Quiz on Policy Gradient Guarantees

By Taha Majlesi

July 17, 2025

1 Multiple Choice Questions (80 Questions)

Instructions: Select the best answer for each question.

1. **What is the primary advantage of policy gradient methods over value-based methods?**

   (a) They are simpler to implement.
   (b) They are more sample efficient.
   (c) They can handle continuous action spaces and learn stochastic policies.
   (d) They always converge to the optimal policy faster.

   Answer: c

2. **What does the objective function $J(\theta)$ in policy gradient methods represent?**

   (a) The probability of reaching a terminal state.
   (b) The expected total discounted reward.
   (c) The immediate reward at the first timestep.
   (d) The variance of the policy's parameters.

   Answer: b

3. **The policy $\pi_\theta(a|s)$ is a function that outputs a:**

   (a) Deterministic action.
   (b) State value.
   (c) Probability distribution over actions.
   (d) Q-value for a state-action pair.

   Answer: c

4. **What is the purpose of the discount factor γ?**

   (a) To make the algorithm more complex.
   (b) To ensure the sum of rewards is finite and to prioritize immediate rewards.
   (c) To increase the learning rate.
   (d) To normalize the policy parameters.

   Answer: b

5. **The core challenge in directly differentiating the objective function $J(\theta)$ is that:**

   (a) The reward function is unknown.
   (b) The expectation is over a distribution $p_\theta(\tau)$ that depends on θ.
   (c) Neural networks cannot be differentiated.
   (d) The state space is too large.

   Answer: b

6. **What is the "log-derivative trick"?**

   (a) A method for taking the logarithm of a negative number.

   (b) A way to approximate the gradient.

   (c) An identity: $\nabla_x f(x) = f(x)\nabla_x \log f(x)$.

   (d) A technique for reducing variance.

   Answer: c

7. **A critical result of applying the log-derivative trick in the Policy Gradient Theorem derivation is that the final gradient expression does not depend on:**

   (a) The policy parameters θ.

   (b) The reward function.

   (c) The environment's dynamics model.

   (d) The discount factor γ.

   Answer: c

8. **The REINFORCE algorithm is also known as:**

   (a) Actor-Critic.

   (b) Q-Learning.

   (c) Monte Carlo Policy Gradient.

   (d) Dynamic Programming.

   Answer: c

9. **How does REINFORCE estimate the policy gradient?**

   (a) By solving the Bellman equation.

   (b) By using a function approximator for the value function.

   (c) By taking a sample mean over a batch of trajectories.

   (d) By using a second-order optimization method.

   Answer: c

10. **In a more effective variant of REINFORCE, the total trajectory reward $R(\tau)$ is replaced by the:**

    (a) Immediate reward $r(s_t, a_t)$.

    (b) Advantage function.

    (c) Reward-to-go $\hat{Q}^{\pi}_{i,t}$.

    (d) State-value function.

    Answer: c

11. **The term $\nabla_\theta \log \pi_\theta(a|s)$ is known as the:**

    (a) Advantage function.

    (b) Score function.

    (c) Value function.

    (d) Policy update.

    Answer: b

12. **What is the major practical drawback of the REINFORCE algorithm?**

    (a) It is biased.

(b) It has high variance in its gradient estimates.

(c) It cannot be used with neural networks.

(d) It only works in deterministic environments.

Answer: b

13. **High variance in REINFORCE is primarily caused by the absolute magnitude of the:**

(a) Policy parameters.

(b) Learning rate.

(c) Score function.

(d) Reward-to-go term.

Answer: d

14. **How is the Advantage Function $A^\pi(s,a)$ defined?**

(a) $V^\pi(s) - Q^\pi(s,a)$

(b) $Q^\pi(s,a) - V^\pi(s)$

(c) $r(s,a) + \gamma V^\pi(s')$

(d) $\sum_t \gamma^t r_t$

Answer: b

15. **Using a baseline in the policy gradient update is a technique for:**

(a) Introducing bias to speed up learning.

(b) Reducing variance without introducing bias.

(c) Making the algorithm off-policy.

(d) Increasing the learning rate.

Answer: b

16. **In an Actor-Critic method, the "actor" refers to the _____ and the "critic" refers to the _____.**

(a) Value function, Policy

(b) Policy, Value function

(c) Environment, Agent

(d) Agent, Environment

Answer: b

17. **If the advantage $A^\pi(s,a)$ is negative, the policy update should:**

(a) Increase the probability of action a.

(b) Decrease the probability of action a.

(c) Keep the probability of action a the same.

(d) Re-evaluate the value function.

Answer: b

18. **Why does subtracting a state-dependent baseline $b(s_t)$ not introduce bias?**

(a) Because the baseline is always positive.

(b) Because the expectation of the baseline term is zero.

(c) Because the learning rate is small.

(d) Because the baseline is learned by a separate network.

Answer: b

19. **Policy gradient methods can be understood as a "soft" form of which classical algorithm?**

    (a) Value Iteration.
    (b) Q-Learning.
    (c) Monte Carlo Tree Search.
    (d) Generalized Policy Iteration (GPI).

    Answer: d

20. **The "evaluation" step in an actor-critic method is analogous to which step in classical Policy Iteration?**

    (a) Policy Improvement.
    (b) Policy Evaluation.
    (c) Greedy Update.
    (d) Initialization.

    Answer: b

21. **The Policy Improvement Identity, $J(\theta') - J(\theta) = \mathbb{E}_{\tau \sim p_{\theta'}(\tau)}[\dots]$, provides a formal guarantee for:**

    (a) Reducing variance.
    (b) Policy improvement.
    (c) Off-policy correction.
    (d) Sample efficiency.

    Answer: b

22. **What is the "chicken-and-egg" problem revealed by the Policy Improvement Identity?**

    (a) We need the new policy to evaluate the expectation, but the evaluation is needed to find the new policy.
    (b) We need the value function to find the policy, and the policy to find the value function.
    (c) We need rewards to learn, but we need to learn to get rewards.
    (d) We need a model to plan, but we need to plan to build a model.

    Answer: a

23. **What statistical technique is used to solve the off-policy evaluation problem in policy gradients?**

    (a) Bootstrapping.
    (b) Conjugate Gradient.
    (c) Importance Sampling (IS).
    (d) Taylor Expansion.

    Answer: c

24. **The importance weight $w(x)$ is defined as:**

    (a) $p(x) - q(x)$
    (b) $p(x) + q(x)$
    (c) $p(x)/q(x)$
    (d) $q(x)/p(x)$

Answer: c

25. **A major problem with importance sampling is that it can lead to:**

    (a) Biased estimates.

    (b) High variance.

    (c) Deterministic policies.

    (d) Slow convergence.

    Answer: b

26. **The surrogate objective $L_\theta(\theta')$ is an approximation of the true performance improvement that can be estimated using data from:**

    (a) The new policy $\pi_{\theta'}$.

    (b) The old policy π_θ.

    (c) A random policy.

    (d) An optimal policy.

    Answer: b

27. **The validity of the surrogate objective approximation hinges on the assumption that:**

    (a) The environment is deterministic.

    (b) The reward function is linear.

    (c) The new policy $\pi_{\theta'}$ is close to the old policy π_θ.

    (d) The learning rate is large.

    Answer: c

28. **Exploding variance in importance sampling is a primary motivation for the concept of a:**

    (a) Value function.

    (b) Trust region.

    (c) Discount factor.

    (d) Replay buffer.

    Answer: b

29. **The theoretical lower bound on performance improvement is given by $J(\theta') - J(\theta) \geq L_\theta(\theta') -$ ErrorTerm, where the error term depends on:**

    (a) The learning rate.

    (b) The number of samples.

    (c) The distance between the old and new policies.

    (d) The size of the neural network.

    Answer: c

30. **What is a "trust region"?**

    (a) A set of states with high value.

    (b) The part of the state space the agent has explored.

    (c) A neighborhood around the current policy where the surrogate objective is a good approximation.

    (d) The set of parameters for which the policy is guaranteed to be optimal.

    Answer: c

31. **Taking a very large policy update step risks:**

    (a) Slowing down learning.

    (b) Leaving the trust region and causing a performance collapse.

    (c) Overfitting to the current batch of data.

    (d) Violating the Bellman equation.

    Answer: b

32. **Trust Region Policy Optimization (TRPO) explicitly constrains the new policy to stay within a trust region defined by:**

    (a) Euclidean distance in parameter space.

    (b) Total Variation distance.

    (c) Kullback-Leibler (KL) divergence.

    (d) The L2 norm of the advantage function.

    Answer: c

33. **TRPO uses which algorithm to efficiently solve its constrained optimization problem?**

    (a) Stochastic Gradient Descent.

    (b) Adam.

    (c) Conjugate Gradient.

    (d) Newton's Method.

    Answer: c

34. **What is the main disadvantage of TRPO?**

    (a) It is not sample efficient.

    (b) It has high variance.

    (c) It is complex to implement.

    (d) It does not guarantee monotonic improvement.

    Answer: c

35. **Proximal Policy Optimization (PPO) is a simpler alternative to:**

    (a) REINFORCE.

    (b) A2C.

    (c) DQN.

    (d) TRPO.

    Answer: d

36. **How does PPO create a trust region?**

    (a) By using a KL-divergence constraint.

    (b) By using a clipped surrogate objective function.

    (c) By using a very small learning rate.

    (d) By using a backtracking line search.

    Answer: b

37. **In the PPO-Clip objective, what does the 'clip' function do?**

    (a) It clips the rewards to be within a certain range.

    (b) It clips the gradients to prevent them from exploding.

(c) It constrains the policy probability ratio $r_t(\theta)$ to be within $[1-\epsilon, 1+\epsilon]$.

(d) It clips the advantage function to be positive.

Answer: c

38. **When the advantage is positive in PPO, the objective's increase is capped. What is the purpose of this?**

(a) To make the algorithm run faster.

(b) To remove the incentive for making the policy update too large.

(c) To ensure the advantage is always positive.

(d) To simplify the implementation.

Answer: b

39. **Compared to TRPO, PPO is:**

(a) More theoretically rigorous but harder to implement.

(b) Less stable but more sample efficient.

(c) Simpler to implement and often has comparable or better performance.

(d) Only applicable to discrete action spaces.

Answer: c

40. **The success of PPO highlights that a _____ algorithm can often have a greater impact than a more rigorous but complex one.**

(a) theoretically pure

(b) second-order

(c) value-based

(d) scalable and practical

Answer: d

41. **The probability of a trajectory $p_\theta(\tau)$ depends on the policy π_θ and the:**

(a) Reward function.

(b) Environment's transition dynamics.

(c) Value function.

(d) Learning rate.

Answer: b

42. **The policy gradient update rule $\theta_{k+1} = \theta_k + \alpha \nabla_\theta J(\theta_k)$ is an instance of:**

(a) Gradient descent.

(b) Gradient ascent.

(c) Newton's method.

(d) The Bellman equation.

Answer: b

43. **The principle of causality in REINFORCE means an action at time t can only influence rewards at times:**

(a) $t' < t$

(b) $t' = t$

(c) $t' \geq t$

(d) For all t'

Answer: c

44. **If all rewards in an environment are positive, what is the issue with the basic REINFORCE update?**

    (a) The gradient will be zero.
    (b) Every action will be reinforced, leading to a noisy signal.
    (c) The algorithm will diverge.
    (d) The log-probability will be undefined.

Answer: b

45. **The advantage function provides a _____ comparison of actions, rather than an absolute one.**

    (a) biased
    (b) absolute
    (c) relative
    (d) noisy

Answer: c

46. **The proof that baselines are unbiased relies on the fact that $\nabla_\theta \sum_a \pi_\theta(a|s)$ equals:**

    (a) 1
    (b) ∞
    (c) -1
    (d) 0

Answer: d

47. **Generalized Policy Iteration (GPI) refers to the interaction between policy evaluation and:**

    (a) Policy initialization.
    (b) Policy improvement.
    (c) Model learning.
    (d) Reward shaping.

Answer: b

48. **The policy improvement step in actor-critic methods is considered "soft" because it:**

    (a) Is not guaranteed to improve the policy.
    (b) Uses a small learning rate.
    (c) Gently pushes the policy in the direction of positive advantages.
    (d) Is computationally inexpensive.

Answer: c

49. **The term $r(s_t, a_t) + \gamma V^{\pi_\theta}(s_{t+1})$ is a one-sample estimate of:**

    (a) $V^{\pi_\theta}(s_t)$
    (b) $A^{\pi_\theta}(s_t, a_t)$
    (c) $Q^{\pi_\theta}(s_t, a_t)$
    (d) $J(\theta)$

Answer: c

50. **The off-policy problem arises because we need to evaluate the new policy but only have samples from the:**

    (a) Optimal policy.
    (b) Random policy.
    (c) Old policy.
    (d) Deterministic policy.

    Answer: c

51. **For importance sampling to be valid, the support of the behavior distribution $q(x)$ must _____ the support of the target distribution $p(x)$.**

    (a) be disjoint from
    (b) be identical to
    (c) cover
    (d) be smaller than

    Answer: c

52. **The surrogate objective $L_\theta(\theta')$ makes the approximation that the _____ does not change significantly.**

    (a) reward function
    (b) policy
    (c) state visitation distribution
    (d) action space

    Answer: c

53. **The bound on the change in state distribution, $|p_{\theta'}(s_t) - p_\theta(s_t)| \leq 2\epsilon t$, shows that the difference grows _____ with the time horizon t.**

    (a) quadratically
    (b) exponentially
    (c) logarithmically
    (d) linearly

    Answer: d

54. **The key theoretical result $J(\theta') - J(\theta) \geq L_\theta(\theta') - \text{ErrorTerm}$ guarantees that maximizing the surrogate objective maximizes a _____ on the true improvement.**

    (a) upper bound
    (b) lower bound
    (c) tight bound
    (d) biased estimate

    Answer: b

55. **TRPO uses a second-order approximation of the KL-divergence, which is the:**

    (a) Hessian matrix.
    (b) Jacobian matrix.
    (c) Fisher Information Matrix.
    (d) Covariance matrix.

    Answer: c

56. **The probability ratio in PPO is defined as $r_t(\theta) =$:**

(a) $\pi_\theta(a_t|s_t) - \pi_{\theta_{\text{old}}}(a_t|s_t)$
(b) $\pi_{\theta_{\text{old}}}(a_t|s_t)/\pi_\theta(a_t|s_t)$
(c) $\pi_\theta(a_t|s_t)/\pi_{\theta_{\text{old}}}(a_t|s_t)$
(d) $\log \pi_\theta(a_t|s_t)$

Answer: c

57. **PPO's clipped objective is optimized using _____ optimization.**

(a) second-order
(b) first-order
(c) derivative-free
(d) constrained

Answer: b

58. **The journey from REINFORCE to PPO shows a continuous effort to balance theoretical rigor, stability, and:**

(a) mathematical elegance.
(b) implementation complexity.
(c) hardware requirements.
(d) biological plausibility. **Answer: b**

59. **Which algorithm provides theoretical guarantees of monotonic improvement under certain conditions?**

(a) REINFORCE
(b) PPO
(c) TRPO
(d) DQN

Answer: c

60. **Which algorithm is considered a heuristic/practical approximation of the TRPO objective?**

(a) REINFORCE
(b) PPO
(c) A2C
(d) DDPG

Answer: b

61. **The term "on-policy" means that the data used for updates is collected using:**

(a) A completely different policy.
(b) The most recent version of the policy being optimized.
(c) A replay buffer of old policies.
(d) An optimal policy.

Answer: b

62. **Which of these algorithms is fully on-policy with no data reuse between updates?**

(a) TRPO
(b) PPO
(c) REINFORCE
(d) DQN

Answer: c

63. **The "soft" update in policy gradients is contrasted with the "hard" greedy update in which algorithm?**

(a) REINFORCE
(b) Classical Policy Iteration
(c) PPO
(d) Actor-Critic

Answer: b

64. **The derivation of the Policy Gradient Theorem removes the dependency on the environment model, making the methods:**

(a) Model-based
(b) Model-free
(c) Value-based
(d) Off-policy

Answer: b

65. **What does a stochastic policy allow an agent to do, which is beneficial in partially observable environments?**

(a) Always choose the best action.
(b) Explore by trying different actions for the same state.
(c) Compute the value function exactly.
(d) Converge faster.

Answer: b

66. **The term $\mathbb{E}_{\tau\sim p_\theta(\tau)}[\cdot]$ denotes an expectation over:**

(a) States
(b) Actions
(c) Trajectories
(d) Rewards

Answer: c

67. **In the Policy Improvement Identity, the advantage function A^{π_θ} is calculated with respect to the:**

(a) New policy $\pi_{\theta'}$
(b) Old policy π_θ
(c) Optimal policy π^*
(d) Random policy

Answer: b

68. **The state visitation distribution $p_\theta(s_t)$ is the distribution of states at timestep t induced by following policy:**

(a) π_θ
(b) $\pi_{\theta'}$
(c) π^*
(d) A uniform random policy

Answer: a

69. **The core idea of a baseline is to center the learning signal around:**

(a) 1
(b) The maximum reward

(c) The average reward
(d) 0

Answer: d

70. **The conjugate gradient algorithm is used in TRPO to avoid explicitly forming and inverting the:**

(a) Policy network.
(b) Value network.
(c) Fisher Information Matrix.
(d) Reward matrix.

Answer: c

71. **In PPO, if the advantage is negative, the objective clips the policy ratio $r_t(\theta)$ from going below:**

(a) 0
(b) $1 + \epsilon$
(c) $1 - \epsilon$
(d) -1

Answer: c

72. **The final conclusion of the report is that _____ has become a de facto standard due to its balance of performance and simplicity.**

(a) REINFORCE
(b) TRPO
(c) PPO
(d) A3C

Answer: c

73. **The "reward-to-go" is an unbiased estimate of:**

(a) $V^\pi(s_t)$
(b) $Q^\pi(s_t, a_t)$
(c) $A^\pi(s_t, a_t)$
(d) $J(\theta)$

Answer: b

74. **The "chicken-and-egg" problem of the Policy Improvement Identity is an example of a(n) _____ problem.**

(a) on-policy
(b) off-policy evaluation
(c) credit assignment
(d) exploration-exploitation

Answer: b

75. **The approximation $p_{\theta'}(s_t) \approx p_\theta(s_t)$ is more likely to be valid if the policy update is:**

(a) Large
(b) Small
(c) Random
(d) Biased

Answer: b

76. **TRPO's use of second-order methods makes it more _____ than PPO.**

(a) simple
(b) stable
(c) complex
(d) biased

Answer: c

77. **The PPO objective function takes the _____ of the unclipped and clipped objectives.**

(a) maximum
(b) minimum
(c) sum
(d) product

Answer: b

78. **The entire theoretical progression in the report aims to solve the _____ problem in policy gradient methods.**

(a) exploration
(b) stability
(c) generalization
(d) memory

Answer: b

2 Explanatory Questions (30 Questions)

Instructions: Provide a detailed explanation for each question.

1. **Question:** Explain the fundamental difference between policy gradient methods and value-based methods. Why are policy gradient methods particularly suited for continuous action spaces?

   Answer: Value-based methods, like Q-learning, learn a value function (e.g., $Q(s, a)$) that estimates the expected return of taking an action in a state. The policy is then derived implicitly from this value function, typically by choosing the action with the highest value. In contrast, policy gradient methods directly parameterize the policy itself, $\pi_\theta(a|s)$, and optimize the parameters θ using gradient ascent on an objective function $J(\theta)$.

   This direct parameterization is crucial for continuous action spaces. In a continuous space, finding the action that maximizes $Q(s, a)$ would require a separate optimization procedure at every single timestep, which is computationally infeasible. Policy gradient methods avoid this by directly outputting the parameters of a probability distribution (e.g., the mean and standard deviation of a Gaussian) from which an action can be sampled, making them naturally applicable to continuous domains.

2. **Question:** What is the Policy Gradient Theorem, and what is the significance of the "log-derivative trick" in its derivation?

   Answer: The Policy Gradient Theorem provides a way to compute the gradient of the expected total reward objective, $J(\theta)$, with respect to the policy parameters, θ. It states that $\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)}[(\sum_t \nabla_\theta \log \pi_\theta(a_t|s_t))R(\tau)]$. The main challenge in deriving this is that the expectation is over a distribution of trajectories, $p_\theta(\tau)$, which itself depends on θ.

   The "log-derivative trick" is the key mathematical step that makes this tractable. The trick uses the identity $\nabla_x f(x) = f(x)\nabla_x \log f(x)$. By applying this to the gradient of the trajectory probability, $\nabla_\theta p_\theta(\tau)$, we get $p_\theta(\tau)\nabla_\theta \log p_\theta(\tau)$. This reintroduces the probability $p_\theta(\tau)$ back into the expression, allowing the entire term to be rewritten as an expectation that can be estimated via sampling. Crucially, this process also eliminates the need to know the environment's dynamics, $p(s_{t+1}|s_t, a_t)$, as the gradient of the log of the dynamics term is zero with respect to θ.

3. **Question:** Describe the REINFORCE algorithm and interpret its update rule.

   Answer: REINFORCE, or Monte Carlo Policy Gradient, is the most direct implementation of the Policy Gradient Theorem. It works in three steps:

   (a) **Generate Samples:** Run the current policy π_θ to collect a set of trajectories.

   (b) **Estimate Gradient:** Use these trajectories to compute a Monte Carlo estimate of the policy gradient. The gradient estimate is $\hat{g} = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T}\nabla_\theta \log \pi_\theta(a_{i,t}|s_{i,t})\hat{Q}^\pi_{i,t}$, where $\hat{Q}^\pi_{i,t}$ is the sampled reward-to-go from that timestep.

   (c) **Update Policy:** Update the policy parameters using gradient ascent: $\theta \leftarrow \theta + \alpha\hat{g}$.

   The update rule has an intuitive interpretation: $\nabla_\theta \log \pi_\theta(a|s)$ is the direction in parameter space that increases the probability of taking action a in state s. This direction is weighted by the reward-to-go $\hat{Q}^\pi$. If $\hat{Q}^\pi$ is high (a good outcome), the update pushes the policy to make that action more likely. If $\hat{Q}^\pi$ is low or negative (a bad outcome), the update pushes the policy to make that action less likely.

4. **Question:** What is the problem of high variance in REINFORCE, and how does using a baseline, specifically the advantage function, address it?

   Answer: The problem of high variance in REINFORCE stems from the reward-to-go term, $\hat{Q}^\pi$. The absolute value of this term can fluctuate wildly depending on the trajectory and the reward structure. For instance, if all rewards are positive, every action is reinforced, making it hard for the algorithm to distinguish between "good" and "very good" actions. This noisy signal leads to unstable updates and slow convergence.

   Using a baseline addresses this by changing the learning signal from an absolute measure to a relative one. The most common baseline is the state-value function, $V^\pi(s)$. Subtracting this from the action-value function gives the **Advantage Function**: $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$. The advantage asks, "How much better was this action than the average action I would take from this state?"

   - If $A^\pi > 0$, the action was better than average and its probability is increased.

- If $A^\pi < 0$, the action was worse than average and its probability is decreased.

This centering of the learning signal around zero provides a much clearer, more stable gradient estimate, significantly reducing variance.

5. **Question:** Prove that subtracting a state-dependent baseline $b(s_t)$ from the policy gradient estimator does not introduce bias.

   Answer: To prove that a baseline $b(s_t)$ does not introduce bias, we must show that the expectation of the term we are adding to the gradient is zero. The added term is proportional to $\nabla_\theta \log \pi_\theta(a_t|s_t)b(s_t)$. We need to show $\mathbb{E}[\nabla_\theta \log \pi_\theta(a_t|s_t)b(s_t)] = 0$.

   Let's analyze the expectation over actions, conditioned on the state s_t:

   $$\mathbb{E}_{a_t\sim\pi_\theta(\cdot|s_t)}[\nabla_\theta \log \pi_\theta(a_t|s_t)b(s_t)]$$

   Since $b(s_t)$ does not depend on the action a_t, we can pull it out of the expectation:

   $$= b(s_t)\mathbb{E}_{a_t\sim\pi_\theta(\cdot|s_t)}[\nabla_\theta \log \pi_\theta(a_t|s_t)]$$

   Now, let's expand the inner expectation:

   $$\mathbb{E}_{a_t\sim\pi_\theta(\cdot|s_t)}[\nabla_\theta \log \pi_\theta(a_t|s_t)] = \sum_{a_t} \pi_\theta(a_t|s_t)\nabla_\theta \log \pi_\theta(a_t|s_t)$$

   Using the log-derivative trick in reverse ($\nabla f = f\nabla \log f$), this becomes:

   $$= \sum_{a_t} \nabla_\theta \pi_\theta(a_t|s_t)$$

   We can swap the sum and the gradient:

   $$= \nabla_\theta \sum_{a_t} \pi_\theta(a_t|s_t)$$

   Since $\pi_\theta(a_t|s_t)$ is a probability distribution, the sum of probabilities over all actions is 1.

   $$= \nabla_\theta(1) = 0$$

   Because the inner expectation is zero, the entire expression is zero. Thus, subtracting a state-dependent baseline does not change the expected gradient and is therefore unbiased.

6. **Question:** Explain the connection between policy gradient methods (specifically actor-critic) and Generalized Policy Iteration (GPI).

   Answer: Generalized Policy Iteration (GPI) is the general algorithmic pattern of alternating between two processes: policy evaluation and policy improvement.

   - **Policy Evaluation:** Given a policy, estimate its value function.
   - **Policy Improvement:** Given a value function, improve the policy (e.g., by acting greedily).

   Actor-critic methods fit this pattern perfectly. The "critic" (the value function network) performs a form of policy evaluation by learning the value (V^π) or advantage (A^π) of the current policy. The "actor" (the policy network) then performs policy improvement by updating its parameters θ based on the critic's evaluation. Instead of a "hard" greedy update like in classical PI, the actor performs a "soft" update, taking a gradient step to make actions with positive advantages more likely. This reveals that actor-critic methods are a continuous, gradient-based instance of the GPI framework.

7. **Question:** What is the Policy Improvement Identity, and what practical challenge does it highlight?

   Answer: The Policy Improvement Identity is a fundamental equation that provides a formal guarantee for policy improvement. It states:

   $$J(\theta') - J(\theta) = \mathbb{E}_{\tau\sim p_{\theta'}(\tau)}\left[\sum_{t=0}^{\infty} \gamma^t A^{\pi_\theta}(s_t, a_t)\right]$$

   This means the improvement in performance from an old policy π_θ to a new policy $\pi_{\theta'}$ is equal to the expected sum of advantages of the new policy's actions, where the advantages are calculated with respect to the old policy.

The practical challenge it highlights is a "chicken-and-egg" problem. The identity guarantees improvement if the right-hand side is positive. However, the expectation is taken over trajectories sampled from the **new policy** ($\tau \sim p_{\theta'}(\tau)$). We cannot evaluate this expectation without having the new policy, but the purpose of the evaluation is to find the new policy in the first place. This is a classic off-policy evaluation problem and is the central motivation for developing surrogate objective functions that can be evaluated using data from the old policy.

8. **Question:** How does Importance Sampling (IS) work, and how is it used to create a surrogate objective function for policy gradients?

   Answer: Importance Sampling (IS) is a technique to estimate the expectation of a function under a target distribution $p(x)$ using samples from a different behavior distribution $q(x)$. It works by re-weighting the samples. The key identity is:

   $$\mathbb{E}_{x\sim p}[f(x)] = \mathbb{E}_{x\sim q}\left[\frac{p(x)}{q(x)}f(x)\right]$$

   The term $p(x)/q(x)$ is the importance weight, which corrects for the mismatch in distributions.

   In policy gradients, we want to evaluate the policy improvement objective, which depends on the new policy $\pi_{\theta'}$, using data from the old policy π_θ. We apply IS to the action-selection part of the objective:

   $$\mathbb{E}_{a\sim\pi_{\theta'}}[A^{\pi_\theta}(s,a)] = \mathbb{E}_{a\sim\pi_\theta}\left[\frac{\pi_{\theta'}(a|s)}{\pi_\theta(a|s)}A^{\pi_\theta}(s,a)\right]$$

   This corrects for the difference in action probabilities. To create a fully tractable surrogate objective, $L_\theta(\theta')$, we make a crucial approximation: we assume the state visitation distribution doesn't change much ($p_{\theta'}(s) \approx p_\theta(s)$). This allows us to write the entire objective as an expectation over data from the old policy:

   $$L_\theta(\theta') = \mathbb{E}_{s\sim p_\theta, a\sim\pi_\theta}\left[\frac{\pi_{\theta'}(a|s)}{\pi_\theta(a|s)}A^{\pi_\theta}(s,a)\right]$$

   Maximizing this surrogate objective is the goal of algorithms like TRPO and PPO.

9. **Question:** What is a "trust region," and why is it a crucial concept for stable policy optimization?

   Answer: A trust region is a neighborhood around the current policy π_θ within which the surrogate objective $L_\theta(\theta')$ is a reliable approximation of the true performance improvement $J(\theta')$.

   The concept is crucial because the surrogate objective relies on two approximations: importance sampling for actions and assuming the state distribution doesn't change. Both of these approximations break down if the new policy $\pi_{\theta'}$ becomes too different from the old policy π_θ. A large policy update can lead to:

   (a) **Exploding Importance Weights:** The ratio $\pi_{\theta'}/\pi_\theta$ can become very large, causing the gradient estimate to have extremely high variance.

   (b) **State Distribution Mismatch:** The actual state distribution under $\pi_{\theta'}$ can be very different from the assumed distribution under π_θ, making the surrogate objective a poor, misleading indicator of true performance.

   By constraining the policy update to stay within a "trust region" (i.e., ensuring $\pi_{\theta'}$ stays close to π_θ), we guarantee that the surrogate objective is a faithful lower bound on the true objective. This prevents catastrophic performance drops and ensures stable, monotonic improvement.

10. **Question:** Describe the optimization problem that Trust Region Policy Optimization (TRPO) solves. What makes it difficult to implement?

    Answer: TRPO formalizes the trust region concept by solving a constrained optimization problem at each update step:

    $$\begin{aligned} \underset{\theta}{\text{maximize}} \quad & L_{\theta_{\text{old}}}(\theta) \\ \text{subject to} \quad & \bar{D}_{\text{KL}}(\pi_{\theta_{\text{old}}}||\pi_\theta) \leq \delta \end{aligned}$$

    It aims to maximize the surrogate objective (L) subject to the constraint that the average KL-divergence between the old and new policies is less than a small hyperparameter δ. The KL-divergence is used as a measure of "distance" between the policies.

    TRPO is difficult to implement because solving this constrained problem directly is hard. The practical solution involves:

(a) **Second-Order Methods:** It approximates the objective linearly and the constraint quadratically (using the Fisher Information Matrix as the Hessian of the KL-divergence).

(b) **Conjugate Gradient Algorithm:** To solve the resulting quadratic problem efficiently without forming and inverting the massive Fisher matrix, it uses the conjugate gradient algorithm to find the update direction.

(c) **Backtracking Line Search:** After finding a direction, it performs a line search to find a step size that satisfies the KL constraint and improves the actual objective.

This reliance on second-order optimization and complex subroutines makes the implementation significantly more challenging than standard first-order methods like SGD or Adam.

11. **Question:** Explain the PPO-Clip objective function and how it emulates a trust region using only first-order optimization.

    Answer: Proximal Policy Optimization (PPO) simplifies TRPO by replacing the hard KL-divergence constraint with a novel clipped surrogate objective. The objective is:

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right]$$

    where $r_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_{\text{old}}}(a_t|s_t)$ is the probability ratio and ϵ is a small hyperparameter (e.g., 0.2).

    This objective emulates a trust region in the following way:

    - **If Advantage $\hat{A}_t > 0$:** The agent wants to increase $r_t(\theta)$ to get more reward. However, the 'clip' function creates an upper bound on the objective at $(1+\epsilon)\hat{A}_t$. Once the policy ratio exceeds $1+\epsilon$, there is no further gain from increasing it. This removes the incentive for making an overly large policy update.
    - **If Advantage $\hat{A}_t < 0$:** The agent wants to decrease $r_t(\theta)$ to reduce the penalty. The 'clip' function creates a lower bound on the objective at $(1-\epsilon)\hat{A}_t$. Once the policy ratio goes below $1-\epsilon$, there is no further gain from decreasing it. This prevents the policy from overreacting to a bad action.

    By taking the 'min' of the normal and clipped objectives, PPO creates a pessimistic bound that discourages large updates, effectively keeping the new policy "proximal" to the old one without needing complex second-order methods. This can be optimized with standard gradient ascent algorithms.

12. **Question:** Compare and contrast TRPO and PPO in terms of theoretical guarantees, implementation complexity, and empirical performance.

    Answer:

    - **Theoretical Guarantees:** TRPO is founded on a rigorous theoretical derivation that guarantees monotonic policy improvement under certain conditions. It directly optimizes a lower bound on performance. PPO, on the other hand, is a heuristic. Its clipped objective is not directly derived from the performance lower bound, so it lacks TRPO's hard guarantees, though it is motivated by the same principles.
    - **Implementation Complexity:** TRPO is highly complex. It requires second-order optimization, the conjugate gradient algorithm, and a backtracking line search. This makes it difficult to implement and debug. PPO is much simpler. It uses a first-order optimization method (like Adam) on a modified objective function, making it easy to integrate into standard deep learning frameworks.
    - **Empirical Performance:** Despite its weaker theoretical guarantees, PPO often achieves comparable or even superior performance to TRPO in practice. Its simplicity allows for faster iteration and easier tuning. The overhead of TRPO's complex update step can sometimes be detrimental. PPO's balance of simplicity, stability, and performance has made it one of the most popular and widely used RL algorithms.

13. **Question:** What is the "state distribution mismatch" problem and why does simply applying importance sampling to actions not fully solve the off-policy problem?

    Answer: The state distribution mismatch problem arises because changing the policy π_θ not only changes the probability of actions in a given state but also changes the distribution of states the agent visits over time, $p_\theta(s_t)$. A different policy leads to different decisions, which leads to different trajectories and thus different future states.

The Policy Improvement Identity, $J(\theta') - J(\theta) = \sum_t \gamma^t \mathbb{E}_{s_t \sim p_{\theta'}(s_t)}[\mathbb{E}_{a_t \sim \pi_{\theta'}(a_t|s_t)}[A^{\pi_\theta}(s_t, a_t)]]$, shows this dependency clearly. The expectation is over both the new action distribution $\pi_{\theta'}$ and the new state distribution $p_{\theta'}$.

Simply applying importance sampling to the inner expectation corrects for the action probabilities: $\mathbb{E}_{a \sim \pi_{\theta'}}[A^{\pi_\theta}] = \mathbb{E}_{a \sim \pi_\theta}[\frac{\pi_{\theta'}}{\pi_\theta} A^{\pi_\theta}]$. However, this does not solve the full off-policy problem because the outer expectation is still over the unknown new state distribution, $s_t \sim p_{\theta'}(s_t)$. To create a tractable surrogate objective, we must make the additional, crucial approximation that $p_{\theta'}(s_t) \approx p_\theta(s_t)$, which is only valid for small policy changes.

14. **Question:** Walk through the derivation of the performance lower bound, $J(\theta') - J(\theta) \geq L_\theta(\theta') - C \cdot D_{\text{TV}}(\pi_\theta, \pi_{\theta'})$. What is the intuition behind this result?

    Answer: The derivation connects the true performance improvement to the surrogate objective.

    (a) **Start with the exact improvement identity:** $J(\theta') - J(\theta) = \sum_t \gamma^t \mathbb{E}_{s_t \sim p_{\theta'}(s_t)}[f(s_t)]$, where $f(s_t)$ is the expected advantage under the new policy.

    (b) **Relate expectations:** We relate the expectation under the new state distribution $p_{\theta'}$ to the old one p_θ. For any function f, $\mathbb{E}_{p'}[f] \geq \mathbb{E}_p[f] - \max|f| \cdot D_{TV}(p, p')$, where D_{TV} is the total variation distance.

    (c) **Bound the state distribution distance:** We first show that the distance between state distributions is bounded by the distance between policies. If $\max_{s,a} |\pi' - \pi| \leq \epsilon$, then $D_{TV}(p'_\theta, p_\theta) \leq C'\epsilon$ for some constant C'.

    (d) **Combine:** Substituting these into the improvement identity gives:

    $$J(\theta') - J(\theta) \geq \sum_t \gamma^t \left(\mathbb{E}_{s_t \sim p_\theta(s_t)}[f(s_t)] - \text{ErrorTerm}\right)$$

    The first term is exactly the surrogate objective $L_\theta(\theta')$. The error term is proportional to the distance between the policies.

    $$J(\theta') - J(\theta) \geq L_\theta(\theta') - C \cdot D_{\text{TV}}(\pi_\theta, \pi_{\theta'})$$

    Intuition: This result provides a formal guarantee. It says that the true improvement is at least as good as the surrogate objective, minus a penalty term. The penalty grows as the new policy moves further away from the old one. Therefore, if we maximize the surrogate objective while keeping the policy change small (i.e., staying in the trust region), we are guaranteed to be maximizing a lower bound on the true performance, ensuring stable improvement.

15. **Question:** Why is the KL-divergence a better measure of "distance" between policies than a simple Euclidean distance in the parameter space θ?

    Answer: Euclidean distance in the parameter space, $||\theta' - \theta||_2$, is a poor measure of how much the policy's behavior actually changes. A small change in the parameters θ could lead to a very large change in the output probability distribution for some states, while a large change in parameters might have a negligible effect for other states. The relationship is non-uniform and depends on the network architecture and activations.

    The Kullback-Leibler (KL) divergence, $D_{KL}(\pi_{\theta_{old}}||\pi_\theta)$, directly measures the difference between the output probability distributions themselves. It quantifies the information lost when using π_θ to approximate $\pi_{\theta_{old}}$. By constraining the average KL-divergence, TRPO ensures that the behavior of the policy does not change too drastically, regardless of how much the parameters themselves have moved. It is a measure of change in the functional output space, which is much more relevant to performance than change in the parameter space.

16. **Question:** What is the role of the 'min' operator in the PPO-Clip objective? Explain its effect for both positive and negative advantages.

    Answer: The 'min' operator is the core of the PPO clipping mechanism. It creates a pessimistic objective that discourages large policy updates.

    The objective is $L^{\text{CLIP}} = \mathbb{E}[\min(r_t \hat{A}_t, \text{clip}(r_t, 1-\epsilon, 1+\epsilon)\hat{A}_t)]$.

    - **Case 1: Advantage $\hat{A}_t$ is positive.** The agent wants to increase the probability of the action, which means increasing the ratio r_t. The second term in the 'min' becomes $(1+\epsilon)\hat{A}_t$. The objective is $\min(r_t\hat{A}_t, (1+\epsilon)\hat{A}_t)$. As r_t increases, the objective increases, but only up to the point where $r_t = 1+\epsilon$. After that, the clipped term becomes smaller, so the 'min' operator ensures the objective is capped at $(1+\epsilon)\hat{A}_t$. This removes the incentive to make r_t excessively large.

- **Case 2: Advantage $\hat{A}_t$ is negative.** The agent wants to decrease the probability of the action, which means decreasing the ratio r_t. The second term in the 'min' becomes $(1-\epsilon)\hat{A}_t$. The objective is $\min(r_t\hat{A}_t, (1-\epsilon)\hat{A}_t)$. Since $\hat{A}_t$ is negative, a smaller r_t makes $r_t\hat{A}_t$ larger (less negative). However, the 'min' operator ensures that the objective is bounded by the clipped term. The agent is penalized for making the ratio too small (below $1-\epsilon$), which prevents an overly aggressive update in response to a single bad action.

In both cases, the 'min' operator creates a conservative objective that is only a faithful representation of the unclipped objective when the policy ratio is within the $[1-\epsilon, 1+\epsilon]$ interval, effectively creating a trust region.

17. **Question:** What is the "credit assignment" problem in reinforcement learning, and how does the reward-to-go formulation in REINFORCE attempt to solve it?

    Answer: The credit assignment problem is the challenge of determining which actions in a long sequence are responsible for the final outcome. If a trajectory yields a high total reward, it's unlikely that every single action was equally good. The problem is how to assign "credit" or "blame" to each individual action.

    The original Policy Gradient Theorem formulation uses the total reward $R(\tau) = \sum_{t=0}^{T} \gamma^t r_t$ to weight the score function for every action in the trajectory. This is poor credit assignment, as it implies an early action is responsible for late rewards.

    The reward-to-go formulation, $\hat{Q}_t^\pi = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$, is a better attempt at solving this. It uses the principle of causality: an action taken at timestep t can only affect rewards from timestep t onwards. By weighting the score function $\nabla_\theta \log \pi_\theta(a_t|s_t)$ only by the sum of future rewards, it correctly assigns zero credit for rewards that occurred before the action was taken. While still noisy, this is a more precise form of credit assignment than using the total reward for all timesteps.

18. **Question:** Describe the conceptual journey from the "Variance Problem" to the "Stability Problem" in the development of policy gradient theory.

    Answer: The conceptual journey reflects a deepening understanding of the challenges in policy gradient methods.

    (a) **The Variance Problem:** The first major hurdle identified with the basic REINFORCE algorithm was the high variance of its gradient estimates, caused by the noisy reward-to-go signal.

    (b) **Solution - Baselines:** The solution was to introduce a baseline, most effectively the state-value function, to create the advantage function. This centered the learning signal and reduced variance by changing the question from "was this outcome good?" to "was this outcome better than average?". This led to Actor-Critic methods.

    (c) **The Off-Policy Problem:** This led to a deeper theoretical view of policy gradients as a form of policy iteration. This view revealed the off-policy "chicken-and-egg" problem: the policy improvement guarantee requires data from the new policy we are trying to find.

    (d) **Solution - Importance Sampling:** The solution was to use importance sampling to correct for the data mismatch, leading to a surrogate objective function that could be optimized with data from the old policy.

    (e) **The Stability Problem:** This solution introduced a new, more severe problem. Importance sampling is only reliable when the new and old policies are similar. If the policy update is too large, the importance weights can explode, and the surrogate objective becomes a poor approximation of true performance, leading to instability and potential performance collapse.

    Thus, the focus shifted from simply reducing variance to ensuring the stability of the entire learning process by constraining the size of the policy update, which is the core idea behind trust region methods.

19. **Question:** Why is PPO often referred to as a "pragmatic" algorithm?

    Answer: PPO is called "pragmatic" because it makes a deliberate trade-off, sacrificing some theoretical purity for massive gains in simplicity, usability, and empirical performance.

    TRPO is the "theoretically pure" algorithm. It is directly derived from the performance lower bound and solves a constrained optimization problem to guarantee monotonic improvement. However, this purity comes at the cost of high implementation complexity (conjugate gradients, line searches).

    PPO takes the core idea of TRPO—constraining the policy update—and finds a much simpler, more practical way to achieve a similar effect. The clipped objective is a clever heuristic, not a

direct consequence of the theory. It doesn't offer the same hard guarantees as TRPO. However, in practice, this heuristic works remarkably well. It provides the necessary stability while being compatible with simple first-order optimizers like Adam. This pragmatic choice to favor a simpler, scalable, and robust heuristic over a complex, theoretically rigorous method is why PPO has had such a large impact and is so widely adopted.

20. **Question:** If you were implementing REINFORCE, what two key improvements from the text would you incorporate to make it more practical? Explain why each is important.

    Answer: To make REINFORCE practical, I would incorporate two crucial improvements discussed in the text:

    (a) **Use Reward-to-Go instead of Total Reward:** Instead of weighting every action's score function in a trajectory by the total reward for the entire trajectory, I would weight the score function at timestep t by the sum of discounted rewards from t until the end of the episode (the reward-to-go, $\hat{Q}_t^\pi$). **Importance:** This is a basic form of credit assignment based on causality. An action at time t cannot possibly have influenced rewards that came before it. Using the reward-to-go provides a slightly less noisy and more accurate signal of the consequences of that specific action, leading to a lower (though still high) variance gradient estimate.

    (b) **Introduce a Value Function Baseline:** I would subtract a learned estimate of the state-value function, $\hat{V}(s_t)$, from the reward-to-go term. The update would use the advantage estimate, $\hat{A}_t = \hat{Q}_t^\pi - \hat{V}(s_t)$. This turns the algorithm into an Actor-Critic method. **Importance:** This is the single most important technique for variance reduction. It centers the learning signal. Instead of reinforcing any action that leads to a positive outcome, it only reinforces actions that lead to outcomes better than the expected average for that state. This dramatically reduces the variance of the gradient estimate, leading to much more stable, faster, and more reliable learning.

21. **Question:** What is the Fisher Information Matrix and what is its role in TRPO?

    Answer: The Fisher Information Matrix (FIM) is a concept from information geometry that measures the amount of information a random variable carries about an unknown parameter of its distribution. In the context of TRPO, it serves as a metric on the space of policy parameters.

    Specifically, the FIM is the second-order derivative (Hessian) of the KL-divergence between two policies, evaluated when the policies are identical. That is, $H_{ij} = \frac{\partial^2}{\partial\theta_i \partial\theta_j} D_{KL}(\pi_{\theta_{old}} || \pi_\theta)|_{\theta=\theta_{old}}$.

    Its role in TRPO is to provide a quadratic approximation to the KL-divergence constraint:

    $$D_{KL}(\pi_{\theta_{old}} || \pi_\theta) \approx \frac{1}{2}(\theta - \theta_{old})^T H (\theta - \theta_{old})$$

    By using this approximation, TRPO converts its complex, non-linear constraint into a simple quadratic one. This allows the optimization problem to be solved efficiently using the conjugate gradient algorithm, which can compute the product $H^{-1}g$ (where g is the policy gradient) without ever needing to explicitly compute, store, or invert the potentially massive FIM, H.

22. **Question:** Can policy gradient methods learn deterministic policies? If so, how, and what is a potential downside?

    Answer: Yes, policy gradient methods can learn deterministic policies. A stochastic policy, like a Gaussian distribution over actions, has a variance or standard deviation parameter. As the agent learns and becomes more certain about the optimal action, the policy network can learn to decrease this variance. In the limit, as the variance approaches zero, the stochastic policy becomes a deterministic one, always outputting the mean action.

    A potential downside of learning a purely deterministic policy is the lack of exploration. If the policy becomes deterministic, it will always take the same action in the same state. If this action is suboptimal, the agent has no way to explore other, potentially better actions. This is why exploration is often encouraged, for example, by adding an entropy bonus to the objective function, which incentivizes the policy to maintain some level of randomness (higher entropy), thus ensuring continued exploration.

23. **Question:** Explain the trade-off between taking a small policy update step versus a large one, in the context of the trust region performance bound.

    Answer: The trade-off is governed by the performance lower bound identity: $J(\theta') - J(\theta) \geq L_\theta(\theta') - C \cdot D_{\text{TV}}(\pi_\theta, \pi_{\theta'})$.

- **Small Step:** If we take a very small step, the distance between policies $D_{\text{TV}}(\pi_\theta, \pi_{\theta'})$ is small. This makes the negative error term negligible. The surrogate objective $L_\theta(\theta')$ becomes a very faithful approximation of the true improvement. This guarantees stable, monotonic improvement but at the cost of being slow. The agent takes tiny, safe steps and may require a huge number of updates to converge.
- **Large Step:** If we try to take a large step to learn faster, the distance D_{TV} becomes large. The error term $C \cdot D_{\text{TV}}$ can grow significantly. It's possible to find a new policy $\pi_{\theta'}$ that has a very high surrogate objective value $L_\theta(\theta')$, but the large negative error term could mean the true performance $J(\theta')$ is actually much worse than the original policy's. This is a catastrophic failure where the agent leaves the trust region and "forgets" what it has learned.

The challenge for modern algorithms like TRPO and PPO is to find the largest possible step that can be taken safely, without the error term overwhelming the gains in the surrogate objective, thus balancing learning speed with stability.

24. **Question:** How does the concept of Generalized Policy Iteration (GPI) provide a conceptual bridge between value-based and policy-based methods?

    Answer: Generalized Policy Iteration (GPI) provides a unifying framework that shows how different RL algorithms are related. GPI is the general idea of two interacting processes, policy evaluation (estimating the value of a policy) and policy improvement (making the policy better based on the values), working together to find an optimal policy.

    - **Classical Value-Based Methods (e.g., Policy Iteration):** These are a direct, "hard" implementation of GPI. They fully evaluate the policy (solve for V^π exactly) and then perform a full, greedy policy improvement.
    - **Actor-Critic Policy-Based Methods:** These are a "soft," approximate implementation of GPI. The critic (value network) performs an approximate policy evaluation by learning V^π from samples. The actor (policy network) performs an approximate policy improvement by taking a gradient step in a direction suggested by the critic.

    By viewing both through the lens of GPI, we see they are not fundamentally different paradigms but rather different implementations of the same core idea. Actor-critic methods are essentially a version of policy iteration that uses function approximation and gradient ascent, making it suitable for large, continuous spaces where exact evaluation and improvement are impossible.

25. **Question:** What is the "support" of a distribution, and why is the condition that the support of the behavior policy must cover the support of the target policy critical for importance sampling?

    Answer: The "support" of a probability distribution is the set of outcomes that have a non-zero probability of occurring. For a policy $\pi(a|s)$, the support is the set of all actions a that the agent might take in state s.

    The condition that the support of the behavior policy $q(x)$ must cover the support of the target policy $p(x)$ means that any outcome possible under p must also be possible under q. Mathematically, if $p(x) > 0$, then $q(x)$ must also be > 0.

    This is critical for importance sampling because the importance weight is $w(x) = p(x)/q(x)$. If an event x could happen under the target policy ($p(x) > 0$) but could never happen under the behavior policy ($q(x) = 0$), the importance weight would be undefined (division by zero). This would mean we have no way of estimating the probability of this event because our samples from q will never include it. In the context of policies, if the target policy could take an action that the behavior policy would never take, we can't estimate the value of that action, and the IS estimate would be invalid and biased.

26. **Question:** If PPO's clipped objective is just a heuristic, why has it become more popular than the theoretically-grounded TRPO?

    Answer: PPO's popularity over TRPO, despite its heuristic nature, stems from a powerful combination of factors that favor practical application over theoretical purity:

    (a) **Simplicity and Ease of Implementation:** This is the biggest factor. PPO can be implemented in a few dozen lines of code on top of a standard actor-critic setup using a first-order optimizer like Adam. TRPO requires complex, non-standard components like the conjugate gradient algorithm and a line search, making it much harder to implement, debug, and integrate.

    (b) **Computational Efficiency:** The second-order calculations in TRPO, even with the conjugate gradient trick, can be more computationally expensive per update than PPO's simple first-order update. This allows PPO to often perform more updates in the same amount of wall-clock time.

(c) **Robust Empirical Performance:** In a wide variety of benchmark environments, PPO has been shown to achieve performance that is just as good, and sometimes better, than TRPO. It successfully captures the essence of the trust region constraint, providing excellent stability without the theoretical and computational overhead.

(d) **Scalability:** PPO's simplicity makes it easier to scale to large-scale distributed settings and adapt to different network architectures (e.g., with shared parameters between actor and critic).

Essentially, PPO hit a "sweet spot" by providing about 90% of the benefit of TRPO's stability with only 10% of the implementation complexity, a trade-off that researchers and practitioners overwhelmingly prefer.

27. **Question:** What is the difference between the Total Variation distance and the KL-divergence, and why might one be preferred over the other in different theoretical contexts?

Answer: Both Total Variation (TV) distance and Kullback-Leibler (KL) divergence are measures of difference between two probability distributions, p and q.

- **Total Variation Distance:** $D_{TV}(p,q) = \frac{1}{2}\sum_x |p(x) - q(x)|$. It is a true metric; it is symmetric ($D_{TV}(p,q) = D_{TV}(q,p)$) and satisfies the triangle inequality. It measures the largest possible difference between the probabilities that the two distributions can assign to the same event. It is bounded between 0 and 1.
- **KL-Divergence:** $D_{KL}(p||q) = \sum_x p(x)\log(p(x)/q(x))$. It is not a true metric; it is not symmetric ($D_{KL}(p||q) \neq D_{KL}(q||p)$) and does not satisfy the triangle inequality. It is unbounded. It measures the expected log-probability ratio, representing the information gain when moving from a prior distribution q to a posterior distribution p.

In the theoretical analysis for the performance lower bound, TV distance is often used because it has convenient properties for bounding the difference in state distributions. However, in the practical implementation of TRPO, KL-divergence is preferred. This is because it has a convenient second-order approximation via the Fisher Information Matrix, which is essential for the conjugate gradient optimization step. Furthermore, the asymmetry of KL-divergence can be useful; $D_{KL}(p_{old}||p_{new})$ has a large penalty if p_{new} assigns near-zero probability to an event that was likely under p_{old}, which is a desirable property for preventing policy collapse.

28. **Question:** How does the entire theoretical framework, from the Policy Gradient Theorem to TRPO, justify the use of multiple optimization steps (epochs) on the same batch of data?

Answer: The theoretical framework provides the justification for reusing data, which improves sample efficiency.

- **REINFORCE:** The basic REINFORCE algorithm is strictly on-policy. The gradient is an expectation under the current policy π_θ. Once you take a single gradient step to get $\pi_{\theta'}$, the data you collected is technically from an old policy, and a strict interpretation would require you to throw it away and collect new data. This is extremely sample inefficient.
- **Surrogate Objective (IS):** The introduction of importance sampling creates the surrogate objective $L_{\theta_{old}}(\theta_{new})$. This objective explicitly measures the performance of a new policy θ_{new} using data collected from a fixed old policy θ_{old}.
- **Trust Region Constraint (TRPO/PPO):** The trust region constraint ensures that the surrogate objective remains a valid approximation of true performance.

The combination of the surrogate objective and the trust region constraint is what justifies data reuse. We can take our batch of data collected with $\pi_{\theta_{old}}$ and perform multiple gradient ascent steps to optimize $L_{\theta_{old}}(\theta_{new})$. As long as each update step keeps the resulting policy $\pi_{\theta_{new}}$ within the trust region (i.e., close to $\pi_{\theta_{old}}$), the gradient of the surrogate objective is still a valid direction for improvement. This allows algorithms like PPO and TRPO to extract much more information from a single batch of experience, dramatically improving sample efficiency compared to a single-update method like REINFORCE.

29. **Question:** Imagine an environment where the optimal policy is highly stochastic (e.g., rock-paper-scissors). Why would a policy gradient method be superior to a value-based method like Q-learning in finding this optimal policy?

Answer: In an environment like rock-paper-scissors, the optimal policy is to play each action with a probability of 1/3. This is a fundamentally stochastic policy.

A traditional value-based method like Q-learning struggles here. It learns the Q-values for each action, $Q(s, \text{rock})$, $Q(s, \text{paper})$, $Q(s, \text{scissors})$. If the opponent is also playing optimally, the expected

return for each action will be the same. The Q-values will converge to be equal: $Q(s, \text{rock}) = Q(s, \text{paper}) = Q(s, \text{scissors})$.

The policy in Q-learning is typically derived by taking the greedy action: $\arg\max_a Q(s, a)$. When all Q-values are equal, the arg max is ill-defined. The agent might deterministically choose one action (e.g., the one with the lowest index), or it might break ties randomly, but it isn't directly learning the optimal 1/3, 1/3, 1/3 distribution.

A policy gradient method, however, directly parameterizes the policy $\pi_\theta(a|s)$. The objective is to maximize expected reward. Through trial and error, the gradient updates will naturally push the parameters θ towards a state where the output distribution is uniform (1/3, 1/3, 1/3), as any other distribution could be exploited by the opponent, leading to lower overall reward. Because policy gradient methods can explicitly represent and optimize for stochastic policies, they are superior for problems where the optimal policy itself is stochastic.

30. **Question:** Summarize the key problem that each of the following concepts solves in the progression of policy gradient theory: (1) Log-derivative trick, (2) Advantage function, (3) Importance sampling, (4) Clipped objective.

    Answer:

    (a) **Log-derivative trick:** Solves the **intractability problem**. It reformulates the gradient of the objective function to remove the dependency on the unknown environment dynamics model and turns it into an expectation that can be estimated from samples.

    (b) **Advantage function:** Solves the **variance problem**. By subtracting a state-value baseline from the reward-to-go, it creates a relative, centered learning signal that dramatically reduces the variance of the gradient estimate, leading to more stable learning.

    (c) **Importance sampling:** Solves the **on-policy problem**. It provides a theoretical mechanism to estimate the performance of a new policy using data collected from an old policy, which is the foundation for improving sample efficiency by reusing data.

    (d) **Clipped objective (PPO):** Solves the **stability problem** in a practical way. It's a heuristic that approximates the effect of TRPO's complex trust region constraint, preventing destructively large policy updates and stabilizing the learning process while being simple to implement with first-order methods.