SPATIAL BUT NOT TEMPORAL REWARD SPARSITY DRIVES SLOW POLICY LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

A primary difference between reinforcement and supervised learning is that rewards are generally sparse in both space and time: few actions yield high reward (spatial sparsity), and rewards may only be provided after temporally extended sequences of actions (temporal sparsity). How these factors impact policy learning dynamics is undertheorized, especially because it is difficult to design a theoretical setting interesting enough to capture relevant phenomenology, but simple enough to treat analytically. In this paper, we propose and analytically solve a minimal model of learning to pursue a moving target which naturally incorporates both spatial and temporal reward sparsity. We show mathematically that spatial sparsity can produce exponential slowdowns in learning, and that temporal sparsity is comparatively benign, only yielding a comparatively short additional phase of pursuit trajectory refinement. We also show that the presence of highly variable environmental state features can—without complex function approximators—produce an implicit curriculum effect which makes learning from sparse rewards polynomially rather than exponentially slow. We provide a wealth of relevant calculations, whose utility is in part that they sharpen our intuitions about reward sparsity.

1 INTRODUCTION

Reward sparsity is one of the primary factors that makes reinforcement learning (RL) (Sutton & Barto, 2018a) generally more difficult than supervised learning (Andrychowicz et al., 2017; Hare, 2019; Ladosz et al., 2022). All else being equal, agents that are provided more frequent and informative feedback about which actions are 'good' ought to learn more quickly. But there are multiple types of reward sparsity, and how these affect learning is poorly understood.

The first, which we call *spatial reward sparsity*, concerns the specificity of 'good' actions: few actions yield high reward. This is true, for example, in robotics and continuous control settings with large action spaces. The second, which we call *temporal reward sparsity*, concerns how often agents are rewarded in temporally extended tasks: reward may only be received after sequences of actions, which produces a temporal credit assignment problem. Episodic tasks where agents receive a reward only at the end, and not at any intermediate time step, are sparse in this sense. This is true of most games, since without auxiliary rewards, agents only receive a reward signal (a win or loss) at the end. Importantly, either kind of reward sparsity can be present without the other.

How do these two types of reward sparsity affect policy learning? Does one cause more difficulties, in the sense of making learning harder or slower, than the other? Existing theoretical work (Saglietti et al., 2022; Lee et al., 2024; Patel et al., 2025) does not usually distinguish between them, and so is inconclusive—in part because it is generally difficult to conduct an exhaustive mathematical analysis unless the setting is taken to be extremely simple.

In this paper, we study the impact of spatial and temporal reward sparsity in the context of a tractable mathematical model of learning to pursue a moving target. This setting is natural because it is easy to implement both types of reward sparsity: the agent must overlap with the target to receive high reward (*spatial sparsity*), and is rewarded mostly based on their final position rather than individual actions (*temporal sparsity*). Although the model we consider is simple—it is linear in various senses—its phenomenology is surprisingly rich, and one can precisely characterize various aspects of its learning dynamics. Overall, we find that spatial reward sparsity determines the main bottleneck of learning when both spatial and temporal sparsity are present. Temporal sparsity mainly yields a

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

separation between when the agent learns to reliably reach the target, and when they refine their trajectory en route to the target, but the convergence rate of the refinement phase remains exponentially fast regardless of the number of time steps. Meanwhile, unless something explicitly or implicitly addresses it, spatial sparsity can yield exponential slowdowns in learning speed. Aside from introducing a theoretical setting in which spatial and temporal reward sparsity can be disentangled and analytically studied, our main contributions are as follows:

- **Spatial sparsity yields slow learning.** We show sparse rewards can cause small differences in initial conditions to produce large increases in learning time, and that this issue worsens with increasing spatial, but not temporal, task dimensionality. (Sec. 3)
- **Temporal sparsity yields trajectory refinement.** We show that temporal sparsity causes agents to learn to reach the target before they optimize their pursuit trajectories, and that the time between these phases is determined by a certain matrix's eigenvalues. (Sec. 4)
- **Learning dynamics yields implicit curriculum effect.** We show that, when there are highly variable environmental state features, an effect due to learning dynamics makes learning polynomially rather than exponentially slow. (Sec. 5 and 6)

Together, these results quantitatively sharpen our intuitions about how sparsity slows learning. They also show that slowdowns can be overcome by explicit (Sec. 5) or implicit (Sec. 6) mechanisms.

2 Mathematical formulation

In this paper, we study reward sparsity in the context of an episodic target pursuit task: the agent must pursue a possibly moving target in $D \geq 1$ spatial dimensions. Our central concern in designing this theoretical setting was incorporating both spatial and temporal reward sparsity while keeping the dynamics of policy learning analytically tractable. Below, we introduce and motivate the task.

Task structure. Each episode lasts for a fixed amount of time $T > 0$ and consists of M time steps of length $\Delta t := T/M$. First, the agent observes a *state vector* $\boldsymbol{z} := (z_1, ..., z_F)^T \in \mathbb{R}^F$ (where $F \geq 1$ is the number of features) that provides information about the target. While we treat this feature vector as an abstract quantity, it is intended to model, e.g., the target's velocity vector. Next, the agent submits a temporal sequence of M velocities $\boldsymbol{u}_0, \boldsymbol{u}_1, ..., \boldsymbol{u}_{M-1} \in \mathbb{R}^D$, after which they arrive at an *endpoint* $\boldsymbol{x}_T := \sum_{t=0}^{M-1} \boldsymbol{u}_t \Delta t \in \mathbb{R}^D$ that corresponds to their integrated velocity vector. Note that the task is *open-loop* since actions do not affect target behavior, which only depends on $\boldsymbol{z}$.

Finally, they are rewarded according to a return $G(\boldsymbol{u}_{0:T}, \boldsymbol{z})$ which depends on both state and the sequence $\boldsymbol{u}_{0:T}$ of control velocities. For simplicity, we assume that the state vector is sampled independently from $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ during each episode, where $\boldsymbol{\Sigma}$ is a fixed $F \times F$ covariance.

Reward structure. The agent's goal is for its endpoint $\boldsymbol{x}_T := \sum_{t=0}^{M-1} \boldsymbol{u}_t \Delta t$ to match the *target endpoint* $\boldsymbol{f}_T(\boldsymbol{z}) \in \mathbb{R}^D$. We assume the return they receive has two components: an *effort term* which penalizes high velocities $\boldsymbol{u}_t$, and a *target proximity term* that incentivizes reaching the target:

$$G(\boldsymbol{u}_{0:T}, \boldsymbol{z}) = G_{\text{effort}} + G_{\text{target}} = \sum_{t=0}^{M-1} -\frac{\alpha}{2} \|\boldsymbol{u}_t\|_2^2 \, \Delta t + g \left(\frac{\kappa}{2\pi}\right)^{D/2} e^{-\frac{\kappa}{2}\|\boldsymbol{x}_T - \boldsymbol{f}_T(\boldsymbol{z})\|_2^2}. \quad (1)$$

Here, $\alpha > 0$ controls the cost of each velocity input $\boldsymbol{u}_t$, $g > 0$ controls the overall size of the target proximity term, and κ controls the *sparsity* of rewards. For large κ, the second term essentially becomes a delta function centered on the target; for small κ, it becomes effectively quadratic. A major objective of our work is to characterize the impact of large κ on learning dynamics.

This task exhibits **spatial sparsity** in the sense that the target is localized in space, and the agent must end the episode within a distance of $1/\sqrt{\kappa}$ from it to receive high reward. It exhibits **temporal sparsity** in the sense that there is a credit assignment problem: most reward depends not on single actions, but essentially on the *average* action, since this is what determines $\boldsymbol{x}_T := \sum_{t=0}^{M-1} \boldsymbol{u}_t \Delta t$.

Target and agent movement structure. We assume that the target endpoint depends linearly on $\boldsymbol{z}$, i.e., $\boldsymbol{f}_T(\boldsymbol{z}) = \boldsymbol{b}^f + \boldsymbol{W}^f \boldsymbol{z}$, where $\boldsymbol{b}^f \in \mathbb{R}^D$ is the *target bias* and $\boldsymbol{W}^f \in \mathbb{R}^{D \times F}$ is the *target weight*

matrix. The agent's average action is assumed to have a similar form, with $\bar{\boldsymbol{u}}(\boldsymbol{z}) = \frac{1}{T}\boldsymbol{b}_t + \frac{1}{T}\boldsymbol{W}_t\boldsymbol{z}$; the control velocity is determined by a *bias* $\boldsymbol{b}_t \in \mathbb{R}^D$ and *weight matrix* $\boldsymbol{W}_t \in \mathbb{R}^{D\times F}$, both of which are learned. Note that *actions associated with different time steps depend on different learned parameters* (for example, $\boldsymbol{b}_t$ vs $\boldsymbol{b}_{t'}$), which is unrealistic but important for tractability.

We assume that the agent's policy $\pi(\boldsymbol{u}_{0:T}|\boldsymbol{z})$ also involves *exploration noise*: during each time step, the action $\boldsymbol{u}_t$ is sampled independently of other actions according to a normal distribution:

$$\pi(\boldsymbol{u}_{0:T}|\boldsymbol{z}) = \prod_{t=0}^{M-1} \pi(\boldsymbol{u}_t|\boldsymbol{z}) \qquad \pi(\boldsymbol{u}_t|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{u}_t; \bar{\boldsymbol{u}}_t(\boldsymbol{z}), \frac{1}{\Delta t}\boldsymbol{B}_t^{-1}) \,. \tag{2}$$

The positive semidefinite matrices $\boldsymbol{B}_t^{-1} \in \mathbb{R}^{D\times D}$ control how exploratory the agent is; in Sec. 5, we will show that choosing them carefully can speed up learning.

Policy learning dynamics. We study an agent which learns a policy for performing this pursuit task via episodic REINFORCE (Williams, 1992; Sutton et al., 1999). If we assume a small learning rate and a large number of episodes, the learning dynamics associated with a parameter θ_t of the time step t mean action $\bar{\boldsymbol{u}}_t(\boldsymbol{z})$ can be approximately described (see Appendix A) by the ordinary differential equation (ODE)

$$\frac{d}{d\tau}\theta_t = \frac{\eta T}{\Delta t}\,\mathbb{E}_{\boldsymbol{z}}\left\{\frac{\partial \bar{\boldsymbol{u}}_t(\boldsymbol{z})^T}{\partial \theta_t}\boldsymbol{B}_t\Delta t\,\mathbb{E}_{\boldsymbol{u}_{0:T}|\boldsymbol{z}}\left[\,(\boldsymbol{u}_t - \bar{\boldsymbol{u}}_t(\boldsymbol{z}))\,G(\boldsymbol{u}_{0:T},\boldsymbol{z})\right]\right\}\,, \tag{3}$$

where $\eta > 0$ is the (effective) learning rate. Although we neglect value learning for simplicity, it would play no role in this setting, since (at least when value is used as a baseline, as in actor-critic methods) value functions do not influence *average* updates (Sutton & Barto, 2018b). Value functions are usually used to reduce variance in learning dynamics, but—under the assumptions that permit the above ODE description—there is no variance in parameter updates.

3 Spatial reward sparsity produces learning slowdown

We begin by studying *pure bias learning*, in which there are no features (i.e., $F = 0$), so $\bar{\boldsymbol{u}}_t = \boldsymbol{b}_t/T$ and $\boldsymbol{f}_T = \boldsymbol{b}^f$. This case is simple while illustrating the basic difficulties associated with sparse rewards—and as we discuss in Sec. 6, it is actually much *harder* to learn than the case with features. Our first question is this: how much do sparser rewards (i.e., higher κ) slow down learning?

Endpoint learning dynamics. Let $\boldsymbol{b} := \sum_{t=0}^{M-1}\boldsymbol{b}_t\frac{\Delta t}{T}$ denote the average learned *endpoint*, which ought to be close to the target endpoint $\boldsymbol{b}^f$ by the end of learning. After performing the average over the agent's action sequence $\boldsymbol{u}_{0:T}$ (see Appendix A), we find that learning dynamics depend on

$$p_{\text{hit}}(\boldsymbol{b}) := \sqrt{\frac{\det\tilde{\boldsymbol{B}}}{(2\pi)^D}}e^{-\frac{1}{2}(\boldsymbol{b}-\boldsymbol{b}^f)^T\tilde{\boldsymbol{B}}(\boldsymbol{b}-\boldsymbol{b}^f)} \quad \text{where} \quad \tilde{\boldsymbol{B}}^{-1} := \frac{1}{\kappa}\boldsymbol{I} + \sum_{t=0}^{M-1}\boldsymbol{B}_t^{-1}\Delta t \in \mathbb{R}^{D\times D}\,, \tag{4}$$

a quantity that we call the *hitting probability*. This quantity modulates the impact of reward sparsity: when it is small, the agent receives little informative reward signal, and learning takes significantly longer. The matrix $\tilde{\boldsymbol{B}}^{-1}$ determines the *effective target size* along each direction of $\mathbb{R}^D$: if it is smaller along a given axis, the agent must stop closer to the target to receive appreciable reward.

We derive the learning dynamics of $\boldsymbol{b}_t$ in Appendix A. But because the sparse reward term depends *only* on the learned endpoint $\boldsymbol{b} := \sum_{t=0}^{M-1}\boldsymbol{b}_t\frac{\Delta t}{T}$, and not on any specific $\boldsymbol{b}_t$, these dynamics are extremely similar to those of the endpoint. We will find it convenient to track $\boldsymbol{b}$ dynamics directly:

$$\frac{d}{d\tau}\boldsymbol{b} = \eta\left\{-\frac{\alpha}{T}\boldsymbol{b} + g\tilde{\boldsymbol{B}}(\boldsymbol{b}^f - \boldsymbol{b})\sqrt{\frac{\det\tilde{\boldsymbol{B}}}{(2\pi)^D}}e^{-\frac{1}{2}(\boldsymbol{b}-\boldsymbol{b}^f)^T\tilde{\boldsymbol{B}}(\boldsymbol{b}-\boldsymbol{b}^f)}\right\}\,. \tag{5}$$

The most salient characteristic of this D-dimensional dynamical system is that it has *two regimes*, which are determined by whether the bias error $E := (\boldsymbol{b}-\boldsymbol{b}^f)^T\tilde{\boldsymbol{B}}(\boldsymbol{b}-\boldsymbol{b}^f)$ is large or small.

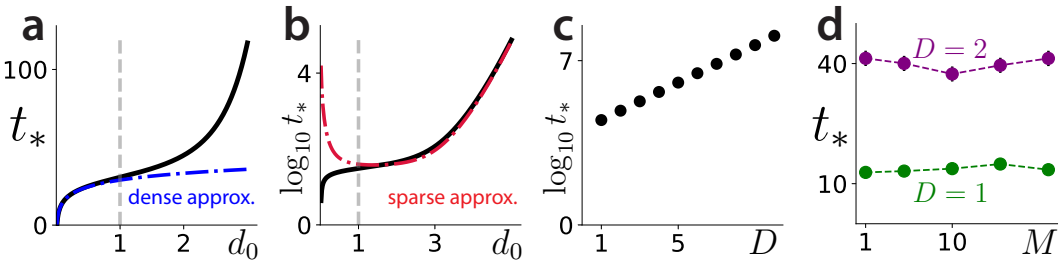

Figure 1: **Spatial sparsity produces exponential slowdown.** Bias units are chosen so that $\tilde{\beta} = 1$. **a.** The convergence time t_* associated with endpoint learning dynamics (black; see Eq. 8) scales poorly with the initial bias error d_0. For small enough d_0, it is approximately logarithmic in d_0 (blue curve); $d_0 = 1$ (gray line) is where the approximation breaks down. **b.** The convergence time is well-approximated as exponential (red; see Eq. 9) when d_0 is large. **c.** Log-convergence-time scales linearly with the spatial dimensionality D under typical assumptions. **d.** Convergence time is insensitive to the number of time steps M under typical assumptions. This panel shows an empirical result ($N = 500$ seeds) for $D = 1$ (green) and $D = 2$ (purple); standard errors are shown but small.

Dense reward regime. When E is somewhat less than 1, so that $\boldsymbol{b}$ is close to the target endpoint,

$$\frac{d}{d\tau}\boldsymbol{b} \approx \eta \left[-\frac{\alpha}{T}\boldsymbol{b} + g\tilde{\boldsymbol{B}}(\boldsymbol{b}^f - \boldsymbol{b}) \sqrt{\det \tilde{\boldsymbol{B}}/(2\pi)^D} \right] , \tag{6}$$

which implies that $\boldsymbol{b}$ converges exponentially quickly to the (unique) optimum, which is near $\boldsymbol{b}^f$ if α/T is small. This occurs on a time scale determined by the eigenvalues of $\frac{\alpha}{T}\boldsymbol{I} + g\sqrt{\det \tilde{\boldsymbol{B}}/(2\pi)^D}\tilde{\boldsymbol{B}}$. Exponentially fast convergence means that learning in this regime is 'easy'.

Sparse reward regime. When E is somewhat *greater* than 1, so that $\boldsymbol{b}$ is far from the target endpoint, the sparse reward term is exponentially suppressed and learning greatly slows down. If we make the simplifying assumption that $\tilde{\boldsymbol{B}} = \tilde{\beta}\boldsymbol{I}$ is isotropic, and assume α/T is small enough to be negligible, we can find (see Appendix B) that the bias error E evolves as

$$\frac{d}{d\tau}E = -2\eta g\tilde{\beta}\left(\frac{\tilde{\beta}}{2\pi}\right)^{D/2} E e^{-E/2} . \tag{7}$$

This ODE implies a convergence time result: the time to go from an initial error $\tilde{\beta}\|\boldsymbol{b}^0 - \boldsymbol{b}^f\|_2^2$ to a final (small) error ϵ^2 equals

$$t_* = \frac{1}{2\eta g\tilde{\beta}}\left(\frac{2\pi}{\tilde{\beta}}\right)^{D/2}\left[\mathrm{Ei}\left(\frac{\tilde{\beta}}{2}\|\boldsymbol{b}^0 - \boldsymbol{b}^f\|_2^2\right) - \mathrm{Ei}(\frac{1}{2}\epsilon^2) \right] , \tag{8}$$

where $\mathrm{Ei}(x) := \int_{-\infty}^{x} \frac{e^y}{y} dy$ is the *exponential integral function*. For small initial errors, t_* depends logarithmically on the initial error (Fig. 1a). But since $\mathrm{Ei}(x) \approx e^x/x$ for large arguments, when the initial endpoint is somewhat far from the target $\boldsymbol{b}^f$,

$$t_* \approx \frac{1}{\eta g\tilde{\beta}^2}\left(\frac{2\pi}{\tilde{\beta}}\right)^{D/2} \frac{e^{\frac{\tilde{\beta}}{2}\|\boldsymbol{b}^0 - \boldsymbol{b}^f\|_2^2}}{\|\boldsymbol{b}^0 - \boldsymbol{b}^f\|_2^2} , \tag{9}$$

where we have neglected the (much smaller) ϵ-dependent term. This result is significant because it says the convergence time scales *exponentially* with the initial bias error: small increases in the initial error will produce large increases in learning time (Fig. 1b). We conclude that reward sparsity—κ being large, which makes $\tilde{\beta}$ large—*substantially slows learning*.

Spatial sparsity, not temporal sparsity, controls exponential slowdown. From Eq. 9, it is clear that the initial endpoint error largely controls the slowness of learning. How does this depend on the spatial dimensionality D and number of time steps M?

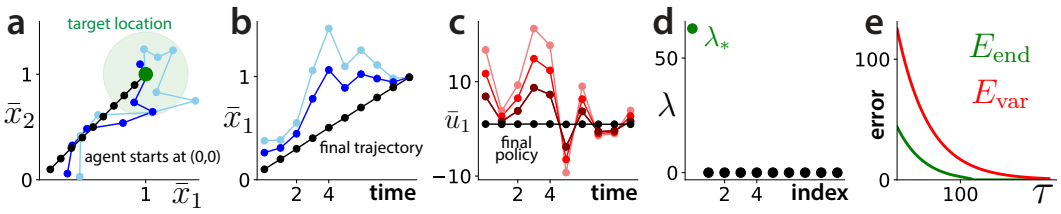

Figure 2: **Temporal reward sparsity delays trajectory refinement.** All simulations use $M = 10$ time steps and $D = 2$. **a.** Average learned trajectory (obtained by summing agent's selected velocities). Examples from three different parts of learning are shown: around when the agent first learns to hit the target (light blue), their trajectory is not straight; later, it becomes straighter (blue), and by the end of learning it is straight (black) since this minimizes the effort penalty. **b.** Same as in a., but trajectory (along the first direction only) visualized as a function of time step. **c.** As learning proceeds, the velocities the agent selects at each time step get increasingly similar (light red to dark red), and by the end they are identical (black). **d.** The matrix $\boldsymbol{A}$ controlling trajectory refinement has a 'gapped' spectrum with one outlier eigenvalue. **e.** Endpoint error (green) and trajectory variance (red) converge at different rates; the agent learns to reach the target before optimizing their path.

To ensure that the endpoint $\boldsymbol{x}_T := \sum_{t=0}^{M-1} \boldsymbol{u}_t \Delta t$ is well-behaved for large M, it is typical to initialize policy parameters so that the variance of $\boldsymbol{x}_T$ is independent of M. For example, choosing $b_{ti}^0 \sim \mathcal{N}(0, \xi^2(T/\Delta t))$ for all $t = 0, 1, ..., M-1$ and $i = 1, ..., D$, we have

$$\mathbb{E}_{\boldsymbol{b}_{0:T}^0}[\boldsymbol{x}_T \boldsymbol{x}_T^T] = \sum_{t,t'} \frac{(\Delta t)^2}{T^2} \mathbb{E}[\boldsymbol{b}_t^0 (\boldsymbol{b}_{t'}^0)^T] = \sum_{t,t'} \frac{(\Delta t)^2}{T^2} \delta_{t,t'} \xi^2 \frac{T}{\Delta t} \boldsymbol{I} = \xi^2 \boldsymbol{I} \, . \tag{10}$$

If we assume that components of the target location are chosen independently according to $b_i^f \sim \mathcal{N}(0, \xi_f^2)$, then a typical value of the initial error is

$$\tilde{\beta} \, \mathbb{E}[\|\boldsymbol{b}^0 - \boldsymbol{b}^f\|_2^2] = \tilde{\beta} \left\{ \mathbb{E}[\|\boldsymbol{b}^0\|_2^2] + \mathbb{E}[\|\boldsymbol{b}^f\|_2^2] \right\} = \tilde{\beta} D \left\{ \xi^2 + \xi_f^2 \right\} \, . \tag{11}$$

If we heuristically write the initial error as $\tilde{\beta} D d_0^2$, we finally obtain a convergence time

$$t_* \approx \frac{1}{\eta g \tilde{\beta}^2 d_0^2} \left(\frac{2\pi}{\tilde{\beta}} e^{\tilde{\beta} d_0^2} \right)^{D/2} \frac{1}{D} \tag{12}$$

which scales *exponentially* in D (Fig. 1c), but does *not* scale with M (Fig. 1d). We conclude that **spatial reward sparsity, not temporal reward sparsity, drives slow policy learning** in this model.

4 Temporal Reward Sparsity Produces Refinement Phase

It may seem strange that the exponential slowdown we just described does not depend at all on our task's temporal extent—intuitively, one expects tasks with more time steps to be harder to learn. In this section, we show that this is indeed the case, but that the difficulties produced by temporal reward sparsity are significantly less severe than those described in the previous section.

Endpoint is learned first. Because the sparse reward term is determined only by the average endpoint $\boldsymbol{b} := \sum_{t=0}^{M-1} \boldsymbol{b}_t \frac{\Delta t}{T}$, in the initial (sparse regime) phase of learning the individual $\boldsymbol{b}_t$ are relatively unconstrained. When $\boldsymbol{b}$ is sufficiently close to the target endpoint $\boldsymbol{b}^f$, so that we obtain *dense regime* dynamics, the $\boldsymbol{b}_t$ evolve according to

$$\frac{d}{d\tau} \boldsymbol{b}_t \approx \eta \left\{ -\frac{\alpha}{T} \boldsymbol{b}_t + g \tilde{\boldsymbol{B}} (\boldsymbol{b}^f - \boldsymbol{b}) \sqrt{\frac{\det \tilde{\boldsymbol{B}}}{(2\pi)^D}} \right\} \, , \tag{13}$$

which is almost identical to Eq. 6. It is only in this second phase, after the endpoint $\boldsymbol{b}$ has been roughly learned (Fig. 2a-b), that individual $\boldsymbol{b}_t$ (i.e., the policy at specific times; Fig. 2c) are refined.

Matrix with a spiked spectrum controls rate of trajectory refinement. Consider an eigendecomposition $\tilde{\boldsymbol{B}} = \boldsymbol{O}\boldsymbol{\Lambda}_B\boldsymbol{O}^T$ of $\tilde{\boldsymbol{B}}$, where $\boldsymbol{O}$ is orthogonal (i.e., $\boldsymbol{O}\boldsymbol{O}^T = \boldsymbol{I}$) and $\boldsymbol{\Lambda}_B$ is diagonal. Let $\tilde{\beta}_i := \Lambda_{B,ii}$ denote its i-th diagonal entry. Eq. 13 can be simplified by defining rotated variables $\tilde{\boldsymbol{b}}_t := \boldsymbol{O}^T\boldsymbol{b}_t$, since the i-th entry of this vector evolves according to (see Appendix C)

$$\frac{d}{d\tau}\tilde{b}_t = \eta\left\{-\frac{\alpha}{T}\tilde{b}_t + g_i\left(\tilde{b}^f - \sum_t \tilde{b}_t\frac{\Delta t}{T}\right)\right\} \qquad g_i := g\sqrt{\det\tilde{\boldsymbol{B}}/(2\pi)^D}\tilde{\beta}_i \tag{14}$$

where we have suppressed the i index of $\tilde{b}_t$ and $\tilde{b}^f$ to ease notation. This reduction is useful because different spatial components of $\tilde{\boldsymbol{b}}_t$ evolve independently; the above ODE describes the evolution of M parameters, each associated with a different time point. Define a matrix $\boldsymbol{A} := \frac{\alpha}{T}\boldsymbol{I} + g_i\frac{\Delta t}{T}\boldsymbol{1}\boldsymbol{1}^T$, a vector $\boldsymbol{y} := (\tilde{b}_0, \tilde{b}_1, ..., \tilde{b}_{M-1})^T \in \mathbb{R}^M$, and $\boldsymbol{y}^* := g_i\tilde{b}^f\boldsymbol{A}^{-1}\boldsymbol{1}$. We can rewrite Eq. 14 as the ODE

$$\frac{d}{d\tau}\boldsymbol{y} = \eta\boldsymbol{A}(\boldsymbol{y}^* - \boldsymbol{y})\,, \tag{15}$$

which we can solve to find that $\boldsymbol{y}(\tau) = \boldsymbol{y}^* + e^{-\eta\boldsymbol{A}\tau}[\boldsymbol{y}(0) - \boldsymbol{y}^*]$. It is interesting that $\boldsymbol{A}$ controls $\tilde{\boldsymbol{b}}_t$ convergence, because it has a 'spiked' spectrum: it has an 'outlier' eigenvector $\boldsymbol{1}$ whose eigenvalue

$$\lambda_* := \frac{\alpha}{T} + g_i \tag{16}$$

is larger than the other eigenvalues, which equal $\lambda_{\text{bulk}} := \alpha/T$ (Fig. 2d). But $\boldsymbol{1}$ being an eigenvector is equivalent to saying that the endpoint $\sum_t \tilde{b}_t\frac{\Delta t}{T}$, which corresponds to this eigenvector up to scaling, is learned earliest. Hence, the gap between endpoint learning and trajectory refinement is controlled by this eigenvalue gap, and in particular the relative sizes of g_i and α/T.

Trajectory error is reduced after bias error. Another way of quantifying the difference between when the agent learns to reach the target, and when they learn to reach the target *efficiently* (i.e., in a way which minimizes the effort term), is to track the total trajectory error, which decomposes as

$$E_{\text{tot}} := \sum_{i=1}^{D}\sum_{t=0}^{M-1}\frac{\Delta t}{T}(b_{ti} - b_i^f)^2 = \sum_{i=1}^{D}(b_i - b_i^f)^2 + \sum_{i=1}^{D}\sum_{t=0}^{M-1}\frac{\Delta t}{T}(b_{ti} - b_i)^2 = E_{\text{end}} + E_{\text{var}}\,, \tag{17}$$

i.e., as a sum of the *endpoint error* and the *variance of actions*. It is the second term which causes the total error to shrink more slowly than the endpoint error (Fig. 2e); it converges on a time scale determined by λ_{bulk}. We conclude that **temporal reward sparsity only somewhat slows learning, since parameters still converge exponentially quickly to their optimal values** (after hitting).

5 Optimal curricula avoid slow learning by increasing hitting

Is it possible to avoid the exponential slowdown due to (spatial) reward sparsity? The answer is *yes*—by dynamically modulating task difficulty to the agent's current ability, it can be avoided. Here, we describe an approach which does this *explicitly*, by changing how exploratory the agent is over time, which is a form of *curriculum learning* (Bengio et al., 2009; Wang et al., 2022). Another approach, which we describe in Sec. 6, accomplishes this *implicitly* through learning dynamics.

According to Eq. 5, the matrix $\tilde{\boldsymbol{B}}^{-1} := [(1/\kappa)\boldsymbol{I} + \sum_{t=0}^{M-1}\boldsymbol{B}_t^{-1}\Delta t]$ determines the *effective target size*, which in turn determines task difficulty, since learning is slow when the hitting probability is small. This means we can effectively modulate the hitting probability and the associated slowdown by choosing the exploration parameters $\boldsymbol{B}_t$ to *ensure that the hitting probability remains large*. Define $\boldsymbol{B}^{-1} := \sum_{t=0}^{M-1}\boldsymbol{B}_t^{-1}\Delta t$, since this is the (controllable) parameter combination that determines $\tilde{\boldsymbol{B}}$. We are interested in adaptively choosing $\boldsymbol{B}^{-1}$ throughout learning in order to speed it up as much as possible. A key fact is that endpoint learning dynamics (Eq. 5) perform gradient ascent on

$$\frac{d}{d\tau}\boldsymbol{b} = \eta\frac{\partial G_{\text{eff}}(\boldsymbol{b},\boldsymbol{B})}{\partial\boldsymbol{b}} \quad \text{where} \quad G_{\text{eff}}(\boldsymbol{b},\boldsymbol{B}) := -\frac{\alpha}{2T}\|\boldsymbol{b}\|_2^2 + g\frac{e^{-\frac{1}{2}(\Delta\boldsymbol{b})^T[(1/\kappa)\boldsymbol{I}+\boldsymbol{B}^{-1}]^{-1}(\Delta\boldsymbol{b})}}{\sqrt{(2\pi)^D\det[(1/\kappa)\boldsymbol{I}+\boldsymbol{B}^{-1}]}}\,, \tag{18}$$

which we call the *effective return*. We can thus define an 'optimal' $\boldsymbol{B}$ as one maximizing G_{eff}.[1]

[1]Note that this only works if we assume α/T is small. Otherwise, this optimization is not sensitive to the effort term, which can lead to the agent getting stuck in a spurious optimum.

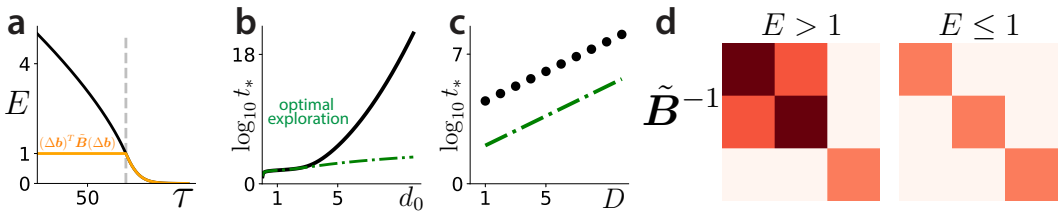

Figure 3: **Optimal curricula avoid slow learning by increasing hitting probability.** Bias units are chosen so that $\kappa = 1$. **a.** When the 'optimal' exploration matrix $\boldsymbol{B}$ (see Eq. 19) is used, the endpoint error $E := \kappa\|\Delta\boldsymbol{b}\|_2^2$ decreases steadily, and the quantity $(\Delta\boldsymbol{b})^T\tilde{\boldsymbol{B}}(\Delta\boldsymbol{b})$ (orange line) is held fixed at 1 until increasing exploration cannot usefully improve the hitting probability (gray line). **b.** Using the optimal $\boldsymbol{B}$ improves the way the convergence time t_* scales with the initial bias error d_0: it is cubic in d_0 (green; see Eq. 23) rather than exponential (black; see Sec. 3). **c.** This result still scales exponentially with the spatial dimensionality D, but note the orders-of-magnitude improvement for any given D. **d.** While the variance-matching condition (Eq. 20) is satisfied, the optimal $\tilde{\boldsymbol{B}}^{-1}$ is a scaled identity matrix plus a low-rank perturbation; when it cannot be satisfied, because $E \leq 1$, the optimal $\tilde{\boldsymbol{B}}^{-1}$ is a scaled identity matrix. Hotter colors indicate larger nonzero entries.

Optimal exploration maximizes hitting probability. Maximizing G_{eff} with respect to $\boldsymbol{B}$, subject to a positive semidefiniteness constraint, yields an 'optimal' matrix (see Appendix D; Fig. 3d)

$$\boldsymbol{B}_*^{-1}(\Delta\boldsymbol{b}) = \begin{cases} \left(1 - \frac{1}{\kappa\|\Delta\boldsymbol{b}\|_2^2}\right)(\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T & \text{if } \kappa\|\Delta\boldsymbol{b}\|_2^2 > 1 \\ \boldsymbol{0} & \text{if } \kappa\|\Delta\boldsymbol{b}\|_2^2 \leq 1 \end{cases} . \tag{19}$$

This result is interesting, since it implies that (if $\kappa\|\Delta\boldsymbol{b}\|_2^2 > 1$) the *variance-matching condition*

$$(\Delta\boldsymbol{b})^T\tilde{\boldsymbol{B}}(\Delta\boldsymbol{b}) = 1 \tag{20}$$

holds. This condition *ensures the hitting probability remains large* (Fig. 3a), as intuition suggests. Moreover, it has a reasonable interpretation: *explore when it improves the hitting probability*.

Optimal exploration yields only polynomially slow learning. The significance of Eq. 20 is that it produces (endpoint) learning dynamics (see Appendix D)

$$\frac{d}{d\tau}\boldsymbol{b} \approx -\frac{\eta g e^{-1/2}}{\kappa^{1/2}}\left(\frac{\kappa}{2\pi}\right)^{D/2}\frac{\Delta\boldsymbol{b}}{\|\Delta\boldsymbol{b}\|_2^3} \tag{21}$$

where we have assumed the effort term is negligibly small. The dynamics of the endpoint error $E := \kappa\|\Delta\boldsymbol{b}\|_2^2$ are then

$$\frac{d}{d\tau}E = -2\kappa^{1/2}\eta g e^{-1/2}\left(\frac{\kappa}{2\pi}\right)^{D/2}\frac{1}{\sqrt{E}} , \tag{22}$$

which implies a convergence time

$$t_* = \frac{1}{3}\frac{1}{\kappa^{1/2}\eta g}\left(\frac{2\pi}{\kappa}\right)^{D/2}\left[\kappa^{3/2}\|\Delta\boldsymbol{b}^0\|_2^3 - 1\right] + \frac{1}{2\eta g\kappa}\left(\frac{2\pi}{\kappa}\right)^{D/2}\left[\text{Ei}(1/2) - \text{Ei}\left(\frac{1}{2}\epsilon^2\right)\right] . \tag{23}$$

Compare this result to our earlier (no curriculum) result, Eq. 12. Notably, the dependence on the initial error $\tilde{\beta}\|\Delta\boldsymbol{b}^0\|_2^2$ is *polynomial* rather than *exponential* (Fig. 3b-c). This implies, **given an appropriate exploration curriculum, sparse rewards need not produce exponential slowdowns.**

6 Feature Variability Yields Beneficial Implicit Curriculum

We are now ready to study the impact of reward sparsity on the more complex case, which involves $F \geq 1$ features and learning an (average) policy $\bar{\boldsymbol{u}}_t(\boldsymbol{z}) = \frac{1}{T}\boldsymbol{b}_t + \frac{1}{T}\boldsymbol{W}_t\boldsymbol{z}$. Because the task now involves episode-to-episode variability that comes from the state vector $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$, and $D \cdot F$ additional parameters to learn, one expects learning to be harder; however, we will show that learning is generally *easier* than if there are no features. In particular, weight learning dynamics can produce an *implicit curriculum learning effect* which avoids the exponential slowdown discussed in Sec. 3 and looks more like the explicit curriculum dynamics discussed in Sec. 5 (see Fig. 4).

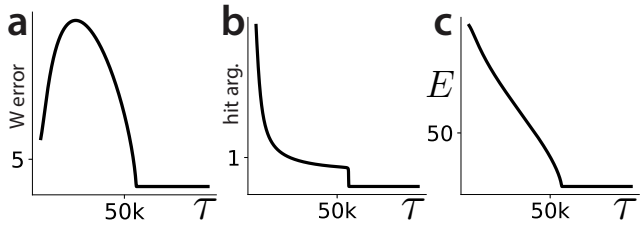

Figure 4: **Weight dynamics produces an implicit curriculum effect. a.** Example weight error trajectory. Note that it transiently increases, and then decreases. **b.** The transient increase in weights ensures that the hitting probability argument $(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1} (\Delta \boldsymbol{b})$ approaches 1. **c.** Bias error decreases substantially around the time that the hitting probability argument is near 1.

Learning dynamics. In this more complex case, learning dynamics read (see Appendix A)

$$\frac{d}{d\tau}\boldsymbol{b}_t = \eta \left\{ -\frac{\alpha}{T}\boldsymbol{b}_t + g\boldsymbol{V}^{-1}(\boldsymbol{b}^f - \boldsymbol{b}) \frac{e^{-\frac{1}{2}(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1}(\Delta \boldsymbol{b})}}{\sqrt{(2\pi)^D \det \boldsymbol{V}}} \right\}$$

$$\frac{d}{d\tau}\boldsymbol{W}_t = \eta \left\{ -\frac{\alpha}{T}\boldsymbol{W}_t \boldsymbol{\Sigma} + g \left[\boldsymbol{I} - \boldsymbol{V}^{-1}(\Delta \boldsymbol{b})(\Delta \boldsymbol{b})^T \right] \boldsymbol{V}^{-1}(\boldsymbol{W}^f - \boldsymbol{W}) \boldsymbol{\Sigma} \, \frac{e^{-\frac{1}{2}(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1}(\Delta \boldsymbol{b})}}{\sqrt{(2\pi)^D \det \boldsymbol{V}}} \right\}$$

where $\boldsymbol{W} := \sum_{t=0}^{M-1} \boldsymbol{W}_t \frac{\Delta t}{T}$ is the weight matrix associated with the average learned endpoint, and

$$\boldsymbol{V} := \tilde{\boldsymbol{B}}^{-1} + (\boldsymbol{W} - \boldsymbol{W}^f)\boldsymbol{\Sigma}(\boldsymbol{W} - \boldsymbol{W}^f)^T \in \mathbb{R}^{D \times D} \tag{24}$$

determines the *effective target size*. Note that this expression is similar to what we had before (Eq. 4), but involves a new weight-dependent term which *grows* as the mismatch between $\boldsymbol{W}$ and $\boldsymbol{W}^f$ increases. In the remainder of this section, we show that this weight-dependent modulation of the effective target size produces an *implicit curriculum learning effect* where agents transiently move weights *away* from the optimal weights in order to increase their rate of hitting the target.

To ease analysis, we will make several simplifying assumptions: (i) the feature covariance matrix is isotropic, i.e., $\boldsymbol{\Sigma} = s\boldsymbol{I}$ for some $s > 0$; (ii) $s > 0$ is so large that the bias parameters are effectively frozen on the time scale of weight dynamics; and (iii) the effort coefficient α/T is small enough that it does not appreciably influence endpoint dynamics. Given these assumptions, the endpoint weight discrepancy $\Delta \boldsymbol{W} := \boldsymbol{W} - \boldsymbol{W}^f$ approximately evolves according to

$$\frac{d}{d\tau}\Delta \boldsymbol{W} \approx -\eta s g p_{\text{hit}} \left[\boldsymbol{I} - \boldsymbol{V}^{-1}(\Delta \boldsymbol{b})(\Delta \boldsymbol{b})^T \right] \boldsymbol{V}^{-1}(\Delta \boldsymbol{W}) \, . \tag{25}$$

Weight dynamics have two fixed points, one of which implements implicit curriculum. To understand weight dynamics, it is convenient to directly track the evolution of $\boldsymbol{V}$. Differentiating,

$$\frac{d}{d\tau}\boldsymbol{V} = -2\eta s g p_{\text{hit}} \left\{ \boldsymbol{I} - \frac{1}{\tilde{\beta}}\boldsymbol{V}^{-1} + \frac{1}{\tilde{\beta}}\boldsymbol{V}^{-1}(\Delta \boldsymbol{b})(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1} - \frac{1}{2}\boldsymbol{V}^{-1}(\Delta \boldsymbol{b})(\Delta \boldsymbol{b})^T - \frac{1}{2}(\Delta \boldsymbol{b})(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1} \right\} . \tag{26}$$

The matrix $\boldsymbol{V}$ stops changing when the right-hand side of this ODE equals zero, i.e., when

$$\boldsymbol{I} - \frac{1}{\tilde{\beta}}\boldsymbol{V}^{-1} + \frac{1}{\tilde{\beta}}\boldsymbol{V}^{-1}(\Delta \boldsymbol{b})(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1} = \frac{1}{2}\boldsymbol{V}^{-1}(\Delta \boldsymbol{b})(\Delta \boldsymbol{b})^T + \frac{1}{2}(\Delta \boldsymbol{b})(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1} \, . \tag{27}$$

This Riccati equation has two solutions. One corresponds to the obvious fixed point where $\boldsymbol{W} = \boldsymbol{W}^f$ (or in terms of $\boldsymbol{V}$, where $\boldsymbol{V}^{-1} = (1/\tilde{\beta})\boldsymbol{I}$), whose presence is apparent even from Eq. 25. But this fixed point becomes *unstable* if $\tilde{\beta}\|\Delta \boldsymbol{b}\|_2^2 > 1$, and the alternative fixed point with

$$\boldsymbol{V}^{-1} = \tilde{\beta}\boldsymbol{I} + \left[\frac{1}{\|\Delta \boldsymbol{b}\|_2^2} - \tilde{\beta} \right] \frac{(\Delta \boldsymbol{b})(\Delta \boldsymbol{b})^T}{\|\Delta \boldsymbol{b}\|_2^2} \tag{28}$$

becomes stable (see Appendix E). Importantly, at this fixed point we have $(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1}(\Delta \boldsymbol{b}) = 1$, which is *precisely the same variance-matching condition we observed in Sec. 5*. Hence, at least at steady state, weight dynamics effectively increase the hitting probability in a manner akin to the explicit curriculum we studied in Sec. 5.

Error-aligned weight component gets magnified and other components get suppressed. What happens away from these fixed points? To simplify our analysis, we will assume that $(\Delta \boldsymbol{b})$ is an eigenvector of $\boldsymbol{V}$; this is true at both fixed points, and if this is not true, one can show that $\boldsymbol{V}$ generally evolves so that this becomes true (see Appendix E). Let λ denote the eigenvalue associated with $\Delta \boldsymbol{b}$, and $\lambda_\perp$ an eigenvalue associated with a different eigenvector. Eq. 26 implies

$$\frac{d}{d\tau}\lambda = -2\eta s g p_{\text{hit}} \left(1 - \frac{1}{\tilde{\beta}}\frac{1}{\lambda}\right)\left(1 - \frac{\|\Delta \boldsymbol{b}\|_2^2}{\lambda}\right) \qquad \frac{d}{d\tau}\lambda_\perp = -2\eta s g p_{\text{hit}} \left(1 - \frac{1}{\tilde{\beta}\lambda_\perp}\right) . \quad (29)$$

The error-associated eigenvalue has two fixed points, one of which corresponds to the nontrivial one, whereas the other eigenvalues have a single fixed point at $\lambda_\perp = 1/\tilde{\beta}$. Hence, as long as $\tilde{\beta}\|\Delta \boldsymbol{b}\|_2^2 > 1$, one eigenvalue gets magnified while others get suppressed, with the net effect being that $(\Delta \boldsymbol{W})(\Delta \boldsymbol{W})^T$ becomes rank-one and error-aligned. Overall, we recover a picture of learning which is similar to, but slightly more complicated than the one studied in Sec. 5: there is an initial phase where weight dynamics move so that the variance-matching condition holds (Eq. 28), and then a phase where bias parameters effectively follow Eq. 21. This continues until the hitting probability becomes small, which causes the cycle to repeat. Sparse rewards only produce a polynomial rather than exponential slowdown, with Eq. 23 holding in the infinite time scale separation limit.

7 Discussion

In this work, we studied a model of learning to pursue a moving target which involves rewards that are sparse in two senses: they are *spatially* sparse, because the agent must be close to the target by the end of an episode to receive appreciable reward; and they are *temporally sparse*, because the agent is rewarded only at the end of episodes, and largely based on the result of a temporal sequence of actions rather than any individual action. We found that spatial sparsity strongly modulated learning dynamics, with small increases in the initial agent-target distance potentially yielding exponential increases in learning time. Meanwhile, we found that the task's temporal extent did not create any additional exponential slowdowns, but rather produced a trajectory refinement phase after learning to reach the target. Here, at least, temporal reward sparsity is much more benign than spatial sparsity.

Inductive biases due to learning dynamics. It is well-known that learning dynamics can produce useful inductive biases. For example, noise associated with stochastic gradient descent has been observed to identify 'flatter' loss landscape optima which may generalize better (Keskar et al., 2017; Smith & Le, 2018; Smith et al., 2020), and neural networks learn 'simpler' functions before more complex ones (Bietti & Mairal, 2019; Canatar et al., 2021; Mingard et al., 2025). The implicit curriculum learning effect we identified is interesting in part because it arises from gradient ascent and a simple (linear) architecture: the critical component appears to be the sparse, non-quadratic return. It is also interesting that what is ostensibly extra task complexity ($F \geq 1$ features and $D \cdot F$ parameters to learn) can make learning substantially easier rather than harder.

Limitations. We do not claim that our model explains the impact of reward sparsity in *all* environments. Rather, we only claim that our model offers an analytically tractable setting in which questions about reward sparsity are easier to pose and study. We expect that in realistic RL settings, other factors come into play. For example, in closed-loop settings, learning dynamics are more complex because previous actions affect future states and actions, which complexifies gradients.

One significant assumption we made is that policies associated with different time steps do not share parameters; this assumption makes learning dynamics approximately linear when rewards are 'dense' (i.e., when κ is small) and keeps them simple in a variety of other ways. We made this assumption because the non-convex nature of the return (Eq. 1) already greatly complexifies learning dynamics, but acknowledge that learning dynamics ought to change when this assumption does not hold. Early analyses of closed versus open-loop learning dynamics have already suggested interesting differences (Ger & Barak, 2025). Another strong assumption we made is that policies are parameterized by a simple linear function approximator. As usual, one expects more complex dynamics with multiple regimes—e.g., rich versus lazy (Dominé et al., 2025)—given deeper networks, especially when nonlinearities are included. However, it is worth emphasizing that the learning dynamics associated with these models are challenging to understand even in supervised settings which lack the additional complexities of RL (Bordelon et al., 2025; Proca et al., 2025).

AI USE STATEMENT

In this work, we used generative AI tools for brainstorming, literature search, to assist with checking the mathematical correctness of our results, and to assist with code generation. The authors have checked all text and results and take full responsibility for them.

REFERENCES

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight Experience Replay. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/453fadbd8a1a3af50a9df4df899537b5-Paper.pdf.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48. ACM, 2009.

Alberto Bietti and Julien Mairal. On the Inductive Bias of Neural Tangent Kernels. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/c4ef9c39b300931b69a36fb3dbb8d60e-Paper.pdf.

Blake Bordelon, Jordan Cotler, Cengiz Pehlevan, and Jacob A. Zavatone-Veth. Dynamically Learning to Integrate in Recurrent Neural Networks. *arXiv e-prints*, art. arXiv:2503.18754, March 2025. doi: 10.48550/arXiv.2503.18754.

Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature Communications*, 12(1):2914, May 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-23103-1. URL https://doi.org/10.1038/s41467-021-23103-1.

Clémentine Carla Juliette Dominé, Nicolas Anguita, Alexandra Maria Proca, Lukas Braun, Daniel Kunin, Pedro A. M. Mediano, and Andrew M Saxe. From lazy to rich: Exact learning dynamics in deep linear networks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ZXaocmXc6d.

Yoav Ger and Omri Barak. Learning dynamics of RNNs in closed-loop environments. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=P63bBMXCIH.

Joshua Hare. Dealing with Sparse Rewards in Reinforcement Learning. *arXiv e-prints*, art. arXiv:1910.09281, October 2019. doi: 10.48550/arXiv.1910.09281.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=H1oyRlYgg.

Pawel Ladosz, Lilian Weng, Minwoo Kim, and Hyondong Oh. Exploration in deep reinforcement learning: A survey. *Information Fusion*, 85:1–22, 2022.

Jin Hwa Lee, Stefano Sarao Mannelli, and Andrew M Saxe. Why do animals need shaping? a theory of task composition and curriculum learning. In *International Conference on Machine Learning*, pp. 26837–26855. PMLR, 2024.

Chris Mingard, Henry Rees, Guillermo Valle-Pérez, and Ard A. Louis. Deep neural networks have an inbuilt Occam's razor. *Nature Communications*, 16(1):220, January 2025. ISSN 2041-1723. doi: 10.1038/s41467-024-54813-x. URL https://doi.org/10.1038/s41467-024-54813-x.

Nishil Patel, Sebastian Lee, Stefano Sarao Mannelli, Sebastian Goldt, and Andrew Saxe. Rl perceptron: Generalization dynamics of policy learning in high dimensions. *Physical Review X*, 15(2): 021051, 2025.

Alexandra Maria Proca, Clémentine Carla Juliette Dominé, Murray Shanahan, and Pedro A. M. Mediano. Learning dynamics in linear recurrent neural networks. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=KGOcrIWYnx.

Luca Saglietti, Stefano Mannelli, and Andrew Saxe. An Analytical Theory of Curriculum Learning in Teacher-Student Networks. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 21113–21127. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/84bad835faaf48f24d990072bb5b80ee-Paper-Conference.pdf.

Samuel Smith, Erich Elsen, and Soham De. On the generalization benefit of noise in stochastic gradient descent. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9058–9067. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/smith20a.html.

Samuel L. Smith and Quoc V. Le. A bayesian perspective on generalization and stochastic gradient descent. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJij4yg0Z.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2nd edition, 2018a. ISBN 978-0262039246.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*, chapter 13. Policy gradient methods. MIT Press, Cambridge, MA, 2nd edition, 2018b. ISBN 978-0262039246.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In S. Solla, T. Leen, and K. Müller (eds.), *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf.

Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4555–4576, 2022. doi: 10.1109/TPAMI.2021.3069908.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, May 1992. ISSN 1573-0565. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

A Derivation of average learning dynamics

In this appendix, we derive the ODEs which govern the policy learning dynamics of our pursuit setting. This amounts to doing a large number of Gaussian integrals. See Sec. 2 for more details on how various aspects of our setting are defined.

A.1 Review of assumptions regarding agent policy

Recall from Sec. 2 that agents must learn a policy

$$\pi(\boldsymbol{u}_{0:T}|\boldsymbol{z}) = \prod_{t=0}^{M-1} \pi(\boldsymbol{u}_t|\boldsymbol{z}) \qquad \pi(\boldsymbol{u}_t|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{u}_t; \bar{\boldsymbol{u}}_t(\boldsymbol{z}), \frac{1}{\Delta t}\boldsymbol{B}_t^{-1}) \tag{30}$$

for selecting a temporal sequence of $M \geq 1$ D-dimensional actions $\boldsymbol{u}_0, \boldsymbol{u}_1, ..., \boldsymbol{u}_{M-1}$ given a state vector $\boldsymbol{z} \in \mathbb{R}^F$. These correspond to the agent's velocity vectors at each time step. At the end of the episode, the agent receives two rewards: one which penalizes the size of these velocities, and one which depends on the distance between their *position* (or *endpoint*)

$$\boldsymbol{x}_T := \sum_{t=0}^{M-1} \boldsymbol{u}_t \Delta t \tag{31}$$

and the location $\boldsymbol{f}_T(\boldsymbol{z})$ of the target. Note that the final position of the agent is defined, as usual, as their integrated velocity vector.

The agent's policy at each time step is Gaussian with a state-dependent mean $\bar{\boldsymbol{u}}_t(\boldsymbol{z})$ and covariance matrix $\boldsymbol{B}_t^{-1}/(\Delta t)$, but only the mean action at each time step is learnable. We assume that the agent's average action depends linearly on $\boldsymbol{z}$, i.e.,

$$\bar{\boldsymbol{u}}_t(\boldsymbol{z}) = \frac{1}{T}\boldsymbol{b}_t + \frac{1}{T}\boldsymbol{W}_t\boldsymbol{z}\ , \tag{32}$$

and that the *bias vector* $\boldsymbol{b}_t \in \mathbb{R}^D$ and *weight matrix* $\boldsymbol{W}_t \in \mathbb{R}^{D\times F}$ associated with each time step's policy are learnable. The factors of $(1/T)$ are included to ensure that the bias vector and weight matrix are both unitless, even though the average action must have units of velocity (i.e., inverse time).

We are effectively in a teacher-student setting, since the target's location is assumed to depend on $\boldsymbol{z}$ as

$$\boldsymbol{f}_T(\boldsymbol{z}) = \boldsymbol{b}^f + \boldsymbol{W}^f\boldsymbol{z} \tag{33}$$

where $\boldsymbol{b}^f \in \mathbb{R}^D$ and $\boldsymbol{W}^f \in \mathbb{R}^{D\times F}$. The average endpoint of the agent, meanwhile, equals

$$\bar{\boldsymbol{x}}_T(\boldsymbol{z}) := \sum_{t=0}^{M-1} \bar{\boldsymbol{u}}_t(\boldsymbol{z})\Delta t = \left(\sum_{t=0}^{M-1} \boldsymbol{b}_t \frac{\Delta t}{T}\right) + \left(\sum_{t=0}^{M-1} \boldsymbol{W}_t \frac{\Delta t}{T}\right)\boldsymbol{z} = \boldsymbol{b} + \boldsymbol{W}\boldsymbol{z} \tag{34}$$

where we have defined the overall (i.e., endpoint-associated) bias and weight matrix via

$$\begin{aligned} \boldsymbol{b} &:= \sum_{t=0}^{M-1} \boldsymbol{b}_t \frac{\Delta t}{T} = \frac{1}{M}\sum_{t=0}^{M-1} \boldsymbol{b}_t \\ \boldsymbol{W} &:= \sum_{t=0}^{M-1} \boldsymbol{W}_t \frac{\Delta t}{T} = \frac{1}{M}\sum_{t=0}^{M-1} \boldsymbol{W}_t\ . \end{aligned} \tag{35}$$

In what follows, we study the learning dynamics of $\boldsymbol{b}_t$ and $\boldsymbol{W}_t$. We may also directly study the learning dynamics of $\boldsymbol{b}$ and $\boldsymbol{W}$, which summarize the endpoint-associated component of the agent's learned policy.

A.2 REINFORCE policy learning

We assume agents learn through episodic REINFORCE, whose general parameter updates for a parameter θ of the policy have the form

$$\Delta\theta = \eta\Delta\tau\ \nabla_\theta \log \pi(\boldsymbol{u}_{0:T}|\boldsymbol{z})\ G(\boldsymbol{u}_{0:T}, \boldsymbol{z})\ , \tag{36}$$

where $\Delta\theta$ is the size of the update after an episode, $G(\boldsymbol{u}_{0:T}, \boldsymbol{z})$ is the return associated with that episode, and $\eta(\Delta\tau) > 0$ is the learning rate.

We make two assumptions which greatly simplify learning dynamics. First, we assume that the total number of episodes (N_E, say) is taken to be extremely large, and the learning rate is chosen to scale inversely with N_E, i.e., $\Delta\tau := 1/N_E$.

This assumption is significant because it allows us to eliminate all randomness associated with parameter updates. Normally, parameter updates would be stochastic due to randomness in the policy and state vector draws (since $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$). But variance in $\Delta\theta$ scales like

$$\text{var}(\Delta\theta) \sim (\Delta\tau)^2 \,, \tag{37}$$

which is too small in the $N_E \to \infty$ limit to contribute to learning. Since there is no variance, we obtain that updates correspond precisely to the *mean update*, i.e.,

$$\Delta\theta = \eta\Delta\tau \, \mathbb{E}_{\boldsymbol{u}_{0:T},\boldsymbol{z}} \left[\nabla_\theta \log \pi(\boldsymbol{u}_{0:T}|\boldsymbol{z}) \, G(\boldsymbol{u}_{0:T}, \boldsymbol{z})\right] \,. \tag{38}$$

In the $\Delta\tau \to 0$ limit, we recover an ODE

$$\frac{d}{d\tau}\theta = \eta \, \mathbb{E}_{\boldsymbol{u}_{0:T},\boldsymbol{z}} \left[\nabla_\theta \log \pi(\boldsymbol{u}_{0:T}|\boldsymbol{z}) \, G(\boldsymbol{u}_{0:T}, \boldsymbol{z})\right] \tag{39}$$

where we use $\tau \geq 0$ to parameterize 'learning time'. In the limit, the number of episodes is so large that it is effectively treated as a continuous variable. It is for this reason that we refer to this as the "continuously-many episodes limit". Since $\Delta\tau$ no longer appears, we will now refer to $\eta > 0$ as the learning rate.

We must now specialize Eq. 39 to our particular choice of policy. To ensure appropriate scaling of the learning dynamics with the number of time steps M, we will take

$$\eta \to \eta \frac{T}{\Delta t} = \eta M \,. \tag{40}$$

Now if θ_t is a parameter of the time step t mean action $\bar{\boldsymbol{u}}_t(\boldsymbol{z})$, then

$$\frac{\partial}{\partial\theta_t} \log \pi(\boldsymbol{u}_{0:T}|\boldsymbol{z}) = \frac{\partial \bar{\boldsymbol{u}}_t(\boldsymbol{z})^T}{\partial\theta_t} \boldsymbol{B}_t \Delta t (\boldsymbol{u}_t - \bar{\boldsymbol{u}}_t(\boldsymbol{z})) \,, \tag{41}$$

which implies that

$$\frac{d}{d\tau}\theta_t = \frac{\eta T}{\Delta t} \, \mathbb{E}_{\boldsymbol{z}} \left\{ \frac{\partial \bar{\boldsymbol{u}}_t(\boldsymbol{z})^T}{\partial\theta_t} \boldsymbol{B}_t \Delta t \, \mathbb{E}_{\boldsymbol{u}_{0:T}|\boldsymbol{z}} \left[\, (\boldsymbol{u}_t - \bar{\boldsymbol{u}}_t(\boldsymbol{z})) \, G(\boldsymbol{u}_{0:T}, \boldsymbol{z})\right] \right\} \,. \tag{42}$$

This is the general ODE which appears in Sec. 2. We can specialize it further to obtain learning dynamics for the biases $\boldsymbol{b}_t$ and weights $\boldsymbol{W}_t$. Since

$$\begin{aligned} \frac{\partial \bar{u}_{tk}(\boldsymbol{z})}{\partial b_{t'j}} &= \frac{1}{T}\delta_{kj}\delta_{tt'} \\ \frac{\partial \bar{u}_{tk}(\boldsymbol{z})}{\partial W_{t',ij}} &= \frac{1}{T}\delta_{ki} z_j \delta_{tt'} \,, \end{aligned} \tag{43}$$

we have

$$\begin{aligned} \frac{d}{d\tau}\boldsymbol{b}_t &= \eta \boldsymbol{B}_t \, \mathbb{E}_{\boldsymbol{z}} \left\{ \, \mathbb{E}_{\boldsymbol{u}_{0:T}|\boldsymbol{z}} \left[\, (\boldsymbol{u}_t - \bar{\boldsymbol{u}}_t(\boldsymbol{z})) \, G(\boldsymbol{u}_{0:T}, \boldsymbol{z})\right] \right\} \\ \frac{d}{d\tau}\boldsymbol{W}_t &= \eta \boldsymbol{B}_t \, \mathbb{E}_{\boldsymbol{z}} \left\{ \, \mathbb{E}_{\boldsymbol{u}_{0:T}|\boldsymbol{z}} \left[\, (\boldsymbol{u}_t - \bar{\boldsymbol{u}}_t(\boldsymbol{z})) \, G(\boldsymbol{u}_{0:T}, \boldsymbol{z})\right] \, \boldsymbol{z}^T \right\} \,. \end{aligned} \tag{44}$$

All that remains to do is to explicitly perform the averages over actions $\boldsymbol{u}_{0:T}$ and states $\boldsymbol{z}$.

A.3 Average over actions

In each parameter update, the same average

$$A = \mathbb{E}_{\boldsymbol{u}_{0:T}|\boldsymbol{z}} \left[\, (\boldsymbol{u}_t - \bar{\boldsymbol{u}}_t(\boldsymbol{z})) \, G(\boldsymbol{u}_{0:T}, \boldsymbol{z})\right] \tag{45}$$

over actions $\boldsymbol{u}_{0:T}$ appears. In this subsection, we compute it explicitly. Recall that the return associated with our setting is (see Sec. 2)

$$G(\boldsymbol{u}_{0:T}, \boldsymbol{z}) = G_{\text{effort}} + G_{\text{target}} = \sum_{t=0}^{M-1} -\frac{\alpha}{2}\|\boldsymbol{u}_t\|_2^2 \, \Delta t + g\left(\frac{\kappa}{2\pi}\right)^{D/2} e^{-\frac{\kappa}{2}\|\boldsymbol{x}_T - \boldsymbol{f}_T(\boldsymbol{z})\|_2^2} \tag{46}$$

where $\alpha > 0$ controls the effort cost, $g > 0$ controls the size of the target-associated reward, and $\kappa > 0$ controls the sparsity of that reward. We can hence break up the overall average into two parts:

$$\begin{aligned} A_{\text{effort}} &:= -\frac{\alpha}{2}\sum_{t'=0}^{M-1} \Delta t \, \mathbb{E}_{\boldsymbol{u}_{0:T}|\boldsymbol{z}}\left[\, (\boldsymbol{u}_t - \bar{\boldsymbol{u}}_t(\boldsymbol{z}))\, \|\boldsymbol{u}_{t'}\|_2^2\right] \\ A_{\text{target}} &:= g\, \mathbb{E}_{\boldsymbol{u}_{0:T}|\boldsymbol{z}}\left[\, (\boldsymbol{u}_t - \bar{\boldsymbol{u}}_t(\boldsymbol{z}))\, \left(\frac{\kappa}{2\pi}\right)^{D/2} e^{-\frac{\kappa}{2}\|\boldsymbol{x}_T - \boldsymbol{f}_T(\boldsymbol{z})\|_2^2}\right] . \end{aligned} \tag{47}$$

Effort term. Evaluating the first average is easy. First, since the $\boldsymbol{u}_t$ are independent of one another, note that the average vanishes unless $t' = t$. Then we have

$$\begin{aligned} A_{\text{effort}} &= -\frac{\alpha}{2}\Delta t\, \mathbb{E}_{\boldsymbol{u}_{0:T}|\boldsymbol{z}}\left[\, (\boldsymbol{u}_t - \bar{\boldsymbol{u}}_t)\, \|\boldsymbol{u}_t\|_2^2\right] \\ &= -\frac{\alpha}{2}\Delta t\, \mathbb{E}_{\boldsymbol{u}_{0:T}|\boldsymbol{z}}\left[\, (\boldsymbol{u}_t - \bar{\boldsymbol{u}}_t)\, \|\boldsymbol{u}_t - \bar{\boldsymbol{u}}_t + \bar{\boldsymbol{u}}_t\|_2^2\right] \\ &= -\frac{\alpha}{2}\Delta t\, \mathbb{E}_{\boldsymbol{u}_{0:T}|\boldsymbol{z}}\left\{\, (\boldsymbol{u}_t - \bar{\boldsymbol{u}}_t)\, \left[\|\boldsymbol{u}_t - \bar{\boldsymbol{u}}_t\|_2^2 + 2(\boldsymbol{u}_t - \bar{\boldsymbol{u}}_t)^T \bar{\boldsymbol{u}}_t + \|\bar{\boldsymbol{u}}_t\|_2^2\right]\, \right\} \\ &= -\alpha\Delta t\, \mathbb{E}_{\boldsymbol{u}_{0:T}|\boldsymbol{z}}\left[(\boldsymbol{u}_t - \bar{\boldsymbol{u}}_t)(\boldsymbol{u}_t - \bar{\boldsymbol{u}}_t)^T\right] \bar{\boldsymbol{u}}_t \\ &= -\alpha\Delta t\, \frac{1}{\Delta t}\boldsymbol{B}_t^{-1}\bar{\boldsymbol{u}}_t \\ &= -\alpha \boldsymbol{B}_t^{-1}\, \bar{\boldsymbol{u}}_t\, . \end{aligned} \tag{48}$$

Target-associated term. The target-associated term is significantly more difficult to treat. To deal with it, we will use several tricks. First, we will introduce the function

$$I(\boldsymbol{J}_{0:T}) := \mathbb{E}_{\boldsymbol{u}_{0:T}}\left[\left(\frac{\kappa}{2\pi}\right)^{D/2} e^{-\frac{\kappa}{2}\|\boldsymbol{x}_T - \boldsymbol{f}_T\|_2^2 + \sum_t \boldsymbol{J}_t^T(\boldsymbol{u}_t - \bar{\boldsymbol{u}}_t)\Delta t}\right] , \tag{49}$$

where the vectors $\boldsymbol{J}_0$, $\boldsymbol{J}_1$, ..., $\boldsymbol{J}_{M-1}$ are 'source terms'. Importantly, the average we care about corresponds to the derivative of I with respect to a $\boldsymbol{J}_t$ evaluated where all source terms are zero:

$$\frac{1}{\Delta t}\left.\frac{\partial I}{\partial \boldsymbol{J}_t}\right|_{\boldsymbol{J}_t = \boldsymbol{0}\ \forall t} = \mathbb{E}_{\boldsymbol{u}_{0:T}}\left[(\boldsymbol{u}_t - \bar{\boldsymbol{u}}_t)\left(\frac{\kappa}{2\pi}\right)^{D/2} e^{-\frac{\kappa}{2}\|\boldsymbol{x}_T - \boldsymbol{f}_T\|_2^2}\right] . \tag{50}$$

Hence, it suffices to compute $I(\boldsymbol{J}_{0:T})$ and then take the appropriate derivative.

Second, we will exploit a useful integral representation of the target-associated reward which comes from the characteristic function of a multivariate Gaussian. We have that

$$\begin{aligned} \left(\frac{\kappa}{2\pi}\right)^{D/2} e^{-\frac{\kappa}{2}\|\boldsymbol{x}_T - \boldsymbol{f}_T\|_2^2} &= \int \frac{d\boldsymbol{p}}{(2\pi)^D} e^{-i\boldsymbol{p}^T(\boldsymbol{x}_T - \boldsymbol{f}_T) - \frac{1}{2\kappa}\boldsymbol{p}^T\boldsymbol{p}} \\ &= \int \frac{d\boldsymbol{p}}{(2\pi)^D} e^{-i\boldsymbol{p}^T\left(\sum_t \boldsymbol{u}_t\Delta t - \boldsymbol{f}_T\right) - \frac{1}{2\kappa}\boldsymbol{p}^T\boldsymbol{p}} . \end{aligned} \tag{51}$$

Using this integral representation, $I(\boldsymbol{J}_{0:T})$ can be written

$$\begin{aligned} I(\boldsymbol{J}_{0:T}) &= \int \frac{d\boldsymbol{p}}{(2\pi)^D} e^{i\boldsymbol{p}^T\boldsymbol{f}_T - \frac{1}{2\kappa}\boldsymbol{p}^T\boldsymbol{p}}\, \mathbb{E}_{\boldsymbol{u}_{0:T}}\left[e^{-i\boldsymbol{p}^T\left(\sum_t \boldsymbol{u}_t\Delta t\right) + \sum_t \boldsymbol{J}_t^T(\boldsymbol{u}_t - \bar{\boldsymbol{u}}_t)\Delta t}\right] \\ &= \int \frac{d\boldsymbol{p}}{(2\pi)^D} e^{i\boldsymbol{p}^T\boldsymbol{f}_T - \frac{1}{2\kappa}\boldsymbol{p}^T\boldsymbol{p} - \sum_t \boldsymbol{J}_t^T\bar{\boldsymbol{u}}_t\Delta t}\, \mathbb{E}_{\boldsymbol{u}_{0:T}}\left[e^{\sum_t(\boldsymbol{J}_t - i\boldsymbol{p})^T\boldsymbol{u}_t\Delta t}\right] \\ &= \int \frac{d\boldsymbol{p}}{(2\pi)^D} e^{i\boldsymbol{p}^T\boldsymbol{f}_T - \frac{1}{2\kappa}\boldsymbol{p}^T\boldsymbol{p} - \sum_t \boldsymbol{J}_t^T\bar{\boldsymbol{u}}_t\Delta t + \sum_t(\boldsymbol{J}_t - i\boldsymbol{p})^T\bar{\boldsymbol{u}}_t\Delta t + \frac{\Delta t}{2}(\boldsymbol{J}_t - i\boldsymbol{p})^T\boldsymbol{B}_t^{-1}(\boldsymbol{J}_t - i\boldsymbol{p})} . \end{aligned} \tag{52}$$

The argument of the exponential can be rewritten as

$$\left(\sum_t \frac{\Delta t}{2} \boldsymbol{J}_t^T \boldsymbol{B}_t^{-1} \boldsymbol{J}_t\right) - i\boldsymbol{p}^T \left(\sum_t (\bar{\boldsymbol{u}}_t + \boldsymbol{B}_t^{-1}\boldsymbol{J}_t)\Delta t - \boldsymbol{f}_T\right) - \frac{1}{2}\boldsymbol{p}^T \left[\frac{1}{\kappa}\boldsymbol{I} + \sum_t \boldsymbol{B}_t^{-1}\Delta t\right] \boldsymbol{p}\,. \quad (53)$$

To ease notation, define the $D \times D$ positive definite matrix

$$\boxed{\tilde{\boldsymbol{B}}^{-1} := \frac{1}{\kappa}\boldsymbol{I} + \sum_t \boldsymbol{B}_t^{-1}\Delta t}\,. \quad (54)$$

Then $I(\boldsymbol{J}_{0:T})$ equals

$$\begin{aligned} I(\boldsymbol{J}_{0:T}) &= e^{\sum_t \frac{\Delta t}{2}\boldsymbol{J}_t^T \boldsymbol{B}_t^{-1}\boldsymbol{J}_t} \int \frac{d\boldsymbol{p}}{(2\pi)^D} e^{-i\boldsymbol{p}^T\left(\sum_t(\bar{\boldsymbol{u}}_t + \boldsymbol{B}_t^{-1}\boldsymbol{J}_t)\Delta t - \boldsymbol{f}_T\right) - \frac{1}{2}\boldsymbol{p}^T\tilde{\boldsymbol{B}}^{-1}\boldsymbol{p}} \\ &= e^{\sum_t \frac{\Delta t}{2}\boldsymbol{J}_t^T \boldsymbol{B}_t^{-1}\boldsymbol{J}_t} \sqrt{\frac{\det\tilde{\boldsymbol{B}}}{(2\pi)^D}} e^{-\frac{1}{2}\left[\sum_t(\bar{\boldsymbol{u}}_t + \boldsymbol{B}_t^{-1}\boldsymbol{J}_t)\Delta t - \boldsymbol{f}_T\right]^T \tilde{\boldsymbol{B}}\left[\sum_t(\bar{\boldsymbol{u}}_t + \boldsymbol{B}_t^{-1}\boldsymbol{J}_t)\Delta t - \boldsymbol{f}_T\right]}\,. \end{aligned} \quad (55)$$

Now that we have evaluated $I(\boldsymbol{J}_{0:T})$, we can take its derivative. It equals

$$\frac{\partial I}{\partial \boldsymbol{J}_t} = \Delta t \left[\boldsymbol{B}_t^{-1}\boldsymbol{J}_t - \boldsymbol{B}_t^{-1}\tilde{\boldsymbol{B}}\left(\sum_t(\bar{\boldsymbol{u}}_t + \boldsymbol{B}_t^{-1}\boldsymbol{J}_t)\Delta t - \boldsymbol{f}_T\right)\right] I(\boldsymbol{J}_{0:T})\,, \quad (56)$$

and when we set all $\boldsymbol{J}_t = \boldsymbol{0}$, we find

$$\begin{aligned} \frac{1}{\Delta t}\left.\frac{\partial I}{\partial \boldsymbol{J}_t}\right|_{\boldsymbol{J}_t = \boldsymbol{0}\ \forall t} &= \boldsymbol{B}_t^{-1}\tilde{\boldsymbol{B}}\left(\boldsymbol{f}_T - \sum_t \bar{\boldsymbol{u}}_t\,\Delta t\right) I(\boldsymbol{J}_{0:T} = \boldsymbol{0}) \\ &= \boldsymbol{B}_t^{-1}\tilde{\boldsymbol{B}}\left(\boldsymbol{f}_T - \sum_t \bar{\boldsymbol{u}}_t\,\Delta t\right) \sqrt{\frac{\det\tilde{\boldsymbol{B}}}{(2\pi)^D}} e^{-\frac{1}{2}(\sum_t \bar{\boldsymbol{u}}_t\Delta t - \boldsymbol{f}_T)^T \tilde{\boldsymbol{B}}(\sum_t \bar{\boldsymbol{u}}_t\Delta t - \boldsymbol{f}_T)}\,. \end{aligned} \quad (57)$$

Recalling that the average learned endpoint is $\bar{\boldsymbol{x}}_T := \sum_t \bar{\boldsymbol{u}}_t \Delta t$, we conclude that

$$\boxed{A_{\text{target}} = g\boldsymbol{B}_t^{-1}\tilde{\boldsymbol{B}}\,(\boldsymbol{f}_T - \bar{\boldsymbol{x}}_T)\sqrt{\frac{\det\tilde{\boldsymbol{B}}}{(2\pi)^D}} e^{-\frac{1}{2}(\bar{\boldsymbol{x}}_T - \boldsymbol{f}_T)^T \tilde{\boldsymbol{B}}(\bar{\boldsymbol{x}}_T - \boldsymbol{f}_T)}}\,. \quad (58)$$

Overall,

$$\boxed{\mathbb{E}_{\boldsymbol{u}_{0:T}|\boldsymbol{z}}\left[(\boldsymbol{u}_t - \bar{\boldsymbol{u}}_t(\boldsymbol{z}))\,G(\boldsymbol{u}_{0:T}, \boldsymbol{z})\right] = -\alpha\boldsymbol{B}_t^{-1}\,\bar{\boldsymbol{u}}_t + g\boldsymbol{B}_t^{-1}\tilde{\boldsymbol{B}}\,(\boldsymbol{f}_T - \bar{\boldsymbol{x}}_T)\sqrt{\frac{\det\tilde{\boldsymbol{B}}}{(2\pi)^D}} e^{-\frac{1}{2}(\bar{\boldsymbol{x}}_T - \boldsymbol{f}_T)^T \tilde{\boldsymbol{B}}(\bar{\boldsymbol{x}}_T - \boldsymbol{f}_T)}}\,. \quad (59)$$

This implies that our REINFORCE ODEs have the general form

$$\boxed{\frac{d}{d\tau}\theta_t = \eta T\,\mathbb{E}_{\boldsymbol{z}}\left\{\frac{\partial \bar{\boldsymbol{u}}_t(\boldsymbol{z})^T}{\partial\theta_t}\left[-\alpha\bar{\boldsymbol{u}}_t + g\tilde{\boldsymbol{B}}\,(\boldsymbol{f}_T - \bar{\boldsymbol{x}}_T)\sqrt{\frac{\det\tilde{\boldsymbol{B}}}{(2\pi)^D}} e^{-\frac{1}{2}(\bar{\boldsymbol{x}}_T - \boldsymbol{f}_T)^T \tilde{\boldsymbol{B}}(\bar{\boldsymbol{x}}_T - \boldsymbol{f}_T)}\right]\right\}}\,. \quad (60)$$

Specializing to $\boldsymbol{b}_t$ and $\boldsymbol{W}_t$, we obtain

$$\begin{aligned} \frac{d}{d\tau}\boldsymbol{b}_t &= \eta\,\mathbb{E}_{\boldsymbol{z}}\left\{-\alpha\bar{\boldsymbol{u}}_t + g\tilde{\boldsymbol{B}}\,(\boldsymbol{f}_T - \bar{\boldsymbol{x}}_T)\sqrt{\frac{\det\tilde{\boldsymbol{B}}}{(2\pi)^D}} e^{-\frac{1}{2}(\bar{\boldsymbol{x}}_T - \boldsymbol{f}_T)^T \tilde{\boldsymbol{B}}(\bar{\boldsymbol{x}}_T - \boldsymbol{f}_T)}\right\} \\ \frac{d}{d\tau}\boldsymbol{W}_t &= \eta\,\mathbb{E}_{\boldsymbol{z}}\left\{\left[-\alpha\bar{\boldsymbol{u}}_t + g\tilde{\boldsymbol{B}}\,(\boldsymbol{f}_T - \bar{\boldsymbol{x}}_T)\sqrt{\frac{\det\tilde{\boldsymbol{B}}}{(2\pi)^D}} e^{-\frac{1}{2}(\bar{\boldsymbol{x}}_T - \boldsymbol{f}_T)^T \tilde{\boldsymbol{B}}(\bar{\boldsymbol{x}}_T - \boldsymbol{f}_T)}\right]\boldsymbol{z}^T\right\}\,. \end{aligned} \quad (61)$$

A.4 THE EFFECTIVE RETURN

A useful feature of our dynamics is that they correspond to gradient ascent up the *effective return*

$$G_{\text{eff}} := \mathbb{E}_{\boldsymbol{z}}\left[-\sum_t \frac{\alpha}{2}\|\bar{\boldsymbol{u}}_t\|_2^2\Delta t + g\sqrt{\frac{\det\tilde{\boldsymbol{B}}}{(2\pi)^D}}\, e^{-\frac{1}{2}(\bar{\boldsymbol{x}}_T-\boldsymbol{f}_T)^T\tilde{\boldsymbol{B}}(\bar{\boldsymbol{x}}_T-\boldsymbol{f}_T)}\right], \tag{62}$$

which only depends on policy parameters rather than on any specific states and actions. In terms of this quantity, the general REINFORCE ODE (Eq. 60) becomes

$$\frac{d}{d\tau}\theta(t) = \frac{\eta T}{\Delta t}\frac{\partial G_{\text{eff}}}{\partial\theta(t)}, \tag{63}$$

and $\boldsymbol{b}_t$ and $\boldsymbol{W}_t$ dynamics (Eq. 61) become

$$\begin{aligned}\frac{d}{d\tau}\boldsymbol{b}_t &= \frac{\eta T}{\Delta t}\frac{\partial G_{\text{eff}}}{\partial \boldsymbol{b}_t}\\ \frac{d}{d\tau}\boldsymbol{W}_t &= \frac{\eta T}{\Delta t}\frac{\partial G_{\text{eff}}}{\partial \boldsymbol{W}_t}.\end{aligned} \tag{64}$$

Simplifying Eq. 61 would ordinarily require taking multiple expectations over $\boldsymbol{z}$, since slightly different expectations appear in $\boldsymbol{b}_t$ and $\boldsymbol{W}_t$ dynamics. But this fact allows us to simultaneously derive both by first taking a *single* expectation over $\boldsymbol{z}$ (to compute G_{eff}), and then taking the appropriate derivatives.

A.5 PURE BIAS LEARNING DYNAMICS

We stop here to note that we already have our final result for pure bias learning, i.e., the case where there are no features and no feature-associated weights. The mean action at time step t is simply $\bar{\boldsymbol{u}}_t = \boldsymbol{b}_t/T$, the average endpoint is $\bar{\boldsymbol{x}}_T = \sum_t \boldsymbol{b}_t\frac{\Delta t}{T} = \boldsymbol{b}$, the target location is fixed at $\boldsymbol{f}_T = \boldsymbol{b}^f$, and learning dynamics read

$$\begin{aligned}\frac{d}{d\tau}\boldsymbol{b}_t &= \eta\left\{-\alpha\bar{\boldsymbol{u}}_t + g\tilde{\boldsymbol{B}}\left(\boldsymbol{f}_T-\bar{\boldsymbol{x}}_T\right)\sqrt{\frac{\det\tilde{\boldsymbol{B}}}{(2\pi)^D}}e^{-\frac{1}{2}(\bar{\boldsymbol{x}}_T-\boldsymbol{f}_T)^T\tilde{\boldsymbol{B}}(\bar{\boldsymbol{x}}_T-\boldsymbol{f}_T)}\right\}\\ &= \eta\left\{-\frac{\alpha}{T}\boldsymbol{b}_t + g\tilde{\boldsymbol{B}}\left(\boldsymbol{b}^f-\boldsymbol{b}\right)\sqrt{\frac{\det\tilde{\boldsymbol{B}}}{(2\pi)^D}}e^{-\frac{1}{2}(\boldsymbol{b}-\boldsymbol{b}^f)^T\tilde{\boldsymbol{B}}(\boldsymbol{b}-\boldsymbol{b}^f)}\right\}.\end{aligned} \tag{65}$$

This follows immediately from Eq. 61, since we can discard the expectation over $\boldsymbol{z}$.

We can suggestively rewrite these dynamics by defining the *hitting probability*

$$\boxed{p_{\text{hit}}(\boldsymbol{b}) := \sqrt{\frac{\det\tilde{\boldsymbol{B}}}{(2\pi)^D}}e^{-\frac{1}{2}(\boldsymbol{b}-\boldsymbol{b}^f)^T\tilde{\boldsymbol{B}}(\boldsymbol{b}-\boldsymbol{b}^f)}}. \tag{66}$$

In terms of it, we have

$$\boxed{\frac{d}{d\tau}\boldsymbol{b}_t = \eta\left\{-\frac{\alpha}{T}\boldsymbol{b}_t + g\tilde{\boldsymbol{B}}\left(\boldsymbol{b}^f-\boldsymbol{b}\right)p_{\text{hit}}(\boldsymbol{b})\right\}}. \tag{67}$$

By averaging the ODEs associated with each $\boldsymbol{b}_t$, we can obtain an ODE which directly tracks the time evolution of the endpoint-associated bias $\boldsymbol{b} = \sum_t \boldsymbol{b}_t\frac{\Delta t}{T}$:

$$\boxed{\frac{d}{d\tau}\boldsymbol{b} = \eta\left\{-\frac{\alpha}{T}\boldsymbol{b} + g\tilde{\boldsymbol{B}}\left(\boldsymbol{b}^f-\boldsymbol{b}\right)p_{\text{hit}}(\boldsymbol{b})\right\}}. \tag{68}$$

Notice that it is the same as the ODE for the individual $\boldsymbol{b}_t$ except for the effort term.

A.6 AVERAGE OVER STATES

We will now determine the effect of state variability on learning by averaging updates over the observed state vector, which is distributed according to $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is a $F \times F$ covariance matrix.

We can do this by computing the effective return

$$G_{\text{eff}} := \mathbb{E}_{\boldsymbol{z}} \left[-\sum_t \frac{\alpha}{2} \|\bar{\boldsymbol{u}}_t\|_2^2 \Delta t + g \sqrt{\frac{\det \tilde{\boldsymbol{B}}}{(2\pi)^D}}\, e^{-\frac{1}{2}(\bar{\boldsymbol{x}}_T - \boldsymbol{f}_T)^T \tilde{\boldsymbol{B}} (\bar{\boldsymbol{x}}_T - \boldsymbol{f}_T)} \right] , \tag{69}$$

and then taking derivatives with respect to $\boldsymbol{b}_t$ and $\boldsymbol{W}_t$. As when we took the average over actions, the average we must do involves two terms: an effort-associated term, and a target-associated term. More precisely, $G_{\text{eff}} = A_{\text{effort}} + A_{\text{target}}$, where

$$\begin{aligned} A_{\text{effort}} &:= -\frac{\alpha \Delta t}{2} \sum_t \mathbb{E}_{\boldsymbol{z}} \left[\|\bar{\boldsymbol{u}}_t\|_2^2 \right] \\ A_{\text{target}} &:= g\, \mathbb{E}_{\boldsymbol{z}} \left[\sqrt{\frac{\det \tilde{\boldsymbol{B}}}{(2\pi)^D}}\, e^{-\frac{1}{2}(\bar{\boldsymbol{x}}_T - \boldsymbol{f}_T)^T \tilde{\boldsymbol{B}} (\bar{\boldsymbol{x}}_T - \boldsymbol{f}_T)} \right] . \end{aligned} \tag{70}$$

Effort term. Once again, evaluating the first average is easy. We have

$$\begin{aligned} A_{\text{effort}} &= -\frac{\alpha \Delta t}{2T^2} \sum_t \mathbb{E}_{\boldsymbol{z}} \left[(\boldsymbol{b}_t + \boldsymbol{W}_t \boldsymbol{z})^T (\boldsymbol{b}_t + \boldsymbol{W}_t \boldsymbol{z}) \right] \\ &= -\frac{\alpha \Delta t}{2T^2} \sum_t \mathbb{E}_{\boldsymbol{z}} \left[\|\boldsymbol{b}_t\|_2^2 + 2\boldsymbol{b}_t^T \boldsymbol{W}_t \boldsymbol{z} + \boldsymbol{z}^T \boldsymbol{W}_t^T \boldsymbol{W}_t \boldsymbol{z} \right] \\ &= -\frac{\alpha \Delta t}{2T^2} \sum_t \left(\, \|\boldsymbol{b}_t\|_2^2 + \text{tr}[\boldsymbol{\Sigma} \boldsymbol{W}_t^T \boldsymbol{W}_t] \, \right) . \end{aligned} \tag{71}$$

Target-associated term. Once again, evaluating the other term is significantly more complicated. Since

$$\bar{\boldsymbol{x}}_T - \boldsymbol{f}_T = \boldsymbol{b} - \boldsymbol{b}^f + (\boldsymbol{W} - \boldsymbol{W}^f)\boldsymbol{z} = \Delta \boldsymbol{b} + \Delta \boldsymbol{W} \boldsymbol{z} \, , \tag{72}$$

where we define the bias and weight discrepancies $\Delta \boldsymbol{b} := \boldsymbol{b} - \boldsymbol{b}^f$ and $\Delta \boldsymbol{W} := \boldsymbol{W} - \boldsymbol{W}^f$, the integral of interest is

$$A_{\text{target}} = g\, \mathbb{E}_{\boldsymbol{z}} \left[\sqrt{\frac{\det \tilde{\boldsymbol{B}}}{(2\pi)^D}}\, e^{-\frac{1}{2}(\Delta \boldsymbol{b} + \Delta \boldsymbol{W} \boldsymbol{z})^T \tilde{\boldsymbol{B}} (\Delta \boldsymbol{b} + \Delta \boldsymbol{W} \boldsymbol{z})} \right] . \tag{73}$$

We will once again exploit a useful integral representation of a multivariate Gaussian to rewrite the reward-associated Gaussian:

$$\sqrt{\frac{\det \tilde{\boldsymbol{B}}}{(2\pi)^D}} e^{-\frac{1}{2}[\Delta \boldsymbol{W} \boldsymbol{z} + \Delta \boldsymbol{b}]^T \tilde{\boldsymbol{B}} [\Delta \boldsymbol{W} \boldsymbol{z} + \Delta \boldsymbol{b}]} = \int \frac{d\boldsymbol{p}}{(2\pi)^D} e^{-i \boldsymbol{p}^T (\Delta \boldsymbol{b} + \Delta \boldsymbol{W} \boldsymbol{z}) - \frac{1}{2} \boldsymbol{p}^T \tilde{\boldsymbol{B}}^{-1} \boldsymbol{p}} \, . \tag{74}$$

Using this integral representation, A_{target} becomes

$$\begin{aligned} A_{\text{target}} &= g \int \frac{d\boldsymbol{p}}{(2\pi)^D} e^{-i \boldsymbol{p}^T \Delta \boldsymbol{b} - \frac{1}{2} \boldsymbol{p}^T \tilde{\boldsymbol{B}}^{-1} \boldsymbol{p}}\, \mathbb{E}_{\boldsymbol{z}} \left[e^{-i \boldsymbol{p}^T \Delta \boldsymbol{W} \boldsymbol{z}} \right] \\ &= g \int \frac{d\boldsymbol{p}}{(2\pi)^D} e^{-i \boldsymbol{p}^T \Delta \boldsymbol{b} - \frac{1}{2} \boldsymbol{p}^T \tilde{\boldsymbol{B}}^{-1} \boldsymbol{p}}\, e^{-\frac{1}{2} \boldsymbol{p}^T (\Delta \boldsymbol{W}) \boldsymbol{\Sigma} (\Delta \boldsymbol{W})^T \boldsymbol{p}} \\ &= g \int \frac{d\boldsymbol{p}}{(2\pi)^D} e^{-i \boldsymbol{p}^T \Delta \boldsymbol{b} - \frac{1}{2} \boldsymbol{p}^T \left[\tilde{\boldsymbol{B}}^{-1} + (\Delta \boldsymbol{W}) \boldsymbol{\Sigma} (\Delta \boldsymbol{W})^T \right] \boldsymbol{p}} \, . \end{aligned} \tag{75}$$

Performing the (Gaussian) integral, we conclude that

$$\boxed{A_{\text{target}} = g \sqrt{\frac{1}{(2\pi)^D \det \boldsymbol{V}}} e^{-\frac{1}{2}(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1} (\Delta \boldsymbol{b})}} \, , \tag{76}$$

where we define the $D \times D$ covariance matrix (or *effective target size*)

$$\boldsymbol{V}(\boldsymbol{W}) := \tilde{\boldsymbol{B}}^{-1} + (\Delta \boldsymbol{W})\boldsymbol{\Sigma}(\Delta \boldsymbol{W})^T \,. \tag{77}$$

This matrix plays a role analogous to the one $\tilde{\boldsymbol{B}}^{-1}$ plays in the case of pure bias learning. Unlike that matrix, this one depends on learned parameters: it is *larger* along directions where $\boldsymbol{W}$ is very different from the target weight matrix $\boldsymbol{W}^f$.

In any case, we conclude that the effective return is

$$\boxed{G_{\text{eff}} = -\sum_t \frac{\alpha \Delta t}{2T^2} \left\{ \, \|\boldsymbol{b}_t\|_2^2 + \text{tr}\left[\boldsymbol{\Sigma} \boldsymbol{W}_t^T \boldsymbol{W}_t\right] \, \right\} + g\sqrt{\frac{1}{(2\pi)^D \det \boldsymbol{V}}} e^{-\frac{1}{2}(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1}(\Delta \boldsymbol{b})}} \,. \tag{78}$$

Derivatives of the effective return. In order to obtain the final form of $\boldsymbol{b}_t$ and $\boldsymbol{W}_t$ dynamics, we must differentiate the effective return with respect to $\boldsymbol{b}_t$ and $\boldsymbol{W}_t$.

The derivative with respect to $\boldsymbol{b}_t$ is simple:

$$\frac{\partial G_{\text{eff}}}{\partial \boldsymbol{b}_t} = -\frac{\alpha \Delta t}{T^2} \boldsymbol{b}_t + \frac{g \Delta t}{T} \boldsymbol{V}^{-1}(\boldsymbol{b}^f - \boldsymbol{b}) \sqrt{\frac{1}{(2\pi)^D \det \boldsymbol{V}}} e^{-\frac{1}{2}(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1}(\Delta \boldsymbol{b})} \,. \tag{79}$$

The derivative with respect to $\boldsymbol{W}_t$ is more complex, since we must differentiate a matrix determinant and inverse with respect to $\boldsymbol{W}_t$. Computing matrix derivatives using standard identities, we find

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{W}} \frac{1}{2} \log \det \boldsymbol{V} &= \boldsymbol{V}^{-1}(\boldsymbol{W} - \boldsymbol{W}^f)\boldsymbol{\Sigma} \\ \frac{\partial}{\partial \boldsymbol{W}} \left[\frac{1}{2} (\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1} (\Delta \boldsymbol{b}) \right] &= -\boldsymbol{V}^{-1}(\Delta \boldsymbol{b})(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1}(\boldsymbol{W} - \boldsymbol{W}^f)\boldsymbol{\Sigma} \,. \end{aligned} \tag{80}$$

This implies, since

$$\log A_{\text{target}} = -\frac{1}{2} \log \det \boldsymbol{V} - \frac{1}{2} (\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1} (\Delta \boldsymbol{b}) + \text{const.} \,, \tag{81}$$

we have

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{W}} \log A_{\text{target}} &= -\boldsymbol{V}^{-1}(\boldsymbol{W} - \boldsymbol{W}^f)\boldsymbol{\Sigma} + \boldsymbol{V}^{-1}(\Delta \boldsymbol{b})(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1}(\boldsymbol{W} - \boldsymbol{W}^f)\boldsymbol{\Sigma} \\ &= \left[\boldsymbol{I} - \boldsymbol{V}^{-1}(\Delta \boldsymbol{b})(\Delta \boldsymbol{b})^T \right] \boldsymbol{V}^{-1}(\boldsymbol{W}^f - \boldsymbol{W})\boldsymbol{\Sigma} \,. \end{aligned} \tag{82}$$

Now we can easily compute that

$$\begin{aligned} \frac{\partial G_{\text{eff}}}{\partial \boldsymbol{W}_t} &= -\sum_{t'} \frac{\alpha \Delta t}{2T^2} \frac{\partial}{\partial \boldsymbol{W}_t} \left(\text{tr}\left[\boldsymbol{\Sigma} \boldsymbol{W}_{t'}^T \boldsymbol{W}_{t'} \right] \right) + \frac{\partial}{\partial \boldsymbol{W}_t} \left[g \sqrt{\frac{1}{(2\pi)^D \det \boldsymbol{V}}} e^{-\frac{1}{2}(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1}(\Delta \boldsymbol{b})} \right] \\ &= -\frac{\alpha \Delta t}{T^2} \boldsymbol{W}_t \boldsymbol{\Sigma} + \frac{g \Delta t}{T} \frac{\partial}{\partial \boldsymbol{W}} \left(\log A_{\text{target}} \right) \sqrt{\frac{1}{(2\pi)^D \det \boldsymbol{V}}} e^{-\frac{1}{2}(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1}(\Delta \boldsymbol{b})} \\ &= -\frac{\alpha \Delta t}{T^2} \boldsymbol{W}_t \boldsymbol{\Sigma} + \frac{g \Delta t}{T} \left[\boldsymbol{I} - \boldsymbol{V}^{-1}(\Delta \boldsymbol{b})(\Delta \boldsymbol{b})^T \right] \boldsymbol{V}^{-1}(\boldsymbol{W}^f - \boldsymbol{W})\boldsymbol{\Sigma} \, \frac{e^{-\frac{1}{2}(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1}(\Delta \boldsymbol{b})}}{\sqrt{(2\pi)^D \det \boldsymbol{V}}} \,. \end{aligned} \tag{83}$$

Final result. Since the dynamics of $\boldsymbol{b}_t$ and $\boldsymbol{W}_t$ correspond to gradient ascent up the effective return, we obtain final ODEs

$$\begin{aligned} \frac{d}{d\tau} \boldsymbol{b}_t &= \eta \left\{ -\frac{\alpha}{T} \boldsymbol{b}_t + g \boldsymbol{V}^{-1}(\boldsymbol{b}^f - \boldsymbol{b}) \frac{e^{-\frac{1}{2}(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1}(\Delta \boldsymbol{b})}}{\sqrt{(2\pi)^D \det \boldsymbol{V}}} \right\} \\ \frac{d}{d\tau} \boldsymbol{W}_t &= \eta \left\{ -\frac{\alpha}{T} \boldsymbol{W}_t \boldsymbol{\Sigma} + g \left[\boldsymbol{I} - \boldsymbol{V}^{-1}(\Delta \boldsymbol{b})(\Delta \boldsymbol{b})^T \right] \boldsymbol{V}^{-1}(\boldsymbol{W}^f - \boldsymbol{W})\boldsymbol{\Sigma} \, \frac{e^{-\frac{1}{2}(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1}(\Delta \boldsymbol{b})}}{\sqrt{(2\pi)^D \det \boldsymbol{V}}} \right\} \,. \end{aligned}$$

Notice that the target-associated term does not depend on t. This implies that the endpoint-associated quantities $\boldsymbol{b} = \sum_t \boldsymbol{b}_t \frac{\Delta t}{T}$ and $\boldsymbol{W} = \sum_t \boldsymbol{W}_t \frac{\Delta t}{T}$ have dynamics

$$\frac{d}{d\tau}\boldsymbol{b} = \eta \left\{ -\frac{\alpha}{T}\boldsymbol{b} + g\boldsymbol{V}^{-1}(\boldsymbol{b}^f - \boldsymbol{b}) \frac{e^{-\frac{1}{2}(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1}(\Delta \boldsymbol{b})}}{\sqrt{(2\pi)^D \det \boldsymbol{V}}} \right\}$$

$$\frac{d}{d\tau}\boldsymbol{W} = \eta \left\{ -\frac{\alpha}{T}\boldsymbol{W}\boldsymbol{\Sigma} + g\left[\boldsymbol{I} - \boldsymbol{V}^{-1}(\boldsymbol{b}^f - \boldsymbol{b})(\boldsymbol{b}^f - \boldsymbol{b})^T\right] \boldsymbol{V}^{-1}(\boldsymbol{W}^f - \boldsymbol{W})\boldsymbol{\Sigma} \, \frac{e^{-\frac{1}{2}(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1}(\Delta \boldsymbol{b})}}{\sqrt{(2\pi)^D \det \boldsymbol{V}}} \right\} .$$

As when we discussed pure bias learning above, it is conceptually helpful to define the *hitting probability*

$$p_{\text{hit}}(\boldsymbol{b}, \boldsymbol{W}) := \frac{e^{-\frac{1}{2}(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1}(\Delta \boldsymbol{b})}}{\sqrt{(2\pi)^D \det \boldsymbol{V}}} , \tag{84}$$

whose size determines whether learning dynamics are fast or slow. In particular, the speed of learning dynamics depends on $(\Delta \boldsymbol{b})^T \boldsymbol{V}^{-1}(\Delta \boldsymbol{b})$: when the learned (endpoint) bias is sufficiently close to its target, learning is fast, and when it is not, learning can be extremely slow.

B CONVERGENCE TIME ANALYSIS FOR PURE BIAS LEARNING

As derived in Appendix A, when there are no features, the endpoint-associated bias $\boldsymbol{b} = \sum_t \boldsymbol{b}_t \frac{\Delta t}{T}$ evolves according to

$$\frac{d}{d\tau}\boldsymbol{b} = \eta \left\{ -\frac{\alpha}{T}\boldsymbol{b} + g\tilde{\boldsymbol{B}}(\boldsymbol{b}^f - \boldsymbol{b}) \sqrt{\frac{\det \tilde{\boldsymbol{B}}}{(2\pi)^D}} e^{-\frac{1}{2}(\boldsymbol{b}-\boldsymbol{b}^f)^T \tilde{\boldsymbol{B}}(\boldsymbol{b}-\boldsymbol{b}^f)} \right\} . \tag{85}$$

Assuming the effort term is negligible, and that $\tilde{\boldsymbol{B}} = \tilde{\beta}\boldsymbol{I}$, we have

$$\boxed{\frac{d}{d\tau}\boldsymbol{b} \approx \eta g \tilde{\beta}(\boldsymbol{b}^f - \boldsymbol{b}) \left(\frac{\tilde{\beta}}{2\pi}\right)^{D/2} e^{-\frac{\tilde{\beta}}{2}\|\boldsymbol{b}-\boldsymbol{b}^f\|_2^2}} . \tag{86}$$

In this appendix, we analytically characterize the convergence time of this ODE.

B.1 EVOLUTION OF BIAS ERROR

The trick is to directly track the time evolution of the *bias error*

$$E := \tilde{\beta}\|\boldsymbol{b} - \boldsymbol{b}^f\|_2^2 . \tag{87}$$

Its time derivative is

$$\frac{d}{d\tau}E = 2\tilde{\beta}(\boldsymbol{b} - \boldsymbol{b}^f)^T \frac{d\boldsymbol{b}}{d\tau} = -2\eta g \tilde{\beta} \left(\frac{\tilde{\beta}}{2\pi}\right)^{D/2} \left[\tilde{\beta}\|\boldsymbol{b} - \boldsymbol{b}^f\|_2^2\right] e^{-\frac{\tilde{\beta}}{2}\|\boldsymbol{b}-\boldsymbol{b}^f\|_2^2} , \tag{88}$$

so it evolves according to the ODE

$$\boxed{\frac{d}{d\tau}E = -2\eta g \tilde{\beta} \left(\frac{\tilde{\beta}}{2\pi}\right)^{D/2} E e^{-E/2}} . \tag{89}$$

We can solve this ODE by separation of variables. If E evolves from an initial error E_0 at $\tau = 0$ to a final error E_f at $\tau = t_*$, then

$$\begin{aligned} \int_{E_0}^{E_f} \frac{e^{E/2}}{E}\, dE &= -2\eta g \tilde{\beta} \left(\frac{\tilde{\beta}}{2\pi}\right)^{D/2} \int_0^{t_*} d\tau \\ \implies \int_{E_0/2}^{E_f/2} \frac{e^u}{u}\, du &= -2\eta g \tilde{\beta} \left(\frac{\tilde{\beta}}{2\pi}\right)^{D/2} t_* \\ \implies \mathrm{Ei}(E_f/2) - \mathrm{Ei}(E_0/2) &= -2\eta g \tilde{\beta} \left(\frac{\tilde{\beta}}{2\pi}\right)^{D/2} t_* , \end{aligned} \tag{90}$$

where $\mathrm{Ei}(x) := \int_{-\infty}^{x} \frac{e^y}{y} dy$ is the *exponential integral function*, a quickly growing special function. Rearranging, we find that

$$\boxed{t_* = \frac{1}{2\eta g \tilde{\beta}} \left(\frac{2\pi}{\tilde{\beta}}\right)^{D/2} \left[\mathrm{Ei}\left(\frac{\tilde{\beta}}{2}\|\boldsymbol{b}^0 - \boldsymbol{b}^f\|_2^2\right) - \mathrm{Ei}\left(\frac{1}{2}\epsilon^2\right) \right]} , \tag{91}$$

where we used $E_f = \epsilon^2$ to denote the final (small) bias error, and $\boldsymbol{b}^0$ to denote the initial endpoint bias.

B.2 Initial bias error in terms of D and M

Recall that a major objective of our work is to characterize how the task's spatial dimensionality D and number of time steps M affect the speed and dynamics of learning. Since convergence time depends only on the initial error $\tilde{\beta}\|\boldsymbol{b}^0 - \boldsymbol{b}^f\|_2^2$ here, answering this question (at least in this setting) amounts to determining how this error scales with D and M.

The relationship between $\boldsymbol{b}^0$ and D and M depends on how we choose to initialize our agent. We consider two typical desiderata here:

- Variance in the initial endpoint $\boldsymbol{x}_T := \sum_{t=0}^{M-1} \boldsymbol{u}_t \Delta t$ must be $\mathcal{O}(1)$ with respect to M.
- Variance in the initial endpoint should scale linearly with D.

The first condition ensures that the initial (average) distance between the agent and the target does not blow up at the number of time steps increases. The second condition is natural because, if initializations along each direction are independent, linear increases in dimensionality ought to linearly increase the overall (squared) distance.

To make things concrete, consider a typical Gaussian initialization for the bias b_{ti} associated with the average action at time step $t \in \{0, ..., M-1\}$ along direction $i \in \{1, ..., D\}$. If

$$b_{ti}^0 \sim \mathcal{N}\left(0, \xi^2 \frac{T}{\Delta t}\right) \tag{92}$$

for some scalar $\xi > 0$ controlling the overall variance in the initialization, then the covariance matrix associated with the endpoint bias $\boldsymbol{b}^0 = \sum_t \boldsymbol{b}_t^0 \frac{\Delta t}{T}$ is

$$\mathbb{E}_{\boldsymbol{b}_{0:T}^0}[\boldsymbol{b}^0(\boldsymbol{b}^0)^T] = \sum_{t,t'} \frac{(\Delta t)^2}{T^2} \mathbb{E}[\boldsymbol{b}_t^0(\boldsymbol{b}_{t'}^0)^T] = \sum_{t,t'} \frac{(\Delta t)^2}{T^2} \delta_{t,t'} \xi^2 \frac{T}{\Delta t} \boldsymbol{I} = \xi^2 \boldsymbol{I} \,, \tag{93}$$

i.e., it is $\mathcal{O}(1)$ with respect to M. But the average norm of $\boldsymbol{b}^0$ depends linearly on D, since

$$\mathbb{E}_{\boldsymbol{b}_{0:T}^0}[\|\boldsymbol{b}^0\|_2^2] = \text{tr}[\xi^2 \boldsymbol{I}] = \xi^2 D \,. \tag{94}$$

Incidentally, the $(T/\Delta t)$ scaling of the variance of b_{ti}^0 was necessary to include, since without it both the covariance and norm would scale like $1/M$. This would cause the agent to always start with parameters near zero in the large M limit.

Analogously, assume the components of the target location $\boldsymbol{b}^f$ along each direction are chosen independently from a Gaussian. Concretely, assume

$$b_i^f \sim \mathcal{N}(0, \xi_f^2) \tag{95}$$

for some $\xi_f > 0$.

Given these Gaussian initializations, a typical value of the initial error is

$$\tilde{\beta}\, \mathbb{E}[\|\boldsymbol{b}^0 - \boldsymbol{b}^f\|_2^2] = \tilde{\beta}\left\{\mathbb{E}[\|\boldsymbol{b}^0\|_2^2] + \mathbb{E}[\|\boldsymbol{b}^f\|_2^2]\right\} = \tilde{\beta} D \left\{\xi^2 + \xi_f^2\right\} \,. \tag{96}$$

In summary, the initial error ought to scale linearly with D but not at all with M. We can heuristically write typical values of the initial error as $\tilde{\beta} D d_0^2$, where d_0 is a 'typical' (i.e., across spatial dimensions) initial distance between the agent's endpoint bias and the target bias.

B.3 Dense reward regime

The convergence time (Eq. 91) behaves substantively differently depending on the size of the initial error E_0; we show this explicitly in this subsection and the next.

Suppose the initial error $\tilde{\beta}\|\boldsymbol{b}^0 - \boldsymbol{b}^f\|_2^2$ is small, so that agent consistently hits the target; because it receives rewards frequently when this condition holds, we call this the *dense reward regime*.

The exponential integral has a small argument expansion

$$\text{Ei}(x) = \gamma + \log x + x + \frac{x^2}{4} + \cdots \tag{97}$$

where $\gamma = 0.5772...$ is the Euler-Mascheroni constant. This implies that the convergence time is approximately

$$\begin{aligned} t_* &\approx \frac{1}{2\eta g\tilde{\beta}} \left(\frac{2\pi}{\tilde{\beta}}\right)^{D/2} \left[\gamma + \log\left(\frac{\tilde{\beta}}{2}\|\boldsymbol{b}^0 - \boldsymbol{b}^f\|_2^2\right) - \gamma - \log\left(\frac{1}{2}\epsilon^2\right) \right] \\ &= \frac{1}{2\eta g\tilde{\beta}} \left(\frac{2\pi}{\tilde{\beta}}\right)^{D/2} \log\left(\frac{\tilde{\beta}\|\boldsymbol{b}^0 - \boldsymbol{b}^f\|_2^2}{\epsilon^2}\right) . \end{aligned} \tag{98}$$

Assuming $E_0 \sim \tilde{\beta} D d_0^2$, this becomes

$$\boxed{t_* \approx \frac{1}{2\eta g\tilde{\beta}} \left(\frac{2\pi}{\tilde{\beta}}\right)^{D/2} \log\left(\frac{\tilde{\beta} D d_0^2}{\epsilon^2}\right)}, \tag{99}$$

i.e., we find that the convergence time scales *logarithmically* with the spatial dimensionality D and squared distance d_0^2. This means that large increases in either quantity do not produce large increases in convergence time; in this sense, learning is relatively easy and fast.

B.4 Sparse reward regime

Suppose the initial error $\tilde{\beta}\|\boldsymbol{b}^0 - \boldsymbol{b}^f\|_2^2$ is large, so that the agent initially only rarely hits the target. We call this the *sparse reward regime*.

The exponential integral has a large argument expansion

$$\mathrm{Ei}(x) = \frac{e^x}{x}\left(1 + \frac{1}{x} + \frac{2}{x^2} + \cdots\right) . \tag{100}$$

This implies that the convergence time is approximately

$$t_* \approx \frac{1}{\eta g\tilde{\beta}^2} \left(\frac{2\pi}{\tilde{\beta}}\right)^{D/2} \frac{e^{\frac{\tilde{\beta}}{2}\|\boldsymbol{b}^0 - \boldsymbol{b}^f\|_2^2}}{\|\boldsymbol{b}^0 - \boldsymbol{b}^f\|_2^2} , \tag{101}$$

where we have neglected the (much smaller) ϵ-dependent term. Assuming $E_0 \sim \tilde{\beta} D d_0^2$, this becomes

$$\boxed{t_* \approx \frac{1}{\eta g\tilde{\beta}^2} \left(\frac{2\pi}{\tilde{\beta}}\right)^{D/2} \frac{e^{\frac{\tilde{\beta}}{2} D d_0^2}}{D d_0^2}}, \tag{102}$$

which means that the convergence time scales *exponentially* with the spatial dimensionality D and the squared distance d_0^2. This means that small increases in either can produce large increases in learning time; in this sense, learning is extremely slow in general.

C TRAJECTORY REFINEMENT AFTER LEARNING TO REACH TARGET

In this appendix, we characterize the refinement of trajectories after agents learn to reliably hit the target. As derived in Appendix A, when there are no features, the biases $\boldsymbol{b}_t$ evolve according to

$$\frac{d}{d\tau}\boldsymbol{b}_t = \eta\left\{-\frac{\alpha}{T}\boldsymbol{b}_t + g\tilde{\boldsymbol{B}}(\boldsymbol{b}^f - \boldsymbol{b})\sqrt{\frac{\det\tilde{\boldsymbol{B}}}{(2\pi)^D}}e^{-\frac{1}{2}(\boldsymbol{b}-\boldsymbol{b}^f)^T\tilde{\boldsymbol{B}}(\boldsymbol{b}-\boldsymbol{b}^f)}\right\}. \tag{103}$$

When $\boldsymbol{b}$ is *sufficiently close* to the target bias $\boldsymbol{b}^f$, it is true that

$$e^{-\frac{1}{2}(\boldsymbol{b}-\boldsymbol{b}^f)^T\tilde{\boldsymbol{B}}(\boldsymbol{b}-\boldsymbol{b}^f)} \approx 1\,, \tag{104}$$

so we obtain a *linear* ODE

$$\boxed{\frac{d}{d\tau}\boldsymbol{b}_t = \eta\left\{-\frac{\alpha}{T}\boldsymbol{b}_t + g\tilde{\boldsymbol{B}}(\boldsymbol{b}^f - \boldsymbol{b})\sqrt{\frac{\det\tilde{\boldsymbol{B}}}{(2\pi)^D}}\right\}}. \tag{105}$$

We will characterize the trajectory refinement phase by solving this ODE explicitly.

C.1 DECOUPLING SPACE- AND TIME-DEPENDENCE

Eq. 105 generically couples b_{ti} (the bias parameter associated with the time step t mean action along direction i) to $b_{t'j}$ for various $t' \neq t$ and $j \neq i$. We can greatly simplify dynamics by rotating them so that they become less coupled to one another.

Let $\tilde{\boldsymbol{B}} = \boldsymbol{O}\boldsymbol{\Lambda}_B\boldsymbol{O}^T$ be an eigendecomposition of $\tilde{\boldsymbol{B}}$, so that $\boldsymbol{O}$ is orthogonal (i.e., $\boldsymbol{O}\boldsymbol{O}^T = \boldsymbol{O}^T\boldsymbol{O} = \boldsymbol{I}$) and $\boldsymbol{\Lambda}_B$ is diagonal. Let $\tilde{\beta}_i = \Lambda_{B,ii}$ denote the i-th eigenvalue of $\tilde{\boldsymbol{B}}$. If we define rotated biases $\tilde{\boldsymbol{b}}_t := \boldsymbol{O}^T\boldsymbol{b}_t$, and the corresponding rotated endpoint $\tilde{\boldsymbol{b}} := \boldsymbol{O}^T\boldsymbol{b}$ and target $\tilde{\boldsymbol{b}}^f := \boldsymbol{O}^T\boldsymbol{b}^f$, then

$$\frac{d}{d\tau}\tilde{\boldsymbol{b}}_t = \eta\left\{-\frac{\alpha}{T}\tilde{\boldsymbol{b}}_t + g\,\boldsymbol{O}^T\tilde{\boldsymbol{B}}\boldsymbol{O}(\tilde{\boldsymbol{b}}^f - \tilde{\boldsymbol{b}})\sqrt{\frac{\det\tilde{\boldsymbol{B}}}{(2\pi)^D}}\right\} = \eta\left\{-\frac{\alpha}{T}\tilde{\boldsymbol{b}}_t + g\,\boldsymbol{\Lambda}_B(\tilde{\boldsymbol{b}}^f - \tilde{\boldsymbol{b}})\sqrt{\frac{\det\tilde{\boldsymbol{B}}}{(2\pi)^D}}\right\}. \tag{106}$$

Since $\boldsymbol{\Lambda}_B$ is diagonal, the spatial components of this ODE decouple, i.e.,

$$\frac{d}{d\tau}\tilde{b}_{ti} = \eta\left\{-\frac{\alpha}{T}\tilde{b}_{ti} + g\tilde{\beta}_i(\tilde{b}_i^f - \tilde{b}_i)\sqrt{\frac{\det\tilde{\boldsymbol{B}}}{(2\pi)^D}}\right\}. \tag{107}$$

To ease notation, define the effective gain

$$g_i := g\tilde{\beta}_i\sqrt{\frac{\det\tilde{\boldsymbol{B}}}{(2\pi)^D}} \tag{108}$$

and leave the spatial index on all quantities implicit. We then have

$$\boxed{\frac{d}{d\tau}\tilde{b}_t = \eta\left\{-\frac{\alpha}{T}\tilde{b}_t + g_i\left(\tilde{b}^f - \sum_{t=0}^{M-1}\tilde{b}_t\frac{\Delta t}{T}\right)\right\}}. \tag{109}$$

C.2 SOLVING THE ODE

A nice feature of Eq. 109 is that it linearly couples the M quantities $\tilde{b}_0, ..., \tilde{b}_{M-1}$, and hence can be recast as a matrix ODE. Define the vector $\boldsymbol{y} := (\tilde{b}_0, ..., \tilde{b}_{M-1})^T \in \mathbb{R}^M$ containing the biases, and define the matrix

$$\boldsymbol{A} := \frac{\alpha}{T}\boldsymbol{I} + g_i\frac{\Delta t}{T}\boldsymbol{1}\boldsymbol{1}^T\,. \tag{110}$$

Eq. 109 is equivalent to the matrix ODE

$$\frac{d}{d\tau}\boldsymbol{y} = \eta(g_i\tilde{b}^f\boldsymbol{1} - \boldsymbol{A}\boldsymbol{y}) = \eta\boldsymbol{A}(\boldsymbol{y}^* - \boldsymbol{y})\,, \tag{111}$$

where we use $\boldsymbol{y}^* := g_i\tilde{b}^f\boldsymbol{A}^{-1}\boldsymbol{1}$ to denote the fixed point.

Fixed point. Since the inverse of $\boldsymbol{A}$ is

$$\boldsymbol{A}^{-1} = \frac{1}{\alpha/T}\boldsymbol{I} - \frac{\frac{g_i\Delta t}{T}}{(\alpha/T)^2}\frac{\mathbf{1}\mathbf{1}^T}{1+\frac{1}{\alpha/T}\frac{g_i\Delta t}{T}M} = \frac{1}{\alpha/T}\left(\boldsymbol{I} - \frac{\frac{g\Delta t}{T}}{\alpha/T+g_i}\mathbf{1}\mathbf{1}^T\right) , \tag{112}$$

the fixed point $\boldsymbol{y}^*$ occurs at

$$\boldsymbol{y}^* := g_i\tilde{b}^f\boldsymbol{A}^{-1}\mathbf{1} = \frac{g_i\tilde{b}^f}{\alpha/T}\left(1-\frac{g_i}{\alpha/T+g_i}\right)\mathbf{1} = \frac{g_i}{\alpha/T+g_i}\tilde{b}^f\ \mathbf{1} . \tag{113}$$

This gets increasingly close to $\tilde{b}^f$ as the coefficient α/T of the effort term gets smaller.

Eigenvalues and eigenvectors. The characteristic polynomial of $\boldsymbol{A}$ is

$$\begin{aligned}\det(\boldsymbol{A}-\lambda\boldsymbol{I}) &= \det\left[\left(\frac{\alpha}{T}-\lambda\right)\boldsymbol{I} + \frac{g_i\Delta t}{T}\mathbf{1}\mathbf{1}^T\right] \\ &= \left(\frac{\alpha}{T}-\lambda\right)^M\left(1+\frac{g_i\Delta t}{T}\frac{M}{\frac{\alpha}{T}-\lambda}\right) \\ &= \left(\frac{\alpha}{T}-\lambda\right)^{M-1}\left(\frac{\alpha}{T}-\lambda+g_i\right) ,\end{aligned} \tag{114}$$

so $\boldsymbol{A}$ has two distinct eigenvalues: a 'bulk' eigenvalue $\lambda_{\text{bulk}} := \alpha/T$ (with multiplicity $M-1$), and an 'outlier' eigenvalue $\lambda_* := \alpha/T + g_i$ (with multiplicity 1). The fact that there is a gap between these eigenvalues, which is determined by the size of $g_i > 0$, implies that the component of the solution along the eigenvector associated with λ_* is learned before other components.

Since $\boldsymbol{A}$ has a particularly simple structure—it is just a scaled identity matrix plus a rank-one perturbation—we can determine the corresponding eigenvectors by inspection. $\mathbf{1}$ is an eigenvector, since

$$\boldsymbol{A}\mathbf{1} = \left(\frac{\alpha}{T}\boldsymbol{I} + \frac{g_i\Delta t}{T}\mathbf{1}\mathbf{1}^T\right)\mathbf{1} = \frac{\alpha}{T}\mathbf{1} + \frac{g_i\Delta t M}{T}\mathbf{1} = \left(\frac{\alpha}{T}+g_i\right)\mathbf{1} . \tag{115}$$

Moreover, $\mathbf{1}$ is the outlier eigenvector which is learned first, since its corresponding eigenvalue is the outlier $\lambda_* = \alpha/T + g_i$. This is significant because the endpoint is proportional to $\mathbf{1}$:

$$\tilde{b} = \sum_t \tilde{b}_t\frac{\Delta t}{T} = \frac{\Delta t}{T}\boldsymbol{y}^T\mathbf{1} . \tag{116}$$

Hence, the endpoint is learned before other trajectory components, with the difference in learning time determined by the eigenvalue gap.

The other eigenvectors are simple: any vector orthogonal to $\mathbf{1}$ is an eigenvector with eigenvalue $\lambda_{\text{bulk}} = \alpha/T$.

Solution. Our matrix ODE (Eq. 111) has the formal solution

$$\boldsymbol{y}(\tau) = e^{-\eta\boldsymbol{A}\tau}\boldsymbol{y}(0) + (\boldsymbol{I}-e^{-\eta\boldsymbol{A}\tau})\boldsymbol{y}^* = \boldsymbol{y}^* + e^{-\eta\boldsymbol{A}\tau}[\boldsymbol{y}(0)-\boldsymbol{y}^*] \tag{117}$$

in terms of the matrix exponential $e^{-\eta\boldsymbol{A}\tau}$. But the simple structure of $\boldsymbol{A}$ allows us to write it more explicitly. The matrix exponential that appears can be rewritten as

$$e^{-\eta\boldsymbol{A}\tau} = e^{-\frac{\eta\tau\alpha}{T}}e^{-\eta\tau g_i\frac{\Delta t}{T}\mathbf{1}\mathbf{1}^T} = e^{-\frac{\eta\tau\alpha}{T}}\left[\boldsymbol{I}+\frac{\Delta t}{T}(e^{-\eta\tau g_i}-1)\mathbf{1}\mathbf{1}^T\right] . \tag{118}$$

Let $\tilde{b} = \frac{\Delta t}{T}\mathbf{1}^T\boldsymbol{y}$ denote the (rotated) learned endpoint, and $\tilde{b}_\perp$ any component of the solution orthogonal to $\mathbf{1}$. We conclude that these evolve according to

$$\begin{aligned}\tilde{b}(\tau) &= e^{-\eta(\alpha/T+g_i)\tau}\ \tilde{b}(0) + (1-e^{-\eta(\alpha/T+g_i)\tau})\frac{g_i}{\alpha/T+g_i}\tilde{b}^f \\ \tilde{b}_\perp(\tau) &= e^{-\eta(\alpha/T)\tau}\ \tilde{b}_\perp(0) .\end{aligned} \tag{119}$$

In words: the non-endpoint components converge to their target value (i.e., zero) on a slower time scale $\sim (\eta\lambda_{\text{bulk}})^{-1} = T/(\eta\alpha)$.

C.3 TOTAL VS ENDPOINT ERROR

Another way to quantify the trajectory refinement effect is to track how bias errors evolve in time. We briefly do this below.

Optimal trajectory. Restoring spatial indices for clarity, our fixed point result above says that the optimal $\tilde{b}_{ti}$ equals

$$\tilde{b}^*_{ti} = \frac{g_i}{\alpha/T + g_i}\tilde{b}^f_i \ . \tag{120}$$

Define the diagonal matrix $\boldsymbol{D}$ whose diagonal entries are

$$D_{ii} := \frac{g_i}{\alpha/T + g_i} \ . \tag{121}$$

In (spatial) vector form, this result reads

$$\tilde{\boldsymbol{b}}^*_t = \boldsymbol{D}\tilde{\boldsymbol{b}}^f \ , \tag{122}$$

which implies that the non-rotated optimal trajectory has

$$\boldsymbol{b}^*_t = \boldsymbol{O}\boldsymbol{D}\boldsymbol{O}^T\boldsymbol{b}^f \ . \tag{123}$$

Importantly, *it does not depend on time*: the trajectory which optimizes the return uses the same velocity input at all time steps, since this choice reaches the target while minimizing the effort term.

Trajectory error decomposition. Since we know that the bias at time step t ought to equal $\boldsymbol{b}^*_t$, we can define the *total trajectory error*

$$E_{\text{tot}} := \sum_{t=0}^{M-1} (\boldsymbol{b}_t - \boldsymbol{b}^*_t)^T \tilde{\boldsymbol{B}} (\boldsymbol{b}_t - \boldsymbol{b}^*_t) \ . \tag{124}$$

Reusing the eigendecomposition $\tilde{\boldsymbol{B}} = \boldsymbol{O}\boldsymbol{\Lambda}_B\boldsymbol{O}^T$, we can rewrite the error as

$$E_{\text{tot}} = \sum_{i=1}^{D} \tilde{\beta}_i \sum_{t=0}^{M-1} (\tilde{b}_{ti} - D_i\tilde{b}^f_i)^2 \ . \tag{125}$$

Let $\tilde{b}_i := \sum_t \tilde{b}_{ti}\frac{\Delta t}{T}$ denote the (rotated) endpoint bias along (rotated) direction i. We immediately have a bias-variance-type decomposition

$$E_{\text{tot}} = \sum_{i=1}^{D} \tilde{\beta}_i \sum_{t=0}^{M-1} (\tilde{b}_{ti} - \tilde{b}_i + \tilde{b}_i - D_i\tilde{b}^f_i)^2 = E_{\text{end}} + E_{\text{var}} \tag{126}$$

where E_{end} is the *endpoint error* (the 'bias' term) and E_{var} is the *trajectory variance* (the 'variance' term), with

$$\begin{aligned} E_{\text{end}} &:= \sum_{i=1}^{D} \tilde{\beta}_i (\tilde{b}_i - D_{ii}\tilde{b}^f_i)^2 = \sum_{i=1}^{D} \tilde{\beta}_i \left[\frac{\Delta t}{T}(\mathbf{1}^T\boldsymbol{y}_i) - D_{ii}\tilde{b}^f_i\right]^2 \\ E_{\text{var}} &:= \sum_{i=1}^{D} \tilde{\beta}_i \sum_{t=0}^{M-1} \frac{\Delta t}{T}(\tilde{b}_{ti} - \tilde{b}_i)^2 = \sum_{i=1}^{D} \tilde{\beta}_i \|\boldsymbol{y}_i - \frac{\Delta t}{T}(\mathbf{1}^T\boldsymbol{y}_i)\mathbf{1}\|_2^2 \ . \end{aligned} \tag{127}$$

Time scales of error evolution. Above, we found that (at least in the 'dense' reward regime) there are two time scales of trajectory learning: learning the *endpoint*, which occurs at a rate $\eta\lambda_*$; and learning all other components, which occurs at a rate $\eta\lambda_{\text{bulk}} \ll \eta\lambda_*$.

We can make this difference manifestly clear if we write $\boldsymbol{y}_i$ in terms of eigenvectors of $\boldsymbol{A}_i$, i.e.,

$$\boldsymbol{y}_i(\tau) = c^i_0(\tau)\mathbf{1} + \sum_{k=1}^{M-1} c^i_k(\tau)\boldsymbol{v}_k \tag{128}$$

where the $\boldsymbol{v}_k$ are normalized vectors orthogonal to each other and orthogonal to $\mathbf{1}$. (Note that we can use the same eigenvectors for all i, since all $\boldsymbol{A}_i$ have the same eigenvectors.) Our matrix ODE (Eq. 111) implies

$$\begin{aligned} \frac{d}{d\tau}c_0^i &= \eta\lambda_*(D_{ii}\tilde{b}_i^f - c_0^i) \\ \frac{d}{d\tau}c_k^i &= -\eta\lambda_{\text{bulk}}c_k^i \ . \end{aligned} \tag{129}$$

Then since $\frac{\Delta t}{T}(\mathbf{1}^T\boldsymbol{y}_i) = c_0^i$, the variance term equals

$$E_{\text{var}} = \sum_{i=1}^{D}\tilde{\beta}_i\|\sum_{k=1}^{M-1}c_k^i(\tau)\boldsymbol{v}_k\|_2^2 = \sum_{i=1}^{D}\tilde{\beta}_i\sum_{k=1}^{M-1}[c_k^i(\tau)]^2 \tag{130}$$

and has

$$\frac{d}{d\tau}E_{\text{var}} = -\sum_{i=1}^{D}\tilde{\beta}_i 2\eta\lambda_{\text{bulk}}\sum_{k=1}^{M-1}[c_k^i(\tau)]^2 = -2\eta\lambda_{\text{bulk}}E_{\text{var}} \ . \tag{131}$$

Meanwhile, the bias term equals

$$E_{\text{end}} = \sum_{i=1}^{D}\tilde{\beta}_i\left[c_0^i(\tau) - D_{ii}\tilde{b}_i^f\right]^2 \tag{132}$$

and has

$$\frac{d}{d\tau}E_{\text{end}} = -2\eta\lambda_*\sum_{i=1}^{D}\tilde{\beta}_i\left[c_0^i(\tau) - D_{ii}\tilde{b}_i^f\right]^2 = -2\eta\lambda_*E_{\text{end}} \ . \tag{133}$$

In summary, the total trajectory error decomposes as a sum of the *endpoint error* and *trajectory variance*, and these evolve at different rates:

$$\boxed{\frac{d}{d\tau}E_{\text{end}} = -2\eta\lambda_*E_{\text{end}} \qquad \frac{d}{d\tau}E_{\text{var}} = -2\eta\lambda_{\text{bulk}}E_{\text{var}}} \ . \tag{134}$$

Moreover, the endpoint error converges the fastest, and the discrepancy between the two rates grows as the coefficient α/T of the effort term gets smaller.

D OPTIMAL CURRICULA FOR PURE BIAS LEARNING

In this appendix, we derive optimal curricula for pure bias learning. If the effort term is negligibly small, the endpoint $\boldsymbol{b} = \sum_t \boldsymbol{b}_t \frac{\Delta t}{T}$ has dynamics

$$\boxed{\frac{d}{d\tau}\boldsymbol{b} = \eta g \tilde{\boldsymbol{B}}(\boldsymbol{b}^f - \boldsymbol{b}) \sqrt{\frac{\det \tilde{\boldsymbol{B}}}{(2\pi)^D}} e^{-\frac{1}{2}(\boldsymbol{b}-\boldsymbol{b}^f)^T \tilde{\boldsymbol{B}}(\boldsymbol{b}-\boldsymbol{b}^f)}} \tag{135}$$

where the $D \times D$ positive definite matrix

$$\tilde{\boldsymbol{B}}^{-1} := \frac{1}{\kappa}\boldsymbol{I} + \sum_t \boldsymbol{B}_t^{-1} \Delta t \tag{136}$$

controls the *effective size* of the target, or equivalently the size of the hitting probability factor. Intuitively, we would like to choose the matrix $\tilde{\boldsymbol{B}}$ so that the hitting probability consistently remains large.

Define the $D \times D$ positive semidefinite matrix $\boldsymbol{B}$ via

$$\boldsymbol{B}^{-1} := \sum_t \boldsymbol{B}_t^{-1} \Delta t \, . \tag{137}$$

The parameter κ (which determines the 'intrinsic size' of the target) cannot be controlled, but we will assume that the exploration matrices $\boldsymbol{B}_t$ can be. Since they only affect dynamics through $\boldsymbol{B}$, we will assume that $\boldsymbol{B}$ is controllable. How should we choose $\boldsymbol{B}$ in order to reduce the convergence time as much as possible? Equivalently, what is the optimal *curriculum* for $\boldsymbol{B}$?

In order to formalize 'as much as possible' here, we exploit the fact that dynamics correspond to gradient ascent up an effective return (see Appendix A). Ignoring the effort term, the effective return is

$$G_{\text{eff}}(\boldsymbol{b}, \boldsymbol{B}) := g\sqrt{\frac{\det \tilde{\boldsymbol{B}}}{(2\pi)^D}} \, e^{-\frac{1}{2}\left(\boldsymbol{b}-\boldsymbol{b}^f\right)^T \tilde{\boldsymbol{B}}\left(\boldsymbol{b}-\boldsymbol{b}^f\right)} \tag{138}$$

and depends on policy parameters entirely through $\boldsymbol{b}$. Note that Eq. 135 can also be written as

$$\frac{d}{d\tau}\boldsymbol{b} = \eta \frac{\partial G_{\text{eff}}}{\partial \boldsymbol{b}} \, . \tag{139}$$

Since the dynamics of $\boldsymbol{b}$ are 'greedy' with respect to this function, in the sense that they move $\boldsymbol{b}_t$ in the direction that increases it the most, we can define an 'optimal' $\boldsymbol{B}$ as one which also maximizes G_{eff}.

D.1 ONE-DIMENSIONAL CASE

For simplicity, we will begin with the $D = 1$ case, which has a scalar exploration parameter

$$\frac{1}{\tilde{\beta}} := \frac{1}{\kappa} + \sum_{t=0}^{M-1} \frac{1}{\beta_t} \Delta t = \frac{1}{\kappa} + \frac{1}{\beta} \, . \tag{140}$$

In one dimension, the effective return is

$$G_{\text{eff}}(b, \beta) = g\sqrt{\frac{\tilde{\beta}}{2\pi}} e^{-\frac{\tilde{\beta}}{2}(b-b^f)^2} \, , \tag{141}$$

and its derivative with respect to $\tilde{\beta}$ equals

$$\frac{\partial G_{\text{eff}}}{\partial \tilde{\beta}} = \frac{g}{\sqrt{2\pi}} \frac{\partial}{\partial \tilde{\beta}} \left[\tilde{\beta}^{1/2} e^{-\frac{\tilde{\beta}}{2}(b-b^f)^2}\right] = \frac{g}{2\sqrt{2\pi}} \left[\frac{1}{\tilde{\beta}} - (b-b^f)^2\right] \tilde{\beta}^{1/2} e^{-\frac{\tilde{\beta}}{2}(b-b^f)^2} \, . \tag{142}$$

Its derivative with respect to β equals

$$\frac{\partial G_{\text{eff}}}{\partial \beta} = \frac{\partial G_{\text{eff}}}{\partial \tilde{\beta}} \frac{\partial \tilde{\beta}}{\partial \beta} \, , \tag{143}$$

but since $\tilde{\beta}$ is a monotonic function of β, these two derivatives have the same critical points. They both equal zero where

$$\boxed{\frac{1}{\tilde{\beta}} = (b - b^f)^2} \,. \tag{144}$$

We call this a *variance-matching condition*, since when it holds the argument of the hitting probability is order one.

The optimal choice of β does not *always* satisfy this condition, however. Since it is equivalent to

$$\frac{1}{\kappa} + \frac{1}{\beta} = (b - b^f)^2 \,, \tag{145}$$

it is impossible to satisfy if $\kappa(b - b^f)^2 < 1$. That is, it cannot be satisfied when b is close enough to its optimal value that exploration cannot improve the hitting probability. In this case, the solution is instead to have no exploration, i.e., $\beta = \infty$.

Finally, the optimal β curriculum in the one-dimensional case is

$$\boxed{\beta_*(\Delta b) = \begin{cases} \frac{1}{(b-b^f)^2 - \frac{1}{\kappa}} & \text{if } \kappa(b - b^f)^2 > 1 \\ \infty & \text{if } \kappa(b - b^f)^2 \leq 1 \end{cases}} \,. \tag{146}$$

Provided the bias error is sufficiently big, i.e., $\kappa(b - b^f)^2 > 1$, using this curriculum yields

$$\tilde{\beta}(b - b^f)^2 = 1 \,, \tag{147}$$

which implies that the hitting probability becomes

$$\boxed{p_{\text{hit}} = \sqrt{\frac{1}{2\pi(b - b^f)^2}} e^{-1/2}} \,. \tag{148}$$

D.2 D-DIMENSIONAL CASE

We will now study the higher-dimensional case—assume $D > 1$. First, to make the optimization slightly more tractable, we will maximize the logarithm of G_{eff}, i.e.,

$$\log G_{\text{eff}} = -\frac{1}{2} \log \det \left[\frac{1}{\kappa} \boldsymbol{I} + \boldsymbol{B}^{-1} \right] - \frac{1}{2} (\Delta \boldsymbol{b})^T \left(\frac{1}{\kappa} \boldsymbol{I} + \boldsymbol{B}^{-1} \right)^{-1} (\Delta \boldsymbol{b}) + \text{const.} \tag{149}$$

Bias error is an eigenvector of the optimal matrix. Second, a change of coordinates will make a subtle conceptual point clearer. Choose a coordinate system where one direction is aligned with $(\Delta \boldsymbol{b})$, and all other directions are orthogonal to it. If we let $\boldsymbol{e}$ denote the corresponding unit vector and $r := \|\Delta \boldsymbol{b}\|_2$, then $\Delta \boldsymbol{b} = r\boldsymbol{e}$. Now write $\tilde{\boldsymbol{B}}$ as a block matrix

$$\tilde{\boldsymbol{B}} = \begin{pmatrix} a & \boldsymbol{c}^T \\ \boldsymbol{c} & \boldsymbol{G} \end{pmatrix} \tag{150}$$

where a is a scalar, $\boldsymbol{c} \in \mathbb{R}^{D-1}$, and $\boldsymbol{G} \in \mathbb{R}^{(D-1)\times(D-1)}$. The determinant of $\tilde{\boldsymbol{B}}$ equals

$$\det \tilde{\boldsymbol{B}} = \det \boldsymbol{G} \left[a - \boldsymbol{c}^T \boldsymbol{G}^{-1} \boldsymbol{c} \right] \,. \tag{151}$$

This means that, in this coordinate system (and in terms of $\tilde{\boldsymbol{B}}$ rather than $\boldsymbol{B}$ for simplicity), $\log G_{\text{eff}}$ equals

$$\log G_{\text{eff}} = \frac{1}{2} \log \det \boldsymbol{G} + \frac{1}{2} \log \left[a - \boldsymbol{c}^T \boldsymbol{G}^{-1} \boldsymbol{c} \right] - \frac{r^2}{2} a + \text{const.} \tag{152}$$

Since $\boldsymbol{G}$ and its inverse are positive definite,

$$\frac{1}{2} \log \left[a - \boldsymbol{c}^T \boldsymbol{G}^{-1} \boldsymbol{c} \right] < \frac{1}{2} \log a \tag{153}$$

whenever $\boldsymbol{c} \neq \boldsymbol{0}$. This implies that we can maximize the effective return by choosing $\boldsymbol{c} = \boldsymbol{0}$, i.e., by not mixing components associated with the bias error $\Delta \boldsymbol{b}$ and those associated with orthogonal directions. Put differently, it is optimal for $\Delta \boldsymbol{b}$ to be an eigenvector of $\tilde{\boldsymbol{B}}$, and hence of $\boldsymbol{B}$.

$\boldsymbol{B}^{-1}$ should be low-rank. Eq. 152 indicates that we can maximize our objective by choosing the determinant of $\boldsymbol{G}$, the matrix describing $\tilde{\boldsymbol{B}}$ along all directions orthogonal to $\Delta\boldsymbol{b}$, to be as large as possible. But since

$$\tilde{\boldsymbol{B}}^{-1} = \frac{1}{\kappa}\boldsymbol{I} + \boldsymbol{B}^{-1} \,, \tag{154}$$

the eigenvalues of $\tilde{\boldsymbol{B}}^{-1}$ are those of $\boldsymbol{B}^{-1}$. If λ is an eigenvalue of $\boldsymbol{B}^{-1}$, then $\lambda + 1/\kappa$ is the corresponding eigenvalue of $\tilde{\boldsymbol{B}}^{-1}$. Hence, we can explicitly compute that the log-determinant of $\tilde{\boldsymbol{B}}$ equals

$$\log\det\tilde{\boldsymbol{B}} = \sum_i \log\left(\frac{1}{\lambda_i + 1/\kappa}\right) \,. \tag{155}$$

The log-determinant of $\boldsymbol{G}$ has this form, except that a single eigenvalue (associated with $\Delta\boldsymbol{b}$) is left out. In any case, it is clear that we can maximize it by taking the λ_i to be as small as possible, with

$$\sum_i \log\left(\frac{1}{\lambda_i + 1/\kappa}\right) < \sum_i \log\left(\kappa\right) \tag{156}$$

for nonzero λ_i. This implies that it is optimal for $\boldsymbol{B}^{-1}$ to be low-rank, and in particular to equal

$$\boldsymbol{B}^{-1} = q(\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T \tag{157}$$

for some $q \geq 0$.

Finishing the optimization. We now only need to choose the q coefficient that appears in Eq. 157. For this choice of $\boldsymbol{B}^{-1}$, we have

$$\tilde{\boldsymbol{B}}^{-1} = \frac{1}{\kappa}\boldsymbol{I} + q(\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T \tag{158}$$

and

$$\tilde{\boldsymbol{B}} = \left(\frac{1}{\kappa}\boldsymbol{I} + q(\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T\right)^{-1} = \kappa\left[\boldsymbol{I} - \frac{q\kappa}{1 + q\kappa\|\Delta\boldsymbol{b}\|_2^2}(\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T\right] \,. \tag{159}$$

Evaluating the relevant determinant and quadratic form, the objective reads

$$\log G_{\text{eff}}(q) = -\frac{1}{2}\log\left[1 + \kappa q\|\Delta\boldsymbol{b}\|_2^2\right] - \frac{1}{2}\frac{\kappa\|\Delta\boldsymbol{b}\|_2^2}{1 + \kappa q\|\Delta\boldsymbol{b}\|_2^2} + \text{const.} \tag{160}$$

Differentiating with respect to q, we find

$$\begin{aligned}
\frac{\partial\log G_{\text{eff}}}{\partial q} &= -\frac{1}{2}\frac{\kappa\|\Delta\boldsymbol{b}\|_2^2}{1 + \kappa q\|\Delta\boldsymbol{b}\|_2^2} + \frac{1}{2}\kappa^2\|\Delta\boldsymbol{b}\|_2^4\frac{1}{(1 + \kappa q\|\Delta\boldsymbol{b}\|_2^2)^2} \\
&= -\frac{1}{2}\frac{\kappa\|\Delta\boldsymbol{b}\|_2^2}{1 + \kappa q\|\Delta\boldsymbol{b}\|_2^2}\left[1 - \frac{\kappa\|\Delta\boldsymbol{b}\|_2^2}{1 + \kappa q\|\Delta\boldsymbol{b}\|_2^2}\right] \\
&= -\frac{1}{2}\frac{\kappa\|\Delta\boldsymbol{b}\|_2^2}{1 + \kappa q\|\Delta\boldsymbol{b}\|_2^2}\left[\frac{1 + \kappa(q-1)\|\Delta\boldsymbol{b}\|_2^2}{1 + \kappa q\|\Delta\boldsymbol{b}\|_2^2}\right] \,.
\end{aligned} \tag{161}$$

This equals zero where

$$q = 1 - \frac{1}{\kappa\|\Delta\boldsymbol{b}\|_2^2} \,. \tag{162}$$

However, as in the one-dimensional case, it is not always possible to satisfy this condition. $\boldsymbol{B}^{-1}$ must be positive semidefinite, so we require that $q \geq 0$; the condition can only be satisfied if

$$\kappa\|\Delta\boldsymbol{b}\|_2^2 \geq 1 \,, \tag{163}$$

i.e., if the agent's bias is far enough from the target bias that increasing exploration can improve the hitting probability.

If it possible to satisfy this condition, the optimal curriculum uses

$$\boldsymbol{B}^{-1} = \left(1 - \frac{1}{\kappa\|\Delta\boldsymbol{b}\|_2^2}\right)(\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T \,, \tag{164}$$

or equivalently the optimal $\tilde{\boldsymbol{B}}$ has

$$\begin{aligned}\tilde{\boldsymbol{B}}^{-1} &= \frac{1}{\kappa}\boldsymbol{I} + \left(1 - \frac{1}{\kappa\|\Delta\boldsymbol{b}\|_2^2}\right)(\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T \\ \tilde{\boldsymbol{B}} &= \kappa\boldsymbol{I} - \left(\kappa - \frac{1}{\|\Delta\boldsymbol{b}\|_2^2}\right)\frac{(\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T}{\|\Delta\boldsymbol{b}\|_2^2}\,.\end{aligned} \tag{165}$$

Additionally, the *variance-matching condition*

$$\boxed{(\Delta\boldsymbol{b})^T\tilde{\boldsymbol{B}}(\Delta\boldsymbol{b}) = 1} \tag{166}$$

holds.

If it is *not* possible to satisfy Eq. 163, then the above derivative with respect to q indicates that $\log G_{\text{eff}}$ is a decreasing function of q. Hence, it is optimal to choose $q = 0$—i.e., to choose no exploration when the agent's policy parameters are close enough that exploration will not help.

Finally, the optimal exploration matrix $\boldsymbol{B}^{-1}$ equals

$$\boxed{\boldsymbol{B}_*^{-1}(\Delta\boldsymbol{b}) = \begin{cases}\left(1 - \frac{1}{\kappa\|\Delta\boldsymbol{b}\|_2^2}\right)(\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T & \text{if } \kappa\|\Delta\boldsymbol{b}\|_2^2 > 1 \\ \boldsymbol{0} & \text{if } \kappa\|\Delta\boldsymbol{b}\|_2^2 \le 1\end{cases}}\,. \tag{167}$$

Note that this result mirrors the result we obtained in the one-dimensional case (Eq. 146).

When the variance-matching condition holds, the determinant of $\tilde{\boldsymbol{B}}^{-1}$ equals

$$\begin{aligned}\det\tilde{\boldsymbol{B}}^{-1} &= \det\left(\frac{1}{\kappa}\boldsymbol{I} + \left(1 - \frac{1}{\kappa\|\Delta\boldsymbol{b}\|_2^2}\right)(\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T\right) \\ &= \det\left(\frac{1}{\kappa}\boldsymbol{I}\right)\left[1 + \kappa\left(1 - \frac{1}{\kappa\|\Delta\boldsymbol{b}\|_2^2}\right)\|\Delta\boldsymbol{b}\|_2^2\right] \\ &= \frac{1}{\kappa^D}\kappa\|\Delta\boldsymbol{b}\|_2^2 \\ &= \frac{\|\Delta\boldsymbol{b}\|_2^2}{\kappa^{D-1}}\,.\end{aligned} \tag{168}$$

This implies that the hitting probability equals

$$\boxed{p_{\text{hit}}(\boldsymbol{b}) := \sqrt{\frac{\det\tilde{\boldsymbol{B}}}{(2\pi)^D}}e^{-\frac{1}{2}(\boldsymbol{b}-\boldsymbol{b}^f)^T\tilde{\boldsymbol{B}}(\boldsymbol{b}-\boldsymbol{b}^f)} = \frac{1}{\kappa^{1/2}}\left(\frac{\kappa}{2\pi}\right)^{D/2}e^{-1/2}\,\frac{1}{\|\Delta\boldsymbol{b}\|_2}}\,. \tag{169}$$

Our one-dimensional result (Eq. 148) is a special case of this one.

D.3 Convergence time using optimal exploration matrix

Suppose we use the optimal curriculum given by Eq. 167. How much better is the convergence time? Bias learning dynamics (Eq. 135) become

$$\begin{aligned}\frac{d}{d\tau}\boldsymbol{b} &= -\eta g\tilde{\boldsymbol{B}}(\Delta\boldsymbol{b})\sqrt{\frac{\det\tilde{\boldsymbol{B}}}{(2\pi)^D}}e^{-\frac{1}{2}(\Delta\boldsymbol{b})^T\tilde{\boldsymbol{B}}(\Delta\boldsymbol{b})} \\ &= -\frac{\eta g e^{-1/2}}{\kappa^{1/2}}\left(\frac{\kappa}{2\pi}\right)^{D/2}\tilde{\boldsymbol{B}}\frac{\Delta\boldsymbol{b}}{\|\Delta\boldsymbol{b}\|_2} \\ &= -\frac{\eta g e^{-1/2}}{\kappa^{1/2}}\left(\frac{\kappa}{2\pi}\right)^{D/2}\frac{\Delta\boldsymbol{b}}{\|\Delta\boldsymbol{b}\|_2^3}\,.\end{aligned} \tag{170}$$

Now define the bias error as $E := \kappa\|\Delta\boldsymbol{b}\|_2^2$. Its time derivative is

$$\begin{aligned}\frac{d}{d\tau}E &= 2\kappa(\Delta\boldsymbol{b})^T\frac{d}{d\tau}(\Delta\boldsymbol{b}) \\ &= -2\kappa\frac{\eta g e^{-1/2}}{\kappa^{1/2}}\left(\frac{\kappa}{2\pi}\right)^{D/2}\frac{1}{\|\Delta\boldsymbol{b}\|_2} \\ &= -2\kappa^{1/2}\eta g e^{-1/2}\left(\frac{\kappa}{2\pi}\right)^{D/2}\frac{1}{\sqrt{E}}\,.\end{aligned} \tag{171}$$

We can solve this ODE via separation of variables. If error at time $t = 0$ is E_0 and the time at $t = t_*$ is E_f, then

$$\begin{aligned} \int_{E_0}^{E_f} \sqrt{E}\, dE &= -2\kappa^{1/2}\eta g e^{-1/2} \left(\frac{\kappa}{2\pi}\right)^{D/2} \int_0^{t_*} dt \\ \implies \frac{2}{3}\, E^{3/2}\Big|_{E_0}^{E_f} &= -2\kappa^{1/2}\eta g e^{-1/2} \left(\frac{\kappa}{2\pi}\right)^{D/2} t_*\,, \end{aligned} \tag{172}$$

which implies that the convergence time equals

$$\boxed{t_* = \frac{1}{3}\frac{1}{\kappa^{1/2}\eta g}\left(\frac{2\pi}{\kappa}\right)^{D/2}\left[\, E_0^{3/2} - E_f^{3/2}\,\right] = \frac{1}{3}\frac{1}{\kappa^{1/2}\eta g}\left(\frac{2\pi}{\kappa}\right)^{D/2}\left[\,\kappa^{3/2}\|\Delta \boldsymbol{b}^0\|_2^3 - 1\,\right]}\,, \tag{173}$$

where we chose $E_f = 1$ since this is the minimum possible value of the error where the variance-matching condition still holds.

If a smaller error is desired, we must consider a second phase of dynamics where $\tilde{\boldsymbol{B}} = \kappa \boldsymbol{I}$. Reusing the result from Appendix B, this equals

$$t_*^{\text{second}} = \frac{1}{2\eta g \kappa}\left(\frac{2\pi}{\kappa}\right)^{D/2}\left[\,\text{Ei}(1/2) - \text{Ei}\left(\frac{1}{2}\epsilon^2\right)\,\right]\,, \tag{174}$$

where we assume the final error equals ϵ^2.

Including both phases, the overall convergence time equals

$$\boxed{t_* = \frac{1}{3}\frac{1}{\kappa^{1/2}\eta g}\left(\frac{2\pi}{\kappa}\right)^{D/2}\left[\,\kappa^{3/2}\|\Delta \boldsymbol{b}^0\|_2^3 - 1\,\right] + \frac{1}{2\eta g\kappa}\left(\frac{2\pi}{\kappa}\right)^{D/2}\left[\,\text{Ei}(1/2) - \text{Ei}\left(\frac{1}{2}\epsilon^2\right)\,\right]}\,. \tag{175}$$

Importantly, this depends on the *cube* of the norm of the initial bias error $\Delta \boldsymbol{b}^0$, whereas our earlier result (see Appendix B) depended *exponentially* on it.

Suppose we assume, as we did in Appendix B, that the initial error scales like $E_0 = \kappa D d_0^2$. Then we have

$$\boxed{t_* = \frac{1}{3}\frac{1}{\kappa^{1/2}\eta g}\left(\frac{2\pi}{\kappa}\right)^{D/2}\left[\,\kappa^{3/2}D^{3/2}d_0^3 - 1\,\right] + \frac{1}{2\eta g\kappa}\left(\frac{2\pi}{\kappa}\right)^{D/2}\left[\,\text{Ei}(1/2) - \text{Ei}\left(\frac{1}{2}\epsilon^2\right)\,\right]}\,. \tag{176}$$

This also scales much better with D than before, since the initialization-related term goes like $D^{3/2}$. Although it still technically scales exponentially with D, this is essentially due to the exponential-in-D scaling of the prefactor that appears in the return (see Sec. 2), so this prefactor is to some extent unavoidable given our formulation.

E ANALYSIS OF THE IMPLICIT CURRICULUM EFFECT

In this appendix, we study the *implicit curriculum effect* discussed in Sec. 6: model weights transiently move away from their target in order to increase the hitting probability, and hence make learning faster overall. We make the same simplifying assumptions we make there: (i) the feature covariance matrix is isotropic, i.e., $\boldsymbol{\Sigma} = s\boldsymbol{I}$ for some $s > 0$; (ii) $s > 0$ is so large that the bias parameters are effectively frozen on the time scale of weight dynamics; and (iii) the effort coefficient α/T is small enough that it does not appreciably influence endpoint dynamics. Given these assumptions, the endpoint weight discrepancy $\Delta\boldsymbol{W} := \boldsymbol{W} - \boldsymbol{W}^f$ approximately evolves according to

$$\boxed{\frac{d}{d\tau}\Delta\boldsymbol{W} \approx -\eta s g p_{\text{hit}} \left[\boldsymbol{I} - \boldsymbol{V}^{-1}(\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T\right] \boldsymbol{V}^{-1}(\Delta\boldsymbol{W})} \tag{177}$$

where, as we derived in Appendix A, the hitting probability factor equals

$$p_{\text{hit}}(\boldsymbol{b}, \boldsymbol{W}) := \frac{e^{-\frac{1}{2}(\Delta\boldsymbol{b})^T\boldsymbol{V}^{-1}(\Delta\boldsymbol{b})}}{\sqrt{(2\pi)^D \det \boldsymbol{V}}} \tag{178}$$

and the covariance matrix $\boldsymbol{V}$ equals

$$\boldsymbol{V} := \tilde{\boldsymbol{B}}^{-1} + (\Delta\boldsymbol{W})\boldsymbol{\Sigma}(\Delta\boldsymbol{W})^T = \frac{1}{\tilde{\beta}}\boldsymbol{I} + s(\Delta\boldsymbol{W})(\Delta\boldsymbol{W})^T . \tag{179}$$

E.1 DYNAMICS OF THE EFFECTIVE COVARIANCE

A useful way to analyze the implicit curriculum effect is to directly track the time evolution of $\boldsymbol{V}$ (Eq. 179). It is straightforward to compute that

$$\frac{d}{d\tau}\boldsymbol{V} = -\eta s g p_{\text{hit}} \left\{ \left[\boldsymbol{I} - \boldsymbol{V}^{-1}(\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T\right] \boldsymbol{V}^{-1}(\boldsymbol{V} - \tilde{\boldsymbol{B}}^{-1}) + (\boldsymbol{V} - \tilde{\boldsymbol{B}}^{-1})\boldsymbol{V}^{-1}\left[\boldsymbol{I} - (\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T\boldsymbol{V}^{-1}\right] \right\} \tag{180}$$

$$= -\eta s g p_{\text{hit}} \left\{ \left[\boldsymbol{I} - \boldsymbol{V}^{-1}(\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T\right] (\boldsymbol{I} - \boldsymbol{V}^{-1}\tilde{\boldsymbol{B}}^{-1}) + (\boldsymbol{I} - \tilde{\boldsymbol{B}}^{-1}\boldsymbol{V}^{-1}) \left[\boldsymbol{I} - (\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T\boldsymbol{V}^{-1}\right] \right\} .$$

Since we are assuming that $\tilde{\boldsymbol{B}} = \tilde{\beta}\boldsymbol{I}$, we can specialize this to

$$\frac{d}{d\tau}\boldsymbol{V} = -\eta s g p_{\text{hit}} \left\{ \left[\boldsymbol{I} - \boldsymbol{V}^{-1}(\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T\right] (\boldsymbol{I} - \frac{1}{\tilde{\beta}}\boldsymbol{V}^{-1}) + (\boldsymbol{I} - \frac{1}{\tilde{\beta}}\boldsymbol{V}^{-1}) \left[\boldsymbol{I} - (\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T\boldsymbol{V}^{-1}\right] \right\} \tag{181}$$

$$= -2\eta s g p_{\text{hit}} \left\{ \boldsymbol{I} - \frac{1}{\tilde{\beta}}\boldsymbol{V}^{-1} + \frac{1}{\tilde{\beta}}\boldsymbol{V}^{-1}(\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T\boldsymbol{V}^{-1} - \frac{1}{2}\boldsymbol{V}^{-1}(\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T - \frac{1}{2}(\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T\boldsymbol{V}^{-1} \right\} .$$

E.2 FIXED POINT OF COVARIANCE DYNAMICS

The covariance matrix $\boldsymbol{V}$ stops changing when the right-hand side of Eq. 181 equals zero, i.e., when

$$\boldsymbol{I} - \frac{1}{\tilde{\beta}}\boldsymbol{V}^{-1} + \frac{1}{\tilde{\beta}}\boldsymbol{V}^{-1}(\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T\boldsymbol{V}^{-1} = \frac{1}{2}\boldsymbol{V}^{-1}(\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T + \frac{1}{2}(\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T\boldsymbol{V}^{-1} . \tag{182}$$

This is an algebraic Riccati equation. To ease notation, use $\boldsymbol{M}$ to denote $\boldsymbol{V}^{-1}$. Let $\boldsymbol{e}$ denote the unit vector parallel to $\Delta\boldsymbol{b}$ and r the magnitude of $\Delta\boldsymbol{b}$, so that $\Delta\boldsymbol{b} = r\boldsymbol{e}$. Finally, let $\boldsymbol{u}$ denote the unit vector parallel to $\boldsymbol{V}^{-1}\boldsymbol{e} = \boldsymbol{M}\boldsymbol{e}$ and v the magnitude of $\boldsymbol{M}\boldsymbol{e}$.

Given this change in notation, our algebraic Riccati equation becomes

$$\boxed{\boldsymbol{I} - \frac{1}{\tilde{\beta}}\boldsymbol{M} + \frac{r^2v^2}{\tilde{\beta}}\boldsymbol{u}\boldsymbol{u}^T = \frac{r^2v}{2}\boldsymbol{u}\boldsymbol{e}^T + \frac{r^2v}{2}\boldsymbol{e}\boldsymbol{u}^T} . \tag{183}$$

Simplifying Riccati equation. Left- and right-multiplying by $\boldsymbol{e}$ yields

$$1 - \frac{1}{\tilde{\beta}} v (\boldsymbol{u}^T \boldsymbol{e}) + \frac{r^2 v^2}{\tilde{\beta}} (\boldsymbol{u}^T \boldsymbol{e})^2 = r^2 v (\boldsymbol{u}^T \boldsymbol{e}) \,. \tag{184}$$

Right-multiplying by $\boldsymbol{e}$ and left-multiplying by $\boldsymbol{u}$ yields

$$\boldsymbol{u}^T \boldsymbol{e} - \frac{1}{\tilde{\beta}} v + \frac{r^2 v^2}{\tilde{\beta}} (\boldsymbol{u}^T \boldsymbol{e}) = \frac{r^2 v}{2} + \frac{r^2 v}{2} (\boldsymbol{u}^T \boldsymbol{e})^2 \,. \tag{185}$$

Rearranging, we have the two conditions

$$\begin{aligned} 0 &= \frac{r^2 v^2}{\tilde{\beta}} (\boldsymbol{u}^T \boldsymbol{e})^2 - \left[\frac{1}{\tilde{\beta}} v + r^2 v \right] (\boldsymbol{u}^T \boldsymbol{e}) + 1 \\ 0 &= \frac{r^2 v}{2} (\boldsymbol{u}^T \boldsymbol{e})^2 - \left[1 + \frac{r^2 v^2}{\tilde{\beta}} \right] (\boldsymbol{u}^T \boldsymbol{e}) + \left[\frac{r^2 v}{2} + \frac{v}{\tilde{\beta}} \right] \end{aligned} \tag{186}$$

The bias error is an eigenvector at the fixed point. We will first use these conditions to show that $\boldsymbol{u}^T \boldsymbol{e} = \pm 1$, i.e., that $\boldsymbol{u}$ and $\boldsymbol{e}$ are parallel, which implies that $\boldsymbol{e}$ is an eigenvector of $\boldsymbol{M}$. Using the quadratic formula to solve the first equation, we find that

$$\begin{aligned} \boldsymbol{u}^T \boldsymbol{e} &= \frac{\tilde{\beta}}{2 r^2 v^2} \left\{ \left[\frac{1}{\tilde{\beta}} v + r^2 v \right] \pm \sqrt{\left[\frac{1}{\tilde{\beta}} v + r^2 v \right]^2 - \frac{4 r^2 v^2}{\tilde{\beta}}} \right\} \\ &= \frac{\tilde{\beta}}{2 r^2 v^2} \left\{ \left[\frac{1}{\tilde{\beta}} v + r^2 v \right] \pm \left[\frac{1}{\tilde{\beta}} v - r^2 v \right] \right\} \,, \end{aligned} \tag{187}$$

i.e., that the two solutions are

$$\boldsymbol{u}^T \boldsymbol{e} = \frac{1}{r^2 v} \,, \frac{\tilde{\beta}}{v} \,. \tag{188}$$

If $\boldsymbol{u}^T \boldsymbol{e} = 1/(r^2 v)$, then the second condition becomes

$$0 = \frac{1}{2 r^2 v} - \left[1 + \frac{r^2 v^2}{\tilde{\beta}} \right] \frac{1}{r^2 v} + \left[\frac{r^2 v}{2} + \frac{v}{\tilde{\beta}} \right] = - \frac{1}{2 r^2 v} + \frac{r^2 v}{2} \,, \tag{189}$$

which implies $r^2 v = 1/(r^2 v)$, i.e., that $r^2 v = 1$. Equivalently, $\boldsymbol{u}^T \boldsymbol{e} = \pm 1$.

If we instead take the other solution $\boldsymbol{u}^T \boldsymbol{e} = \tilde{\beta}/v$ to be true, the second condition becomes

$$\begin{aligned} 0 &= \frac{r^2 v}{2} \frac{\tilde{\beta}^2}{v^2} - \left[1 + \frac{r^2 v^2}{\tilde{\beta}} \right] \frac{\tilde{\beta}}{v} + \left[\frac{r^2 v}{2} + \frac{v}{\tilde{\beta}} \right] \\ &= \frac{r^2}{2} \frac{\tilde{\beta}^2}{v} - \frac{\tilde{\beta}}{v} + \frac{v}{\tilde{\beta}} - \frac{1}{2} r^2 v \\ &= \frac{\tilde{\beta}}{v} \left(\frac{r^2}{2} \tilde{\beta} - 1 \right) - \frac{v}{\tilde{\beta}} \left(\frac{r^2}{2} \tilde{\beta} - 1 \right) \\ &= \left(\frac{r^2}{2} \tilde{\beta} - 1 \right) \left(\frac{\tilde{\beta}}{v} - \frac{v}{\tilde{\beta}} \right) \,. \end{aligned} \tag{190}$$

Importantly, we would like equality to hold *regardless* of r. Since in most cases it will not be true that $r^2 = 2/\tilde{\beta}$, we ought to take $v^2 = \tilde{\beta}^2$. But then $\boldsymbol{u}^T \boldsymbol{e} = \pm 1$. Hence, either way we obtain that $\boldsymbol{u}$ and $\boldsymbol{e}$ are parallel, so $\boldsymbol{e}$ is an eigenvector of $\boldsymbol{M}$ when the algebraic Riccati equation above holds.

Expression for the fixed point. If $\boldsymbol{e}$ is an eigenvector of $\boldsymbol{M}$, the Riccati equation above (Eq. 184 reduces to

$$\boldsymbol{I} - \frac{1}{\tilde{\beta}} \boldsymbol{M} + \frac{r^2 v^2}{\tilde{\beta}} \boldsymbol{e} \boldsymbol{e}^T = r^2 v \, \boldsymbol{e} \boldsymbol{e}^T \,, \tag{191}$$

which implies

$$\boldsymbol{M} = \tilde{\beta}\boldsymbol{I} + r^2 v \left[v - \tilde{\beta}\right] \boldsymbol{e}\boldsymbol{e}^T \,. \tag{192}$$

Note that v is the eigenvalue associated with $\boldsymbol{e}$, since (given how we have defined things) $\boldsymbol{M}\boldsymbol{e} = v\boldsymbol{u} = v\boldsymbol{e}$. Consistency requires that

$$v = \tilde{\beta} + r^2 v(v - \tilde{\beta}) \,, \tag{193}$$

which is equivalent to the quadratic equation

$$r^2 v^2 - \left[1 + \tilde{\beta} r^2\right] v + \tilde{\beta} = 0 \,. \tag{194}$$

The solutions of this equation are

$$v = \frac{1}{2r^2}\left\{1 + \tilde{\beta}r^2 \pm \sqrt{\left[1 + \tilde{\beta}r^2\right]^2 - 4r^2\tilde{\beta}}\right\} = \frac{1}{2r^2}\left\{1 + \tilde{\beta}r^2 \pm \left[1 - \tilde{\beta}r^2\right]\right\} \,, \tag{195}$$

or

$$v = \frac{1}{r^2} \,, \tilde{\beta} \,. \tag{196}$$

These two choices of v correspond to qualitatively different solutions. If $v = \tilde{\beta}$, then $\boldsymbol{M} = \tilde{\beta}\boldsymbol{I}$ is proportional to the identity matrix. But the other v solution implies

$$\boldsymbol{M} = \tilde{\beta}\boldsymbol{I} + \left[\frac{1}{r^2} - \tilde{\beta}\right] \boldsymbol{e}\boldsymbol{e}^T \,. \tag{197}$$

This solution is not valid for *all* possible values of r. To see this, note that since $\boldsymbol{M} = \boldsymbol{V}^{-1}$, we have

$$\boldsymbol{V} = \frac{1}{\tilde{\beta}}\left[\boldsymbol{I} + \left[\frac{1}{\tilde{\beta}r^2} - 1\right]\boldsymbol{e}\boldsymbol{e}^T\right]^{-1} = \frac{1}{\tilde{\beta}}\left[\boldsymbol{I} - \tilde{\beta}r^2\left[\frac{1}{\tilde{\beta}r^2} - 1\right]\boldsymbol{e}\boldsymbol{e}^T\right] = \frac{1}{\tilde{\beta}}\left[\boldsymbol{I} - \left[1 - \tilde{\beta}r^2\right]\boldsymbol{e}\boldsymbol{e}^T\right] \,. \tag{198}$$

Now since $\boldsymbol{V} = (1/\tilde{\beta})\boldsymbol{I} + (\Delta\boldsymbol{W})(\Delta\boldsymbol{W})^T$, it must be true that

$$(\Delta\boldsymbol{W})(\Delta\boldsymbol{W})^T = -\frac{1}{\tilde{\beta}}\left[1 - \tilde{\beta}r^2\right]\boldsymbol{e}\boldsymbol{e}^T \tag{199}$$

is positive semidefinite. But this can only happen when

$$\tilde{\beta}r^2 \geq 1 \,. \tag{200}$$

If we change notation, this is just

$$\tilde{\beta}\|\Delta\boldsymbol{b}\|_2^2 \geq 1 \,, \tag{201}$$

i.e., the condition we encountered in Sec. 5 and Appendix D (note that if $\boldsymbol{B}$ is controllable, we should replace $\tilde{\beta}$ with κ). In words: the agent's bias must be far enough from the target bias that increasing exploration can increase the hitting probability.

We conclude that there are two fixed points of $\boldsymbol{V}$. One has $\boldsymbol{V} = (1/\tilde{\beta})\boldsymbol{I}$ (i.e., the weight discrepancy is zero), and the other has

$$\boldsymbol{V} = \frac{1}{\tilde{\beta}}\boldsymbol{I} + \left(1 - \frac{1}{\tilde{\beta}\|\Delta\boldsymbol{b}\|_2^2}\right)(\Delta\boldsymbol{b})(\Delta\boldsymbol{b})^T \tag{202}$$

but requires that $\tilde{\beta}\|\Delta\boldsymbol{b}\|_2^2 \geq 1$. It is important to point out that this result is essentially the same as Eq. 167, the optimal choice of $\boldsymbol{B}^{-1}$.

E.3 Error direction gets magnified while other directions decay

Let us assume for now that $\Delta \boldsymbol{b}$ is an eigenvector of $\boldsymbol{V}$ (and $\boldsymbol{V}^{-1}$), since, as we just found, this is true at the fixed point. Since $\boldsymbol{V}$ is positive definite, and in particular symmetric, its other eigenvectors are orthogonal to $\Delta \boldsymbol{b}$.

Let λ denote the eigenvalue of $\boldsymbol{V}$ corresponding to $\Delta \boldsymbol{b}$. Let $\boldsymbol{v}$ denote an eigenvector orthogonal to $\Delta \boldsymbol{b}$ and $\lambda_\perp$ its corresponding eigenvalue. Then our equation for $\boldsymbol{V}$ dynamics (Eq. 181) implies

$$\begin{aligned} \frac{d}{d\tau}\lambda &= -\eta s g p_{\text{hit}} \left\{ 2\left[1 - \frac{1}{\tilde{\beta}}\frac{1}{\lambda} + \frac{1}{\tilde{\beta}}\frac{\|\Delta \boldsymbol{b}\|_2^2}{\lambda^2}\right] - 2\frac{\|\Delta \boldsymbol{b}\|_2^2}{\lambda} \right\} \\ &= -2\eta s g p_{\text{hit}} \left(1 - \frac{1}{\tilde{\beta}}\frac{1}{\lambda}\right)\left(1 - \frac{\|\Delta \boldsymbol{b}\|_2^2}{\lambda}\right) . \end{aligned} \tag{203}$$

This system has two fixed points:

$$\lambda = \frac{1}{\tilde{\beta}} \, , \; \|\Delta \boldsymbol{b}\|_2^2 \, . \tag{204}$$

The one with $\lambda = 1/\tilde{\beta}$ corresponds to the trivial fixed point with $\boldsymbol{V} = (1/\tilde{\beta})\boldsymbol{I}$. The other corresponds to the nontrivial fixed point (Eq. 202) at which the variance-matching condition holds.

Importantly, it is clear from the above ODE that the trivial fixed point is *unstable* if it is true that $\tilde{\beta}\|\Delta \boldsymbol{b}\|_2^2 > 1$; hence, the nontrivial fixed point is the only stable one in that case.

Meanwhile, eigenvectors orthogonal to $\Delta \boldsymbol{b}$ have dynamics

$$\frac{d}{d\tau}\lambda_\perp = -2\eta s g p_{\text{hit}} \left(1 - \frac{1}{\tilde{\beta}\lambda_\perp}\right) \tag{205}$$

with only a single fixed point at $\lambda_\perp = 1/\tilde{\beta}$. In other words, the eigenvalues of $\boldsymbol{V}$ associated with orthogonal directions only decay toward the fixed point where the corresponding $\Delta \boldsymbol{W}$ component equals zero.

E.4 Evolution of eigenbasis

Suppose $\Delta \boldsymbol{b}$ is *not* an eigenvector of $\boldsymbol{V}$. Can we still say something about its approach to either fixed point? It turns out that, if we explicitly compute the dynamics of the eigenbasis of $\boldsymbol{V}$, we can show that one eigenvector generally orients in the direction of $\Delta \boldsymbol{b}$.

Assume $\boldsymbol{V} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$. Then

$$\frac{d}{d\tau}\boldsymbol{V} = \dot{\boldsymbol{U}}\boldsymbol{\Lambda}\boldsymbol{U}^T + \boldsymbol{U}\dot{\boldsymbol{\Lambda}}\boldsymbol{U}^T + \boldsymbol{U}\boldsymbol{\Lambda}\dot{\boldsymbol{U}}^T \, . \tag{206}$$

Relevant facts are that

$$\boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{I} \implies \dot{\boldsymbol{U}}\boldsymbol{U}^T = -\boldsymbol{U}\dot{\boldsymbol{U}}^T \implies \sum_k \dot{U}_{ik}U_{jk} = -\sum_k U_{ik}\dot{U}_{jk} \tag{207}$$

$$\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I} \implies \dot{\boldsymbol{U}}^T\boldsymbol{U} = -\boldsymbol{U}^T\dot{\boldsymbol{U}} \implies \sum_k \dot{U}_{ki}U_{kj} = -\sum_k U_{ki}\dot{U}_{kj} \, . \tag{208}$$

Using the above results, and the additional fact that $\boldsymbol{V}^{-1} = \boldsymbol{U}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}^T$, we can rewrite $\boldsymbol{V}$ dynamics (Eq. 181) as

$$\frac{d}{d\tau}\boldsymbol{V} = -\eta s g p_{\text{hit}}\boldsymbol{U}\left\{2\left[\boldsymbol{I} - \frac{1}{\tilde{\beta}}\boldsymbol{\Lambda}^{-1} + \frac{1}{\tilde{\beta}}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}^T(\Delta \boldsymbol{b})(\Delta \boldsymbol{b})^T\boldsymbol{U}\boldsymbol{\Lambda}^{-1}\right] - \boldsymbol{\Lambda}^{-1}\boldsymbol{U}^T(\Delta \boldsymbol{b})(\Delta \boldsymbol{b})^T\boldsymbol{U} - (\Delta \boldsymbol{b})(\Delta \boldsymbol{b})^T\boldsymbol{U}\boldsymbol{\Lambda}^{-1}\right\}\boldsymbol{U}^T.$$

Define $\boldsymbol{y} := \boldsymbol{U}^T\Delta \boldsymbol{b}$. Then

$$\frac{d}{d\tau}\boldsymbol{V} = -\eta s g p_{\text{hit}}\boldsymbol{U}\left\{2\left[\boldsymbol{I} - \frac{1}{\tilde{\beta}}\boldsymbol{\Lambda}^{-1} + \frac{1}{\tilde{\beta}}\boldsymbol{\Lambda}^{-1}\boldsymbol{y}\boldsymbol{y}^T\boldsymbol{\Lambda}^{-1}\right] - \boldsymbol{\Lambda}^{-1}\boldsymbol{y}\boldsymbol{y}^T - \boldsymbol{y}\boldsymbol{y}^T\boldsymbol{\Lambda}^{-1}\right\}\boldsymbol{U}^T \, .$$

Putting everything together, we have

$$\boldsymbol{U}^T\dot{\boldsymbol{U}}\boldsymbol{\Lambda} + \dot{\boldsymbol{\Lambda}} + \boldsymbol{\Lambda}\dot{\boldsymbol{U}}^T\boldsymbol{U} = -\eta s g p_{\text{hit}}\left\{2\left[\boldsymbol{I} - \frac{1}{\tilde{\beta}}\boldsymbol{\Lambda}^{-1} + \frac{1}{\tilde{\beta}}\boldsymbol{\Lambda}^{-1}\boldsymbol{y}\boldsymbol{y}^T\boldsymbol{\Lambda}^{-1}\right] - \boldsymbol{\Lambda}^{-1}\boldsymbol{y}\boldsymbol{y}^T - \boldsymbol{y}\boldsymbol{y}^T\boldsymbol{\Lambda}^{-1}\right\} \, .$$

Eigenvalue dynamics. The (i, j) component of the left-hand side equals

$$\sum_k U_{ki}\dot{U}_{kj}\lambda_j + \delta_{ij}\dot{\lambda}_i + \sum_k \dot{U}_{ki}U_{kj}\lambda_i \,, \tag{209}$$

so its diagonal equals $\delta_{ij}\dot{\lambda}_i$because the other parts cancel. This means

$$\dot{\lambda}_i = -2\eta sgp_{\text{hit}}\left\{1 - \frac{1}{\tilde{\beta}}\frac{1}{\lambda_i} + \frac{1}{\tilde{\beta}}\frac{y_i^2}{\lambda_i^2} - \frac{y_i^2}{\lambda_i}\right\} = -2\eta sgp_{\text{hit}}\left(1 - \frac{y_i^2}{\lambda_i}\right)\left(1 - \frac{1}{\tilde{\beta}\lambda_i}\right) \,. \tag{210}$$

Note that this is almost what we had before for the eigenvector aligned with $\Delta\boldsymbol{b}$, except this time the component y_i of $\Delta\boldsymbol{b}$ along this (not necessarily aligned) eigenvector appears instead.

Eigenvector dynamics. Meanwhile, the off-diagonal entries are

$$\begin{aligned}\sum_k U_{ki}\dot{U}_{kj}\lambda_j + \sum_k \dot{U}_{ki}U_{kj}\lambda_i &= -2\eta sgp_{\text{hit}}\left\{\frac{1}{\tilde{\beta}}\frac{y_i y_j}{\lambda_i\lambda_j} - \frac{1}{2}\left(\frac{1}{\lambda_i} + \frac{1}{\lambda_j}\right)y_i y_j\right\} \\ &= -\eta sgp_{\text{hit}}\left\{\frac{2}{\tilde{\beta}} - (\lambda_i + \lambda_j)\right\}\frac{y_i y_j}{\lambda_i\lambda_j} \,.\end{aligned} \tag{211}$$

Define $\boldsymbol{\Omega} := \boldsymbol{U}^T\dot{\boldsymbol{U}}$. Then

$$\boldsymbol{\Omega}^T = \dot{\boldsymbol{U}}^T\boldsymbol{U} = -\boldsymbol{U}^T\dot{\boldsymbol{U}} = -\boldsymbol{\Omega} \,, \tag{212}$$

so we can rewrite the off-diagonal dynamics as

$$\Omega_{ij}(\lambda_j - \lambda_i) = -\eta sgp_{\text{hit}}\left\{\frac{2}{\tilde{\beta}} - (\lambda_i + \lambda_j)\right\}\frac{y_i y_j}{\lambda_i\lambda_j} \,. \tag{213}$$

Now dynamics of $\boldsymbol{y}$ follow

$$\frac{d}{d\tau}\boldsymbol{y} = \frac{d}{d\tau}\boldsymbol{U}^T\Delta\boldsymbol{b} = \dot{\boldsymbol{U}}^T\boldsymbol{U}\boldsymbol{y} = -\boldsymbol{\Omega}\boldsymbol{y} \,, \tag{214}$$

or in terms of components,

$$\dot{y}_i = -\sum_{j\neq i}\eta sgp_{\text{hit}}\left\{\frac{2}{\tilde{\beta}} - (\lambda_i + \lambda_j)\right\}\frac{1}{(\lambda_i - \lambda_j)}\frac{y_i y_j^2}{\lambda_i\lambda_j} \,. \tag{215}$$

Winner-take-all dynamics. Because $\boldsymbol{\Omega}$ is skew-symmetric, there is a conservation law: the norm of $\boldsymbol{y}$ never changes. To see this, observe that

$$\frac{d}{d\tau}\frac{\boldsymbol{y}^T\boldsymbol{y}}{2} = -\boldsymbol{y}^T\boldsymbol{\Omega}^T\boldsymbol{y} - \boldsymbol{y}^T\boldsymbol{\Omega}\boldsymbol{y} = 0 \,. \tag{216}$$

Hence, the sum of the squares of the y_i must always add up to

$$\boldsymbol{y}^T\boldsymbol{y} = (\Delta\boldsymbol{b})^T\boldsymbol{U}\boldsymbol{U}^T(\Delta\boldsymbol{b}) = \|\Delta\boldsymbol{b}\|_2^2 \,. \tag{217}$$

This has the following consequence. If one of the y_i is somewhat larger than the rest, the dynamics of its corresponding eigenvalue λ_i (Eq. 210) will tend to ensure that $\lambda_i \approx y_i^2$. If the other eigenvalues are somewhat small, for example because the corresponding y_j are small, then the dynamics of the larger y_i (Eq. 215) will likely have a positive right-hand side. Hence, the larger y_i will get even larger, until eventually $|y_i| \approx \|\Delta\boldsymbol{b}\|_2$. In this sense, we obtain 'winner-take-all' dynamics when there is an imbalance in the y_i.

We can see this most explicitly in the 2D case. If we only have y_i and y_j with $y_i^2 + y_j^2 = \|\boldsymbol{y}\|_2^2$, then there are two ODEs

$$\begin{aligned}\dot{y}_i &= -\eta sgp_{\text{hit}}\left\{\frac{2}{\tilde{\beta}} - (\lambda_i + \lambda_j)\right\}\frac{1}{(\lambda_i - \lambda_j)}\frac{y_i y_j^2}{\lambda_i\lambda_j} \\ \dot{y}_j &= -\eta sgp_{\text{hit}}\left\{\frac{2}{\tilde{\beta}} - (\lambda_i + \lambda_j)\right\}\frac{1}{(\lambda_j - \lambda_i)}\frac{y_j y_i^2}{\lambda_i\lambda_j} \,.\end{aligned} \tag{218}$$

By our conservation law, $y_j^2 = \|\boldsymbol{y}\|_2^2 - y_i^2$, so

$$\dot{y}_i = \eta sgp_{\text{hit}}\left\{\lambda_i + \lambda_j - \frac{2}{\tilde{\beta}}\right\}\frac{1}{(\lambda_i - \lambda_j)}\frac{y_i(y^2 - y_i^2)}{\lambda_i\lambda_j} \,. \tag{219}$$

If $\|\boldsymbol{y}\|_2^2$ is bigger than $2/\tilde{\beta}$, and $y_i^2 \neq y_j^2$, then at least one y_i^2 is bigger than $1/\tilde{\beta}$. Then at least one λ_i will be pushed toward the nontrivial fixed point. Then Eq. 219 indicates that the bigger y_i will increase until $y_i^2 = \|\boldsymbol{y}\|_2^2$.