The Policy Gradient Theorem

Infinite-Horizon Expected Discounted Reward

Sai Sampath Kedari
sampath@umich.edu

Contents

1 The Objective and Why Its Gradient Is Hard **2**
1.1 What θ Controls . . . 2
1.2 Following a Single Step, and Then the Whole Trajectory . . . 2
1.3 Setup . . . 3
1.4 Notation and Dimensions . . . 3

2 Differentiating the Value: Two Terms **4**
2.1 The Recursion in Stacked Form . . . 5

3 Unrolling the Recursion **5**

4 From the Jacobian to the Visitation Form **6**
4.1 The Row Vector $\rho_0^\top P_{\pi_\theta}^t$ . . . 6
4.2 Collecting the Discounted Sum . . . 7
4.3 Normalizing the Weights into a Distribution . . . 7

5 The Theorem **8**
5.1 From a Sum to an Expectation: the Likelihood-Ratio Step . . . 8

6 Reading the Gradient **8**
6.1 The Score: How the Policy Responds to θ . . . 9
6.2 The Value: Which Actions to Favour . . . 11
6.3 Putting the States Together . . . 12

7 Where the Visitation Shift Went **13**
7.1 The Term We Were Afraid Of . . . 13
7.2 It Was Absorbed into "Use Q, not r" . . . 13

A The Matrix Expansion: $\rho_0^\top P_{\pi_\theta}^t$ as a Row of State Probabilities **15**

B The Discounted Visitation Measure $\eta_\theta(s)$ **15**

1 The Objective and Why Its Gradient Is Hard

We have a policy $\pi_\theta(a \mid s)$ parameterized by $\theta \in \mathbb{R}^{d'}$, a fixed start-state distribution ρ_0, and the discounted objective

$$J(\theta) \;=\; \sum_s \rho_0(s)\, V^{\pi_\theta}(s).$$

The objective is a single number, and we want its gradient so we can move θ in the direction that raises the return. Before deriving anything, let us see what one small change to θ actually does. That is what makes the gradient look hard.

1.1 What θ Controls

The parameter θ controls one thing: the policy. For each state s, the policy $\pi_\theta(\cdot \mid s)$ is a probability distribution over the actions. Since it is a distribution, the probabilities add up to one. So if we change θ and push more probability onto one action, that probability has to come from somewhere: the other actions lose exactly as much as this one gains. The total stays one; the mass just moves around between actions.

That is the only thing θ does. In particular, θ does not appear in the environment. The environment is the rule $p(s', r \mid s, a)$: given a state s and an action a, it tells us the chance of each next state s' and reward r. That rule belongs to the world, not to us. We cannot change it, and there is no θ inside it. The only knob we have is θ, and it is connected only to the policy.

1.2 Following a Single Step, and Then the Whole Trajectory

Sit at a state s and watch what happens in one step.

- We pick an action from our policy: $a \sim \pi_\theta(\cdot \mid s)$. This is the only place θ is used.
- We give that action to the environment, and the environment gives us back a reward r and the next state s'.

Now change θ a little. The probabilities over actions at s move, so we are likely to pick a different action than before. We give this different action to the environment, and the environment gives us back a different reward and a different next state. The environment did nothing new. It followed the same rule as always. We just gave it a different action, and a different action leads to a different result. So changing *which action we pick* is what changed *where we land*.

This does not stop after one step. The next state s' is a new place where the policy picks an action, and the policy uses the same θ we just changed. So at s' we again pick a slightly different action, and again land in a slightly different state s'', and the same thing happens at s'', all the way down the chain. One small change to θ therefore changes the whole path: every state we visit later, and every reward we collect along the way, shifts. And $V^{\pi_\theta}(s)$, the value we are trying to differentiate, is just the discounted sum of all those rewards over the whole path. This is why θ reaches into the entire future.

The two effects, and why the gradient looks hard. Changing θ has two effects.
The first is *which actions we pick*. This is a change in the action probabilities, and its local sensitivity is measured by $\nabla_\theta \pi_\theta(a \mid s)$, which we can compute exactly because we built the policy ourselves.
The second is *which states we end up in*: by changing our actions we change where the chain goes, so we change how often we visit each state over the whole future. That second effect is the worry, because "where we go" came out of the environment $p(s', r \mid s, a)$, which we do not know. It looks like the gradient must contain a piece measuring how our visits shift, and measuring that seems to need the environment, which we cannot differentiate. If that were really required, policy gradient would be impossible without a model of the world.

The policy gradient theorem says this worry is unfounded. We will show that $\nabla_\theta J$ comes out using only $\nabla_\theta \pi_\theta$, the easy effect, weighted by how often we visit each state, with no derivative of the environment and no derivative of the visits anywhere in it. The change in where we go is real, but we never have to differentiate it; we get it for free by running the policy and seeing where we end up. The rest of this report makes that precise.

1.3 Setup

The state space is finite,

$$\mathcal{S} = \{s_1, \dots, s_n\}, \qquad n := |\mathcal{S}|,$$

and the action space is finite. From the joint environment kernel we define the transition probability and the expected one-step reward,

$$p(s' \mid s, a) := \sum_r p(s', r \mid s, a), \qquad r(s, a) := \sum_{s', r} r \, p(s', r \mid s, a),$$

so that the value and action-value satisfy

$$V^{\pi_\theta}(s) = \sum_a \pi_\theta(a \mid s) \, Q^{\pi_\theta}(s, a), \qquad Q^{\pi_\theta}(s, a) = r(s, a) + \gamma \sum_{s'} p(s' \mid s, a) \, V^{\pi_\theta}(s').$$

With $\gamma < 1$ and an infinite horizon, $Q^{\pi_\theta}(s, a)$ is time-independent: from any state the remaining horizon is always infinite, so "steps left" never changes with the clock, and we may carry a single $Q^{\pi_\theta}(s, a)$ with no time index throughout. Episodic tasks fit this framework without change: model an absorbing terminal state as a self-loop with zero reward.

Unless stated otherwise, every trajectory probability below is taken under $S_0 \sim \rho_0$, actions drawn from π_θ, and the fixed environment dynamics.

1.4 Notation and Dimensions

Orientation matters in the matrix manipulations ahead, so we fix it once.

Scalars. The quantities $\pi_\theta(a \mid s)$, $V^{\pi_\theta}(s)$, $Q^{\pi_\theta}(s, a)$, $r(s, a)$, and $J(\theta)$ are all real numbers.

Gradients are columns. Differentiating any scalar with respect to $\theta \in \mathbb{R}^{d'}$ gives a column vector:

$$\nabla_\theta \pi_\theta(a \mid s), \quad \nabla_\theta V^{\pi_\theta}(s), \quad \nabla_\theta Q^{\pi_\theta}(s, a), \quad \nabla_\theta J(\theta) \in \mathbb{R}^{d' \times 1}.$$

Distributions are rows. The start distribution is

$$\rho_0^\top = \begin{bmatrix} \rho_0(s_1) & \cdots & \rho_0(s_n) \end{bmatrix} \in \mathbb{R}^{1\times n}.$$

State functions are columns. Stacking the state-values gives

$$\mathbf{V}_\theta := \begin{bmatrix} V^{\pi_\theta}(s_1) \\ \vdots \\ V^{\pi_\theta}(s_n) \end{bmatrix} \in \mathbb{R}^{n\times 1}, \qquad \text{so} \qquad J(\theta) = \rho_0^\top \mathbf{V}_\theta.$$

Jacobians. The map $\theta \mapsto \mathbf{V}_\theta$ sends $\mathbb{R}^{d'}$ to $\mathbb{R}^n$, so its derivative is a matrix:

$$D_\theta \mathbf{V}_\theta := \begin{bmatrix} \nabla_\theta V^{\pi_\theta}(s_1)^\top \\ \vdots \\ \nabla_\theta V^{\pi_\theta}(s_n)^\top \end{bmatrix} \in \mathbb{R}^{n\times d'},$$

one row per state, one column per parameter coordinate. This is the Jacobian of a vector-valued map; it is not a gradient, and it should not be written $\nabla_\theta \mathbf{V}_\theta$. For the scalar objective the same rule gives a row vector, which is exactly the transpose of the gradient:

$$D_\theta J(\theta) = \nabla_\theta J(\theta)^\top \in \mathbb{R}^{1\times d'}.$$

We will derive $\nabla_\theta J(\theta)^\top$ and transpose at the end.

2 Differentiating the Value: Two Terms

Differentiate $V^{\pi_\theta}(s) = \sum_a \pi_\theta(a \mid s)\, Q^{\pi_\theta}(s,a)$ by the product rule. Both factors carry θ, so

$$\nabla_\theta V^{\pi_\theta}(s) = \underbrace{\sum_a \nabla_\theta \pi_\theta(a \mid s)\, Q^{\pi_\theta}(s,a)}_{\textbf{(A)}} + \underbrace{\sum_a \pi_\theta(a \mid s)\, \nabla_\theta Q^{\pi_\theta}(s,a)}_{\textbf{(B)}} \in \mathbb{R}^{d'\times 1}.$$

In (A) the scalar $Q^{\pi_\theta}(s,a)$ weights the column $\nabla_\theta \pi_\theta(a \mid s)$; in (B) the scalar $\pi_\theta(a \mid s)$ weights the column $\nabla_\theta Q^{\pi_\theta}(s,a)$.

What (A) and (B) mean. **(A)** freezes every downstream action-value $Q^{\pi_\theta}(s,a)$ as a fixed number and asks: as θ moves and probability mass slides between actions, how much does the average value at s change from that re-aiming of mass alone? This is the *reinforce* term: push mass toward high-value actions.
(B) freezes the action probabilities at s and asks: how much does the value of the future itself move, because Q^{π_θ} is computed under the same θ we are perturbing? This is the downstream ripple, the same "where we go" worry from Section 1, now appearing inside the derivative.

Now expand (B). In $Q^{\pi_\theta}(s,a) = r(s,a) + \gamma \sum_{s'} p(s' \mid s,a) V^{\pi_\theta}(s')$ the reward r and the transition p are environment quantities with no θ in them, so they differentiate to zero:

$$\nabla_\theta Q^{\pi_\theta}(s,a) = \gamma \sum_{s'} p(s' \mid s,a)\, \nabla_\theta V^{\pi_\theta}(s') \in \mathbb{R}^{d'\times 1}.$$

Because the environment does not depend on θ,

$$\nabla_\theta Q^{\pi_\theta}(s,a) = \gamma \sum_{s'} p(s' \mid s,a)\, \nabla_\theta V^{\pi_\theta}(s').$$

Differentiating Q^{π_θ} produces another $\nabla_\theta V^{\pi_\theta}$ one step later, never a $\nabla_\theta p$. The transition probabilities appear only as averaging weights.

Substitute (B) back and define the policy-induced one-step transition and the *local reinforce vector*

$$P_{\pi_\theta}(s' \mid s) := \sum_a \pi_\theta(a \mid s)\, p(s' \mid s,a) \;\in\; \mathbb{R}, \qquad \phi_\theta(s) := \sum_a \nabla_\theta \pi_\theta(a \mid s)\, Q^{\pi_\theta}(s,a) \;\in\; \mathbb{R}^{d' \times 1}.$$

Here $\phi_\theta(s)$ is exactly term (A) at state s: the gain in value from re-aiming the action probabilities at s, with every downstream Q^{π_θ} held fixed. It is local in *where the derivative lands*, since the derivative hits only $\pi_\theta(a \mid s)$ at the current state. It is *not* local in reward information, because the weight $Q^{\pi_\theta}(s,a)$ already carries the entire discounted downstream future.

The differentiation now collapses to a recursion, one column equation per state:

$$\nabla_\theta V^{\pi_\theta}(s) = \phi_\theta(s) + \gamma \sum_{s'} P_{\pi_\theta}(s' \mid s)\, \nabla_\theta V^{\pi_\theta}(s') \;\in\; \mathbb{R}^{d' \times 1}.$$

2.1 The Recursion in Stacked Form

Stack the per-state reinforce vectors the same way we stacked the values:

$$\Phi_\theta := \begin{bmatrix} \phi_\theta(s_1)^\top \\ \vdots \\ \phi_\theta(s_n)^\top \end{bmatrix} \in \mathbb{R}^{n \times d'}, \qquad [P_{\pi_\theta}]_{ij} = P_{\pi_\theta}(s_j \mid s_i), \quad P_{\pi_\theta} \in \mathbb{R}^{n \times n}.$$

Each row of P_{π_θ} is a probability distribution over next states, so P_{π_θ} is *row-stochastic*. The per-state recursion then becomes a single matrix equation for the Jacobian:

$$D_\theta \mathbf{V}_\theta = \Phi_\theta + \gamma\, P_{\pi_\theta}\, D_\theta \mathbf{V}_\theta, \qquad D_\theta \mathbf{V}_\theta,\ \Phi_\theta \in \mathbb{R}^{n \times d'}, \quad P_{\pi_\theta} \in \mathbb{R}^{n \times n}.$$

The shapes check: $P_{\pi_\theta}\, D_\theta \mathbf{V}_\theta$ is $(n \times n)(n \times d') = n \times d'$, matching Φ_θ. The transition matrix acts on the state index alone; the parameter index is untouched. This is a fixed-point equation for the Jacobian $D_\theta \mathbf{V}_\theta$, analogous to a Bellman equation but for the parameter sensitivity of the value vector rather than the value itself.

3 Unrolling the Recursion

The recursion is linear in $D_\theta \mathbf{V}_\theta$. Substitute it into its own right-hand side once, then again, and after K rounds

$$D_\theta \mathbf{V}_\theta = \sum_{t=0}^{K} \gamma^t\, P_{\pi_\theta}^t\, \Phi_\theta \;+\; \gamma^{K+1} P_{\pi_\theta}^{K+1} D_\theta \mathbf{V}_\theta,$$

where every term is an $n \times d'$ matrix. Bound the leftover tail in the matrix max-norm $\|A\|_\infty = \max_i \sum_j |A_{ij}|$, which is submultiplicative. Since P_{π_θ} is row-stochastic, so is every power of it, hence $\|P_{\pi_\theta}^{K+1}\|_\infty = 1$ and

$$\left\|\gamma^{K+1} P_{\pi_\theta}^{K+1} D_\theta \mathbf{V}_\theta\right\|_\infty \;\le\; \gamma^{K+1} \left\|D_\theta \mathbf{V}_\theta\right\|_\infty \;\xrightarrow[K\to\infty]{}\; 0,$$

because $\gamma < 1$. What remains is the geometric solution

$$D_\theta \mathbf{V}_\theta = \sum_{t=0}^{\infty} \gamma^t \, P_{\pi_\theta}^t \, \Phi_\theta \qquad (\in \mathbb{R}^{n \times d'}).$$

This is the only place the infinite-horizon discount is used in the unrolling: it is what kills the leftover. The standing assumptions are finite state and action spaces, bounded rewards, $\gamma < 1$, a policy differentiable in θ, and the resulting $D_\theta \mathbf{V}_\theta$ bounded.

(B) is just (A) at the next state. The ripple introduced no new kind of object. $\nabla_\theta V^{\pi_\theta}(s')$ inside (B) opens, by the same product rule, into "(A) at s'" plus "(B) at s'," and (B) at s' reaches one step further. Unrolling forever expresses the entire value-function Jacobian as a discounted accumulation of the local reinforce term Φ_θ, evaluated at every state and weighted by the probability of reaching it.

4 From the Jacobian to the Visitation Form

Since $J(\theta) = \rho_0^\top \mathbf{V}_\theta$ and ρ_0 does not depend on θ, differentiating gives

$$\nabla_\theta J(\theta)^\top = \rho_0^\top \, D_\theta \mathbf{V}_\theta \;\in\; \mathbb{R}^{1 \times d'},$$

a product of shapes $(1 \times n)(n \times d')$. This is the Jacobian of the scalar objective, equal to the transpose of the gradient we are after. Substituting the unrolled solution and carrying the row vector inside the sum,

$$\nabla_\theta J(\theta)^\top = \rho_0^\top \Big[\sum_{t=0}^{\infty} \gamma^t P_{\pi_\theta}^t\Big] \Phi_\theta = \Big[\sum_{t=0}^{\infty} \gamma^t \, \rho_0^\top P_{\pi_\theta}^t\Big] \Phi_\theta.$$

Everything now hinges on the row vector $\rho_0^\top P_{\pi_\theta}^t$, the start distribution pushed t steps forward through the chain.

4.1 The Row Vector $\rho_0^\top P_{\pi_\theta}^t$

The matrix $P_{\pi_\theta}^t$ is the t-step transition matrix: its (i, j) entry is $\mathbb{P}(S_t{=}s_j \mid S_0{=}s_i)$. Left-multiplying by the start distribution marginalizes out the starting state, by the law of total probability:

$$\big(\rho_0^\top P_{\pi_\theta}^t\big)_j = \sum_i \rho_0(s_i) \, \mathbb{P}(S_t{=}s_j \mid S_0{=}s_i) = \mathbb{P}(S_t = s_j).$$

Writing $\mu_t^\top := \rho_0^\top P_{\pi_\theta}^t$, this says

$$\mu_t^\top = \begin{bmatrix} \mathbb{P}(S_t{=}s_1) & \cdots & \mathbb{P}(S_t{=}s_n) \end{bmatrix} \;\in\; \mathbb{R}^{1 \times n},$$

the unconditional distribution of the chain's position at time t. The full bookkeeping is in Appendix A.

4.2 Collecting the Discounted Sum

Substituting and summing entrywise over t,

$$\nabla_\theta J(\theta)^\top = \underbrace{\Big[\sum_{t=0}^{\infty} \gamma^t \mu_t^\top\Big]}_{1\times n} \underbrace{\Phi_\theta}_{n\times d'}.$$

The bracket is a row of n scalar weights. Define the j-th of them as

$$\eta_\theta(s) := \sum_{t=0}^{\infty} \gamma^t \, \mathbb{P}(S_t{=}s) \;\in\; \mathbb{R}, \qquad \boldsymbol{\eta}_\theta^\top := \big[\eta_\theta(s_1) \quad \cdots \quad \eta_\theta(s_n)\big] \in \mathbb{R}^{1\times n},$$

the *discounted visitation measure*, which depends on θ through the chain. Then

$$\nabla_\theta J(\theta)^\top = \boldsymbol{\eta}_\theta^\top \, \Phi_\theta = \sum_s \eta_\theta(s)\, \phi_\theta(s)^\top \;\in\; \mathbb{R}^{1\times d'}.$$

Row i of Φ_θ is $\phi_\theta(s_i)^\top$, so multiplying by the visitation row scales each row by its weight and sums the rows. Transposing both sides returns the column gradient:

$$\boxed{\nabla_\theta J(\theta) = \sum_s \eta_\theta(s)\, \phi_\theta(s) \;\in\; \mathbb{R}^{d'\times 1}.}$$

What the product is saying. The row $\boldsymbol{\eta}_\theta^\top$ holds, for each state, the discounted total time the chain spends there. The matrix Φ_θ holds, in row s, the transposed local reinforce vector. The product scales each state's reinforce direction by how much discounted time we spend in that state and sums the results into a single direction in parameter space.

4.3 Normalizing the Weights into a Distribution

The weights $\eta_\theta(s)$ are nonnegative but do not sum to one, so they form a measure, not a distribution. Their total mass is

$$\sum_s \eta_\theta(s) = \sum_{t=0}^{\infty} \gamma^t \underbrace{\sum_s \mathbb{P}(S_t{=}s)}_{=1} = \frac{1}{1-\gamma},$$

because at each fixed time the chain is somewhere. Dividing by this total mass gives the *discounted visitation distribution*

$$d_\theta(s) := (1-\gamma)\,\eta_\theta(s), \qquad \sum_s d_\theta(s) = 1, \qquad \eta_\theta(s) = \frac{1}{1-\gamma}\, d_\theta(s).$$

As a row vector, $\mathbf{d}_\theta^\top = (1-\gamma)\,\boldsymbol{\eta}_\theta^\top \in \mathbb{R}^{1\times n}$. Equivalently,

$$\eta_\theta(s) = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t \, \mathbb{1}\{S_t = s\}\Big],$$

the expected discounted number of visits to s; this reading is developed in Appendix B.

5 The Theorem

Substituting $\eta_\theta(s) = \frac{1}{1-\gamma} d_\theta(s)$ and expanding $\phi_\theta(s)$:

$$\nabla_\theta J(\theta) = \frac{1}{1-\gamma} \sum_s d_\theta(s) \sum_a \nabla_\theta \pi_\theta(a \mid s)\, Q^{\pi_\theta}(s,a) \;\in\; \mathbb{R}^{d' \times 1}.$$

The constant $\frac{1}{1-\gamma}$ is positive and independent of θ, so it sets only the scale, not the direction, and is absorbed into the step size.

5.1 From a Sum to an Expectation: the Likelihood-Ratio Step

The boxed form is a weighted sum over states and actions. To turn it into something we can *sample*, we rewrite it as an expectation. The tool is the likelihood-ratio identity, which is just $\nabla_\theta \log f = \nabla_\theta f / f$ rearranged:

$$\nabla_\theta \pi_\theta(a \mid s) \;=\; \pi_\theta(a \mid s)\, \nabla_\theta \log \pi_\theta(a \mid s), \qquad \nabla_\theta \log \pi_\theta(a \mid s) \in \mathbb{R}^{d' \times 1}.$$

The point of this rewrite is that it manufactures a scalar factor of $\pi_\theta(a \mid s)$ out front. Substituting,

$$\nabla_\theta J(\theta) = \frac{1}{1-\gamma} \sum_s d_\theta(s) \sum_a \pi_\theta(a \mid s)\, \nabla_\theta \log \pi_\theta(a \mid s)\, Q^{\pi_\theta}(s,a).$$

Now read the two weighted sums as nested expectations: the outer $\sum_s d_\theta(s)$ averages over $S \sim d_\theta$, and the inner $\sum_a \pi_\theta(a \mid s)$ averages over $A \sim \pi_\theta(\cdot \mid S)$.

$$\nabla_\theta J(\theta) = \frac{1}{1-\gamma}\, \mathbb{E}_{S \sim d_\theta,\, A \sim \pi_\theta(\cdot \mid S)} \big[\, \nabla_\theta \log \pi_\theta(A \mid S)\, Q^{\pi_\theta}(S,A) \,\big] \;\in\; \mathbb{R}^{d' \times 1}.$$

Two distributions, nested: the outer $S \sim d_\theta$ is *where you are*, the discounted visitation set by ρ_0 and the dynamics; the inner $A \sim \pi_\theta(\cdot \mid S)$ is *what you do there*, the policy itself.

Why the expectation form is the one that matters. An expectation is the one shape we can estimate without computing anything in closed form: draw samples, evaluate the bracket, average. Both draws are things we already do by running the policy, and the bracket is built from the analytic score and a value we can estimate from returns. This is why every policy gradient algorithm starts from this line.

6 Reading the Gradient

The gradient is

$$\nabla_\theta J(\theta) \;=\; \sum_s \eta_\theta(s)\, \phi_\theta(s) \;=\; \frac{1}{1-\gamma} \sum_s d_\theta(s)\, \phi_\theta(s),$$

a sum over states of two pieces multiplied together: a number $\eta_\theta(s)$ and a vector $\phi_\theta(s)$. We read them one at a time.

$\eta_\theta(s)$*: how often we are there.* The discounted number of times we visit s over the whole run. A visit now counts as one; a visit at time t counts as γ^t. So $\eta_\theta(s)$ is large for states we reach often and early, small for states we reach rarely or only far in the future.

$\phi_\theta(s)$: what tuning the policy at s does. This is term (A) of Section 2, written at the action level:

$$\phi_\theta(s) = \mathbb{E}_{A\sim\pi_\theta(\cdot|s)}\big[\, Q^{\pi_\theta}(s,A)\,\nabla_\theta \log \pi_\theta(A \mid s)\,\big].$$

It is built from one object, the score $\nabla_\theta \log \pi_\theta(A \mid s)$, weighted by the value $Q^{\pi_\theta}(s,A)$. We look at the score first, then at what the weighting does.

Reading the next two figures. They assume $d' = 1$, so the score is a single number that can be positive or negative and drawn as a slope. They also draw a continuous action space, so $\pi_\theta(\cdot \mid s)$ is a density and sums over actions become integrals. The theorem itself holds for any d' and for discrete actions as derived above; the figures only trade generality for something we can draw.

6.1 The Score: How the Policy Responds to θ

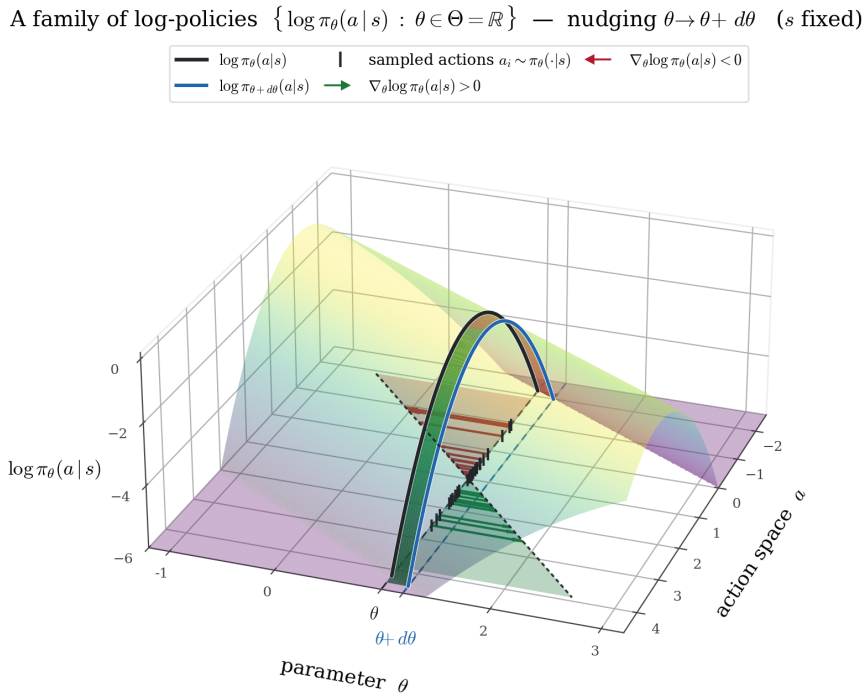


Figure 1: A fixed state s. Each slice at a fixed θ is a distribution over the action space $\mathcal{A}$, drawn on a log scale, so the surface is the whole family $\{\log \pi_\theta(\cdot \mid s) : \theta \in \mathbb{R}\}$. Black is the current $\log \pi_\theta(\cdot \mid s)$; blue is $\log \pi_{\theta+d\theta}(\cdot \mid s)$, one small step in θ away. The arrows are the score $\nabla_\theta \log \pi_\theta(a \mid s)$, the slope of the surface in the θ direction: green where the log-probability rises with θ, red where it falls.

We are standing at one state s and keeping it fixed. For that state the policy $\pi_\theta(\cdot \mid s)$ is a distribution over the action space $\mathcal{A}$. But the figure does not show a single distribution. It shows a whole family of them, one for every value of the parameter,

$$\big\{\, \pi_\theta(\cdot \mid s) \;:\; \theta \in \Theta \,\big\},$$

drawn on a log scale so the tails stay visible. The parameter θ is nothing more than an index into this family: it names which distribution over $\mathcal{A}$ we are using right now. Moving θ does not touch

the actions and does not touch the world; it just slides us from one distribution in the family to a neighbouring one.

Pick the current value of θ. That single value is one slice through the surface, the black curve. Reading along it from left to right is our policy at s right now. Now move θ by a hair. That gives the neighbouring slice, the blue curve, sitting just beside the black one. Same state, same actions, nothing else touched: only the index moved.

Here is the whole point of the picture. Lay the blue curve over the black one and read it action by action: where the blue sits above the black, that action's probability rose when we moved θ, marked green; where it sits below, the probability fell, marked red. That green-and-red reading is the score $\nabla_\theta \log \pi_\theta(a \mid s)$, the slope of the surface in the θ direction. Steeply rising is a large positive score; steeply falling is a large negative one; flat is a score near zero. It is the gradient of the *log*-policy, which is exactly why we drew the log surface and not the policy itself.

Step back and see what this really is. Before it is a tool for anything, the score answers a single question about the family of distributions: as I move the parameter, how does my distribution over actions respond? It reports, action by action, which ones the distribution is pouring probability onto and which ones it is pulling probability off. That is a statement about the shape of the family itself, and it is worth holding onto before we ask the score to do any work for us.

Only now read it as control. If we want the green actions to carry more probability, we move θ one way; if we want the red actions to carry more, we move it the other. But we do not get to pick actions one at a time. Because θ indexes the *whole* distribution, a single step raises every green action and lowers every red one together.

And on its own the score picks no winner. The actions it raises are balanced by the actions it lowers, so averaged over the actions the policy actually takes it comes out to zero,

$$\mathbb{E}_{A\sim\pi_\theta(\cdot|s)}\big[\,\nabla_\theta \log \pi_\theta(A \mid s)\,\big] = 0 \qquad \text{for every } \theta.$$

The score is a complete description of how θ reshapes the distribution, yet as a direction to move it is neutral: it tells us how probability *can* be moved, not which way it *should* go.

So the score hands us a steering wheel and no destination. What breaks the tie is how good the actions are, the value $Q^{\pi_\theta}(s, a)$: the next figure takes this same score and weights each action by its value.

6.2 The Value: Which Actions to Favour

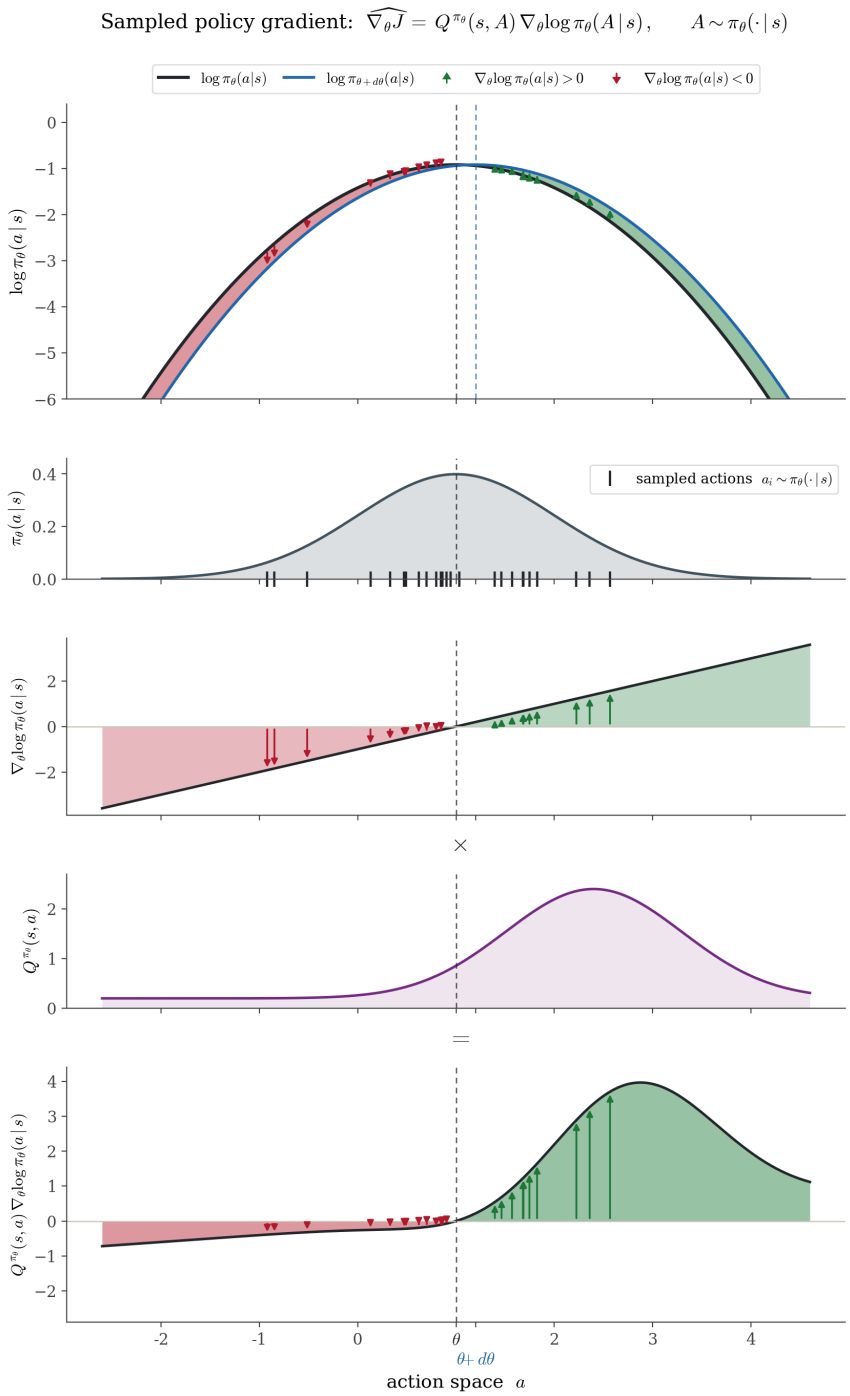


Figure 2: The same state, read as a multiplication from top to bottom. The score (third panel) has its ups and downs balanced. The value $Q^{\pi_\theta}(s,a)$ (fourth panel) is not flat, it is larger on the better actions. Multiplying the two tips the balance: the bottom panel, $Q^{\pi_\theta}(s,a)\,\nabla_\theta \log \pi_\theta(a \mid s)$, leans toward the high-value actions. Its average over the sampled actions is $\phi_\theta(s)$.

The score left us with a steering wheel and no destination. It tells us how the knob moves each action's probability, but not which actions deserve more. The missing piece is the value $Q^{\pi_\theta}(s, a)$: how good it is to take action a in state s and then carry on with the policy. This is the knowledge that finally says where to drive.

The figure stacks the two pieces as a multiplication, read top to bottom: the policy slices from before, the policy density with tick marks for the actions we actually sampled, the score on its own as a rising line that is as much below the axis as above it, the value $Q^{\pi_\theta}(s, a)$ which is emphatically *not* flat, and finally their product.

Now watch what the multiplication did. In the score panel the up-part and the down-part balanced. But the value is large on the good side and small on the poor side. Multiply them, and the up-arrows on the good actions get stretched tall while the arrows on the poor actions get squashed small. The balance breaks. The bottom panel no longer cancels: it leans clearly toward the high-value actions.

That lopsided total, averaged over the sampled actions, is $\phi_\theta(s)$. The score said which way each action moves; the value said which actions we want; the product says which way to actually turn the knob.

One caution, because it is tempting to picture this wrongly. We do not find the single best action and chase only it. Every action takes part: each one's score is scaled by its own value, and all of them are summed. It is the whole weighted sum that tips one way, not any single action's arrow.

What we are really reaching for. Ideally we would pour all the probability onto the single best action $a^\star = \arg\max_a Q^{\pi_\theta}(s, a)$ and forget the rest. We cannot do that directly. The only thing we hold is the knob θ, and turning it moves the whole distribution at once. So $\phi_\theta(s)$ is the first-order local improvement direction obtained by redistributing action probability at s with the action-values held fixed. Take the step, resample, take it again, and over many updates the policy leans harder and harder toward $a^\star$. We keep the policy spread out rather than jumping straight there so that we go on trying other actions and learning their values.

One last observation, and it plants a seed for a later report. Because the score's ups and downs cancel, only the *differences* in value between actions matter, never the overall level. Add the same constant to every $Q^{\pi_\theta}(s, a)$ and $\phi_\theta(s)$ does not budge, since that constant multiplies the mean-zero score and washes out. What drives the step is that some actions beat the state's average value, and that excess, $Q^{\pi_\theta}(s, a) - V^{\pi_\theta}(s)$, is the advantage. Isolating it is the subject of the actor-critic baseline report.

6.3 Putting the States Together

Everything so far lived at one state. But the agent passes through many states, and one parameter vector has to serve all of them at once. The full gradient handles this in the plainest way possible, by adding up the per-state directions, each weighted by how much that state matters:

$$\nabla_\theta J(\theta) = \sum_s \eta_\theta(s)\, \phi_\theta(s).$$

So a state pulls on the gradient only when *both* factors are sizeable. A state we sit in constantly but already act well in has large $\eta_\theta(s)$ and $\phi_\theta(s) \approx 0$; a state where our actions are badly wrong

but which we almost never reach has the reverse. Only states that are both visited often and still improvable move the gradient.

The gradient in one sentence. At each state the score says how the parameter moves each action and the value says which actions are worth favouring; their product is a direction that pushes probability toward the better actions there; and the policy gradient adds up those per-state directions, each weighted by how often we visit the state.

We just treated $\phi_\theta(s)$ as a *local* gain at s, holding the future fixed, yet turning the knob really does move everything downstream. Where did that downstream effect go? The next section answers exactly that.

7 Where the Visitation Shift Went

Section 6 read $\phi_\theta(s)$ as a local gain with the future held fixed. But turning the knob moves the whole trajectory, so it also changes $\eta_\theta(s)$, how often we reach each state. Why did that effect never show up in the gradient?

7.1 The Term We Were Afraid Of

Write the objective with the immediate reward pulled out, $\bar{r}_\theta(s) := \sum_a \pi_\theta(a \mid s)\, r(s,a)$:

$$J(\theta) = \sum_s \eta_\theta(s)\, \bar{r}_\theta(s).$$

A naive product rule predicts *two* terms:

$$\nabla_\theta J = \underbrace{\sum_s \eta_\theta(s)\, \nabla_\theta \bar{r}_\theta(s)}_{\text{channel 1: what you do}} + \underbrace{\sum_s \nabla_\theta \eta_\theta(s)\, \bar{r}_\theta(s)}_{\text{channel 2: where you go}}.$$

The second term is the nightmare. It is not a $\nabla_\theta p$ term, since the environment carries no θ, but computing $\nabla_\theta \eta_\theta$ still means tracking how the policy-induced transition P_{π_θ} moves with θ through every power $P_{\pi_\theta}^t$ that builds up η_θ. That needs the model. The policy gradient theorem never computes it.

7.2 It Was Absorbed into "Use Q, not r"

The recursion never produced that two-term split. It produced the single form $\nabla_\theta J = \sum_s \eta_\theta(s)\, \phi_\theta(s)$, with $\eta_\theta(s)$ appearing only as an *undifferentiated* weight. The reason is the trail it came down: in $D_\theta \mathbf{V}_\theta = \Phi_\theta + \gamma P_{\pi_\theta} D_\theta \mathbf{V}_\theta$ the transition matrix entered undifferentiated, and unrolling turned its powers into the state distributions $\rho_0^\top P_{\pi_\theta}^t$, and hence into the visitation row. No transition was ever differentiated.

The visitation shift did not vanish. Subtracting the two expressions for the same gradient, the naive one above and the theorem $\nabla_\theta J = \sum_s \eta_\theta(s) \sum_a \nabla_\theta \pi_\theta(a \mid s) Q^{\pi_\theta}(s,a)$, and noting that $\nabla_\theta \bar{r}_\theta(s) = \sum_a \nabla_\theta \pi_\theta(a \mid s)\, r(s,a)$, gives:

$$\underbrace{\sum_s \nabla_\theta \eta_\theta(s)\, \bar{r}_\theta(s)}_{\text{channel 2}} \;=\; \sum_s \eta_\theta(s) \sum_a \nabla_\theta \pi_\theta(a \mid s) \left[\, Q^{\pi_\theta}(s,a) - r(s,a) \,\right].$$

The would-be $\nabla_\theta \eta_\theta$ correction equals the future-reward part $Q^{\pi_\theta} - r$, weighted by the visitation. It was absorbed into promoting the immediate reward r to the full Q^{π_θ} inside ϕ_θ.

So the two views, "r plus a $\nabla_\theta \eta_\theta$ correction" and "Q^{π_θ} with no correction," are the same gradient. Writing Q^{π_θ} instead of r *is* accounting for where you go.

The aha. You never *predict* how your policy reshapes your future states; you simply *live* it. Run the policy and the states you land in already *are* the reshaped distribution. The one piece you could not have computed without a model, how your visitation shifts, you never compute at all, because it is already folded into crediting each action with the full return Q^{π_θ} instead of the immediate reward r, and the return is something you can sample. This gives the *exact* gradient, not an approximation.

This is what lets policy gradient run on pure interaction. No $\nabla_\theta \eta_\theta$ and no $\nabla_\theta p$ survive, so every ingredient is either analytic or sampleable: the score we built ourselves, the value from sampled returns or a learned critic, and the visitation from simply running the policy.

A The Matrix Expansion: $\rho_0^\top P_{\pi_\theta}^t$ as a Row of State Probabilities

This appendix supports the body's claim that $\rho_0^\top P_{\pi_\theta}^t$ is the row of unconditional time-t state probabilities. Everything here is on the state index alone; the parameter dimension plays no role until the final multiplication by Φ_θ.

One step. The entry (i, j) of P_{π_θ} is the one-step probability of moving from s_i to s_j under the policy:

$$P_{\pi_\theta} = \begin{pmatrix} \mathbb{P}(S_1{=}s_1 \mid S_0{=}s_1) & \cdots & \mathbb{P}(S_1{=}s_n \mid S_0{=}s_1) \\ \vdots & \ddots & \vdots \\ \mathbb{P}(S_1{=}s_1 \mid S_0{=}s_n) & \cdots & \mathbb{P}(S_1{=}s_n \mid S_0{=}s_n) \end{pmatrix}.$$

Each row is a probability distribution over next states.

t **steps.** By Chapman–Kolmogorov, $P_{\pi_\theta}^t$ holds the t-step transition probabilities:

$$P_{\pi_\theta}^t = \begin{pmatrix} \mathbb{P}(S_t{=}s_1 \mid S_0{=}s_1) & \cdots & \mathbb{P}(S_t{=}s_n \mid S_0{=}s_1) \\ \vdots & \ddots & \vdots \\ \mathbb{P}(S_t{=}s_1 \mid S_0{=}s_n) & \cdots & \mathbb{P}(S_t{=}s_n \mid S_0{=}s_n) \end{pmatrix}.$$

Left-multiply by the start distribution. A row vector times a matrix is a row vector, whose j-th entry sums the start probability of each state s_i times the probability of being at s_j after t steps from s_i:

$$\big(\rho_0^\top P_{\pi_\theta}^t\big)_j = \sum_i \mathbb{P}(S_0{=}s_i)\,\mathbb{P}(S_t{=}s_j \mid S_0{=}s_i) = \mathbb{P}(S_t = s_j),$$

which is the law of total probability.

$$\rho_0^\top P_{\pi_\theta}^t = \begin{bmatrix} \mathbb{P}(S_t{=}s_1) & \cdots & \mathbb{P}(S_t{=}s_n) \end{bmatrix} \in \mathbb{R}^{1\times n}.$$

This collapsed row is the whole point. We began with a matrix of *conditional* t-step probabilities and a row of start probabilities. Multiplying them marginalizes out the start: the result no longer conditions on S_0, it is the plain distribution of the chain's position at time t. Multiplying it by Φ_θ, whose row i is $\phi_\theta(s_i)^\top$, then produces the probability-weighted sum of the transposed per-state vectors used in the body.

B The Discounted Visitation Measure $\eta_\theta(s)$

The coefficient $\eta_\theta(s) = \sum_{t=0}^{\infty} \gamma^t \mathbb{P}(S_t{=}s)$ carries the gradient's dependence on the start distribution and the dynamics. It has no parameter dimension, which is exactly why it appears as an undifferentiated scalar weight on each $\phi_\theta(s)$.

From probabilities to an indicator count. Fix a state s and define the indicator $\mathbb{1}\{S_t = s\}$, equal to one if the chain is in s at time t and zero otherwise. Its expectation is the occupancy probability, $\mathbb{E}[\mathbb{1}\{S_t = s\}] = \mathbb{P}(S_t = s)$. Substituting and pulling the expectation outside the sum by linearity,

$$\eta_\theta(s) = \sum_{t=0}^{\infty} \gamma^t\, \mathbb{P}(S_t = s) = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t\, \mathbb{1}\{S_t = s\}\Big].$$

Reading the inside. For a single trajectory, $\sum_t \mathbb{1}\{S_t = s\}$ is literally the number of times that trajectory lands in s. The factor γ^t re-weights each visit by when it happened: a visit on step 0 counts as a full one, a visit ten steps in counts as γ^{10}, and visits in the far future count for almost nothing.

$\eta_\theta(s)$ is the **expected discounted number of visits to** s, averaged over trajectories started from ρ_0 and run under π_θ. When finite, the undiscounted quantity $\mathbb{E}[\sum_t \mathbb{1}\{S_t = s\}]$ is the plain expected visit count; the γ^t makes it a discounted count in which the early part of the chain dominates.

Why discounting localizes it. The weights $1, \gamma, \gamma^2, \ldots$ sum to $\frac{1}{1-\gamma}$ and decay to negligible after roughly that many steps. So $\eta_\theta(s)$ effectively counts visits only within that early window. A state reached only deep in the chain contributes almost nothing, regardless of how often it is visited later. This is the precise sense in which the discounted objective forgets the far future: not by ignoring it in the return, but by weighting the visitation so that the early, start-anchored portion of the trajectory carries the mass.