC learning in a multi-shot setting is equivalently capable of single-shot learning asymptotically.*

Proof. We begin by reviewing some essential aspects of PAC learning [35]. The PAC learning framework is concerned with providing sample complexity bounds on the learning of function classes from data. These function classes $\mathcal{C}$ are comprised of "target concepts" $c(\cdot)$, which map from an input space $\mathcal{X}$ to some output space $\mathcal{Y}$. We assume the input space to be the same as the exploration domain of our maximally diffusive stochastic control process. In this abstract framework, a learning algorithm is successful if it is able to find a hypothesis $h(\cdot) : \mathcal{X} \to \mathcal{Y}$ within some class of functions $\mathcal{H}$ that matches the target concept. To determine whether this is the case, we define the generalization error or risk associated with a hypothesis.

Definition 1.3. *Given a hypothesis $h \in \mathcal{H}$, a target concept $c \in \mathcal{C}$, and an underlying data distribution $\mathcal{D}$, the generalization error or risk of h is defined as*

$$R(h) = P_{x\sim\mathcal{D}}[h(x) \neq c(x)] = E_{x\sim\mathcal{D}}[\mathbf{1}_{h(x)\neq c(x)}].$$

We note that in the above definition $\mathbf{1}_{h(x)\neq c(x)}$ is an indicator function that evaluates to 1 when the condition in the subscript is met and is 0 otherwise. We now state the formal definition of PAC-learnability.

Definition 1.4. *A concept class $\mathcal{C}$ is said to be PAC-learnable if there exists an algorithm $\mathcal{A}$ and a polynomial function $poly(\cdot, \cdot)$ such that for any $\epsilon > 0$ and $\delta > 0$, for all data distributions $\mathcal{D}$ on $\mathcal{X}$ and for any target concept $c \in \mathcal{C}$, the following holds for any sample size $N \geq poly(1/\epsilon, 1/\delta)$:*

$$P[R(h) \leq \epsilon] \geq 1 - \delta.$$

In other words, a class of functions is PAC-learnable if an algorithm can produce a function that recreates the input-output mapping of an arbitrary target function with high probability (at least $1 - \delta$) and low error (at most ϵ). With these definitions in hand, we may now prove how ergodicity enables single-shot PAC learning.

A crucial assumption underlying the concept of PAC-learnability is the independence and identical distribution (*i.i.d.*) of data samples. Ideally, an agent (or algorithm) would exhaustively sample from all regions of the data distribution $\mathcal{D}$ simultaneously and then produce a hypothesis from this spatial ensemble of data samples. This is equivalent to multi-shot learning—an ensemble of several agents are initialized in parallel to gather experience and feed a learning process. When we instead consider an embodied single-shot learning process, an agent such as a robot must navigate the exploration domain in order to gather samples in what is now a time-ordered sequential sampling process. In general, such a sampling process does not produce *i.i.d.* data [36]. However, ergodicity can give us a way around this through Birkhoff's well-known pointwise ergodic theorem [26], which we restate below:

Theorem 1.4. *(Birkhoff's Ergodic Theorem) Let f be a measurable observable with $E[|f|] < \infty$, and M be an ergodic measure-preserving map on a measure space $(\mathcal{X}, \mathcal{F}, \mathcal{D})$. Then with probability 1:*

$$\lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} f(M^t x) = E_{x\sim\mathcal{D}}[f].$$

Informally, this theorem states that the ensemble averages of observable functions of an ergodic process are equal to their time averages asymptotically. In general, we can think of the ergodic map M as a discrete-time representation of the dynamics of some dynamical system, where its superscript implies applying the map M iteratively t times. However, here we can substitute this generic definition with the transition operators M_{max} or M^V_{max} induced by the Markov chains of our discretized stochastic control processes, as defined in Corollary 1.1.2 and Theorem 1.2, respectively. Crucially, we have proven that both M_{max} and M^V_{max} are ergodic maps, which allows us use them along with Birkhoff's theorem. Then, from the expression of generalization error in Definition 1.3, we proceed by applying Birkhoff's theorem to the learning dynamics of a maximally diffusive stochastic control process:

$$R(h) = E_{x\sim\mathcal{D}}[\mathbf{1}_{h(x)\neq c(x)}] \tag{29}$$

$$= \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} \mathbf{1}_{h(M^t_{max}x)\neq c(M^t_{max}x)}. \tag{30}$$

This result shows that due to the ergodicity of maximally diffusive stochastic control processes, the generalization error of single-shot PAC learning (i.e., Eq. 30) is asymptotically equal to that of multi-shot PAC learning (i.e., Eq. 29). Thus, given an ergodic agent capable of multi-shot learning (in the sense of Definition 1.4), we have shown that it can asymptotically achieve the same generalization error in a single-shot setting, which concludes our proof. □

Instead of providing guarantees on the ergodicity of reinforcement learning generally, our proof examines the relationship between ergodicity and PAC learning, as others have done in similar contexts [37]. Our theorem states that the generalization error of any PAC learning algorithm is equivalent in single-shot and multi-shot settings. We note that this equivalently applies to empirical risk minimization problems since empirical risk is also a valid observable under Birkhoff's ergodic theorem. Hence, rather than applying to a particular class of reinforcement learning problems, our result applies to all algorithms formally capable of PAC learning, which includes many reinforcement learning algorithms (e.g., [38–42]).

As a final note, the following theorem also follows directly from our proof of Theorem 1.3.

Theorem 1.5. *A stochastic control process (Definition 1.2) achieving maximum diffusion exploration (in the sense of Eq. 23) and capable of PAC learning is asymptotically seed-invariant.*

Proof. The proof follows directly from the ergodicity of maximally diffusive PAC learners and the application of Birkhoff's ergodic theorem in our proof of Theorem 1.3. In providing guarantees that any single-shot PAC learner asymptotically attains the same generalization risk as a statistical ensemble of multi-shot learners, we also proved that ergodic PAC learners are able to learn from any random initialization (see Eq. 29), which establishes seed-invariance and concludes our proof. □

We note that we have focused on providing PAC guarantees because the use of deep learning architectures makes providing formal model-specific guarantees very challenging. Hence, by instead opting for model-independent guarantees, such as those that PAC provides, we can work around this limitation. Nonetheless, our results in the main text suggest that our guarantees hold empirically when deep learning architectures are applied. Outside of PAC learning, maximally diffusive sampling processes will more broadly lead to the same single-shot and multi-shot learning outcomes in learning algorithms that preserve ergodicity, such as ergodic mirror descent [43] or incremental subgradient methods [44].

1.6 Minimizing path free energy produces diffusive gradient descent

To develop further intuition about the sense in which systems satisfying the statistics of Eq. 26 are achieving goal-directed exploratory behavior, we can examine the maximum likelihood trajectory of our minimum free energy path distribution. To do this, we begin by calculating the negative

log-likelihood of $P^V_{max}[x(t)]$

$$\begin{aligned} -\log[P^V_{max}[x(t)]] &= \int_{-\infty}^{\infty} V[x(t)] + \frac{1}{2}\dot{x}(t)^T \mathbf{C}^{-1}[x(t)]\dot{x}(t)dt \\ &= \int_{-\infty}^{\infty} \mathcal{H}(t, x(t), \dot{x}(t))dt \\ &= \int_{-\infty}^{\infty} \dot{x}(t)^T \mathbf{C}^{-1}[x(t)]\dot{x}(t) - \mathcal{L}(t, x(t), \dot{x}(t))dt, \end{aligned} \tag{31}$$

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:

$$\mathcal{A} = \int_{-\infty}^{\infty} \mathcal{L}(t, x(t), \dot{x}(t))dt = \int_{-\infty}^{\infty} V[x(t)] - \frac{1}{2}\dot{x}(t)^T \mathbf{C}^{-1}[x(t)]\dot{x}(t)dt. \tag{32}$$

Assuming that our potential is differentiable, which the rest of our analysis does not require, we can find the dynamics of the maximum likelihood trajectory by using the Euler-Lagrange equations:

$$\begin{aligned} 0 &= \nabla_x \mathcal{L} - \frac{d}{dt}\big[\nabla_{\dot{x}} \mathcal{L}\big] \\ &= \nabla_x V[x(t)] - \frac{1}{2}\dot{x}(t)^T \nabla_x \mathbf{C}^{-1}[x(t)]\dot{x}(t) - \Big[-\ddot{x}(t)^T \mathbf{C}^{-1}[x(t)] - \dot{x}(t)^T \nabla_x \mathbf{C}^{-1}[x(t)]\dot{x}(t)\Big] \\ &= \nabla_x V[x(t)] + \ddot{x}(t)^T \mathbf{C}^{-1}[x(t)] + \frac{1}{2}\dot{x}(t)^T \nabla_x \mathbf{C}^{-1}[x(t)]\dot{x}(t) \end{aligned} \tag{33}$$

which we can rearrange into our final expression,

$$\ddot{x}(t) = -\mathbf{C}[x(t)]\Big[\nabla_x V[x(t)] + \frac{1}{2}\dot{x}(t)^T \nabla_x \mathbf{C}^{-1}[x(t)]\dot{x}(t)\Big] \tag{34}$$

This last expression represents the maximum likelihood dynamics of a system whose trajectories satisfy our minimum free energy path statistics. 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. 34 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 local exploration rate (or controllability, when applicable).

To simplify these dynamics further, we can make one of two assumptions: either that our local exploration rate varies slowly over space (at least relative to $\nabla_x V$), or that our system dynamics are linearizable. Taken together, our assumptions imply that $\nabla_x \mathbf{C} \approx \mathbf{0}$, which leads to a simplification of the final expression in Eq. 34. For the sake of making a connection to controllability, consider simplifying the maximum likelihood dynamics by assuming their linearizability. For this class of dynamics, Eq. 8 tells us that our system's controllability properties do not vary abruptly over state-space, as discussed in Supplementary Note 1.1. For systems with fixed or quasi-static morphologies this assumption holds well. Then, we have the following simplified dynamics:

$$\begin{aligned} \ddot{x}(t) &= -\mathbf{C}[x(t)]\nabla_x V[x(t)] \\ &= -W(t, t_0)\nabla_x V[x(t)]. \end{aligned} \tag{35}$$

By inspection we see that these second order dynamics resemble those of inertial gradient descent [45–48], 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 noisy exploration, suggesting that under our theoretical framework for maximum diffusion there is no formal trade-off between exploration and exploitation asymptotically, as we discuss in the following section. However, over finite time horizons we do not

expect this to be the case, as discussed in the main text. Nonetheless, this motivation will form the basis of our approach to optimization and learning in following sections.

For a moment, we consider the implications of a learning agent satisfying maximally diffusive trajectory statistics in the presence of a goal-encoding potential field (e.g., a reward function). Because such an agent will asymptotically realize the same path statistics as an ensemble of agents initialized from any initial condition, ergodicity formally requires robustness to randomized conditions of the agent and environment even outside of the PAC formalism, which we term "environmental seed-invariance," and state in the following proposition.

Proposition 1.1. *A stochastic control process (Definition 1.2) achieving maximum diffusion exploration in a potential (in the sense Eq. 26) is environmentally seed-invariant asymptotically.*

Proof. The proof of this proposition follows directly from the ergodicity of maximally diffusive agents in Theorem 1.2, but can be also motivated by our derivation in this section. The ergodicity of maximally diffusive agents formally requires that agents realize the same behavior regardless of their initialization, which alone guarantees environmental seed-invariance. □

Beyond ergodicity, however, our results in this section show that realizing maximally diffusive exploration with respect to a potential field spontaneously leads to goal-directed behavior regardless of how the agent and its environment are initialized. Their behavior will lead to the same outcomes in maximum likelihood—gradient descent on the potential. Moreover, because the nth moments of the maximally diffusive path distribution are zero for all $n > 2$—that is, our distribution is not heavy-tailed—we also know that maximum likelihood trajectories are representative of typical agent behavior. In other words, as long as our agent is controllable, we can reliably expect the same outcomes across different random realizations of the agent and its environment, which also establishes environmental seed-invariance. As discussed in the main text, formally guaranteeing seed-invariance in the sense that is usually entailed by the reinforcement learning community would require including deep learning models of agent dynamics, policies, and rewards in our analysis. However, analyzing such systems is exceedingly challenging and out of the scope of our work, which limits us to providing either model-independent seed-invariance guarantees, as we do in Theorem 1.5, or environmental seed-invariance guarantees, as we do above. Nonetheless, as our results in the main text show, our approach may be able to overcome these issues in practice.

As a final note on the derivations carried out throughout all of Supplementary Note 1, we point out that most of the work we have done largely amounts to formulating an exploration problem and deriving the optimal trajectory statistics for a sufficiently broad class of agents, $P_{max}[x(t)]$, as well as exploring its formal properties. We chose to tailor our modelling assumptions to capture the behavior of embodied agents—a class of agents historically underexplored in reinforcement learning theorycrafting—only considering agents whose trajectories are continuous. In doing so, we paid particular attention to the ways in which an agent's properties (i.e., its controllability-induced temporal correlations) affect its ability to generate optimal path statistics. However, it is entirely possible for disembodied agents 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 1 still holds when we remove the continuity constraint (except for the analysis in Supplementary Note 1.6). 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. 28, 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 2.1, as we discuss in the following section. We note that it is only under these conditions that agents can completely overcome correlations between state transitions. Moreover, all of the control and policy synthesis results we derive in the following sections will still hold and work well in a broad range of disembodied reinforcement learning applications, except for those in Supplementary Notes 2.4 and 2.5. While agents satisfying these statistics will still achieve sequential sampling that is asymptotically *i.i.d.*, the connection to the statistical mechanics of diffusion processes will no longer hold.

2 Synthesizing maximally diffusive trajectories

Throughout the previous section, we have been studying the properties of a theoretical agent whose trajectories spontaneously satisfy the 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*.

2.1 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 achieve these statistics. To this end, we recall that in Supplementary Note 1.2 we defined a path probability measure for an arbitrary stochastic control process, $P_{u(t)}[x(t)]$. Equipped with this measure, 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 1 and the system's current path distribution. Equivalently, we can express this as,

$$\underset{u(t)}{\text{argmin}}\ D_{KL}(P_{u(t)}[x(t)]||P^V_{max}[x(t)]), \tag{36}$$

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, we can factor our objective in the following way:

$$\begin{aligned}
D_{KL}(P_{u(t)}||P^V_{max}) &= \int_{\mathcal{F}} P_{u(t)}[x(t)] \log \frac{P_{u(t)}[x(t)]}{P^V_{max}[x(t)]} \mathcal{D}x(t) \\
&= \int_{\mathcal{F}} P_{u(t)}[x(t)] \Big[\log P_{u(t)}[x(t)] - \log P^V_{max}[x(t)] \Big] \mathcal{D}x(t) \\
&= \int_{\mathcal{F}} P_{u(t)}[x(t)] \Big[\log P_{u(t)}[x(t)] - \log P_{max}[x(t)] + \int_{-\infty}^{\infty} V[x(t)]dt \Big] \mathcal{D}x(t) \\
&= \langle V[x(t)] \rangle_{P_{u(t)}} + D_{KL}(P_{u(t)}[x(t)]||P_{max}[x(t)]),
\end{aligned} \tag{37}$$

where we used Eq. 27 to arrive at our final expression. Now, we can rewrite our control synthesis problem as the following

$$\underset{u(t)}{\text{argmin}}\ \langle V[x(t)] \rangle_{P_{u(t)}} + D_{KL}(P_{u(t)}[x(t)]||P_{max}[x(t)]), \tag{38}$$

or equivalently

$$\underset{u(t)}{\text{argmin}}\ E_{P_{u(t)}} \big[L[x(t), u(t)] \big] + D_{KL}(P_{u(t)}[x(t)]||P_{max}[x(t)]), \tag{39}$$

where we replace our potential with a cost function $L[x(t), u(t)] = \int l(x(t), u(t))dt$ 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 agent's 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 an agent is capable of achieving maximally diffusive statistics, 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 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.

2.2 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 [33]. In SOC, the objective is to find a policy $\pi(\cdot|\cdot)$ over control actions conditioned on the current state that optimizes the expected cost of a given cost-per-stage function over a time-horizon of fixed duration (although extending to an infinite horizon setting can be trivially done through the inclusion of a discount factor). The standard discrete time formulation of the SOC problem is

$$\pi^* = \underset{\pi}{\operatorname{argmin}}\, E_{(x_{1:N}, u_{1:N})\sim P_\pi}\Big[\sum_{t=1}^{N} l(x_t, u_t)\Big], \tag{40}$$

where $l(\cdot,\cdot)$ is a discretized running cost and the expectation is taken with respect to the trajectory measure 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 1.2) requires careful treatment. Whereas our definition of a path distribution allowed us to express a distribution directly over the trajectories of our agent, 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 because it makes the KL divergence in Eq. 36 ill-posed—the agent's distribution and our maximally diffusive distribution are now defined over different domains. To solve this issue, we introduce the following distributions:

$$\begin{aligned} P_\pi[x_{1:N}, u_{1:N}] &= \prod_{t=1}^{N} p(x_{t+1}|x_t, u_t)\pi(u_t|x_t) \\ P^l_{max}[x_{1:N}, u_{1:N}] &= \prod_{t=1}^{N} p_{max}(x_{t+1}|x_t)e^{-l(x_t,u_t)}, \end{aligned} \tag{41}$$

where $p_{max}(x_{t+1}|x_t) \propto \exp\big[-\frac{1}{2}(x_{t+1}-x_t)^T\mathbf{C}^{-1}[x_t](x_{t+1}-x_t)\big]$ is the discretized maximally diffusive conditional measure. The second of these distributions was analytically derived in Eq. 28, 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. 36 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. 36 as,

$$\underset{\pi}{\operatorname{argmin}}\, D_{KL}(P_\pi[x_{1:N}, u_{1:N}]||P^l_{max}[x_{1:N}, u_{1:N}]). \tag{42}$$

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:

$$D_{KL}(P_\pi||P^l_{max}) = E_{P_\pi}\left[\sum_{t=1}^{N} l(x_t, u_t) + \alpha \log \frac{p(x_{t+1}|x_t, u_t)\pi(u_t|x_t)}{p_{max}(x_{t+1}|x_t)}\right]. \tag{43}$$

With this result we are now able to write our final expression for an equivalent SOC representation of the KL control problem in Eq. 36:

$$\pi^*_{\text{MaxDiff}} = \underset{\pi}{\text{argmin}}\, E_{(x_{1:N}, u_{1:N})\sim P_\pi}\Big[\sum_{t=1}^{N} \hat{l}(x_t, u_t)\Big], \tag{44}$$

with

$$\hat{l}(x_t, u_t) = l(x_t, u_t) + \alpha \log \frac{p(x_{t+1}|x_t, u_t)\pi(u_t|x_t)}{p_{max}(x_{t+1}|x_t)}, \tag{45}$$

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.

Before we conclude this section, we return to the role that temporal correlations and controllability play in the generation of maximally diffusive trajectories. To this end, we first formalize and define a particular notion of controllability in the context of MDPs that was partially introduced in [49], implicit in the results of [50], and explicitly called out in [33].

Definition 2.1. *A state transition model, $p(x_{t+1}|x_t, u_t)$, in an MDP, $(\mathcal{X}, \mathcal{U}, p, r)$, is fully controllable when there exists a policy, $\pi : \mathcal{U} \times \mathcal{X} \to [0, \infty)$, such that:*

$$p_\pi(x_{t+1}|x_t) = E_{u_t\sim\pi(\cdot|x_t)}[p(x_{t+1}|x_t, u_t)] \tag{46}$$

and

$$D_{KL}\Big(p_\pi(x_{t+1}|x_t)\Big|\Big|\nu(x_{t+1}|x_t)\Big) = 0, \quad \forall t \in \mathbb{Z}^+ \tag{47}$$

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 2.1. 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 controlled state transition model as

$$p_\pi(x_{t+1}|x_t) = \int_{\mathcal{U}} p(x_{t+1}|x_t, u_t)\pi(u_t|x_t)du_t, \tag{48}$$

where the integral over control actions arises from the expectation in Eq. 46. Alternatively, in the deterministic case the agent's state transition model is given by

$$p_\pi(x_{t+1}|x_t) = \int_{\mathcal{U}} p(x_{t+1}|x_t, u_t)\delta(u_t - \tau_\pi(x_t))du_t = p(x_{t+1}|x_t, \tau_\pi(x_t)), \tag{49}$$

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 given deterministic function [33].

Equipped with our definition of full controllability, we may now shed a light on the relationship between our MaxDiff RL framework and the broader literature on maximum entropy reinforcement learning (MaxEnt RL) [7, 8, 51], and present one of our main theorems.

Theorem 2.1. *MaxEnt RL is a special case of MaxDiff RL with the added assumption that state transitions are decorrelated.*

Proof. Our goal in this proof will be to take the MaxDiff RL objective function in Eq. 44 and explore its relationship to the MaxEnt RL objective. We begin our proof by algebraically manipulating the MaxDiff RL objective function in Eq. 44:

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

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. 41. 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:

$$\begin{aligned}
&\leq E_{P_\pi}\left[\sum_{t=1}^{N} l(x_t,u_t)\right] + \sum_{t=1}^{N} E_{(x_t,u_t)\sim p,\pi}\Big[\alpha\log\pi(u_t|x_t)\Big] + \sum_{t=1}^{N} E_{x_t\sim p}\left[\alpha\log\frac{E_{u_t\sim\pi}[p(x_{t+1}|x_t,u_t)]}{p_{max}(x_{t+1}|x_t)}\right] \\
&\leq E_{P_\pi}\left[\sum_{t=1}^{N} l(x_t,u_t)\right] + \sum_{t=1}^{N} E_{(x_t,u_t)\sim p,\pi}\Big[\alpha\log\pi(u_t|x_t)\Big] + \sum_{t=1}^{N} E_{x_t\sim p}\left[\alpha\log\frac{p_\pi(x_{t+1}|x_t)}{p_{max}(x_{t+1}|x_t)}\right] \\
&\leq E_{P_\pi}\left[\sum_{t=1}^{N} l(x_t,u_t) + \alpha\log\pi(u_t|x_t) + \alpha D_{KL}\big(p_\pi(x_{t+1}|x_t)\big|\big|p_{max}(x_{t+1}|x_t)\big)\right],
\end{aligned} \tag{50}$$

where we also used the definition of $p_\pi(x_{t+1}|x_t)$ from Eq. 46.

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 paths, or in general. Our derivation throughout Supplementary Note 1 achieves this in the context of agents with continuous paths. Therein, we proved that the least-correlated continuous agent paths uniquely satisfy maximally diffusive 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 2.1). Given full controllability, the optimum of Eq. 50 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. 50 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:

$$\pi^* = \underset{\pi}{\text{argmin}}\, E_{(x_{1:N},u_{1:N})\sim P_\pi}\Big[\sum_{t=1}^{N}\hat{l}_c(x_t,u_t)\Big], \tag{51}$$

with

$$\hat{l}_c(x_t,u_t) = l(x_t,u_t) + \alpha\log\pi(u_t|x_t), \tag{52}$$

or equivalently, we can write Eq. 51 as a maximization by replacing the cost with a reward function:

$$\hat{r}_c(x_t,u_t) = r(x_t,u_t) + \alpha\mathcal{H}(\pi(u_t|x_t)), \tag{53}$$

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 correlations in their state transitions, which includes all physically-embodied agents, as well as many disembodied agents. Moreover, this also proves that maximizing policy entropy does not decorrelate state transitions in general. □

In contrast to MaxEnt RL, when the system induces temporal correlations the MaxDiff RL objective continues to prioritize effective exploration by decorrelating state transitions and encouraging the system to satisfy maximally diffusive trajectory statistics. 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 agent's properties do not induce temporal correlations on their own. Moreover, when the agent is capable of satisfying $D_{KL}(p_\pi|p_{max}) = 0$, the agent's state transition dynamics cancel out of the MaxDiff RL objective in Eq. 44. This suggests that successful MaxDiff RL policies will achieve a kind of generalizability across agent embodiments, as we illustrated in Figure 4 of the main text and in Supplementary Movie 3. This should not come as a complete surprise based on our analysis in Supplementary Note 1.6, which shows that the maximum likelihood paths of maximally diffusive dynamics evolve along gradients that optimize their controllability.

An interesting aside is that the MaxDiff RL objective formally requires model-based techniques to optimize because of its dependence on the system's transition model. 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. 44, it is technically possible to extend our results to model-free algorithms, as we show in the following sections.

2.3 Alternative synthesis approach via entropy maximization

For convenience, but without lack of generality, here we begin by limiting ourselves to deriving controllers in the absence of a potential or cost function. In Supplementary Note 2.1, we derived a synthesis approach based on KL control that optimizes exploration and task performance by making agents satisfy maximally diffusive trajectory statistics. Alternatively, we can use the fact that in Supplementary Note 1 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 via infinite velocities. 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,

$$\underset{u(t)}{\text{argmax}}\ S[P_{u(t)}[x(t)]], \tag{54}$$

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:

$$\begin{aligned}
&\max_{u(t)} S[P_{u(t)}[x(t)]], \quad \max_{u_{1:N-1}} S\Big[\prod_{t=1}^{N} p(x_{t+1}|x_t, u_t)\Big], \\
&\max_{\pi} S[P_{\pi}[x(t), u(t)]], \quad \max_{\pi} S\Big[\prod_{t=1}^{N} p(x_{t+1}|x_t, u_t)\pi(u_t|x_t)\Big],
\end{aligned} \tag{55}$$

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 maximum entropy formulation we have produced in this section, leads to

$$\begin{aligned}
&\underset{u(t)}{\text{argmin}}\ E_{P_{u(t)}}\big[L[x(t), u(t)]\big] - \alpha S[P_{u(t)}[x(t)]], \\
&\underset{\pi}{\text{argmin}}\ E_{P_{\pi}}\big[L[x(t), u(t)]\big] - \alpha S[P_{\pi}[x(t), u(t)]]
\end{aligned} \tag{56}$$

and

$$\begin{aligned}
&\underset{u_{1:N}}{\text{argmin}}\ E_{P_{u_{1:N}}}\Big[\sum_{t=1}^{N} l(x_t, u_t) - \alpha S[p(x_{t+1}|x_t, u_t)]\Big], \\
&\underset{\pi}{\text{argmin}}\ E_{P_{\pi}}\Big[\sum_{t=1}^{N} l(x_t, u_t) - \alpha S[p(x_{t+1}|x_t, u_t)\pi(u_t|x_t)]\Big]
\end{aligned} \tag{57}$$

also being formally equivalent to Eq. 36. While the different objectives listed in Eqs. 55-57 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. 55-57, but is better suited to model-free optimizations.

2.4 Simplified synthesis via local entropy maximization

As currently written, optimizing any of the objectives specified thus far requires access to a model capable of assessing the likelihood of our stochastic control process' trajectories. 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 optimally diffusive transition model. We formalize this optimistic assumption by asserting that our agent's path statistics are of the following form,

$$P^{L}_{u(t)}[x(t)] = \frac{1}{Z}\exp\Big[-\frac{1}{2}\int_{-\infty}^{\infty} \dot{x}(t)^T \mathbf{C}^{-1}_{u(t)}[x(t)]\dot{x}(t)dt\Big], \tag{58}$$

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 $S[P^{L}_{u(t)}[x(t)]]$ instead of the more general $S[P_{u(t)}[x(t)]]$, we merely change the direction from which our system approaches the true variational optimum. Furthermore, we note that it is still the case that achieving the true optimum is only possible when the system is controllable (see Supplementary Note 1.1).

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 a finite path as a collection of N random variables, where $x_{1:N}$ is the collection of variables comprising the path, and the conditional measures are as described in previous sections. We only care to describe $P_{max}[x_{1:N}]$ up

to proportionality because constant offsets will not affect the behavior of the optimal controller. To proceed, we make use of the chain rule of the conditional entropies of random variables. For the reader's convenience, we state the chain rule as it is commonly formulated below:

$$S[P[x_{1:N}]] = \sum_{t=1}^{N} S[p(x_{t+1}|x_{1:t})]. \tag{59}$$

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

$$S[P_{max}[x_{1:N}]] = \sum_{t=1}^{N} S[p_{max}(x_{t+1}|x_t)] \propto \frac{1}{2}\sum_{t=1}^{N} \log\det \mathbf{C}[x_t], \tag{60}$$

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.

Our expression for the entropy of the maximally diffusive trajectory distribution is of a simple form that only depends on the optimal covariance statistics of the process locally. The matrix $\mathbf{C}[\cdot]$ expresses the optimal covariance statistics achievable within the constraints imposed on the local rate of exploration of a given system, which in many cases are formally related to their controllability properties. Thus, given trajectory statistics $P^L_{u(t)}[x(t)]$, matching the entropy of the maximally diffusive trajectory distribution 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 is only possible when the system is controllable. To achieve this we can optimize either,

$$\underset{u(t)}{\text{argmax}}\ \frac{1}{2}\int_{-\infty}^{\infty} \log\det \mathbf{C}_{u(t)}[x(t)]dt, \quad \text{or} \quad \underset{\pi}{\text{argmax}}\ E_{(x_{1:N},u_{1:N})\sim P_\pi}\Big[\frac{1}{2}\sum_{t=1}^{N} \log\det \mathbf{C}_{\pi}[x_t]\Big]. \tag{61}$$

When a potential or cost encoding a task is introduced we instead have,

$$\begin{aligned} &\underset{u(t)}{\text{argmin}}\ \langle V[x(t)]\rangle_{P^L_{u(t)}} - \frac{\alpha}{2}\int_{-\infty}^{\infty} \log\det \mathbf{C}_{u(t)}[x(t)]dt, \quad \text{or} \\ &\underset{u(t)}{\text{argmin}}\ E_{P^L_{u(t)}}\big[L[x(t),u(t)]\big] - \frac{\alpha}{2}\int_{-\infty}^{\infty} \log\det \mathbf{C}_{u(t)}[x(t)]dt, \end{aligned} \tag{62}$$

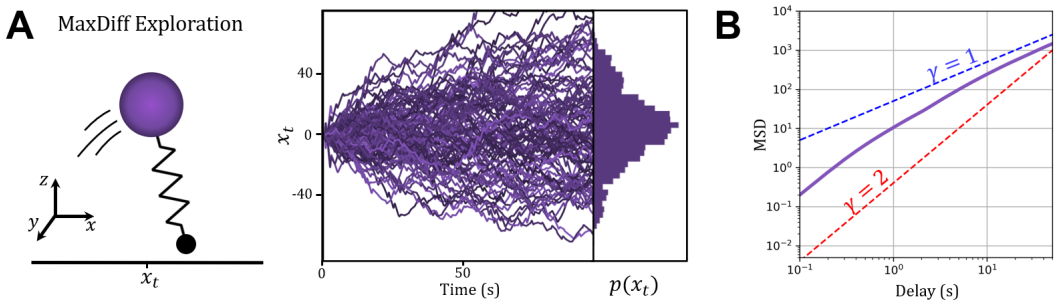
or their discretized variants expressed with respect to policies instead of controllers.

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

$$\underset{\pi}{\text{argmax}}\ E_{(x_{1:N},u_{1:N})\sim P_\pi}\Big[\sum_{t=1}^{N} r(x_t,u_t) + \frac{\alpha}{2}\log\det \mathbf{C}_{\pi}[x_t]\Big], \tag{63}$$

which, again, satisfies the same optimum as our previous objectives and is formally equivalent to them within a variational neighborhood of the optimum. This objective is the one that we used to derive all results in the main text. While it may seem that evaluating $\mathbf{C}_{\pi}[x_t]$ still requires access to predictive system rollouts in a model-based fashion, we first note that $\mathbf{C}_{\pi}[x_t]$ can be empirically estimated from data, and second that local autoregressive estimates of the agent's state covariance statistics can also be used instead. Thus, one could alternatively define $\hat{\mathbf{C}}[x_t] = Cov[x_{t-w:t}]$ in terms of short trajectory windows from time $t-w$ to the current time t for all $t - w > 0$—in other words, looking backward in time instead of forward. So long as samples from this window represent a sufficiently small δt of the system's state transition history, this poses no issues to our optimization and the theoretical guarantees our framework provides. In this sense, the MaxDiff RL objective in Eq. 63 can be implemented in model-free settings.

While we will explore the properties of MaxDiff trajectory synthesis in a variety of problem domains outside of deep RL in the following section, for now discuss the properties of the optimization problem in Eq. 61 (and by extension the one in Eq. 63). We begin by noting that this objective function is concave and submodular due to the properties of the log-determinant, which means that the optimization can be efficiently solved in polynomial time [14, 52]. Nonetheless, computational



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.

challenges arise due to the need to calculate the state covariance matrix and its determinant at every point along the agent's paths. Thus, here we list some details relevant to the numerics of implementing our objective function in an RL pipeline. First, the fact that maximum entropy sample paths are Markovian and that the objective is submodular means that, rather than optimizing the log-determinant over the entire integral, a practitioner may break down the integral and instead optimize the objective locally in an iterative fashion. Second, in practice we may not always have guarantees on the full-rankness of the covariance matrix, which would make its determinant evaluate to zero, thereby creating numerical stability issues for the resulting algorithm. To remedy this (without coordinate changes), 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 $\mathbf{C}[x(t)]$. However, it is important to note that this effectively restricts the exploration to an M-dimensional subspace of the full domain. Finally, we note that one can optimize the logarithm of the trace of $\mathbf{C}[x(t)]$ as an approximation that drastically reduces the complexity of computing the determinant in high dimensional optimizations. However, this approximation can only formally produce equivalent results to the log-determinant when system states vary independently from one another (i.e., when $\mathbf{C}[x(t)]$ is diagonal), which is generally not the case. Nonetheless, this assumption is routinely made out of convenience in much of the conditional variational autoencoder literature (e.g., [53]), so it may be of help to a practitioner at the cost of some added distance to the assumptions underlying our formal guarantees.

2.5 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 obey maximally diffusive statistics via MaxDiff trajectory synthesis 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 non-smooth 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 tracks whether the system is in contact with the ground or in the air. The SLIP dynamics are the following:

$$\dot{x} = f(x,u) = \begin{cases} f_c(x,u), & \text{if } l_c < l_0 \\ f_a(x,u), & \text{otherwise} \end{cases},$$

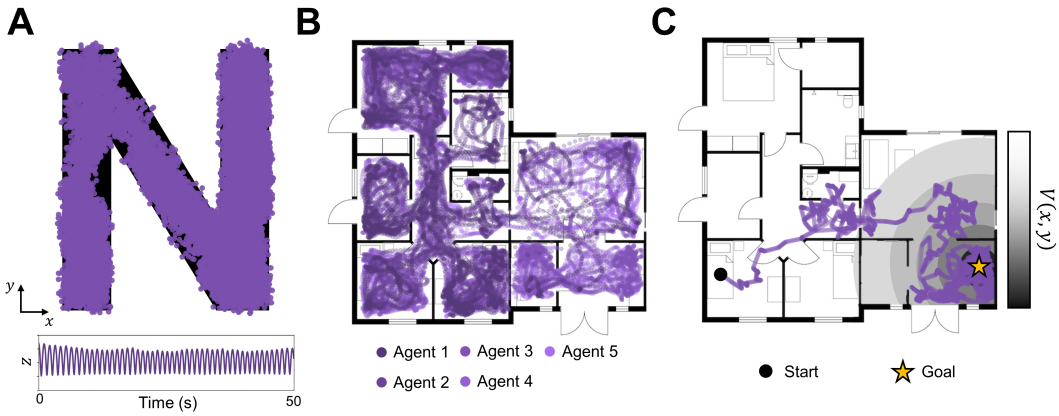
$$f_c(x,u) = \begin{bmatrix} \dot{x}_h \\ \frac{(k(l_0-l_s)+u_c)(x_h-x_t)}{ml_c} \\ \dot{y}_h \\ \frac{(k(l_0-l_c)+u_c)(y_h-y_t)}{ml_c} \\ \dot{z}_h \\ \frac{(k(l_0-l_c)+u_c)(z_h-z_t)}{ml_c} - g \\ 0 \\ 0 \end{bmatrix}, \; f_a(x,u) = \begin{bmatrix} \dot{x}_h \\ 0 \\ \dot{y}_h \\ 0 \\ \dot{z}_h \\ -g \\ \dot{x}_h + u_{t_x} \\ \dot{y}_h + u_{t_y} \end{bmatrix}, \tag{64}$$

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 1. 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. 61. In Supplementary Fig. 3 we illustrate the results of this process. Supplementary Fig. 3(a) depicts 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,

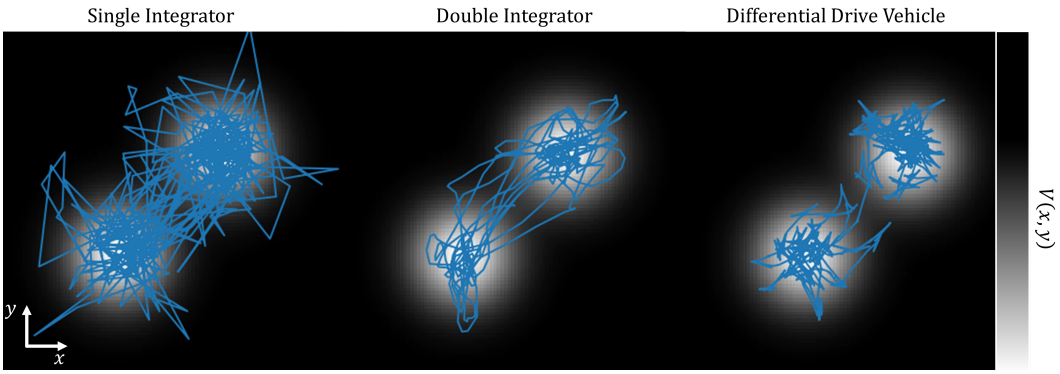


Supplementary Figure 4: **SLIP maximally diffusive exploration in various settings**. **a**, Undirected maximally diffusive exploration in a constrained **N**-shaped environment. The boundaries of the environment, as well as safety constraints, are established through the use of control barrier functions, 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.

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 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,

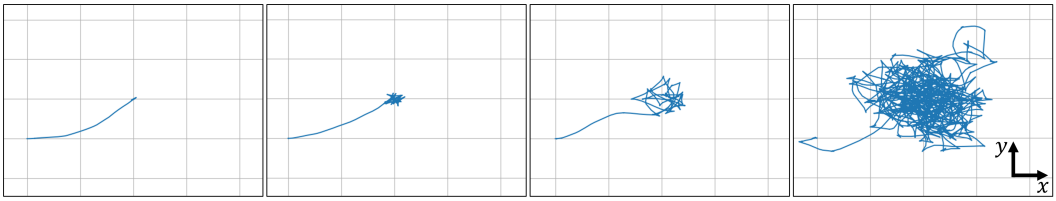


Supplementary Figure 5: **Directed maximally diffusive exploration of bimodal potential across systems**. (left panel) The single integrator is a linear system whose velocities are directly determined by the controller. Hence, its sample paths behave exactly as free Brownian particles in a potential. (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.

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 [34]. 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 Eqs. 61 and 62, 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 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.



$\alpha = 0$ $\alpha = 1$ $\alpha = 10$ $\alpha = 100$

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.

3 Reinforcement learning implementation details

3.1 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 models 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 networks. Both the reward function and transition model are optimized using Adam [62].The model 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 [8], 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 [8], 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.

3.2 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.

3.3 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` model 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 model is 1000 and modified tail density is 100.

3.4 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.

3.5 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.

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
	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
	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	$[x, y, \dot{x}, \dot{y}]$	$[x, y, \dot{x}, \dot{y}]$	$[x, y, z]$	$[x, y, \dot{x}, \dot{y}]$
		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		$[x, y, \dot{x}, \dot{y}]$	$[x, y, \dot{x}, \dot{y}]$	$[x, y, \dot{x}, \dot{y}]$
		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 parameters. All weights are diagonal matrices with the values specified.

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.

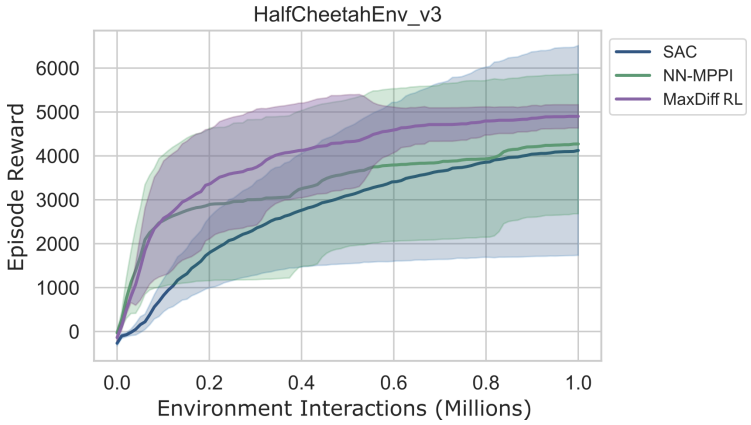
Movie 2: **Robustness of MaxDiff RL across random seeds.** Here, we depict an application of MaxDiff RL to MuJoCo's swimmer environment, comparing with alternative state-of-the-art MaxEnt RL algorithms, NN-MPPI and SAC. We observe that the performance of MaxDiff RL beats the state-of-the-art and does 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: **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 RL algorithms train with a system with a given set of physical properties, and are deployed on a system with different properties than those trained. We find that unlike alternative approaches, MaxDiff RL remains task capable across agent embodiments.

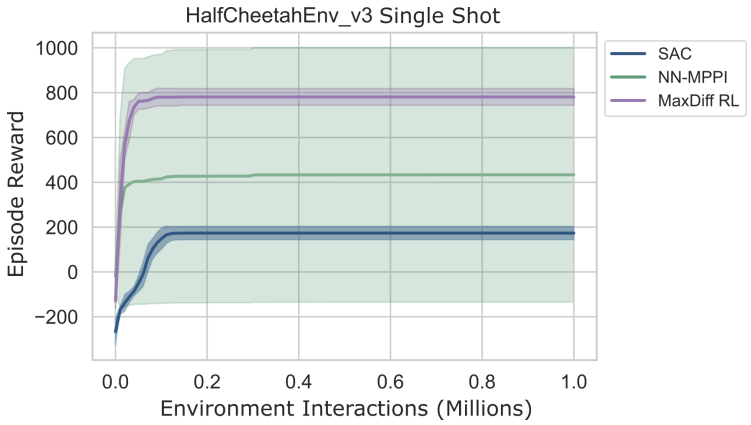
Movie 4: **Single-shot learning in MaxDiff RL agents.** Here, we depict an application of MaxDiff RL to MuJoCo's swimmer environment under a significant modification. Agents are unable to reset their environment, which requires all algorithms to learn to solve the task in a single deployment. First, we show representative snapshots of agents using models learned in single-shot deployments, and observe that MaxDiff RL still achieves state-of-the-art seed-invariant performance. Then, for MaxDiff RL we also show a complete playback of a single representative single-shot learning trial. We stagger the playback so 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 across 10 swimmers. In doing so, we can visualize the entirety of the single-shot learning process in real time.

*All supplementary movies are available in the following YouTube Playlist.

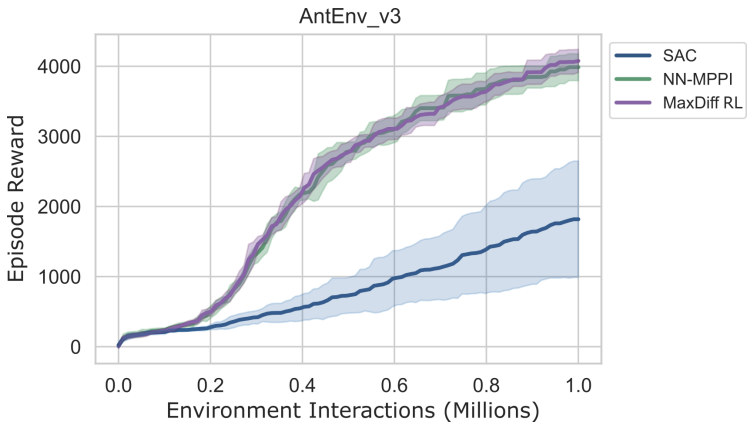
Supplementary figures



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 (10 seeds each).



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 (10 seeds each).



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 (10 seeds each). 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.

Supplementary references

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[2] Annalisa T. Taylor, Thomas A. Berrueta, and Todd D. Murphey. Active learning in robotics: A review of control principles. *Mechatronics*, 77:102576, 2021. doi: 10.1016/j.mechatronics.2021.102576.

[3] Julian Ibarz, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. How to train your robot with deep reinforcement learning: Lessons we have learned. *The International Journal of Robotics Research*, 40(4-5):698–721, 2021. doi: 10.1177/0278364920987859.

[4] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022. doi: 10.1126/scirobotics.abk2822.

[5] Michael Bloesch, Jan Humplik, Viorica Patraucean, Roland Hafner, Tuomas Haarnoja, Arunkumar Byravan, Noah Yamamoto Siegel, Saran Tunyasuvunakool, Federico Casarini, Nathan Batchelor, Francesco Romano, Stefano Saliceti, Martin Riedmiller, S. M. Ali Eslami, and Nicolas Heess. Towards real robot learning in the wild: A case study in bipedal locomotion. *Proceedings of the 5th Conference on Robot Learning*, 164:1502–1511, 2022.

[6] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, 2002.

[7] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. *Proceedings of the International Conference on Machine Learning (ICML)*, 70:1352–1361, 2017.

[8] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 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.

[9] Oswin So, Ziyi Wang, and Evangelos A. Theodorou. Maximum entropy differential dynamic programming. In *2022 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3422–3428, 2022. doi: 10.1109/ICRA46639.2022.9812228.

[10] Eduardo D. Sontag. *Mathematical Control Theory: Deterministic Finite Dimensional Systems*, volume 6. Springer, 2013. ISBN 9781461205777.

[11] J. P. Hespanha. *Linear Systems Theory: Second Edition*. Princeton University Press, 2018. ISBN 9780691179575.

[12] E. D. Sontag. *Kalman's Controllability Rank Condition: From Linear to Nonlinear*, pages 453–462. Springer, 1991. ISBN 9783662085462. doi: 10.1007/978-3-662-08546-2_25.

[13] Fabrizio L. Cortesi, Tyler H. Summers, and John Lygeros. Submodularity of energy related controllability metrics. In *2014 IEEE Conference on Decision and Control (CDC)*, pages 2883–2888, 2014. doi: 10.1109/CDC.2014.7039832.

[14] Tyler H. Summers, Fabrizio L. Cortesi, and John Lygeros. On submodularity and controllability in complex dynamical networks. *IEEE Transactions on Control of Network Systems*, 3(1): 91–101, 2016. doi: 10.1109/TCNS.2015.2453711.

[15] Kenji Kashima. Noise response data reveal novel controllability Gramian for nonlinear network dynamics. *Scientific Reports*, 6(1):27300, 2016.

[16] Hannes Risken. Fokker-planck equation. In *The Fokker-Planck Equation*, pages 63–95. Springer, 1996. ISBN 978-3-642-96807-5.

[17] D. Mitra. *W* matrix and the geometry of model equivalence and reduction. *Proceedings of the Institution of Electrical Engineers*, 116:1101–1106, 1969.

[18] Anastasios Tsiamis and George J. Pappas. Linear systems can be hard to learn. *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 2903–2910, 2021. doi: 10.1109/CDC45484.2021.9682778.

[19] Anastasios Tsiamis, Ingvar M Ziemann, Manfred Morari, Nikolai Matni, and George J. Pappas. Learning to control linear systems can be hard. In *Proceedings of 35th Conference on Learning Theory (COLT)*, volume 178, pages 3820–3857, 2022.

[20] Robert G. Gallager. *Stochastic Processes: Theory for Applications*. Cambridge University Press, 2013. ISBN 9781107039759.

[21] Marian Grendar. Entropy and effective support size. *Entropy*, 8(3):169–174, 2006. doi: 10.3390/e8030169.

[22] E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106:620–630, 1957.

[23] Purushottam D. Dixit, Jason Wagoner, Corey Weistuch, Steve Pressé, Kingshuk Ghosh, and Ken A. Dill. Perspective: Maximum caliber is a general variational principle for dynamical systems. *The Journal of Chemical Physics*, 148(1):010901, 2018.

[24] Pavel Chvykov, Thomas A. Berrueta, Akash Vardhan, William Savoie, Alexander Samland, Todd D. Murphey, Kurt Wiesenfeld, Daniel I. Goldman, and Jeremy L. England. Low rattling: A predictive principle for self-organization in active collectives. *Science*, 371(6524):90–95, 2021.

[25] Mehran Kardar. *Statistical Physics of Fields*. Cambridge University Press, 2007. ISBN 9780511815881.

[26] Calvin C. Moore. Ergodic theorem, ergodic theory, and statistical mechanics. *Proceedings of the National Academy of Sciences*, 112(7):1907–1911, 2015. doi: 10.1073/pnas.1421798112.

[27] Richard P. Feynman, Albert R. Hibbs, and Daniel F. Styer. *Quantum Mechanics and Path Integrals*. Dover Publications, 2010.

[28] Xavier Michalet and Andrew J Berglund. Optimal diffusion coefficient estimation in single-particle tracking. *Physical Review E*, 85(6):061916, 2012.

[29] Denis Boyer, David S. Dean, Carlos Mejía-Monasterio, and Gleb Oshanin. Optimal estimates of the diffusion coefficient of a single Brownian trajectory. *Physical Review E*, 85(3):031136, 2012.

[30] Sebastian B. Thrun. Efficient exploration in reinforcement learning. Technical report, Carnegie Mellon University, 1992.

[31] L. M. Miller, Y. Silverman, M. A. MacIver, and T. D. Murphey. Ergodic exploration of distributed information. *IEEE Transactions on Robotics*, 32(1):36–52, 2016.

[32] A. Mavrommati, E. Tzorakoleftherakis, I. Abraham, and T. D. Murphey. Real-time area coverage and target localization using receding-horizon ergodic exploration. *IEEE Transactions on Robotics*, 34(1):62–80, 2018.

[33] Konrad Rawlik, Marc Toussaint, and Sethu Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference. *Proceedings of Robotics: Science and Systems (RSS)*, pages 353–361, 2012.

[34] Xudong Wang, Weihua Deng, and Yao Chen. Ergodic properties of heterogeneous diffusion processes in a potential well. *The Journal of Chemical Physics*, 150(16):164121, 2019. doi: 10.1063/1.5090594.

[35] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning, (2nd Edition)*. Adaptive Computation and Machine Learning series. MIT Press, 2018. ISBN 9780262351362.

[36] Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. On the sample complexity of the linear quadratic regulator. *Foundations of Computational Mathematics*, 20(4):633–679, 2020. doi: 10.1007/s10208-019-09426-y.

[37] Cosma Shalizi and Aryeh Kontorovich. Predictive pac learning and process decompositions. *Advances in Neural Information Processing Systems (NeurIPS)*, 26, 2013.

[38] Alexander L Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L Littman. Pac model-free reinforcement learning. *Proceedings of the 23rd International Conference on Machine learning (ICML)*, pages 881–888, 2006.

[39] Tor Lattimore and Marcus Hutter. PAC bounds for discounted MDPs. In Nader H. Bshouty, Gilles Stoltz, Nicolas Vayatis, and Thomas Zeugmann, editors, *Algorithmic Learning Theory*, pages 320–334. Springer, 2012. ISBN 978-3-642-34106-9.

[40] Mohammad Gheshlaghi Azar, Rémi Munos, and Hilbert J. Kappen. Minimax PAC bounds on the sample complexity of reinforcement learning with a generative model. *Machine Learning*, 91(3):325–349, 2013. doi: 10.1007/s10994-013-5368-1.

[41] Akshay Krishnamurthy, Alekh Agarwal, and John Langford. PAC reinforcement learning with rich observations. *Advances in Neural Information Processing Systems (NeurIPS)*, 29, 2016.

[42] Andrew J. Wagenmaker, Max Simchowitz, and Kevin Jamieson. Beyond no regret: Instance-dependent PAC reinforcement learning. *Proceedings of 35th Conference on Learning Theory (COLT)*, 178:358–418, 2022.

[43] John C. Duchi, Alekh Agarwal, Mikael Johansson, and Michael I. Jordan. Ergodic mirror descent. *SIAM Journal on Optimization*, 22(4):1549–1578, 2012.

[44] Angelia Nedic and Dimitri P. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM Journal on Optimization*, 12(1):109–138, 2001. doi: 10.1137/S1052623499362111.

[45] Yurii E. Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269:543–547, 1983.

[46] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145–151, 1999.

[47] Hedy Attouch and Alexandre Cabot. Asymptotic stabilization of inertial gradient dynamics with time-dependent viscosity. *Journal of Differential Equations*, 263(9):5412–5458, 2017. doi: https://doi.org/10.1016/j.jde.2017.06.024.

[48] Hedy Attouch, Zaki Chbaniand, and Hassan Riahi. Rate of convergence of the Nesterov accelerated gradient method in the subcritical case $\alpha \leq 3$. *ESAIM: COCV*, 25:2, 2019. doi: 10.1051/cocv/2017083.

[49] Emanuel Todorov. Linearly-solvable markov decision problems. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 19, pages 1369–1376. MIT Press, 2007.

[50] Emanuel Todorov. Efficient computation of optimal actions. *Proceedings of the National Academy of Sciences*, 106(28):11478–11483, 2009. doi: 10.1073/pnas.0710743106.

[51] Benjamin Eysenbach and Sergey Levine. Maximum entropy RL (provably) solves some robust RL problems. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

[52] Alex Kulesza and Ben Taskar. *Determinantal Point Processes for Machine Learning*. Now Publishers Inc., 2012. ISBN 1601986289.

[53] Ahalya Prabhakar and Todd Murphey. Mechanical intelligence for learning embodied sensor-object relationships. *Nature Communications*, 13(1):4108, 2022. doi: 10.1038/s41467-022-31795-2.

[54] Christian Scholz, Soudeh Jahanshahi, Anton Ldov, and Hartmut Löwen. Inertial delay of self-propelled particles. *Nature Communications*, 9(1):5156, 2018. doi: 10.1038/s41467-018-07596-x.

[55] Manoj Srinivasan and Andy Ruina. Computer optimization of a minimal biped model discovers walking and running. *Nature*, 439(7072):72–75, 2006. doi: 10.1038/nature04113.

[56] Alexander R. Ansari and Todd D. Murphey. Sequential action control: Closed-form optimal control for nonlinear and nonsmooth systems. *IEEE Transactions on Robotics*, 32(5):1196–1214, 2016. doi: 10.1109/TRO.2016.2596768.

[57] A. Ames, J. Grizzle, and P. Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *2014 IEEE Conference on Decision and Control (CDC)*, 2014.

[58] Wolfgang Kliemann. Recurrence and invariant measures for degenerate diffusions. *Annals of Probability*, 15(2):690–707, 1987.

[59] Nawaf Bou-Rabee and Houman Owhadi. Ergodicity of langevin processes with degenerate diffusion in momentums. *International Journal of Pure and Applied Mathematics*, 45(3): 475–490, 2008.

[60] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

[61] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

[62] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[63] Justin A. Boyan. Least-squares temporal difference learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 49–56, 1999.

[64] Doina Precup, Richard S. Sutton, and Sanjoy Dasgupta. Off-policy temporal-difference learning with function approximation. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 417–424, 2001.