Model-Based Reinforcement Learning: A Comprehensive Question Bank

By Taha Majlesi

July 17, 2025

Contents

1 Multiple Choice Questions (80 Questions)

This section contains 80 multiple-choice questions based on the provided text. Each question has four possible answers. The correct answer is indicated in the answer key at the end of this section.

1. What is the fundamental distinction between model-free and model-based RL?
   (a) The type of reward signal used.
   (b) Whether the agent learns a policy or a value function.
   (c) Whether the agent explicitly learns a model of the environment's dynamics.
   (d) The complexity of the neural network architecture.
2. A model-free agent treats the environment as a:
   (a) "White box" with known rules.
   (b) "Black box" to be learned through trial and error.
   (c) Deterministic system.
   (d) Solved problem.
3. What is the primary goal of Model-Based Reinforcement Learning's (MBRL) first stage?
   (a) To directly optimize the policy parameters.
   (b) To learn a model of the environment's dynamics.
   (c) To execute a random policy for exploration.
   (d) To calculate the final cumulative reward.
4. The analogy of learning to ride a bicycle purely by practice, without studying physics, best describes:
   (a) Model-Based RL (MBRL).
   (b) Model-Free RL (MFRL).
   (c) Trajectory Optimization.
   (d) System Identification.
5. In the context of MBRL, what is a "dynamics model"?
   (a) A function that directly outputs the optimal action.
   (b) An approximation of the environment's dynamics, predicting the next state and reward.
   (c) The agent's final policy.
   (d) A record of all past experiences.
6. In which of the following scenarios would a "known model" most likely be available?
   (a) A robot learning to walk in the real world.
   (b) A self-driving car navigating a new city.

(c) A program learning to play the game of Chess.

(d) An AI learning to interact with a human.

7. What is the primary motivation for using MBRL over MFRL?

   (a) Higher asymptotic performance.

   (b) Lower computational cost during planning.

   (c) Superior sample efficiency.

   (d) Simpler implementation.

8. The ability to generate "hypothetical" or "imaginary" experiences is a direct benefit of:

   (a) Having a learned model of the world.

   (b) Using a model-free approach.

   (c) Q-Learning.

   (d) Having a large replay buffer.

9. What is the most significant challenge in MBRL?

   (a) Inefficient exploration.

   (b) The "reality gap" caused by model inaccuracies.

   (c) The inability to adapt to new tasks.

   (d) High computational cost during learning.

10. The "reality gap" refers to the discrepancy between:

    (a) The agent's policy and the optimal policy.

    (b) The predicted reward and the actual reward.

    (c) The learned model and the true environment.

    (d) The sample efficiency of MBRL and MFRL.

11. Why do MFRL methods often achieve better asymptotic performance than MBRL methods?

    (a) They learn faster.

    (b) They are not constrained by the bias of an imperfect model.

    (c) They require less memory.

    (d) They use more complex neural networks.

12. An imperfect model in MBRL acts as a:

    (a) Performance bottleneck.

    (b) Catalyst for faster learning.

    (c) Way to guarantee convergence.

    (d) Replacement for a reward function.

13. Which of the following is listed as an example of a model-based algorithm?
    (a) Proximal Policy Optimization (PPO).
    (b) Soft Actor-Critic (SAC).
    (c) Q-Learning.
    (d) Dyna-Q.
14. According to the comparative table, which method has higher flexibility to adapt to new goals by re-planning?
    (a) Model-Free RL (MFRL).
    (b) Model-Based RL (MBRL).
    (c) Both are equally flexible.
    (d) Neither is flexible.
15. System Identification is a process of:
    (a) Using general models like RNNs to learn dynamics.
    (b) Fitting the parameters of a pre-defined model structure.
    (c) Identifying the optimal policy directly.
    (d) Identifying which states are most valuable.
16. In a deterministic environment, a given sequence of actions will produce:
    (a) A distribution over possible trajectories.
    (b) Exactly one trajectory of states.
    (c) A random outcome.
    (d) An error unless the model is perfect.
17. In a stochastic environment, why is calculating the expected cumulative reward often intractable?
    (a) The rewards are unknown.
    (b) It requires integrating over all possible future paths.
    (c) The action space is too small.
    (d) The model is always deterministic.
18. What defines an open-loop controller?
    (a) It continuously uses feedback to adjust its actions.
    (b) It determines a fixed sequence of actions at the outset and executes it without feedback.
    (c) It adapts its plan based on the current state at every timestep.
    (d) It uses a learned policy network.
19. The Persian parable of the deaf man visiting his sick friend illustrates the core weakness of:

(a) Closed-loop control.

(b) Model Predictive Control (MPC).

(c) Open-loop control.

(d) Model-free learning.

20. What is the primary advantage of open-loop systems?

(a) High accuracy and robustness to disturbances.

(b) Simplicity, low cost, and fast response time.

(c) Ability to self-correct errors.

(d) Suitability for highly stochastic environments.

21. A thermostat is a classic example of:

(a) An open-loop control system.

(b) A closed-loop (feedback) control system.

(c) A model-free agent.

(d) A random shooting method.

22. What is the core idea behind Model Predictive Control (MPC)?

(a) To generate a single, perfect open-loop plan.

(b) To turn an open-loop planner into a closed-loop controller by repeatedly re-planning.

(c) To eliminate the need for a dynamics model.

(d) To use only past data to select actions.

23. In the MPC cycle, what does the agent do after planning an optimal action sequence?

(a) Executes the entire sequence.

(b) Executes only the first action of the sequence.

(c) Discards the plan and samples a random action.

(d) Waits for the environment to change before acting.

24. The strategy of re-planning from the latest observed state in MPC provides:

(a) A guarantee of finding the globally optimal policy.

(b) Simplicity and lower computational cost.

(c) Robustness against unforeseen events and model errors.

(d) A way to avoid learning a model.

25. According to the comparison table, which control system is more prone to oscillations but ensures long-term accuracy?

(a) Open-Loop Control.

(b) Closed-Loop Control.

(c) Both are equally stable.

(d) Neither can oscillate.

26. The planning problem is abstracted as finding an action sequence A that maximizes an objective function $J(A)$, which represents the:

(a) Model's accuracy.

(b) Expected cumulative reward.

(c) Exploration bonus.

(d) Policy's entropy.

27. Why can't we typically use standard gradient-based optimization methods for the planning objective $J(A)$?

(a) The action space is discrete.

(b) The objective function is always linear.

(c) We do not have access to its gradient, as it's evaluated via simulation.

(d) Gradient-based methods are too slow.

28. The "curse of dimensionality" in planning refers to:

(a) The difficulty of learning from high-dimensional state inputs like pixels.

(b) The vast search space created by long time horizons and high-dimensional actions.

(c) The problem of having too many reward signals.

(d) The instability of optimizers in more than three dimensions.

29. What is the most straightforward derivative-free optimization algorithm for planning?

(a) Cross-Entropy Method (CEM).

(b) Monte Carlo Tree Search (MCTS).

(c) Random Shooting.

(d) Gradient Ascent.

30. What is the primary limitation of the random shooting method?

(a) It is difficult to parallelize.

(b) It requires a differentiable model.

(c) It is grossly inefficient and explores the search space blindly.

(d) It can only be used in deterministic environments.

31. How does the Cross-Entropy Method (CEM) improve upon random shooting?

(a) It uses gradients to guide the search.

(b) It iteratively refines a sampling distribution to focus on high-reward regions.

(c) It builds a search tree to evaluate actions sequentially.

(d) It uses a perfect model of the environment.

32. In CEM, the "elites" are:

(a) The actions with the lowest reward.

(b) A subset of randomly chosen action sequences.

(c) The top-performing fraction of the sampled action sequences.

(d) The parameters of the initial sampling distribution.

33. What is the core "learning step" in the CEM algorithm?

(a) Sampling N candidate action sequences.

(b) Evaluating the reward for each sequence.

(c) Updating the parameters of the sampling distribution to fit the elite set.

(d) Selecting the single best action sequence.

34. In CEM, the sampling distribution over action sequences is typically a:

(a) Uniform distribution.

(b) Multivariate Gaussian distribution.

(c) Bernoulli distribution.

(d) Poisson distribution.

35. The "refit" step in CEM is mathematically equivalent to:

(a) Maximum Likelihood Estimation (MLE).

(b) Gradient Descent.

(c) Value Iteration.

(d) Policy Iteration.

36. For a Gaussian sampling distribution in CEM, the new mean is updated to be the:

(a) Mean of all sampled sequences.

(b) Mean of the elite action sequences.

(c) Mean from the previous iteration.

(d) A zero vector.

37. What is a key advantage of CEM mentioned in the text?

(a) It is guaranteed to find the global optimum.

(b) It is not sensitive to hyperparameters.

(c) The evaluation of candidate trajectories is highly parallelizable.

(d) It is a closed-loop planner by nature.

38. What is a listed "con" or weakness of CEM?
    (a) It is difficult to implement.
    (b) It cannot be used for continuous control problems.
    (c) It can struggle in very high-dimensional search spaces.
    (d) It is less efficient than random shooting.
39. To be used for robust control in a stochastic environment, CEM must be:
    (a) Run for a single iteration.
    (b) Combined with a model-free algorithm.
    (c) Wrapped in an MPC scheme.
    (d) Used with a discrete action space.
40. The CEM algorithm iteratively minimizes the cross-entropy between its sampling distribution and an ideal distribution that would:
    (a) Generate only low-reward samples.
    (b) Generate only high-reward samples.
    (c) Have maximum variance.
    (d) Be uniform over the action space.
41. How does MCTS construct its search tree?
    (a) Symmetrically, exploring all branches equally.
    (b) Asymmetrically, focusing on the most promising sequences of actions.
    (c) By only expanding the single best node at each level.
    (d) Randomly, without using results from past simulations.
42. What are the four steps of an MCTS iteration, in order?
    (a) Selection, Expansion, Simulation, Backpropagation.
    (b) Expansion, Selection, Simulation, Backpropagation.
    (c) Simulation, Backpropagation, Selection, Expansion.
    (d) Selection, Simulation, Expansion, Backpropagation.
43. In MCTS, the "Simulation" or "Rollout" step is used to:
    (a) Select the best child node to traverse.
    (b) Expand the tree with new nodes.
    (c) Estimate the value of a newly created node.
    (d) Update the statistics of parent nodes.
44. The UCB1 formula is used in which step of MCTS?

(a) Selection.

(b) Expansion.

(c) Simulation.

(d) Backpropagation.

45. The UCB1 formula is designed to balance:

(a) Sample efficiency and asymptotic performance.

(b) Model accuracy and planning cost.

(c) Exploration and exploitation.

(d) Open-loop and closed-loop control.

46. In the UCB1 formula, the term $\frac{V_i}{n_i}$ represents:

(a) The exploration bonus.

(b) The exploitation term (average value of the node).

(c) The visit count of the parent node.

(d) The exploration constant C.

47. The exploration bonus in the UCB1 formula is large for nodes that have been:

(a) Visited frequently.

(b) Proven to have high value.

(c) Visited infrequently relative to their parent.

(d) Identified as terminal states.

48. How did algorithms like AlphaZero and MuZero enhance classic MCTS?

(a) By removing the selection step.

(b) By using random rollouts exclusively.

(c) By integrating learned policy and value networks.

(d) By applying it to continuous action spaces.

49. In the AlphaZero approach, what is used to provide a more informed estimate of a leaf node's quality, replacing random rollouts?

(a) A learned value network.

(b) A learned policy network.

(c) The UCB1 formula.

(d) A deeper, random rollout.

50. MCTS is particularly well-suited for problems with:

(a) Continuous action spaces.

(b) No clear reward signal.

(c) Discrete action spaces and a deep, sequential structure.

(d) A very short time horizon.

51. What does it mean for MCTS to be an "anytime" algorithm?

(a) It can be used at any time of day.

(b) It can be stopped at any point to return the current best action.

(c) It takes the same amount of time for every decision.

(d) It can only be used for real-time applications.

52. A key limitation of standard MCTS is:

(a) It cannot manage the exploration-exploitation trade-off.

(b) It is not parallelizable.

(c) It is difficult to apply to continuous or very large action spaces.

(d) It is less effective than random search in discrete domains.

53. Trajectory Optimization (TO) is a technique for finding an:

(a) Open-loop solution to an optimal control problem.

(b) Closed-loop policy directly.

(c) Model-free value function.

(d) Estimate of the environment's dynamics.

54. Which of the following is NOT a key component of a TO problem?

(a) Cost Function.

(b) Dynamics Constraints.

(c) A learned policy network.

(d) Boundary Conditions.

55. Minimizing $\int u(t)^2 dt$ in a TO problem is an objective used to encourage:

(a) Maximum speed.

(b) Minimum travel time.

(c) Smoothness of control inputs.

(d) Obstacle avoidance.

56. How do "Direct Methods" in TO solve the optimal control problem?

(a) By deriving the first-order necessary conditions for optimality.

(b) By transcribing it into a finite-dimensional nonlinear programming (NLP) problem.

(c) By using Monte Carlo Tree Search.

(d) By learning a value function.

57. "Shooting Methods" and "Direct Collocation" are two types of:

(a) Indirect Methods.

(b) Direct Methods.

(c) Model-Free Methods.

(d) Tree Search Methods.

58. What is the primary difficulty of using "Indirect Methods" in TO?

(a) They are computationally very slow.

(b) They produce inaccurate solutions.

(c) They are difficult to use due to non-intuitive co-states and a small region of convergence.

(d) They cannot handle constraints.

59. Generating dynamic walking and running gaits for humanoid robots is a key application of:

(a) Random Shooting.

(b) Q-Learning.

(c) Trajectory Optimization.

(d) Model-Free RL exclusively.

60. Indirect methods use Pontryagin's Minimum Principle to transform the optimization problem into a:

(a) Nonlinear programming problem.

(b) Two-point boundary value problem (TPBVP).

(c) Monte Carlo simulation.

(d) Simple regression problem.

61. How is the Cross-Entropy Method (CEM) described in the conceptual map?

(a) A type of indirect method.

(b) An advanced, iterative shooting method.

(c) A heuristic tree search algorithm.

(d) The overarching mathematical framework.

62. What is the standard and essential technique for deploying open-loop planners like CEM and MCTS in dynamic environments?

(a) Running them for more iterations.

(b) Using a larger batch size.

(c) Model Predictive Control (MPC).

(d) Using a simpler model.

63. For a problem with a discrete action space like a board game, which planner is often the superior choice?
    (a) Cross-Entropy Method (CEM).
    (b) Monte Carlo Tree Search (MCTS).
    (c) Direct Collocation.
    (d) Random Shooting.
64. For a problem with a continuous action space like robot joint torques, which planner is more naturally applicable?
    (a) Monte Carlo Tree Search (MCTS).
    (b) Cross-Entropy Method (CEM).
    (c) A basic UCB1 algorithm.
    (d) A discrete state-action planner.
65. Which of the following is NOT listed as a future direction in the text?
    (a) Differentiable Trajectory Optimization.
    (b) Uncertainty-Aware Planning.
    (c) A return to simpler, model-free methods.
    (d) Hybrid Approaches combining different methods.
66. What is the goal of "Differentiable Trajectory Optimization"?
    (a) To make planners faster by removing differentiation.
    (b) To learn model or cost function parameters by backpropagating task loss through the planner.
    (c) To prove that all planners are differentiable.
    (d) To apply planners only to differentiable environments.
67. Uncertainty-aware planning methods explicitly account for:
    (a) Aleatoric uncertainty (inherent randomness).
    (b) Epistemic uncertainty (model uncertainty).
    (c) Uncertainty in the reward function.
    (d) Uncertainty in the action space.
68. What is the relationship between Trajectory Optimization (TO) and Shooting Methods?
    (a) TO is a specific type of shooting method.
    (b) Shooting methods are a specific type of direct method within the TO framework.
    (c) They are unrelated concepts.
    (d) TO is used to initialize shooting methods.

69. The text states that MCTS's performance relies on having access to a:
    (a) Perfect simulator or model.
    (b) Differentiable model.
    (c) Continuous action space.
    (d) Very large batch of samples.
70. What is the key difference between single shooting and multiple shooting?
    (a) Single shooting is more robust.
    (b) Multiple shooting breaks the trajectory into segments to improve numerical stability.
    (c) Single shooting optimizes states, while multiple shooting optimizes controls.
    (d) Multiple shooting is an indirect method.
71. The "objective mismatch" in MBRL refers to the problem where:
    (a) The model is trained to be accurate, but this doesn't guarantee good task performance.
    (b) The agent's objective changes over time.
    (c) The reward function does not reflect the true goal.
    (d) The planner and the policy have different objectives.
72. What is the purpose of the "backpropagation" step in MCTS?
    (a) To calculate gradients for a neural network.
    (b) To propagate the result of a simulation back up the tree to update node statistics.
    (c) To expand the tree with new actions.
    (d) To select the most promising branch to explore.
73. A key weakness of MCTS is its difficulty in handling continuous action spaces. Why?
    (a) The UCB1 formula only works for discrete values.
    (b) Expanding a node would require creating an infinite number of children.
    (c) Continuous actions make the rollouts too slow.
    (d) The backpropagation step cannot handle continuous values.
74. Which method is described as a "guess and check" approach?
    (a) MCTS.
    (b) CEM.
    (c) Random Shooting.
    (d) Trajectory Optimization.
75. The final plan in CEM can be taken as:
    (a) The single best sample from the first iteration.

(b) The mean of the final sampling distribution.

(c) A random sample from the initial distribution.

(d) The action with the highest variance.

76. What is a primary reason for using a diagonal covariance matrix in CEM?

(a) To increase the complexity of the model.

(b) To ensure the model is always accurate.

(c) For simplicity and numerical stability.

(d) To guarantee faster convergence.

77. The "receding horizon" strategy is another name for:

(a) MCTS.

(b) CEM.

(c) Open-loop control.

(d) MPC.

78. Which of the following is an application of open-loop control?

(a) Automotive cruise control.

(b) A timed washing machine cycle.

(c) An air conditioning system.

(d) Industrial robotics.

79. The policy in a closed-loop system is state-dependent, represented as:

(a) $a_1, \ldots, a_T$.

(b) $a_t \sim \pi(a_t|s_t)$.

(c) $J(A)$.

(d) $s_{t+1} = f(s_t, a_t)$.

1.1 Answer Key for Multiple Choice Questions

1. (c)	8. (a)	15. (b)	22. (b)	29. (c)
2. (b)	9. (b)	16. (b)	23. (b)	30. (c)
3. (b)	10. (c)	17. (b)	24. (c)	31. (b)
4. (b)	11. (b)	18. (b)	25. (b)	32. (c)
5. (b)	12. (a)	19. (c)	26. (b)	33. (c)
6. (c)	13. (d)	20. (b)	27. (c)	34. (b)
7. (c)	14. (b)	21. (b)	28. (b)	35. (a)

36. (b)
37. (c)
38. (c)
39. (c)
40. (b)
41. (b)
42. (a)
43. (c)
44. (a)
45. (c)
46. (b)
47. (c)
48. (c)
49. (a)
50. (c)
51. (b)
52. (c)
53. (a)
54. (c)
55. (c)
56. (b)
57. (b)
58. (c)
59. (c)
60. (b)
61. (b)
62. (c)
63. (b)
64. (c)
65. (b)
66. (a)
67. (b)
68. (a)
69. (b)
70. (b)
71. (c)
72. (b)
73. (c)
74. (d)
75. (b)
76. (b)

2 Explanatory Questions and Answers (30 Questions)

This section provides 30 questions that require detailed, explanatory answers. A complete answer is provided for each question based on the source text.

1. Compare and contrast Model-Based RL (MBRL) and Model-Free RL (MFRL) in terms of sample efficiency and asymptotic performance.

Answer: MBRL generally has much higher sample efficiency than MFRL. This is because MBRL learns a model of the environment, which can then be used to generate a vast number of simulated or "imaginary" experiences for planning. This reduces the need for costly and time-consuming interactions with the real world. However, MFRL often achieves better asymptotic (final) performance. The reason is that the learned model in MBRL is almost never perfect. This "model bias" acts as a performance bottleneck, as the agent finds a policy that is optimal for its flawed model, not the true environment. In contrast, an MFRL agent learns directly from the true environment, and while it may take much longer, it is not constrained by model bias and can potentially converge to the true optimal policy.

2. Explain the concept of the "reality gap" in MBRL and describe its consequences.

Answer: The "reality gap" is the discrepancy between the agent's learned model of the world and the true, underlying dynamics of the real environment. Since the model is learned from finite data, it is almost always an imperfect approximation. The primary consequence of this gap is that even minor inaccuracies in the model can accumulate and compound over long prediction horizons. This can lead to catastrophic failures where the planner discovers and exploits a flaw in the model, devising a strategy that appears highly successful in simulation but is disastrous when executed in the real world. This model-induced bias places a ceiling on the agent's ultimate performance.

3. Describe the two-stage process of Model-Based Reinforcement Learning using the example of a robot learning to throw a basketball.

Answer: The two-stage process of MBRL is (1) learn a model, and (2) use the model to plan. In the basketball example, a model-based robot would first perform a few throws to collect data. In the first stage, it uses this data to learn a model of physics—how factors like throwing force, angle, and gravity affect the ball's trajectory. This learned model encapsulates the environment's dynamics. In the second stage, the robot uses this internal physics model to plan. It can simulate thousands of different throws internally without actually performing them, calculating the optimal throwing force and angle that would maximize the probability of the ball going into the hoop. This is far more sample-efficient than a model-free approach, which would require thousands of real throws to learn by trial and error.

4. What is the critical distinction between open-loop and closed-loop control? Use the Persian parable to illustrate the weakness of one approach.

Answer: The critical distinction is the use of feedback. An open-loop controller determines a complete, fixed sequence of actions at the outset and executes it without any subsequent feedback from the environment. A closed-loop controller, in contrast, continuously uses feedback by observing the current state of the environment to make real-time adjustments to its actions.

The Persian parable powerfully illustrates the weakness of open-loop control. A deaf man pre-plans an entire conversation with his sick friend, assuming a specific sequence of responses. When he asks, "How are you?" his friend, near death, replies, "I am dying!" Sticking to his rigid, open-loop plan, the deaf man cheerfully responds, "Thank God!" This highlights the core weakness: open-loop control is brittle and can lead to absurd or catastrophic outcomes because it cannot adapt when the world deviates from the expected sequence of events.

5. Explain the process of Model Predictive Control (MPC) and why it is essential for making open-loop planners robust.

Answer: Model Predictive Control (MPC), or Receding Horizon Control, is a strategy that transforms an open-loop planner into a robust, closed-loop controller. It works through a simple, iterative process:

(a) From the current state s_t, use the open-loop planner to find an optimal action sequence over a finite horizon.

(b) Execute only the *first* action of that sequence in the real environment.

(c) Observe the resulting new state, s_{t+1}.

(d) Discard the rest of the planned sequence and repeat the entire process from the new state.

MPC is essential because open-loop planners (like Random Shooting or CEM) produce a fixed plan that is optimal only if the model is perfect and the environment is deterministic. In reality, model errors and environmental stochasticity will cause the system to deviate from the planned trajectory. By constantly re-planning from the latest observed state, MPC allows the agent to continuously correct for these deviations, making it reactive and robust to unforeseen events. It effectively gives the agent the benefits of closed-loop feedback control while leveraging the power of open-loop planning algorithms.

6. What is the "curse of dimensionality" in the context of planning, and why does it make the Random Shooting method inefficient?

Answer: In the context of planning, the "curse of dimensionality" refers to the exponential growth of the search space as the planning horizon and the dimensionality of the action vector increase. For instance, if an agent plans over a horizon of $T = 50$ timesteps with a 10-dimensional continuous action vector at each step, the total decision variable (the action sequence) is a 500-dimensional vector.

This makes the Random Shooting method highly inefficient. Random Shooting is a "guess and check" approach that samples candidate action sequences from a fixed distribution. In a low-dimensional space, random guessing might eventually find a decent solution. However, in a high-dimensional space, the volume of the space is so vast that the probability of randomly stumbling upon a high-reward region is infinitesimally small. The method explores the space blindly, and its inefficiency grows exponentially with the dimension of the problem.

7. Describe the core algorithm of the Cross-Entropy Method (CEM) for planning.

Answer: The Cross-Entropy Method (CEM) is an iterative, population-based optimization algorithm for planning. It proceeds as follows:

(a) **Initialize:** Define a parameterized probability distribution over action sequences, typically a multivariate Gaussian with a mean μ and covariance Σ.

(b) **Sample:** Draw a population of N candidate action sequences from this distribution.

(c) **Evaluate:** For each sequence, use the dynamics model to simulate the outcome and calculate its total reward.

(d) **Select Elites:** Sort the sequences by their rewards and select the top-performing fraction (e.g., the top 10%) to form the "elite set."

(e) **Refit:** Update the parameters of the sampling distribution (μ and Σ) to best fit the elite set. This is done via Maximum Likelihood Estimation, where the new mean becomes the sample mean of the elites, and the new covariance becomes their sample covariance.

(f) **Iterate:** Repeat the Sample-Evaluate-Select-Refit loop for a fixed number of iterations or until the distribution converges.

This process iteratively shifts the sampling distribution towards regions of the search space that produce higher rewards.

8. Explain the mathematical update rules for the mean and covariance in CEM when using a Gaussian distribution.

Answer: In CEM, the "refit" step involves finding the new parameters $(\mu_{\text{new}}, \Sigma_{\text{new}})$ of the Gaussian distribution that maximize the likelihood of observing the elite set of action sequences $\{A_1, \ldots, A_M\}$. This Maximum Likelihood Estimation (MLE) has a closed-form solution.

Mean Update: The new mean, $\hat{\mu}_{\text{new}}$, is simply the sample mean (the average) of the M elite action sequences. This intuitively pulls the center of the search distribution towards the average of the best solutions found in the current iteration.

$$\hat{\mu}_{\text{new}} = \frac{1}{M} \sum_{j=1}^{M} A_j$$

Covariance Update: The new covariance, $\hat{\Sigma}_{\text{new}}$, is the sample covariance of the elite action sequences around their new mean. This adjusts the spread and orientation of the search distribution to match the spread of the best solutions.

$$\hat{\Sigma}_{\text{new}} = \frac{1}{M} \sum_{j=1}^{M} (A_j - \hat{\mu}_{\text{new}})(A_j - \hat{\mu}_{\text{new}})^T$$

9. Describe the four fundamental steps of a single iteration in the Monte Carlo Tree Search (MCTS) algorithm.

Answer: A single iteration of MCTS consists of a four-step loop:

(a) **Selection:** Starting at the root node (the current state), the algorithm traverses down the existing tree. At each node, it uses a tree policy (like UCB1) to select a child node that optimally balances exploiting known good moves and exploring less-certain moves. This continues until a leaf node (a node that has not yet been expanded) is reached.

(b) **Expansion:** If the selected leaf node is not a terminal state, the tree is expanded by creating one or more new child nodes corresponding to valid, unexplored actions from that state.

(c) **Simulation (Rollout):** From one of these newly created child nodes, a simulation is run to estimate its value. In classic MCTS, this is done by following a simple, fast policy (e.g., choosing actions randomly) until a terminal state is reached. The outcome of this simulation (e.g., win/loss) is the value estimate.

(d) **Backpropagation (Backup):** The result of the simulation is propagated back up the tree along the path traversed during the selection phase. The statistics of every node on this path are updated: its visit count (n) is incremented, and its total value (V) is updated with the simulation result.

10. Explain the two components of the UCB1 formula and how they manage the exploration-exploitation trade-off in MCTS.

Answer: The UCB1 formula, used in the selection step of MCTS, is:

$$\text{UCB1}(\text{node}_i) = \underbrace{\frac{V_i}{n_i}}_{\text{Exploitation}} + \underbrace{C\sqrt{\frac{\ln N}{n_i}}}_{\text{Exploration}}$$

The two components manage the trade-off as follows:

(a) **Exploitation Term ($\frac{V_i}{n_i}$):** This is the average value of the node, calculated as the total value (V_i) from all simulations that passed through it, divided by its visit count (n_i). This term favors selecting nodes that have historically performed well, thereby exploiting known good moves.

(b) **Exploration Term ($C\sqrt{\frac{\ln N}{n_i}}$):** This term encourages exploration. It is large for nodes that have been visited infrequently (small n_i) relative to their parent's visit count (N). This bonus pushes the algorithm to try less-certain moves to see if they might be even better than the current best options. The logarithm ensures that the bonus grows slowly, so that eventually, choices are driven more by the exploitation term.

The constant C tunes the balance between these two competing objectives.

11. How do modern algorithms like AlphaZero integrate learned neural networks into the MCTS framework?

Answer: Algorithms like AlphaZero and MuZero enhance classic MCTS by replacing its less-informed components with powerful, learned neural networks, creating a symbiotic cycle of planning and learning.

(a) **Learned Value Network ($v_\theta(s)$):** Instead of performing a high-variance random rollout from a new leaf node, the algorithm uses a value network to evaluate the state at that node. This provides a much more stable and informed estimate of the node's quality.

(b) **Learned Policy Network ($\pi_\theta(a|s)$):** The selection step is no longer guided solely by UCB1 statistics. A policy network provides prior probabilities for which moves are likely

to be promising from a given state. This prior is incorporated into the selection formula (e.g., PUCT), guiding the search towards more plausible moves from the very beginning.

The MCTS search then acts as a powerful policy improvement operator. The improved action distribution (from the root node's visit counts) and value estimate from the search are used as training targets to update the networks, which in turn enables an even stronger search in the next iteration.

12. What is Trajectory Optimization (TO) and what are its four key components?

Answer: Trajectory Optimization (TO) is a formal discipline from optimal control theory. It is the process of computing a time-indexed history of states, $x(t)$, and control inputs, $u(t)$, that minimizes (or maximizes) a performance objective while satisfying a set of constraints. It is fundamentally a technique for finding an open-loop solution to an optimal control problem.

The four key components that define a TO problem are:

(a) **Cost Function (Objective):** A functional that quantifies the performance of a trajectory, such as minimizing energy consumption, travel time, or control effort.

(b) **Dynamics Constraints:** The differential equations that govern the system's motion (e.g., $\dot{x} = f(x, u)$), which must be satisfied at all times.

(c) **Path Constraints:** Inequalities that must hold throughout the trajectory, such as joint limits on a robot or obstacle avoidance constraints.

(d) **Boundary Conditions:** Equality constraints on the state at the beginning and end of the trajectory, such as starting and ending at rest at specific positions.

13. Briefly differentiate between Direct and Indirect methods in Trajectory Optimization.

Answer: Direct and Indirect methods are two major families of algorithms for solving Trajectory Optimization problems.

Direct Methods work by transcribing the infinite-dimensional optimal control problem into a finite-dimensional nonlinear programming (NLP) problem. They do this by discretizing the trajectory in time (creating "knot points") and then using a numerical optimizer to solve for the state and control variables at these points. Shooting methods and direct collocation are common examples. They are generally more versatile and easier to use than indirect methods.

Indirect Methods take a different approach. They use the calculus of variations (specifically, Pontryagin's Minimum Principle) to derive the first-order necessary conditions for optimality. This transforms the optimization problem into a two-point boundary value problem (TPBVP). While these methods can be extremely fast and accurate, they are notoriously difficult to use in practice because they require a good initial guess for non-intuitive "co-state" variables and have a small region of convergence.

14. How do the planning algorithms discussed (Random Shooting, CEM, MCTS) fit into the broader framework of Trajectory Optimization?

Answer: These algorithms can be understood as points on a spectrum within the broader field of optimal control and Trajectory Optimization (TO).

- **Trajectory Optimization (TO)** is the overarching formal framework for finding an optimal state-control history.
- **Direct Methods** are a class of TO algorithms that discretize the problem.
- **Shooting Methods** are a specific type of direct method where control inputs are the decision variables.
- **Random Shooting** is the most basic, non-iterative shooting method.
- **Cross-Entropy Method (CEM)** is an advanced, iterative shooting method that uses elite sampling to intelligently guide the search for the optimal control sequence.
- **Monte Carlo Tree Search (MCTS)** is an alternative planning approach, distinct from shooting methods. It can be seen as a heuristic tree-search method for solving the optimal control problems defined by TO, particularly in discrete or sequential domains.

15. When choosing a planner, what are the key problem characteristics to consider, and which planner (CEM or MCTS) is better suited for different action spaces?

Answer: The key problem characteristics to consider when choosing a planner include the action space, problem structure, stochasticity, computational budget, and availability of gradient information.

The most critical distinction for choosing between CEM and MCTS is the **action space**:

- **MCTS** is the superior choice for problems with **discrete action spaces** (e.g., board games like Go, grid worlds). Its tree-based expansion works naturally by creating a child node for each discrete action. Applying it to continuous spaces is challenging as it would require creating an infinite number of children.
- **CEM** is more naturally applicable to problems with **continuous action spaces** (e.g., robot joint torques, vehicle steering angles). It operates by sampling from and fitting a continuous probability distribution (like a Gaussian) over the entire high-dimensional action sequence, which is a natural fit for continuous control.

16. What is meant by an "anytime" algorithm, and which of the discussed planners has this property?

Answer: An "anytime" algorithm is an algorithm that can be interrupted at any point during its computation and still provide a valid, albeit potentially suboptimal, solution. The quality of the solution generally improves as the algorithm is allowed to run for longer.

Of the planners discussed, **Monte Carlo Tree Search (MCTS)** is a prime example of an anytime algorithm. The MCTS loop iteratively refines the value estimates and visit counts in its search tree. If it is stopped prematurely, one can still select the best action based on the current statistics (e.g., the most visited child of the root node). The longer it runs, the more simulations are performed, and the more reliable this choice becomes. This is highly valuable in time-constrained applications.

17. Why is high sample efficiency a paramount advantage in fields like robotics?

Answer: High sample efficiency is paramount in robotics because real-world trials (samples) are often extremely costly, time-consuming, and potentially dangerous. Unlike in a simulation where millions of trials can be run for free, each real-world trial on a physical robot consumes energy, causes wear and tear on mechanical parts, takes a significant amount of time to execute, and poses a risk of damage to the robot or its environment. For example, a robot learning to walk could fall and break. By using a model-based approach with high sample efficiency, the robot can learn a model from a small number of safe, real-world interactions and then use that model to simulate thousands of "imaginary" trials internally to find a good policy, drastically reducing the cost and risk associated with learning.

18. Explain the "objective mismatch" problem and how differentiable trajectory optimization aims to solve it.

Answer: The "objective mismatch" problem in MBRL arises from the two-stage learning process. The model is typically trained to minimize a model-learning objective, such as the mean squared error between predicted and actual next states. The planner then uses this model to find a plan that maximizes a task-reward objective. The mismatch is that a model that is highly accurate on the learning objective (e.g., low prediction error on average) is not guaranteed to produce good performance on the actual task. The planner might exploit a region where the model is inaccurate but predicts high rewards.

Differentiable Trajectory Optimization aims to solve this by creating an end-to-end learning pipeline. If the entire planning process is differentiable, one can backpropagate the final task loss (the true objective) all the way through the planner and back to the parameters of the dynamics model. This directly optimizes the model's parameters to make it better for planning the specific task, thus resolving the mismatch by aligning the model learning objective with the task performance objective.

19. What is the difference between a deterministic and a stochastic dynamics model?

Answer: The difference lies in the predictability of the outcome for a given state-action pair.

- A **deterministic dynamics model** assumes that for a given state s and action a, there is only one possible next state s'. The model is a simple function, $s' = f(s, a)$. There is no randomness in the transition.
- A **stochastic dynamics model** assumes that for a given state s and action a, there is a probability distribution over possible next states. The model takes the form $s' \sim p(s'|s, a)$. The outcome is not fixed, and the same action in the same state can lead to different results due to inherent randomness in the environment.

20. Why is CEM considered an "embarrassingly parallel" task?

Answer: CEM is considered an "embarrassingly parallel" task because its most computationally expensive step—the evaluation of candidate action sequences—can be split into completely independent sub-tasks with no need for communication between them. In the "Evaluate" step, the algorithm must simulate the outcome for N different action sequences. Each of these simulations is independent of all the others. Therefore, one can easily distribute these N simulations across multiple CPU cores or even multiple machines. Each core/machine runs its simulation,

computes the total reward, and sends the result back. This allows the algorithm to scale almost linearly with the number of available processors, making it very fast on modern hardware.

21. What is the role of the exploration constant 'C' in the UCB1 formula?

Answer: The exploration constant 'C' in the UCB1 formula, $V_i/n_i + C\sqrt{\ln N/n_i}$, is a hyperparameter that tunes the balance between exploitation and exploration. A higher value of 'C' increases the weight of the exploration term, causing the MCTS algorithm to be more optimistic about less-explored nodes and thus explore more widely. A lower value of 'C' makes the algorithm more conservative, causing it to focus more on the exploitation term and prefer nodes that have already shown high average rewards. The optimal value of 'C' is problem-dependent and often needs to be tuned empirically to achieve the best performance.

22. What is the fundamental difference in approach between CEM and MCTS for planning?

Answer: The fundamental difference is how they explore the search space of action sequences. CEM is a "trajectory-at-a-time" method. It samples and evaluates entire action sequences, $A = (a_1, \ldots, a_T)$, at once. It then refines its sampling distribution based on which complete trajectories were successful. It does not consider the sequential nature of the problem during its search; it treats the entire sequence as a single point in a high-dimensional space.

MCTS, in contrast, is a "step-at-a-time" method. It incrementally builds a search tree, making decisions sequentially. It uses rollouts to estimate the value of taking an action from a state and uses this information to intelligently focus its search deeper into promising branches of the tree. This makes it particularly effective for problems where early decisions have profound long-term consequences.

23. What are "path constraints" in Trajectory Optimization, and can you provide an example?

Answer: In Trajectory Optimization, "path constraints" are inequalities that must be satisfied at every point throughout the trajectory, not just at the start or end points. They define a "safe" or "valid" region of operation for the system. For example, when planning a path for a robot arm, a path constraint could be that the angle of each joint must remain within its physical limits (e.g., $-\pi/2 \leq \theta_1 \leq \pi/2$). Another common example is an obstacle avoidance constraint, which would require that the distance between any part of the robot and any known obstacle must always be greater than zero.

24. Why might a planner using a learned model discover a strategy that is successful in simulation but disastrous in reality?

Answer: This happens because the planner's objective is to find a sequence of actions that maximizes reward *according to the learned model*. If the learned model has inaccuracies or flaws (which it always does), the planner can inadvertently discover and exploit these flaws. For example, the model might have learned that passing through a wall is possible in a certain way or that a specific action yields an impossibly high reward due to a modeling error. The planner, being an optimization algorithm, will find a plan that takes advantage of this "loophole" because it appears to be a highly successful strategy in the simulated world of the model. When this plan is executed in the real environment where the laws of physics are unforgiving, the strategy fails, potentially catastrophically.

25. What is the "backpropagation" or "backup" step in MCTS and what information is updated?

Answer: The backpropagation or backup step is the final step in an MCTS iteration. After a simulation (rollout) is completed from a newly expanded node, its outcome (e.g., +1 for a win, -1 for a loss) is propagated back up the tree along the exact path of nodes that was traversed during the initial "selection" step. For every node on this return path, two key statistics are updated:

(a) **Visit Count (n):** The visit count for the node is incremented by one. This records that one more simulation has passed through this node.

(b) **Total Value (V):** The outcome of the simulation is added to the node's total value. This accumulates the results of all simulations that have involved this node.

This updated information is then used in the UCB1 calculation for future selection steps, making the search more informed over time.

26. What is the conceptual link between CEM and importance sampling?

Answer: The conceptual link is that CEM can be viewed as an adaptive procedure for finding a good proposal distribution for importance sampling, specifically in the context of rare-event simulation. The optimization problem of finding a high-reward action sequence can be reframed as estimating the probability of a "rare event" – the event that a randomly sampled action sequence A has a reward $J(A)$ exceeding some high threshold. Importance sampling is a technique to estimate such probabilities efficiently by sampling from a different proposal distribution where the event is more common. CEM provides a principled, iterative method for finding such a proposal distribution. It iteratively updates its sampling distribution to minimize the cross-entropy (or KL divergence) between it and an ideal (but unknown) distribution that would only generate these high-reward, "rare" samples.

27. Explain the trade-off between complexity/cost and accuracy/robustness in open-loop vs. closed-loop control.

Answer: The trade-off is fundamental to control systems design.

- **Open-Loop Control** is simple and inexpensive. It doesn't require sensors or complex controllers to process feedback, leading to a lower component count and cost. However, this simplicity comes at the cost of accuracy and robustness. Since it cannot monitor its output or correct for errors, it is highly susceptible to external disturbances and cannot guarantee precision.
- **Closed-Loop Control** is highly accurate and robust. It uses feedback from sensors to continuously compare the actual output with the desired output and make real-time adjustments. This allows it to compensate for disturbances and maintain high precision. However, this performance comes at the cost of increased complexity and expense, as it requires sensors, controllers, and feedback mechanisms. It can also be prone to instability or oscillations if not designed carefully.

28. What is a "shooting method" in Trajectory Optimization?

Answer: A shooting method is a type of direct method for solving a trajectory optimization

problem. In this approach, the control inputs over time are the primary decision variables for the optimizer. The method works by "shooting" out a trajectory from an initial state by integrating the system dynamics forward in time using a given sequence of control inputs. The optimizer's job is to find the control sequence that results in a trajectory that satisfies all constraints (like boundary conditions) and minimizes the cost function. The system dynamics are treated as a "black box" that takes a control sequence and produces a trajectory outcome. Random Shooting and CEM are examples of shooting methods.

29. Why is MCTS considered particularly well-suited for problems with a deep, sequential structure?

Answer: MCTS is well-suited for problems with a deep, sequential structure (like board games) because its search mechanism explicitly models this structure. It builds a game tree where nodes represent states and edges represent actions. This allows it to reason about long-term consequences. By performing deep rollouts and backpropagating the results, it can assign credit for a distant future outcome (like winning a game) to the early decisions that led to it. Its asymmetric tree growth, guided by the UCB1 policy, allows it to selectively search much deeper along relevant lines of play, ignoring unpromising branches. This is far more effective than methods like CEM that evaluate an entire sequence at once without explicitly reasoning about the value of intermediate states.

30. What is the goal of "Uncertainty-Aware Planning" and why is it important?

Answer: The goal of Uncertainty-Aware Planning is to explicitly account for the agent's uncertainty about its learned model during the planning process. Standard planners typically assume the learned model is a perfect representation of reality, which can lead to two problems: the agent might be overly cautious and fail to explore, or it might be overconfident in a flawed part of its model and make catastrophic mistakes.

Uncertainty-aware planners, by tracking model uncertainty (epistemic uncertainty), can make more intelligent decisions. They can actively plan to take actions that reduce uncertainty in important parts of the state space, leading to safer and more efficient exploration. For example, an uncertainty-aware planner might choose an action that is slightly suboptimal according to its current model but has the added benefit of providing valuable information that will improve the model for future decisions. This is crucial for safe and robust operation in unknown environments.