Chapter 5
Performance Evaluations

Abstract In this chapter, we discuss performance evaluation metrics and methods in details. Performance evaluation of reinforcement learning systems or algorithms are important to system monitoring and development improvements. It can be carried out in both offline and online. Offline evaluation evaluates the agent's performance on one or multiple fixed datasets of experiences. Online performance evaluation assesses the agent's performance in real-time interaction with the environment. Benchmarking is a common process to compare the performance of a newly developed reinforcement learning algorithm or reinforcement learning system to that of the established ones. It's generally carried out offline but some literatures also refer A/B testing as benchmarking in the sense that the method to be compared with is the reference one. Performance evaluation through simulator and emulator is another way to assess the reinforcement learning performance, and the effectiveness of this method itself heavily depends on the quality of the simulator or emulator used.

5.1 Evaluation Metrics

The key metrics for evaluating the performance of reinforcement learning (RL) algorithms encompass several crucial aspects, such as reward, success rate, learning curve, sample efficiency, exploration versus exploitation, and convergence. Each of these metrics plays a vital role in providing insights into how well an RL agent is performing in its designated environment. By carefully considering these metrics alongside appropriate evaluation methods, you can effectively assess the performance of your RL agents and make informed decisions regarding their development, fine-tuning, and deployment in real-world applications.

Understanding these metrics not only enables researchers and practitioners to gauge the effectiveness of their RL approaches but also helps in identifying areas that require improvement. For instance, a high reward might indicate that the agent is performing well, while a low success rate could suggest that the agent struggles to complete tasks consistently. The learning curve provides a visual representation of how quickly an agent learns over time, while sample efficiency reveals how effectively the agent learns from the data it collects. Lastly, balancing exploration

and exploitation is crucial for the agent to discover new strategies while maximizing its performance. We list the general definitions of these important metrics below:

- Reward
  - Average Reward: Average reward is a fundamental metric in reinforcement learning that reflects the performance of an agent over time. It is calculated by dividing the cumulative reward obtained by the agent by the total number of episodes or steps it has taken. There are two widely recognized forms of average reward: the average reward per episode and the average reward per step. The average reward per episode is determined by taking the total reward acquired during a complete episode and dividing it by the total number of episodes the agent has completed. Conversely, the average reward per step is calculated by dividing the cumulative reward by the total number of steps taken throughout those episodes. These metrics provide valuable insights into the efficiency and effectiveness of the agent's learning process, revealing how well the agent is performing in its environment.
  - Discounted Reward: The concept of discounted reward is essential in reinforcement learning, as it helps in valuing immediate rewards more than those received in the distant future. The discounted reward is computed as the sum of rewards received over time, where each reward is multiplied by a discount factor, typically denoted as γ. This factor is crucial because it exponentially decreases the weight of future rewards, which allows the agent to prioritize short-term gains while still considering long-term objectives. By employing a consistent discount factor, researchers and practitioners can strike a balance between immediate and future rewards, enabling the agent to develop strategies that are both effective in the short run and sustainable in the long run.
  - Average Discounted Reward: The average discounted reward provides a comprehensive overview of the agent's performance by averaging the discounted rewards received per episode or per step. This metric is particularly useful for evaluating how well an agent can optimize its decision-making over time while taking into account the diminishing importance of future rewards.
  - Cumulative Reward: The cumulative reward represents the total reward that an agent accumulates over a defined number of episodes or steps. It serves as a key performance indicator, allowing researchers and practitioners to assess the overall success of the agent's strategies and behaviors in the given environment. The cumulative reward is instrumental in understanding the long-term effectiveness of the agent's actions and can guide further refinements in its learning algorithms.
- Success Rate: The percentage of episodes in which the agent successfully achieves a predefined goal or reaches a specific state, indicating the effectiveness of the agent's strategy and decision-making capabilities. A higher success rate suggests that the agent is effectively learning and adapting to its environment.
- Episode Length: This metric refers to the average number of steps taken per episode, offering insights into the complexity of the tasks being tackled. Longer

episode lengths may indicate more intricate challenges, while shorter lengths might suggest simpler tasks.
- Learning Curve: A graphical representation that illustrates how the agent's performance, such as average reward or success rate, improves over time. This curve allows us to visualize the learning process and understand whether the agent is effectively acquiring new skills and knowledge.
- Sample Efficiency: This term describes the amount of experience, quantified by the number of interactions with the environment, required for the agent to achieve a desired level of performance. High sample efficiency is crucial in environments where obtaining samples is costly or time-consuming.
- Exploration vs. Exploitation: This concept encompasses the trade-off between exploring new actions to uncover potentially better rewards and exploiting known good actions to maximize immediate returns. Striking the right balance is essential for achieving optimal performance in various environments.
- Convergence: This refers to whether the agent's performance stabilizes over time or continues to improve indefinitely. A convergent agent will reach a performance plateau, while one that continues to improve demonstrates ongoing learning and adaptation to its environment.

Among these metrics, cumulative reward, Success Rate, learning curve, sample efficiency, and convergence are used frequently as performance metrics in various reinforcement learning systems. These metrics provide valuable insights into the effectiveness and efficiency of the learning algorithms employed. However, there are additional considerations that can be utilized as performance metrics, depending on the specific applications and contexts in which the reinforcement learning models are deployed. Understanding these additional metrics can offer a more comprehensive evaluation of a system's performance and its ability to meet the desired objectives. We briefly describe each of these considerations below, highlighting their significance and how they contribute to a more nuanced understanding of reinforcement learning performance in diverse scenarios. By incorporating these additional metrics into the evaluation process, researchers and practitioners can develop more robust and effective reinforcement learning systems tailored to their unique requirements and challenges.

- Task Complexity:The difficulty of the reinforcement learning (RL) task can significantly influence the choice of metrics used for evaluation as well as the expected performance outcomes. More complex tasks may require more sophisticated evaluation metrics to capture the nuances of agent behavior and performance effectively.
- Environment Dynamics:The nature of the environment, whether it is deterministic, stochastic, continuous, or discrete, can profoundly impact the evaluation process. Different dynamics can lead to variations in agent performance and behavior, which necessitate tailored evaluation strategies to accurately assess the agent's capabilities.
- Agent Architecture:The complexity and design of the RL agent, including its neural network architecture and the algorithms it employs, can substantially affect its

performance. Variations in architecture can lead to different learning efficiencies, generalization capabilities, and overall effectiveness in solving tasks.
- Hyperparameters:The choice of hyperparameters, such as learning rate, discount factor, and exploration strategies, can significantly influence the agent's behavior and learning efficiency. The tuning of these parameters is crucial, as they can dramatically alter the learning trajectory and final performance of the RL agent in various environments.

There are also specific metrics for different tasks. We present several examples for specific tasks as below:

- Navigation Tasks
  - Distance Traveled: This metric represents the total distance covered by the agent during its operation. It provides valuable insight into the efficiency of the agent's navigation and movement strategies. A shorter distance traveled can indicate a more efficient route, while a longer distance might suggest unnecessary detours or inefficiencies in the agent's pathfinding algorithms. Understanding the distance traveled can help in optimizing the agent's performance in various scenarios, whether it's a robotic vehicle navigating through a complex environment or a software agent processing data across a network.
  - Goal Completion Time: This metric measures the time taken for the agent to successfully reach its designated goal. It serves as an important indicator of the agent's responsiveness and operational speed. A shorter goal completion time suggests that the agent is effectively executing its tasks and adapting to changing conditions, whereas a longer time might highlight potential areas for improvement in decision-making or processing. Analyzing goal completion time can provide insights into the agent's overall efficiency and capability, allowing for better performance tuning and refinement of algorithms.
  - Energy Consumption: In tasks that involve physical agents, such as robots or drones, energy consumption is a crucial metric to consider. It reflects the amount of energy expended by the agent while performing its tasks, which can have significant implications for operational costs and sustainability. Monitoring energy consumption helps in understanding the efficiency of the agent's movements and actions. By analyzing this data, developers can optimize the agent's performance to minimize energy use, extend operational time, and ultimately enhance the longevity of the agent's components, leading to better overall performance in real-world applications.

- Control Tasks
  - Control Error : Control error refers to the degree of deviation from the desired control signal, which can significantly impact the performance and accuracy of a system. This error is crucial for understanding how well a control system is functioning. It is essential to minimize control error to ensure that the system behaves as intended and achieves its objectives effectively.
  - Stability : Stability is the capacity of a system to maintain a stable state over time, resisting disturbances or fluctuations that might cause it to deviate from its

intended behavior. A stable system ensures reliable performance and predictable outcomes, which are fundamental in various applications, from engineering to finance.
  - Safety Metrics : In safety-critical applications, such as autonomous vehicles or medical devices, specific safety metrics are vital for assessing performance and risk. Metrics like collision rate, the frequency of violations of operational constraints, and response time to emergencies are essential for evaluating the overall safety and reliability of the system. Monitoring these metrics helps identify potential hazards and implement necessary corrective actions to enhance safety measures.

- Game-Playing tasks
  - Win Rate: The percentage of games the agent wins.
  - Score: The total score achieved by the agent in the game.

Most of these performance metrics and considerations can be either used in offline performance evaluation, online performance evaluation, or both.

5.2 Offline Performance Evaluation

The offline evaluation is usually more straight-forward than online evaluation. The evaluation is carried out on offline datasets that are usually predefined before the training. For RL problem with episodic behaviors, many episodes in the testing dataset are conducted with different initial conditions which are usually randomized and preset parameters like the exploration rate ϵ, for upto a time limit or number limit to avoid the experiment/evaluation runs for too long.

Evaluating the performance of an RL agent offline also presents unique challenges due to the lack of real-time interaction. Here are some common approaches.

- On-Policy Evaluation
  - Direct Policy Evaluation (DPE): This method entails the direct assessment of a given policy using an offline dataset, which is a collection of past observations and actions taken by the policy or similar policies in a controlled environment. While DPE can provide valuable insights into the effectiveness of a policy, it is important to note that its accuracy can be compromised if the offline dataset does not adequately represent the true environment in which the policy will be implemented. This discrepancy can lead to biased evaluations, resulting in potentially misleading conclusions about the policy's performance. Therefore, careful consideration must be given to the dataset's characteristics, ensuring that it captures the diversity and dynamics of the real-world scenario.
  - Importance Sampling: This technique is a statistical method used to adjust the weights of samples in the dataset based on their likelihood of occurrence under the current policy being evaluated. By reweighting the samples, importance sampling aims to mitigate bias that may arise from discrepancies between the

training and evaluation policies. However, it is crucial to acknowledge that while this method can help reduce bias, it may also experience high variance, particularly when the policies being compared are significantly different. This high variance can lead to instability in the estimates, making it challenging to draw reliable conclusions.

  - Doubly Robust Estimators: This approach ingeniously combines the principles of Direct Policy Evaluation (DPE) and importance sampling to achieve a more robust estimation of policy performance. By leveraging both techniques, doubly robust estimators seek to reduce bias and variance, resulting in more reliable and stable evaluations. This method is particularly valuable in scenarios where one of the two components may be poor in quality, as it ensures that the overall estimator remains consistent as long as one of the components is correctly specified. Thus, doubly robust estimators provide a powerful tool for practitioners seeking to evaluate and refine their policies effectively.

- Off-Policy Evaluation
  - Q-value-based Methods : These methods focus on estimating the Q-values associated with the current policy by leveraging an offline dataset. Q-values, which represent the expected future rewards for taking specific actions in particular states, are crucial for evaluating the performance of a policy. By calculating these values, one can assess how well the policy is likely to perform in various scenarios without needing to deploy it in a live environment. This approach enables researchers and practitioners to identify potential improvements in the policy through analysis of the Q-values derived from historical data, thereby facilitating a more informed decision-making process regarding policy adjustments and refinements.
  - Behavior Cloning : This technique involves training a supervised learning model to replicate the actions taken by the original policy as observed in the offline dataset. By doing so, the model learns to imitate the decision-making process of the original policy. This method serves as a valuable tool for indirectly evaluating the performance of the original policy by comparing the outcomes achieved by the behavior-cloned model with those from the offline dataset. It provides insights into the effectiveness of the original policy and highlights areas where it might fall short, guiding future improvements.
  - Counterfactual Estimators : Counterfactual estimators are powerful tools that allow for the evaluation of different actions under the current policy by estimating potential outcomes that could have occurred had alternative actions been taken. This method enables a more nuanced evaluation by providing insights into how changing certain decisions might affect overall performance. By simulating various scenarios based on historical data, counterfactual estimators help in understanding the implications of different strategies, thereby enhancing the ability to optimize the policy effectively.

- Simulation-Based Evaluation

  - Synthetic Environments: Create simulated environments that mimic the real-world environment and evaluate the agent's performance in these simulations.

- Benchmarking
  - Standard Datasets: To facilitate a comprehensive evaluation of various offline reinforcement learning (RL) algorithms, it is essential to utilize publicly available datasets. These datasets serve as benchmarks that provide a common ground for comparing the performance and efficiency of different algorithms under controlled conditions. By relying on these established datasets, researchers can gain insights into the strengths and weaknesses of their approaches, allowing for a clearer understanding of how each algorithm performs across diverse scenarios and challenges.
  - Real-World Applications: In addition to theoretical evaluations, it is crucial to assess the performance of offline RL agents in real-world tasks, as this helps in determining their practical applicability and effectiveness. By applying these agents to real-world problems, researchers can better understand their capabilities and limitations in dynamic environments. Evaluating offline RL in practical scenarios not only tests the robustness of the algorithms but also highlights their potential for addressing complex, real-life challenges across various industries, thereby enhancing their overall credibility and usefulness.

Offline evaluation of reinforcement learning systems encounters a range of significant challenges and considerations that must be carefully addressed to ensure reliable results. Key issues include data quality, which refers to the accuracy and relevance of the datasets used for evaluation. Additionally, distribution shift poses a critical problem; this occurs when the data used for training differs from that encountered in real-world applications, potentially leading to suboptimal performance. Evaluation bias is another concern, as it can skew results and misrepresent the effectiveness of the reinforcement learning algorithms. Furthermore, computational cost is a crucial factor, as the resources required for comprehensive evaluations can be substantial, impacting both time and budget constraints. Addressing these challenges is essential for developing robust and effective reinforcement learning systems that perform well in practical scenarios.

- Data Quality: The quality of the offline dataset is crucial for accurate evaluation in any machine learning framework. An offline dataset that is biased or incomplete can lead to misleading results, which can significantly impact the model's effectiveness and real-world applicability. For instance, if certain populations or scenarios are underrepresented, the derived insights may not generalize well, resulting in poor performance when the model is deployed in practical settings. Therefore, it is essential to ensure that the dataset is not only comprehensive but also representative of the various conditions the model will encounter.
- Distribution Shift: The distribution of states and actions present in the offline dataset may not align with the distribution observed in real-world environments. This discrepancy, known as distribution shift, can lead to performance degradation when the model is applied outside the controlled conditions of the offline

dataset. When the model encounters scenarios that were not well-represented during training, its predictions may become unreliable, thus necessitating robust techniques to address this challenge.
- Evaluation Bias: The choice of evaluation method can introduce bias, which can significantly affect the accuracy and validity of the results obtained from model assessments. Different evaluation techniques may focus on distinct aspects of performance, and selecting the wrong method can paint an inaccurate picture of a model's capabilities. It is vital to adopt comprehensive evaluation strategies that mitigate bias and encompass various performance metrics to ensure an accurate assessment.
- Computational Cost: Some evaluation methods can be computationally expensive, particularly when dealing with large datasets or complex models. This cost can become a limiting factor, especially in resource-constrained environments, thereby impacting the feasibility of using certain evaluation strategies. Efficient evaluation techniques that balance thoroughness and computational efficiency are essential to enable timely assessments without sacrificing the quality of the insights gained.

By carefully considering these factors and selecting appropriate evaluation techniques, you can effectively assess the performance of offline RL agents and make informed decisions about their development and deployment.

5.2.1 On-Policy Evaluation

On-policy evaluation (OPE) methods are fundamental for assessing the performance of a policy by utilizing data collected from the specific target policy in question. This approach is particularly relevant in reinforcement learning (RL) problems where data is plentiful and not sparse. In such cases, OPE data is typically gathered using various data sampling methods to ensure both feasibility and efficiency in the evaluation process. However, it is essential to recognize a critical distinction between OPE data and OPE sampling concerning the broader context of policy evaluation. Specifically, OPE sampling might not accurately reflect the expected distribution of on-policy data, particularly after only observing a limited number of trajectories. This discrepancy can significantly impede the effectiveness and efficiency of data-driven policy evaluations and lead to suboptimal decision-making.

In certain scenarios, the reliance on on-policy sampling can lead to inefficiencies, resulting in sample data that is biased away from the anticipated on-policy data distribution. To enhance data efficiency, we present a thorough analysis that illustrates how non-independent and identically distributed (non-i.i.d.), off-policy sampling techniques can yield data that better aligns with the expected distribution of on-policy data. This alignment is crucial for improving the accuracy and reliability of Monte Carlo estimators employed during policy evaluation. By employing these advanced off-policy sampling methods, we can optimize the learning process, leading to improved performance and effectiveness of reinforcement learning algorithms.

To further elucidate this perspective, consider a straightforward yet illustrative example. Imagine a target policy that frequently visits a particular state where it executes action A with a probability of 0.2 and action B with a probability of 0.8. When conducting on-policy sampling, after five visits to this state, we might observe action A occurring twice and action B occurring three times. This observation deviates from the expected frequencies of one occurrence for action A and four for action B, underscoring the inherent randomness associated with on-policy sampling. The variability in outcomes showcases the potential for significant fluctuations from theoretical expectations.

In contrast, by employing off-policy data collection methods and deterministically tracking the expected action proportions as dictated by the target policy, we can achieve the exact expected action frequencies—one for action A and four for action B. While this situation utilizes off-policy sampling, it produces data that arguably aligns more closely with on-policy behavior than that generated by traditional on-policy sampling methods. This observation raises thought-provoking questions about the reliability and effectiveness of various sampling strategies in policy evaluation. It emphasizes the necessity of understanding how different approaches can influence the quality and utility of the data derived from these strategies, ultimately affecting the performance of reinforcement learning algorithms and their applications in real-world scenarios. Through this deeper understanding, we not only improve policy evaluation but also contribute to the advancement of more robust RL methodologies.

In the realm of policy evaluation, a common problem setting is a specific evaluation policy tasked with estimating the expected return that would be realized when implementing this evaluation policy on a chosen task of interest. This problem holds significant importance for the high-confidence deployment of reinforcement learning (RL)-trained policies. In various RL applications, particularly in fields such as robotics, healthcare, and autonomous systems, the significance of data-efficient policy evaluation cannot be overstated. Practitioners and researchers alike desire to achieve the most accurate estimates while minimizing the amount of data collected, as excessive data gathering can be time-consuming, financially burdensome, and resource-intensive. The challenge is further compounded in scenarios where the cost of data collection is high, or where operating in real-world environments entails risks and uncertainties.

While extensive research has been conducted on how to efficiently leverage a set of already collected data—known as the off-policy policy evaluation problem—there remains an implicit assumption within the RL community that, when available, on-policy data is inherently more valuable than off-policy data. This belief is rooted in the notion that on-policy data is generated by the evaluation policy itself, thereby providing a more direct measure of its performance. However, this assumption can overlook the potential advantages of off-policy data, particularly in situations where on-policy data is scarce or difficult to obtain.

When data can be collected in an on-policy manner, the Monte Carlo estimator becomes a powerful tool. It computes a mean return estimate by utilizing trajectories that are sampled independently and identically distributed (i.i.d.) through the execution of the evaluation policy. In an ideal scenario, where an infinite number of

trajectories are gathered, the empirical proportions of each trajectory will converge to their true probabilities under the evaluation policy, leading the estimate to approach the true expected return. However, in reality, the limitations of any finite sample size come into play, causing the empirical proportions of each trajectory to likely diverge from the true probabilities, thereby introducing error into the estimates.

This sampling error is an inherent characteristic of i.i.d. sampling. Specifically, the probability of each new trajectory remains unaffected by the trajectories that occurred previously, which means that the only method to ensure that the empirical distribution aligns with the true probability is to gather a sufficiently large dataset. This leads to the realization that it is only in the theoretical limit that on-policy sampling yields data that is precisely on-policy. In practice, the finite nature of sample sizes can lead to variability and uncertainty in the estimates, necessitating the exploration of alternative methods to improve policy evaluation accuracy, such as utilizing off-policy data more effectively or employing more sophisticated statistical techniques to mitigate the effects of sampling error. Furthermore, robust evaluation frameworks are essential for ensuring that RL policies can be deployed safely and effectively, as the consequences of inaccurate evaluations can lead to suboptimal decision-making and unintended negative outcomes in real-world applications.

The observations made thus far prompt an intriguing question: "Can non-i.i.d., off-policy trajectory sampling lead to a faster convergence of the empirical distribution of trajectories to the expected on-policy distribution?" The answer to this question is affirmative. Various methods have been developed to address this problem by adapting the behavior of the data-collecting policy, taking into account the data that has already been collected when selecting future actions. Specifically, we introduce Robust On-Policy Sampling (ROS), a novel approach that produces an empirical distribution of data that converges more rapidly to the expected on-policy trajectory distribution when compared to traditional on-policy sampling methods.

To delve deeper, let us explore the underlying mechanics of ROS and its advantages over conventional techniques. Traditional on-policy sampling methods collect data solely based on the current policy, restricting their ability to leverage previously gathered information effectively. This limitation can lead to slow convergence rates, especially in complex environments where the number of possible trajectories is vast. In contrast, ROS intelligently modifies the data collection strategy by incorporating historical trajectory data to inform future actions. Through this adaptive framework, ROS not only improves sampling efficiency but also enhances the overall stability of the learning process.

5.2.1.1 Problem Formulation

We formally define the problem of policy evaluation, a crucial aspect of reinforcement learning. In the policy evaluation problem, we are given a policy to be evaluated, denoted as π_e, for which we aspire to estimate $v(\pi_e)$. Algorithms designed for policy evaluation typically involve two main steps: collecting data and computing an

estimate based on that data. The data is gathered by executing a behavior policy that may or may not align with the evaluation or target policy.

In instances of on-policy evaluation, the behavior policy is identical to the evaluation policy, ensuring that the data collected is directly relevant to the policy in question. Conversely, in off-policy evaluation situations, the behavior policy diverges from the evaluation policy, which necessitates more sophisticated techniques to adjust for this discrepancy. The culmination of this process involves calculating a performance metric through a policy evaluation estimator (PE). This estimator maps a set of trajectories to a scalar-valued estimator of $v(\pi_e)$, commonly referred to as the average policy return (APR).

Consequently, the overarching goal of policy evaluation is to conduct the evaluation with a low mean squared error (MSE) on APR. Achieving this goal is paramount, as a lower MSE indicates a more accurate representation of the policy's performance, thus providing a more reliable basis for decision-making in reinforcement learning contexts. By utilizing ROS, we anticipate improvements not only in the speed of convergence but also in the robustness of the resulting estimates, thereby enhancing the effectiveness of the policy evaluation process.

$$MSE[PE] := E[(PE(D) - v(\pi_e))^2 | D \sim \pi_b], \tag{5.1}$$

where π_b is the behavior policy that is run to collect D and PE is a generic policy evaluation estimator.

One kind of common policy evaluation method is Monte Carlo Policy Evaluation (MCPE). Before we delve into specific on-policy evaluation algorithms, it is vital to elucidate how an estimator that leverages on-policy data can significantly benefit from the incorporation of off-policy sampling methodologies. To be more specific, our focus shifts to the Monte Carlo estimator, where we examine a scenario in which we have already amassed a data set, denoted as $D1$, consisting of various trajectories collected from previous interactions with the environment. Our goal is to gather an additional set of trajectories, referred to as $D2$, and subsequently compute the Monte Carlo estimate utilizing the combined data set $D1 \cup D2$.

It is important to highlight that $D1$ represents a fixed set of trajectories that have already been observed, while $D2$ acts as a random variable representing the trajectories that have yet to be collected. This raises a crucial question: how should we go about collecting $D2$ to ensure minimal Mean Squared Error (MSE) in our policy evaluation when applying the Monte Carlo estimator? Our analysis within this section indicates that employing independent and identically distributed (i.i.d.) sampling of trajectories according to policy π_e may not be the most optimal approach available. This insight paves the way for exploring alternative sampling methods that could enhance the accuracy and performance of the Monte Carlo estimator.

To further elucidate the implications of our findings, we need to consider the nature of the data collected in $D2$. In many real-world scenarios, the environment is complex and dynamic, which means that the trajectories collected may be influenced by a multitude of factors, including changes in the state of the environment and the

actions taken by the policy. Therefore, the quality and relevance of the trajectories we collect in $D2$ are paramount for achieving a precise evaluation of the policy.

In this context, the formulation of the Monte Carlo estimator using the combined data set $D1 \cup D2$ can be articulated as follows:

$$\hat{V}\pi = \frac{1}{N} \sum i = 1^N G_i \tag{5.2}$$

where G_i represents the returns obtained from the trajectories in the combined data set. By strategically selecting the trajectories in $D2$ and ensuring they complement $D1$, we can improve the overall estimate of the policy's value, leading to more informed decision-making in policy optimization. This exploration underscores the importance of not only gathering data but also ensuring that the data collection process is tailored to the specific objectives of the policy evaluation task at hand.

$$MC(D_1 \cup D_2) := \underbrace{\frac{1}{n} \sum_{i=1}^{nD_1} g(h_i)}_{\text{fixed value}} + \underbrace{\frac{1}{n} \sum_{i=1}^{nD_2} g(H_i)}_{\text{random variable}}, \tag{5.3}$$

where $nD1$ and nD_2 are the number of trajectories in datasets $D1$ and D_2, respectively, and $n = nD1 + nD_2$. This estimator is commonly referred to as the data-conditioned Monte Carlo estimator (DCMCE).

Understanding the Monte Carlo estimator as a composite of a fixed quantity and a random quantity significantly enhances our comprehension of its statistical properties. By adopting this viewpoint, we can delve deeper into the analysis of how the estimator behaves under various sampling scenarios, which is crucial for a robust application in practical settings. For example, the Monte Carlo estimator has been well-established as unbiased when operating under on-policy sampling conditions. However, the situation becomes considerably more intricate when we turn our attention to its data-conditioned estimate. This complexity becomes particularly evident in the following proposition.

Theory 1. The data-conditioned Monte Carlo estimator is inherently biased when utilizing on-policy sampling from dataset $D2$, unless the condition $MC(D_1) = v(\pi_e)$ is satisfied or if D_1 is devoid of any data (i.e., $D_1 = \emptyset$). This finding highlights a significant limitation in the application of the estimator, emphasizing that its accuracy is heavily dependent on the relationship between the datasets involved.

The implications of this bias are important for researchers and practitioners who rely on Monte Carlo methods for estimating value functions or other related metrics in reinforcement learning and other fields. It suggests that careful consideration must be given to the datasets being utilized, as the interaction between them can lead to skewed results. Furthermore, it raises questions about the robustness of conclusions drawn from such estimators in real-world applications. For instance, if $D1$ contains

trajectories that are significantly different from those in D_2, the bias introduced can lead to misleading interpretations and poor decision-making.

Moreover, the understanding of this bias necessitates a deeper investigation into the conditions under which the DCMCE can be reliably applied. Future research could focus on developing methods to mitigate this bias or explore alternative estimators that can accommodate more complex relationships between datasets. Such advancements could greatly enhance the practical utility of Monte Carlo methods in diverse applications, further solidifying their role as essential tools in the statistical analysis of trajectories.

Notes that Theory 1 remains valid even when $D1$ is obtained through on-policy sampling methods. In scenarios where D_1 is collected under on-policy sampling, the Monte Carlo estimator retains its unbiased status when considering all conceivable realizations of $D1$. This means that regardless of the method used to gather data, as long as the sampling adheres to the principles of on-policy procedures, the estimator remains reliable. However, it is crucial to note that once the trajectories in D_1 are fixed, the potential values they could have assumed become irrelevant. This fixation alters the estimator's statistical behavior, emphasizing the intricate balance between sampling methods and the resulting estimates. Therefore, understanding these dynamics is essential for accurate statistical inference within the framework of Monte Carlo methods, as the assumptions and methodologies underlying the data collection will significantly affect the quality of the estimates derived from them.

Moreover, we can indeed reduce the bias of the data-conditioned Monte Carlo estimator by collecting the dataset $D2$ with a policy that is different from the evaluation policy π_e. This approach allows for the potential mitigation of bias that may arise when the data collected does not adequately represent the underlying state-action distributions. To illustrate this concept more clearly, we conclude this section with an example that demonstrates how such an approach can be beneficial in practice. Consider a simple one-step Markov Decision Process (MDP) featuring only one state, denoted as s, and two possible actions, $a0$ and a_1. The return for taking action $a0$ is 2, while the return for action a_1 is notably higher at 4. The evaluation policy is defined such that $\pi_e(a_0|s) = \pi_e(a_1|s) = 0.5$, indicating that both actions are equally likely under this policy.

Now, let's suppose that after sampling three trajectories, the dataset $D1$ consists of two instances of $\{s, a_0, 2\}$ and one instance of $\{s, a1, 4\}$. It is important to note that action a_0 is over-sampled relative to its true probability in state s, while action $a1$ is under-sampled. This means that the empirical distribution of actions taken in D_1 does not match the theoretical distribution posited by the evaluation policy. If we proceed to collect an additional trajectory using the evaluation policy π_e, the expected value of the Monte Carlo estimate can be calculated as follows:

In this case, the new trajectory collected will provide a clearer representation of the true returns associated with each action, thereby refining our overall estimate. By adjusting our sampling strategy to encompass a range of policies rather than strictly following the evaluation policy, we can enhance the quality of our data, leading to more accurate and less biased Monte Carlo estimates. This example underscores the importance of strategic sampling in Monte Carlo methods, especially in environ-

ments where the distribution of returns may vary significantly based on the actions taken. Thus, one can see that the interplay between the sampling policy and the evaluation policy is not merely a theoretical concern but a practical consideration that can greatly influence the efficacy of statistical inference in reinforcement learning contexts.

$$\frac{1}{4}(2 + 2 + 4 + 2\pi_e(a_0) + 4\pi_e(a_1)) = \frac{11}{4} = 2.75. \tag{5.4}$$

The actual true value, denoted as $v(\pi_e)$, is 3. This figure serves as a benchmark for evaluating the performance of various estimation techniques. Thus, conditioned on the prior data we have, we can see that the Monte Carlo estimate is biased in expectation, as discussed in Theory 1. It's important to understand that this bias can significantly impact decision-making processes and policy evaluations in reinforcement learning scenarios. However, if we were to select the behavior policy such that $\pi_b(a_1) = 1$, we would achieve a situation where neither action is over-sampled nor under-sampled. This adjustment is crucial for ensuring that our estimates are reliable and reflective of the true underlying dynamics of the environment. In this case, the expected value of the Monte Carlo estimate would equal the exact true value, providing a more accurate representation of the system we are analyzing. This alignment between the estimate and the true value is essential for the development of effective policies and strategies in various applications of reinforcement learning.

$$\frac{1}{4}(2 + 2 + 4 + 4) = \frac{12}{4} = 3. \tag{5.5}$$

This example effectively highlights the significance of adapting the behavior policy to take into account the previously collected data, which can substantially lower the expected finite-sample error in policy evaluation processes. The importance of this adaptation lies in its ability to refine the decision-making process, ultimately leading to more accurate predictions and better overall outcomes. In the forthcoming section, we will introduce an innovative adaptive data collection method that dynamically adjusts the behavior policy based on the data that has already been observed. This method not only considers past data but also strategically incorporates it into future decision-making, ensuring that every new piece of information contributes to the refinement of the policy. The adjustment aims to minimize the mean squared error (MSE) of the Monte Carlo estimate, which is a crucial aspect of improving the reliability of our evaluations. By leveraging all available observed data, we can significantly enhance the statistical properties of our estimates. Such an approach promises to enhance the accuracy and efficiency of policy evaluation in various applications, including finance, healthcare, and machine learning. As we progress, we will delve deeper into the mechanisms that underpin this adaptive method, illustrating its potential to revolutionize the way we approach policy evaluation and data collection in complex environments. Through this exploration, we aim to showcase

the transformative impact that thoughtful data adaptation can have on real-world decision-making processes.

5.2.1.2 Robust On-Policy Data Collection

We now describe a method that adjusts the data-collecting behavior policy online to minimize sampling error in the data used by Monte Carlo estimators. This method is referred to as the Robust On-Policy Monte Carlo Estimator (dsilver2022robust) [75]. Specifically, let Dt denote all trajectories observed up to time-step t of the current trajectory, which includes the partial current trajectory as well. At time-step t, our method sets the behavior policy with the specific goal of reducing the current sampling error. This sampling error is defined as the divergence between $Pr(h|\pi_e)$ and $Pr(h|D_t)$.

Our method can be initiated in two ways: starting with $Dt = \emptyset$ or beginning with some pre-existing trajectories in a setting like that described in the preceding section. To effectively reduce sampling error when collecting future trajectories, we seek to adjust the behavior policy in a manner that increases the probability of under-sampled trajectories—specifically, those h for which $Pr(h|Dt) <; Pr(h|\pi_e)$. Unfortunately, we face a significant challenge because the trajectory distributions are inherently unknown, primarily due to the transition function, P, also remaining unknown. To overcome this limitation, we will focus on increasing the probability of under-sampled actions.

Let $\pi_D : S \times A \rightarrow [0, 1]$ denote the empirical policy that represents the proportion of times each action was taken in each state within Dt. If $\pi_D(a|s) <; \pi_e(a|s)$, it indicates that action a has appeared less frequently in the data than would be expected under the policy π_e. Therefore, we should increase the probability of taking action a in state s for future data collection endeavors. When both the state and action spaces are finite, π_D can be computed exactly as the maximum likelihood policy under D_t.

This method provides a systematic approach to improve data efficiency by ensuring that underrepresented actions are given higher priority in future sampling. The implications of this are profound, as it allows for a more informed exploration of the state-action space, ultimately contributing to a more robust and accurate estimation of the underlying policy. By continually refining the behavior policy in light of the observed data, we can mitigate the effects of sampling bias and enhance the overall performance of the Monte Carlo estimator. This iterative adjustment process not only aids in reducing sampling error but also fosters greater adaptability in dynamic environments where the true distribution of trajectories may shift over time. Through this adaptive mechanism, we can strive towards achieving optimal performance in reinforcement learning tasks, ensuring that the models developed are both resilient and capable of learning effectively from their experiences.

$$\pi_D := \arg\max_{\pi} L(\pi), \quad L(\pi) := \sum_{h \in D_t} \sum_{t'=0}^{l-1} \log \pi(a_{t'}|s_{t'}), \tag{5.6}$$

In the realm of reinforcement learning, particularly when dealing with Markov Decision Processes (MDPs) of considerable size, the task of optimizing policies becomes increasingly complex. The argmax operation, which identifies the policy that maximizes expected returns, is typically performed across a vast set of potential policies. As the scale of the MDP expands, the need for function approximation emerges as a practical necessity. This requirement introduces additional layers of complexity to the computation and online updating of the behavior policy, denoted as π_D, especially as new data is collected over time.

Fortunately, by adopting a crucial supplementary assumption, we can navigate this computational complexity. Specifically, we assume that our exploration policy, π_e, is contained within a class of differentiable, parameterized policies. These policies are characterized by a parameter vector, Θ, which lies in the vector space $\mathbb{R}^d$. This assumption is not overly restrictive and is applicable to a wide range of reinforcement learning scenarios. It facilitates the use of various representations for the policy, including tabular methods, linear function approximators, and even more complex neural network architectures.

To streamline our analysis, we denote the particular parameter values associated with the exploration policy π_e as Θe. This notation allows us to focus on the specific configuration of the policy during our discussions. In the following subsection, we will delve deeper into the implications of our chosen parameterization. We will illustrate how the gradient of the log-likelihood function, evaluated at the parameter vector Θ_e, specifically $\nabla_\Theta L(\pi_\Theta)|\Theta = \Theta_e$, can serve as a powerful tool for implementing effective changes to the behavior policy.

By leveraging this gradient information, we can systematically adjust the action probabilities in a manner that mitigates sampling errors, thereby enhancing the overall performance of our reinforcement learning algorithm. This procedure not only optimizes our policy but also ensures that the learning process remains grounded in the underlying data, allowing for more robust and efficient exploration of the state-action space.

Robust On-policy Sampling

Robust On-Policy Sampling (ROS) innovatively reduces sampling error by adapting the behavior policy using a single step of gradient descent on the log-likelihood at each time-step. This sophisticated methodology helps refine the policy effectively, thereby enhancing the performance of reinforcement learning algorithms. Throughout this discussion, we denote the gradient of the log-likelihood evaluated at the current evaluation policy parameters as $\nabla_\Theta \mathcal{L}$. It is crucial to recognize that the log-likelihood function, $\mathcal{L}$, provides explicit guidance for modifying Θe in a manner that increases the probability of actions that have been over-sampled in relation to their expected probability under the evaluation policy Θ_e.

Moreover, $\mathcal{L}$ serves a dual purpose; it also indicates how to decrease the probability of actions that have been over-sampled when the behavior policy $\pi_D(a|s)$ exceeds the evaluation policy $\pi_e(a|s)$. This duality of $\mathcal{L}$ allows the ROS algorithm to adeptly and effectively alter the evaluation policy π_e, ensuring that the distribution

π_D aligns closely with π_e. Importantly, this alignment is achieved without the necessity of explicitly computing π_D. At every time-step, the ROS framework computes the log-likelihood $\mathcal{L}$ using all state-action pairs that have been observed in previous interactions, enabling a robust learning process. Following this computation, the evaluation policy parameters are updated using a single step of gradient descent, which ensures that under-sampled actions are assigned a higher probability than they would typically be given under the evaluation policy π_e.

The pseudocode for the ROS algorithm is detailed in Algorithm **??**. Initially, the algorithm computes $\mathcal{L}$ based on previously collected trajectories, provided that such data exists. Subsequently, ROS actively interacts with the specified Markov Decision Process (MDP) to gather an additional n trajectories. During each action selection, the behavior policy parameters are determined according to the expression $\Theta_e - \alpha \nabla_\Theta \mathcal{L}(\pi_\Theta)|\Theta = \Theta_e$. Following this, the algorithm calculates $\nabla_\Theta log \pi_\Theta(A|s)|\Theta = \Theta_e$ and updates $\nabla_\Theta \mathcal{L}(\pi_\Theta)|\Theta = \Theta_e$. Once these updates are completed, the selected action is executed in the environment, rewards are received, and the agent transitions to the next state. It is noteworthy that the process of updating $\mathcal{L}$ requires per-timestep computation that scales linearly with the number of policy parameters, while remaining constant irrespective of the size of the dataset D. This efficiency is not just a theoretical advantage; it is crucial for the practical implementation of the ROS algorithm in a variety of real-world applications, where computational resources may be limited and the need for efficient algorithms is paramount. The design of the ROS algorithm, therefore, combines theoretical robustness with practical efficiency, making it a valuable tool in the reinforcement learning toolkit.

ROS Convergence

We present a thorough theoretical analysis that underpins the efficacy of the Reinforcement Learning method known as ROS (Rapid Optimal Sampling). This analysis is essential for understanding how ROS can significantly improve the process of policy evaluation in Markov Decision Processes (MDPs). Firstly, we highlight that ROS converges to the expected state visitation frequencies under the optimal policy π_e. This convergence is crucial because it ensures that, over time, the agent's experience aligns closely with the ideal behavior dictated by the optimal policy, allowing for more accurate learning and decision-making.

Second, we demonstrate a vital aspect of ROS: for a fixed state, the distribution $\pi_D(\dot{|}s)$ converges to the optimal policy distribution $\pi_e(\dot{|}s)$ at a significantly faster rate when employing the ROS method compared to traditional on-policy sampling methods. This accelerated convergence is a critical factor in enhancing the efficiency of policy evaluation processes. Traditional methods often suffer from slow convergence rates, which can hinder the ability to quickly adapt to the optimal policy, particularly in complex environments. In contrast, ROS leverages a more effective sampling strategy that accelerates this process, allowing for quicker and more reliable updates to the policy, thereby facilitating more efficient learning.

Finally, we introduce a comprehensive upper bound on the squared error between the Monte Carlo estimate and the true value function $v(\pi_e)$. This bound is expressed

Algorithm 24 Robust On-Policy Sampling

Input: Evaluation policy π_e parameterized with Θ_e; step size α, previously collected trajectories to be used for policy evaluation, D_1 (possibly empty), number of trajectories to collect n.
Output: Data set of trajectories.

for each episode **do**
  $k \leftarrow numberofstate - actiontuplesinD_1$
  $\nabla_\Theta \mathcal{L} \leftarrow \frac{1}{k} \sum_{(s,a)\in D_1} \nabla_\Theta log\pi_\Theta(a|s)|_{\Theta=\Theta_e}$
  $D \leftarrow D_1$
  for each trajectory **do**
    $s_0 \sim d_0$
    for each time step **do**
      $\Theta_b \leftarrow \Theta_e - \alpha \nabla_\Theta \mathcal{L}$
      $a_t \leftarrow A \sim \pi_{\Theta_b}(\dot{|}s_t)$
      $\nabla_\Theta \mathcal{L} \leftarrow \frac{k}{k+1} \nabla_\Theta \mathcal{L} + \frac{1}{k+1} \nabla_\Theta log\pi_\Theta(a_t|s_t)|_{\Theta=\Theta_e}$
      $s_{t+1} \sim P(\dot{|}s_t, a_t), r_t \leftarrow R(s_t, a_t)$
    end for$D \leftarrow D \cup (s_0, a_0, r_0, \ldots, s_{l-1}, a_{l-1}, r_{l-1})$
  end for
end for
Return $\mathcal{D}$

in terms of the sampling error, providing a clear quantitative measure of how errors in sampling can affect policy evaluation outcomes. This formulation illustrates how the faster convergence achieved by ROS positively impacts the Mean Squared Error (MSE) of policy evaluation, making it a powerful tool for practitioners seeking to optimize their reinforcement learning strategies.

These significant results are based on the following foundational assumption.

Assumption 1: The discrete state space of the MDP is structured as a directed acyclic graph (DAG). More specifically, the states within the set S can be partitioned into l disjoint subsets St, each indexed by the episode step. The transition function is defined such that if $P(s'|s, a) >; 0$, it must follow that $s \in S_t$ and $s' \in S_{t+1}$. This structure facilitates a clear progression of states over time, ensuring that no cycles are formed.

It is important to note that Assumption 1 is relatively mild, as any finite-horizon MDP can easily be transformed into a DAG by incorporating the current time-step into the state representation. This flexibility underscores the broad applicability of our theoretical findings, making them relevant for a wide range of reinforcement learning scenarios.

Assumption 2: The Reinforcement Learning algorithm, known as ROS, utilizes a step-size that approaches infinity ($\alpha \rightarrow \infty$). This key feature of ROS facilitates rapid adaptation and responsiveness to the environment, allowing the algorithm to effectively explore and exploit available states and actions. Furthermore, the behavior

policy is parameterized through a softmax function, mathematically represented as $\pi_\theta(a|s) \propto e^{\theta_{s,a}}$. In this formulation, for each state s and action a, a specific parameter $\theta_{s,a}$ is designated. The structure of this policy is integral to the learning process, as it incorporates a mechanism for balancing exploration and exploitation. As formally demonstrated in the subsequent sections, this assumption guarantees that ROS consistently selects the most under-sampled action in each state, thereby maximizing the exploration of less frequently visited actions.

To elucidate further, we introduce the notation $d_t^\pi(s)$, which represents the probability of visiting state s at episode time t while strictly adhering to policy π. This notation is essential for quantifying how often states are visited during the learning process. Additionally, we define $dt^n(s)$ as the empirical frequency of visits to state s at episode time t after observing n distinct trajectories. This empirical frequency serves as a critical benchmark for evaluating the effectiveness of the exploration strategy employed by ROS. Understanding the relationship between these two definitions is crucial for analyzing the convergence and efficiency of our approach within the Markov Decision Process (MDP) framework.

Theory 1: Under assumptions 1 and 2, along with the action selection mechanism of ROS, we assert that ${d_n}^t(s)$ converges to ${d_\pi}^t(s)$ with probability 1 for all $s \in S$ and for any time step $0 \leq t \leq l$. This convergence property is fundamental, as it ensures that the empirical distribution of state visits aligns with the theoretical distribution dictated by the policy, thereby confirming that our algorithm is not only exploring but also learning effectively from its interactions with the environment. This is particularly important in complex environments where the state space can be vast, and efficient exploration strategies are paramount for achieving optimal performance. The implications of this convergence extend to the overall efficacy of the learning process, suggesting that the ROS algorithm is equipped to handle a wide range of scenarios effectively. This robustness is vital for applications in dynamic environments where adaptability is crucial.

$$\lim_{n\to\infty} {d_n}^t(s) = {d_\pi}^t(s), \forall s \in S, 0 \leq t \leq l. \tag{5.7}$$

Theory 2: Let s be a specific state that is visited m times during the process of data collection. For our analysis, we will assume that the set $|\mathcal{A}|$ is greater than or equal to 2, indicating that at least two actions are available for selection at any state. Under Assumption 2, we find that the Kullback-Leibler divergence $D_{KL}(\pi_D(\cdot|s)||\pi(\cdot|s)) = Op(\frac{1}{m^2})$ when employing the ROS action selection method. This reduction in divergence is crucial because it suggests that as we collect more samples, the estimates of the policy derived from our dataset D converge more quickly towards the optimal policy π_e.

Remark 2 highlights that the second term in the bound presented in Theorem 3 represents the KL divergence between the policy derived from the dataset D, denoted as π_D, and the optimal policy π_e. The significance of Theorem 2 is that it informs us that this KL-divergence will decrease at a faster rate when employing the ROS action selection method, which is designed to balance exploration and exploitation

effectively. In contrast, the first term in this upper-bound equation captures the KL-divergence between the empirical state distribution and the true state distribution. This divergence is influenced by two primary sources of sampling error: one arising from action selection and the other from the transition and initial state distributions.

Furthermore, observing the implications of different sampling strategies is vital. For instance, when implementing on-policy sampling, we have:

$$D_{KL}(\pi_D(\cdot|s)||\pi(\cdot|s)) = Op(\frac{1}{m}). \tag{5.8}$$

Here, Op denotes the concept of stochastic boundedness, which is crucial for understanding the behavior of the distributions involved in this context. This distinction underscores the efficiency of ROS over on-policy sampling in terms of convergence speed.

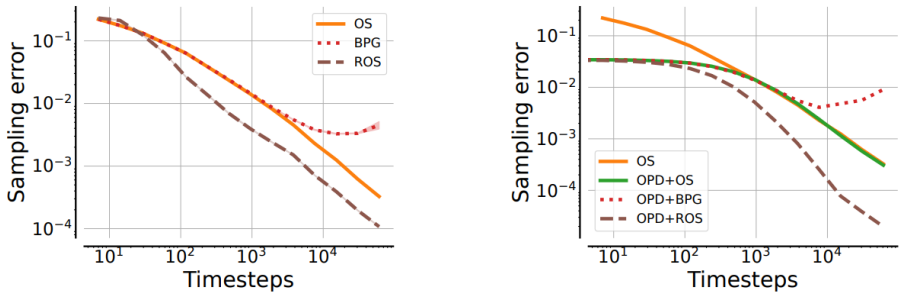
Moreover, the theoretical findings are backed by rigorous policy evaluation experiments conducted across both finite and continuous-valued state and action space domains. These experiments validate the assumptions made in our theories, demonstrating the robustness and applicability of our proposed methodologies in real-world scenarios.

Theorem 3: Assume that s is an element of the state space $\mathcal{S}$, and a is an element of the action space $\mathcal{A}$ such that the reward $R(s,a)$ is less than or equal to $R_{\max}$. Under these conditions, we can assert that the squared error in the Monte Carlo estimate utilizing the dataset D can be upper-bounded by a specific expression. This expression characterizes the relationship between the empirical and true distributions, providing essential insights into the convergence properties of the Monte Carlo estimates. The upper bound reveals how the discrepancies between estimated and actual rewards diminish as the number of samples increases, further emphasizing the importance of sample size in achieving reliable estimates in reinforcement learning frameworks.

$$(v(\pi_e) - MC(\mathcal{D}))^2 \leq \sum_{t=0}^{l-1} \gamma^{2t} {R_{max}}^2 \sqrt{2KL({d_n}^t||{d_{\pi_e}}^t) + 2\mathbf{E}_{s\sim {d_n}^t}[KL(\pi_D(\dot{|}s)||\pi_e(\dot{|}s))]}. \tag{5.9}$$

Through this detailed examination, we aim to bridge the gap between theoretical constructs and practical implementations, allowing researchers and practitioners alike to harness the benefits of advanced sampling methods in their reinforcement learning tasks. The implications of our findings extend beyond mere theoretical interest; they offer actionable insights that can enhance the efficiency and effectiveness of learning algorithms in various applications.

While the former type of error tends to decline at a faster rate under ROS (Return on Sampling) action selection, the latter type decreases at a similar rate for both ROS sampling and on-policy sampling approaches. This observation suggests that while ROS demonstrates a theoretical advantage in reducing certain types of



(a) Without initial data (b) With initial data

Fig. 5.1: Sampling error (KL) curves of data collection in the GridWorld domain are presented in this detailed analysis. The focus of this study is to explore the nuances and implications of various sampling strategies in a structured environment, as represented by the GridWorld framework. Each strategy employed is meticulously followed to collect data across an extensive range of 2^{12} trillion steps, ensuring a comprehensive and robust dataset for thorough evaluation. This massive scale of data collection is crucial, as it allows for a more nuanced understanding of the performance of different sampling strategies, particularly in the context of their effectiveness and reliability. To enhance statistical reliability, all results are averaged over 200 trials, which serves to mitigate the influence of outliers and random variations that may occur in smaller sample sizes. The use of shading in the graphs indicates one standard error interval, providing a clear visual representation of uncertainty and variability in the data. This method of visualization is particularly important for interpreting the results, as it allows researchers to quickly identify the degree of confidence in the estimations presented. Figures (a) and (b) illustrate the sampling error curves of data collection without and with initial data, respectively. These figures effectively highlight the differences in performance under each condition, emphasizing how initial data can influence the learning process. The axes in these figures are log-scaled to facilitate better interpretation of the results and to clearly depict trends across a wide range of values, allowing for an insightful analysis of the underlying patterns in the data. This approach not only aids in comprehension but also enhances the overall clarity of the findings presented in this study. [80].

errors, the practical benefits may not be as pronounced in dynamic environments characterized by high levels of stochasticity. Consequently, the theoretically faster rate of reduction in sampling error associated with action selection under ROS may be somewhat diminished when faced with these unpredictable elements within the environment. Such stochastic factors can introduce significant variability that ultimately overshadows the advantages that the ROS strategy may offer in more stable settings. This leads to a complex interaction between the sampling strategy and the environment's inherent uncertainty.

Moreover, understanding the implications of this interaction is essential for practitioners who seek to optimize their decision-making processes in real-world appli-

cations. The experimental results illustrated in Fig. 5.1 (a) serve to complement and support this theoretical observation, providing empirical evidence that aligns with the derived theoretical insights. These findings highlight the importance of carefully considering the characteristics of the environment when selecting a sampling strategy, as the presence of stochasticity can fundamentally alter the effectiveness of the ROS approach. By examining these results, researchers can gain a deeper insight into the conditions under which various action selection methods may be most beneficial, ultimately leading to more informed decisions in the design of algorithms and systems.

5.2.1.3 Experiment Studies

In this study [75], we aim to conduct a comprehensive empirical investigation of the Robust Off-Policy Evaluation (ROS) method specifically within the context of various policy evaluation problems encountered in reinforcement learning. Our primary objective is to delve deep into the intricacies of this method and its applicability in real-world scenarios. We seek to address several pertinent questions that can significantly inform and guide both future research and practical applications in the field of reinforcement learning. These questions focus on the effectiveness of the ROS method compared to traditional evaluation techniques, its ability to handle different types of data distributions, and its robustness in the face of noisy or incomplete information. By exploring these aspects, we hope to contribute valuable insights that enhance the understanding and implementation of reinforcement learning strategies, ultimately paving the way for more advanced and reliable systems in diverse domains.

- Does ROS significantly reduce sampling error when compared to traditional on-policy sampling techniques? This question investigates the effectiveness of the Robust Off-Policy Sampling (ROS) approach in minimizing the discrepancies that can arise during the sampling process. By analyzing how ROS performs relative to conventional on-policy methods, we can gain insights into the advantages it may offer in terms of accuracy and reliability. Understanding the extent of sampling error reduction is crucial for practitioners in the field, as it could inform their choice of techniques in various applications, ultimately leading to better decision-making and enhanced performance in complex environments.
- Does ROS lead to a decrease in the mean squared error (MSE) of policy evaluation, particularly when starting with both off-policy data and scenarios without such data? This inquiry aims to explore the implications of using ROS for evaluating policies in different contexts. By comparing the MSE when employing ROS against other methods, it becomes possible to assess its effectiveness in deriving accurate policy evaluations. Furthermore, examining scenarios with and without off-policy data helps to understand the robustness of ROS across various data availability situations. The outcome of this analysis will provide valuable insights into the benefits of utilizing ROS for enhancing policy evaluation processes, which is an essential aspect of reinforcement learning and decision-making frameworks.

Experiment Settings

To explore these questions, we design a comprehensive series of policy evaluation experiments that span across four distinct domains, each of which encompasses both discrete and continuous state and action spaces. These domains are carefully chosen to represent a diverse range of challenges and scenarios in the field of reinforcement learning. The first domain is a classic multi-armed bandit problem, as outlined in the foundational work by Sutton and Barto in 1988, which has become a cornerstone for understanding decision-making processes under uncertainty. In this scenario, agents must choose from multiple options, each with unknown payout distributions, making it a quintessential example of exploration versus exploitation dilemmas.

The second domain we investigate is the well-known Gridworld scenario, which serves as a benchmark for evaluating reinforcement learning algorithms. In this environment, agents navigate a grid-based world, encountering various states and rewards, thus testing their ability to learn optimal policies through trial and error. The third domain is the CartPole environment, which presents a more dynamic setting where an agent must balance a pole on a moving cart by applying forces in either direction. This task introduces continuous action spaces and requires the agent to develop precise control strategies to achieve stability.

Furthermore, we extend our exploration to the Continuous CartPole environment, which adds complexity by allowing for a continuous range of actions rather than discrete ones. This is particularly significant as it challenges the agent to learn more nuanced control policies, enhancing the applicability of our findings to real-world problems where continuous actions are prevalent.

For the comparisons across these domains, we primarily utilize on-policy sampling (OS) of independent and identically distributed (i.i.d.) trajectories. By applying the Monte Carlo estimator, we compute the final policy value estimate, which we denote as OS-MC. This approach allows us to evaluate the performance of various policies based on observed trajectories, providing a robust framework for our analysis. Additionally, we conduct comparisons with the Behavior Policy Gradient (BPG) method, which identifies a minimum variance behavior policy for the ordinary importance sampling (OIS) policy value estimator.

The specifics of how the evaluation policy (denoted as π_e) and its corresponding value ($\mathrm{v}(\pi_e)$) were determined are elaborated in writing under the next section. This detailed explanation ensures that readers have access to all relevant methodological details necessary for understanding the experimental framework, thus enhancing the clarity and transparency of our research process. By providing these insights, we aim to contribute to the ongoing discourse on effective policy evaluation techniques in reinforcement learning, ultimately advancing the field's methodologies and practical applications.

Pre-trained Evaluation Policy

Each domain necessitates the development of a comprehensive evaluation policy, which we denote as π_e. This policy is critical for assessing the performance and effectiveness of the actions taken within that domain. In the three specific domains

that feature a discrete action space, we implement softmax policies that take the following form:

This approach allows us to efficiently manage the selection of actions based on their associated probabilities, ensuring that our evaluation process remains both systematic and robust. By leveraging softmax policies, we can effectively balance exploration and exploitation, which is vital for optimizing decision-making in dynamic environments.

$$\pi_\Theta(a|s) \propto \frac{e^{{\omega_a}^T \phi(s)}}{\sum_{b \in \mathcal{A}} e^{{\omega_b}^T \phi(s)}} \tag{5.10}$$

where ϕ is a one-hot encoding operator specifically designed for domains characterized by a discrete state space. This is particularly relevant in environments such as Bandit problems and the GridWorld, where the states can be distinctly identified and represented. In contrast, for domains that exhibit a continuous state space, such as CartPole, we utilize a more complex approach by employing a feed-forward neural network. This neural network allows us to effectively capture the nuances and variations present in continuous states, enabling more sophisticated decision-making processes. Specifically, for the ContinuousCartPole, our formulation is expressed as follows:

This approach ensures that we can leverage the strengths of different representation methods tailored to the nature of the state space, allowing for improved learning and performance in various scenarios. By combining these techniques, we can effectively address the challenges posed by both discrete and continuous environments in reinforcement learning contexts.

$$\pi_\Theta(a|s) := \mathcal{N}\left(a; {\omega_\mu}^T \phi(s), {\omega_\sigma}^T \phi(s)\right)^2, \tag{5.11}$$

Here, we denote ϕ as a function of the state that is determined by a feed-forward neural network. This function is crucial as it serves to map the state space to action probabilities, thereby enabling the policy to make informed decisions. The set of policy parameters, represented as Θ, encompasses all the parameters associated with the policy in question. For simpler environments such as Bandit and GridWorld, Θ consists solely of the vectors ω_a, which dictate the action probabilities for each possible action. However, in more complex settings like CartPole and ContinuousCartPole, Θ also includes the weights and biases of the neural network, reflecting the additional complexity introduced by the neural network's architecture.

When ϕ is defined as a neural network, its architecture is meticulously designed to ensure optimal performance. It is structured to include a batch normalization layer as the first layer, which helps to stabilize and accelerate the training process by normalizing the inputs to each layer. Following this layer, we have two hidden layers, each comprising 64 hidden units. These hidden layers utilize the ReLU (Rectified Linear Unit) activation function, which has been shown to enhance learning effectiveness

by mitigating issues related to vanishing gradients and allowing for faster convergence during training. We implement our neural network using PyTorch, a powerful library that provides flexible tools for building and training deep learning models. Additionally, for the necessary linear algebra computations, we utilize NumPy, a fundamental library that offers efficient operations on arrays and matrices.

In all domains we explore, we apply the REINFORCE algorithm to train the policy model effectively. This algorithm is a classic reinforcement learning approach that utilizes Monte Carlo methods to update policy parameters based on received rewards. During the training phase, we select a policy snapshot to serve as the evaluation policy. This evaluation policy is designed to achieve returns that surpass those generated by a uniformly random policy, although it still remains substantially far from achieving full convergence. To compute $v(\pi_e)$, we employ on-policy sampling to gather a substantial total of 10^6 trajectories. From this extensive dataset, we subsequently compute the Monte Carlo estimate of $v(\pi_e)$, thereby obtaining a robust measure of the policy's performance. This comprehensive approach ensures a nuanced understanding of how the policy operates across diverse domains, facilitating insights that can guide future enhancements and optimizations in our reinforcement learning strategies.

Policy Evaluation without Initial Data

We first conduct a series of experiments in a setting devoid of initial data, where all data is gathered entirely from scratch. This approach is fundamental to understanding the performance of various methods in environments that do not provide any pre-existing knowledge or experience. To facilitate our analysis, we denote T as the average length of a trajectory within each specific domain. Throughout our experiments, we accumulate a total of $2^{12}T$ environment steps using each of the methods under investigation. This extensive data collection allows us to perform a robust evaluation of the different techniques in various settings.

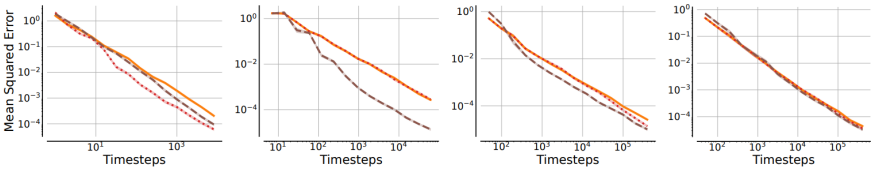
Moreover, we compute relevant metrics at intervals of $2^1, 2^2, \ldots, 2^{12}$ trajectories, ensuring a thorough examination of the performance over time. It is crucial to emphasize that we specify the number of environment steps rather than the number of trajectories in our empirical results to ensure clarity in our methodology. This decision is made to provide a clearer understanding of the relationship between the sampling strategy and the accumulation of data points, which is particularly important in environments with differing dynamics and complexity.

For the Bandit domain, we find that $T = 1$; for GridWorld, $T = 7.42$; for CartPole, $T = 48.48$; and for CartPoleContinuous, $T = 49.56$. These varying trajectory lengths highlight the distinct characteristics of each domain and set the stage for our comparative analysis. The hyper-parameter settings employed across all experiments are detailed in Appendix E for reference, providing transparency and allowing for reproducibility in future research.

In the initial phase of our analysis, we aim to verify that the Reinforcement Learning with ROS (Reinforcement Learning with Robust Off-Policy Sampling) approach effectively reduces sampling error when compared to traditional on-policy sampling methods. To quantify sampling error, we utilize the Kullback-Leibler (KL)

divergence between the estimated policy π_e and a parametric maximum likelihood estimate of the target policy π_D derived from the observed data. A comprehensive definition of this measure, along with an alternative metric that yields qualitatively similar results, is provided in our supplementary materials.

Due to constraints on space, we present this outcome exclusively for the Grid-World domain in Fig. **??**(a); however, the results obtained for other domains exhibit qualitatively similar trends and can be examined in more detail in the appendices. As illustrated in Fig. **??**(a), it is evident that when employing ROS, the sampling error diminishes at a faster rate than with standard on-policy sampling (OS). This finding underscores the effectiveness of our proposed method in improving learning efficiency. Additionally, it is not surprising to observe that the Behavior Policy Gradient (BPG) method results in an increase in sampling error. This outcome arises because BPG is inherently an off-policy method, which adjusts the behavior policy away from π_e, leading to potential misalignment with the target policy. Collectively, these results contribute significantly to addressing our first empirical inquiry, laying the groundwork for subsequent analyses and discussions.



(a) Bandit (b) GridWorld (c) CartPole (d) CartPoleContinuous

Fig. 5.2: MSE of policy evaluation in the context of the 'without initial data' setting. In this scenario, policy evaluation is conducted on the data that has been collected from each strategy employed. These resulting curves effectively illustrate the MSE of the estimates, where a lower value is preferred as it indicates better performance. The vertical axis represents the MSE values, while the horizontal axis indicates the number of environment steps taken, with both axes being log-scaled for clarity and better visualization. Additionally, the shaded areas on the graph signify one standard error, providing a sense of the variability and reliability of the estimates presented [80].

These results provide a comprehensive answer to our first empirical question and substantiate our theoretical assertion that non-independent and identically distributed (non-i.i.d.) off-policy sampling has the potential to accelerate the convergence of the empirical data distribution towards the anticipated on-policy distribution at a rate that surpasses previous observations. This insight is critical for advancing our understanding of sampling methodologies within the realm of policy evaluation. Ultimately, the central focus of this paper is on the imperative task of reducing sampling error, which plays a pivotal role in achieving lower mean squared error (MSE) in the context of policy evaluation.

As depicted in Fig. **??**, our proposed method, which we have designated as ROS (Robust Off-Policy Sampling), demonstrates a remarkable capacity to reduce MSE when juxtaposed with traditional off-policy sampling (OS) methods and the Baseline Policy Gradient (BPG) approach across all evaluated domains. The data presented not only addresses our second empirical question but also lends robust support to the notion that a strategic reduction in sampling error is directly correlated with a significant decrease in the MSE of the Monte Carlo estimator utilized for policy evaluation. This relationship underscores the importance of refining sampling strategies to enhance the fidelity of policy evaluation processes.

Moreover, the implications of these findings extend beyond academic interest; they have substantial practical ramifications for improving the accuracy and reliability of policy evaluation methods across a wide range of applications, including but not limited to reinforcement learning, robotics, and decision-making systems. By effectively addressing the challenges associated with sampling error, our research lays the groundwork for future advancements in the field, paving the way for more efficient and accurate policy evaluation techniques that can be applied in diverse and complex environments.

Policy Evaluation with Initial Data

Our next set of experiments considers a setting with initial data, wherein a comprehensive set of 100 trajectories has already been gathered. These trajectories represent valuable insights into the environment and its dynamics, and our objective is to leverage these trajectories to enhance our policy value estimates. The importance of this initial data cannot be overstated, as it serves as a foundational element for our ongoing research and experimentation efforts. These trajectories were collected through independent and identically distributed (i.i.d.) off-policy sampling, utilizing a behavior policy that exhibits slight deviations from the target policy, denoted as π_e. This experimental framework is designed to mirror a scenario where π_e has recently undergone an update from a preceding policy. This scenario is crucial in understanding how policy updates can affect performance and learning efficiency. In such a case, it becomes essential to exploit the off-policy data that was accumulated from the older policy while simultaneously integrating new data that will be collected in the ongoing experimental phase.

To maximize the utility of the existing data, we will also gather an additional $2*12T$ steps of environment interaction through each method employed in our experiments. This step is crucial since it allows us to further refine our policy and improve its performance based on the most recent interactions with the environment. It is important to note that we do not include the initial 100 trajectories in the overall data tally when calculating the cumulative data we will analyze. For the Reinforcement Off-policy Sampling (ROFFS) method, we will utilize the OPD to initialize the gradient $\nabla_\Theta \mathcal{L}$. This initialization is vital for ensuring that our learning algorithm starts from a well-informed position, thereby potentially accelerating convergence. Our expectation is that the ROS will effectively gather new data, which, when combined with the OPD, will yield an aggregate dataset that appears as though it was originally collected using the policy π_e. This innovative approach allows us to harness

previously collected data while simultaneously adapting to newer information, ultimately leading to more robust policy improvement and enhanced decision-making capabilities in our experiments. By carefully analyzing the interactions and updates, we aim to contribute valuable findings to the field of reinforcement learning and off-policy evaluation.

To evaluate the performance of the Reinforcement Learning method known as ROS, we will conduct a thorough comparison against a range of established baseline methods. This comparative analysis is essential for understanding the strengths and weaknesses of the ROS algorithm in diverse scenarios. The first baseline in our evaluation, denoted as (OPD + OS)-MC, involves the collection of additional data through the Off-policy Sampling (OS) method. This data is then subjected to a Monte Carlo (MC) estimator that processes the entire dataset. The use of MC allows us to estimate the expected returns based on the sampled trajectories, providing a robust benchmark for comparison.

The second baseline, referred to as (OPD + OS)-(WIS + MC), takes a different approach by implementing a Weighted Importance Sampling (WIS) technique. This method derives an estimate from the Off-policy Data (OPD) and subsequently integrates this WIS estimate with a Monte Carlo estimate that is grounded in on-policy data. The combination of WIS and MC allows us to leverage the strengths of both off-policy and on-policy data, potentially resulting in more accurate policy value estimates.

The third baseline, (OPD + BPG)-OIS, involves the collection of supplementary data utilizing the Behavior Policy Gradient (BPG) method. In this case, we employ ordinary importance sampling (OIS) as the estimator across the entirety of the available data. This approach allows us to assess how effectively BPG can enhance the learning process when combined with OIS, providing valuable insights into the interplay between different sampling techniques and their impact on policy evaluation.

Lastly, we examine the (OS - MC) method, which substitutes the initial 100 trajectories with those gathered using the OS method, and then continues to collect the remainder of the data using OS as well, relying on the Monte Carlo estimator for analysis. It is important to note that in the (OS - MC) scenario, the initial 100 trajectories will be included in the total data collected, which may influence the overall results. By conducting these comprehensive experiments, we aim to derive meaningful insights into the effectiveness of various approaches. Our goal is to enhance our understanding of how both off-policy and on-policy data can be leveraged to improve policy value estimates, ultimately leading to more efficient and effective reinforcement learning methodologies. Through this comparative study, we hope to contribute valuable knowledge to the field of reinforcement learning, highlighting the nuanced advantages and potential limitations of each baseline method in relation to the ROS approach.

Fig. **??**(b) illustrates that the sampling error decreases most rapidly for the Reinforcement Learning with ROS method as additional data is progressively collected. This observation is pivotal in understanding the efficiency of different sampling methods in reinforcement learning settings. To assess policy evaluation more com-

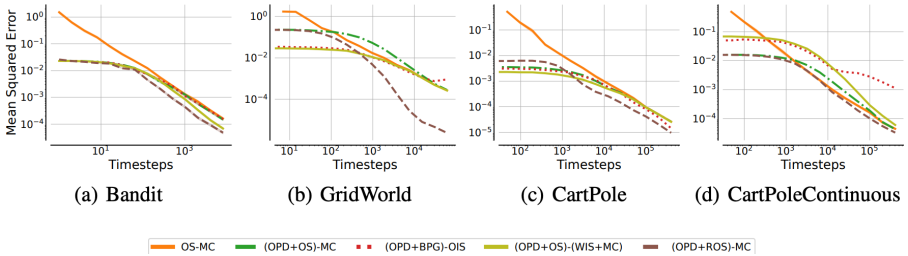


Fig. 5.3: The Mean Squared Error (MSE) of policy evaluation is calculated within the context of the initial data setting, which serves as a foundation for assessing the effectiveness of various strategies. In this evaluation process, policy assessment is carried out using the data that has been meticulously collected from each specific strategy. This is complemented by a small, yet representative set of initial data that has been gathered off-policy, which adds an additional layer of robustness to the analysis. It is essential to highlight that the axes and confidence intervals used in this analysis are consistent with those presented in Fig. **??**. This consistency ensures that the results are not only comparable but also clear and interpretable for stakeholders. By adopting this dual approach, researchers and practitioners can achieve a comprehensive assessment of policy performance that leverages the strengths of both on-policy and off-policy data. Such a thorough evaluation is critical for making informed decisions and optimizing strategies in various contexts, ultimately leading to improved outcomes and enhanced understanding of policy implications. [80].

prehensively, we present the Mean Squared Error (MSE) for varying amounts of data in Fig. **??**, along with detailed numerical values for the final MSE. Our observations reveal that the initial data tends to result in an immediate reduction in MSE; however, this comes at the cost of injecting bias into the estimates. This phenomenon is critical to consider, as it highlights the trade-off between the immediate benefits of data collection and the long-term accuracy of the estimates derived from that data.

Notably, the (OPD+OS)-Monte Carlo (MC) method struggles to mitigate this bias effectively, showcasing its limitations in environments where data quality is paramount. In contrast, the (OPD+ROS)-MC method can successfully address the bias through strategic data collection practices, thereby enhancing the reliability of the estimates produced. Overall, this result underscores the efficacy of ROS in gathering additional data that not only reduces sampling error in the aggregate dataset but also yields lower MSE estimates when compared to other data collection approaches. This reinforces the notion that not all sampling strategies are created equal, and the choice of method can significantly impact the quality of policy evaluation.

Intuitively, the Off-Policy Sampling (OS) method requires a significantly greater number of samples to dilute the bias that arises from using the Off-Policy Distribution (OPD) in the Monte Carlo estimator. This necessity for a larger sample size can be a limiting factor in practical applications, where computational resources and

time may be constrained. In contrast, ROS possesses the capability to correct the empirical off-policy distribution, aligning it more closely with the expected on-policy distribution. This alignment allows the Monte Carlo estimator to function effectively without necessitating any off-policy corrections, which simplifies the evaluation process and reduces computational overhead.

The comparison with OS-MC further highlights the potential of ROS for effectively correcting an off-policy empirical distribution to better reflect the expected on-policy distribution. It is important to note that OS-MC utilizes 100 fewer trajectories than the other baseline methods. However, even when factoring in the initial 100 off-policy trajectories in the total data for all methods, ROS ultimately achieves a lower MSE in comparison to OS-MC. In this regard, ROS has adeptly transformed an initially biased dataset by selectively collecting the appropriate trajectories. This strategic selection creates the appearance that the evaluation policy had collected all trajectories from the outset, thereby enhancing the credibility of the results and demonstrating the power of informed data collection strategies in reinforcement learning.

Summary

From the experiment results, it is safe to conclude that: 1. Reinforcement Learning with Off-Policy Sampling (ROS) effectively reduces sampling error within finite datasets, anD˙2. as a direct consequence of this reduction, it significantly lowers the Mean Squared Error (MSE) of policy value estimates when compared to conventional independent and identically distributed (i.i.d.) on-policy sampling techniques. This advancement in policy evaluation methodology offers a substantial improvement in both the efficiency and accuracy of reinforcement learning (RL) applications across a wide variety of domains. Enhanced accuracy in policy evaluation not only leads to better decision-making but also contributes to more robust learning processes, allowing for quicker adaptation to new information and changes in the environment.

5.2.2 Off-Policy Evaluation

Off-policy evaluation (OFFPE) methods are designed to evaluate and predict the performance of a reinforcement learning policy based on historical data that may have been generated by a different policy. The significance of off-policy evaluation cannot be overstated, particularly in applications where the deployment of a suboptimal or harmful policy can lead to dangerous and costly consequences. For instance, consider a deployed policy that determines which advertisement to show to a user visiting a website. If this policy is ineffective, it may result in poor user experience and lost revenue for the business. Similarly, when it comes to healthcare, a deployed policy that suggests medical treatments could have dire implications if it leads to incorrect or harmful treatment options for patients.

Additionally, personalized educational systems that suggest tailored curricula for students also highlight the importance of effective off-policy evaluation. In this context, the quality of the developed policy can significantly impact a student's learning journey and overall performance. Therefore, it is crucial to predict how well a new policy will perform without the need for immediate deployment, allowing for an analysis of potential risks and benefits beforehand. Moreover, off-policy evaluation methods can facilitate the assessment of existing policies in the face of changing environments, ensuring that the policies remain effective and relevant over time. This capability is essential for maintaining high standards in service delivery in fields where lives and well-being are at stake, underscoring the critical role that OFFPE plays in the advancement of reinforcement learning applications.

The problem of off-policy policy evaluation (OFFPE) is a significant challenge in reinforcement learning and can be defined in several interconnected ways. At the core of OFFPE is the evaluation of a specific policy, denoted as π_e, which is the policy we wish to assess. Alongside this evaluation policy, we have a set of historical data, represented as D, which consists of previously collected state-action-reward tuples. Additionally, we assume the existence of an approximate model of the system being studied, which is crucial for understanding the dynamics of the environment.

Our primary objective is to produce an estimator, denoted $\hat{v}(D)$, which aims to approximate the value function $v(\pi_e)$ associated with the evaluation policy. The value function itself quantifies the expected return that can be achieved by following the policy π_e from any given state. To ensure the efficacy of our estimator, we seek to minimize the mean squared error (MSE) between our estimator and the true value function. This can be mathematically expressed as $MSE(\hat{v}(D), v(\pi_e)) := E[\hat{v}(D) - v(\pi_e)]^2$. In this context, capital letters are utilized to signify random variables, meaning that all random components within expected values are consistently represented by capitalized letters (for instance, D is treated as a random variable).

Furthermore, we operate under the assumption that the underlying process generating states, actions, and rewards conforms to the structure of a Markov Decision Process (MDP). MDPs provide a powerful framework for modeling decision-making problems where outcomes are partly random and partly under the control of a decision-maker. However, it is crucial to note that the initial state distribution, transition function, and reward function remain unknown. Despite this uncertainty, we assume that the evaluation policy, π_e, along with the behavior policies πi for i in the set $1, ..., n$, and the discount parameter γ, are known. This foundational understanding sets the stage for addressing the complexities inherent in off-policy evaluation, enabling us to explore various methodologies and algorithms that can effectively derive accurate estimates based on historical data and the specified evaluation policy.

By leveraging the available information, researchers aim to enhance decision-making processes in a variety of applications, from robotics to adaptive control systems. Effective OFFPE techniques can lead to improved performance in areas such as automated driving, where assessing the safety and efficiency of different driving strategies is paramount. Furthermore, in healthcare, OFFPE can aid in evaluating treatment policies based on historical patient data, leading to better patient outcomes.

Overall, the challenges posed by OFFPE are critical to advancing reinforcement learning methodologies, ultimately contributing to the development of intelligent systems that can adapt to complex, dynamic environments.

5.2.2.1 Doubly Robust Estimator

The doubly robust (DR) estimator, as discussed in the work by Thomas et al. [85], represents a significant advancement in the field of statistical estimation, particularly in the context of Markov Decision Processes (MDPs). This innovative estimator is designed to provide an unbiased estimate of the value function $v(\pi_e)$, and it has demonstrated both promising empirical and theoretical results that have garnered attention from researchers and practitioners alike. The DR estimator achieves this by leveraging an approximate model of an MDP, which effectively reduces the variance associated with the unbiased estimates generated through ordinary importance sampling techniques. By incorporating model-based and model-free components, the DR estimator capitalizes on the strengths of both approaches, providing a more robust framework for estimation in complex environments.

The term "doubly robust" refers to the estimator's ability to deliver reliable estimates under two distinct scenarios: first, if the model used in the estimation process is accurate, and second, if the behavior policies are known. This characteristic is particularly valuable in real-world applications where perfect knowledge of all parameters is often unrealistic. The robustness of the estimator is noteworthy—if the model is inaccurate, the estimator maintains its unbiased nature, although it may exhibit high variance, leading to an increased mean squared error. Conversely, if the behavior policies are not known but the model itself is accurate, the doubly robust estimator tends to yield lower error rates, thus providing a safety net for practitioners who may face uncertainty in their model specifications.

This dual reliability is one of the reasons why doubly robust estimators have gained traction in the statistical community since their introduction by Rotnitzky and Robins in 1995. The versatility of the DR estimator makes it applicable not only in MDPs but also in various fields such as causal inference, machine learning, and epidemiology, where understanding the impact of interventions or treatments is crucial. As the field of statistical estimation continues to evolve, the DR estimator stands out as a powerful tool, offering practitioners a means to obtain reliable estimates in the presence of model misspecifications or incomplete information. The ongoing research into improving and extending the capabilities of doubly robust estimators is expected to further enhance their utility in diverse applications, solidifying their role in modern statistical methodologies.

The foundational work that ultimately led to the development of the Doubly Robust (DR) estimator for Markov Decision Processes (MDPs) was significantly influenced by earlier advancements in doubly robust estimators tailored for bandit problems. This historical connection sheds light on the rationale behind the original derivation of the DR estimator being limited to a finite horizon setting. In this context, the total time horizon is predetermined and known, with L representing the

maximum number of steps taken, where L is finite. Such a limitation resulted in a recursive formulation of the DR estimator, which, while mathematically sound, can be somewhat challenging to interpret and apply in practical scenarios. The recursive nature of this formulation can create barriers to understanding, particularly for practitioners who may not have a deep statistical background or familiarity with complex mathematical constructs.

In response to these challenges, we have undertaken the task of rederiving the DR estimator for MDPs from a fresh perspective, presenting it as an application of control variates. This new approach is particularly notable for its inherent flexibility, as it does not impose any constraints on the time horizon of the processes under consideration. This expanded applicability is crucial for real-world situations where the length of the decision-making horizon may vary significantly. Moreover, our derivation offers a more intuitive and straightforward, non-recursive definition of the estimator, which can be expressed simply as ${w_t}^i = \frac{{\rho^i}_t}{n}$. This reinterpretation significantly enhances the accessibility and usability of the DR estimator across various applications, paving the way for its broader implementation in both research and practical scenarios.

By refining the understanding and application of the DR estimator, we are contributing to the ongoing efforts aimed at improving statistical methodologies within decision-making frameworks. In practical terms, this means that researchers and practitioners alike can leverage the DR estimator more effectively, leading to better-informed decisions in complex environments. To illustrate this point, let us consider a scenario in which π_i represents a set of known policies, and H_i denotes the trajectory generated with policy π_i. We can denote the experience dataset as $D := (H_i, \pi_i)i = 1^n$, comprising the generated experience trajectories. It is important to note that it is permissible for π_i to be equal to π_j. With this framework established, we can define the importance weight ρ_t as the ratio between the probability of the first t steps of H under the evaluation policy π_e and its probability under the behavior policy π_b. This definition lays the groundwork for understanding the operational mechanics behind the DR estimator in practical applications, thus bridging the gap between theoretical development and real-world utility.

$$\rho_t(H, \pi_e, \pi_b) := \Pi_{i=0}{}^t \frac{\pi_e(A_{H_i}|S_{H_i})}{\pi_b(A_{H_i}|S_{H_i})}, \tag{5.12}$$

In other words, the same known policy can generate multiple experience trajectories. The doubly robust estimator on experience dataset D is defined as:

$$DR(D) := \sum_{i=1}^{n} \sum_{t=0}^{\infty} \gamma^t {\omega_t}^i R_{t H_i} \tag{5.13}$$

$$- \sum_{i=1}^{n} \sum_{t=0}^{\infty} \gamma^t ({\omega_t}^i \hat{q}_{\pi_e}(S_{t,H_i}, A_{t,H_i} - \omega_{t-1}^i \hat{v}_{\pi_e}(S_{t,H_i}))), \tag{5.14}$$

where ${\omega_t}^i = \frac{{\rho_t}^i}{n}$ represents the step-length normalized importance sampling weight. While unbiasedness may initially appear to be an attractive characteristic of an estimator, its importance is somewhat diminished when the primary objective is to minimize the mean squared error (MSE). The MSE of an estimator, represented as $\hat{\Theta}$, for a statistic Θ, can be broken down into two key components: its variance and its squared bias. This crucial relationship can be articulated mathematically as follows:

$$\text{MSE}(\hat{\Theta}) = \text{Var}(\hat{\Theta}) + \left(\text{Bias}(\hat{\Theta})\right)^2 \tag{5.15}$$

In this equation, the variance quantifies the degree of variability in the estimator, while the squared bias measures the extent to which the estimator deviates from the true parameter value. Understanding this decomposition is vital for statisticians and researchers, as it enables them to make informed decisions regarding the trade-offs between bias and variance. In many practical situations, an estimator may be biased but have a significantly lower variance, leading to a smaller overall MSE. Consequently, practitioners often focus on optimizing MSE rather than strictly pursuing an unbiased estimator, recognizing that the ultimate goal is to achieve the most reliable and accurate estimates possible in their specific context.

$$MSE(\hat{\Theta}, \Theta) = E[(\Theta - \hat{\Theta})^2] = Var(\hat{\Theta}) + Bias(\hat{\Theta})^2, \tag{5.16}$$

In statistical estimation, the concept of bias is defined as $Bias(\hat{\Theta}) := E[\hat{\Theta}] - \Theta$, where $\hat{\Theta}$ is the estimator and Θ is the true parameter value. This definition highlights a critical aspect of statistical inference: the difference between the expected value of the estimator and the true parameter value. However, in practical applications, the situation is often more complex than simply aiming for an unbiased estimator. The quest for the optimal estimator frequently involves a careful balancing act between bias and variance, a relationship known as the bias-variance trade-off.

Minimizing Mean Squared Error (MSE) is one of the primary objectives in statistical estimation, and it becomes evident that this objective cannot always be achieved by striving for zero bias alone. In fact, an estimator may carry a certain degree of bias but can still yield a lower overall MSE due to a significant reduction in variance. This means that the estimator's predictions are more consistent, even if they are not perfectly centered around the true parameter value. A common scenario in which this trade-off is evident arises in small sample sizes, where the variance of an estimator can be excessively high, leading to unreliable estimates.

As we delve deeper into the implications of this trade-off, it becomes apparent that strong asymptotic consistency is a more desirable property than mere unbiasedness, particularly when the goal is to minimize MSE. Strong asymptotic consistency requires that the MSE of an estimator converges almost surely to zero as the number of observations increases. This characteristic guarantees that as we collect more data, our estimator becomes increasingly reliable and accurate. In many practical

situations, this reliability is paramount, as it reinforces the notion that allowing for a small amount of bias can sometimes enhance overall performance in real-world applications.

To summarize, while unbiasedness is certainly a critical factor in the evaluation of estimators, it is the sophisticated understanding of the bias-variance trade-off that ultimately informs effective estimation strategies. By recognizing that a small bias may lead to a lower overall MSE, statisticians can make more informed decisions about which estimators to use in various contexts, ensuring better performance and more reliable outcomes in their analytical endeavors. This nuanced perspective encourages practitioners to consider the broader implications of their estimation choices, ultimately leading to improved methodologies and results in statistical practice.

5.2.2.2 Weighted Doubly Robust Estimator

The weighted doubly robust (WDR) estimator emerges from an intriguing application of a straightforward yet well-established extension to importance sampling estimators. This connection is then seamlessly integrated into the framework of the doubly robust (DR) estimator, resulting in a novel guided importance sampling method that significantly enhances traditional estimation techniques. The primary advantage of this innovative approach lies in its ability to provide a better balance of the bias-variance trade-off, despite the fact that this extension does not directly target optimization of these competing aspects. Remarkably, the WDR estimator retains the critical property of asymptotic consistency, making it a robust choice for various statistical applications.

To delve into the specifics, the WDR method is fundamentally grounded in the principles of weighted importance sampling, which stands in contrast to the conventional method of ordinary importance sampling. This distinction is not merely academic; it has practical implications. Weighted importance sampling often leads to improved efficiency and robustness in statistical estimation, particularly in situations where the available data may be sparse or unevenly distributed. This characteristic is especially beneficial in real-world applications, where data collection can be challenging, and the resulting datasets may exhibit significant variability. For those interested in exploring the advantages of weighted importance sampling further, there is a wealth of literature available. These studies elucidate the numerous benefits that this approach can offer, enhancing the quality of statistical estimation and inference.

Formally, the definition of WDR aligns closely with that of the DR estimator, with the unique distinction of employing an importance sampling weight denoted as ${\omega_t}^i = \frac{{\rho_t}^i}{\sum j=1^n {\rho_t}^j}$. According to the law of large numbers, the denominator of ${\omega_t}^i$ will converge to the sample size n as the number of observations increases. This convergence has significant implications for the performance of the estimator. Specifically, when considering a single behavior policy denoted as π_b, we observe

that the expected value of the WDR estimator shifts from the value associated with the behavior policy, $v(\pi_b)$, towards the value associated with the target policy, $v(\pi_e)$, as the number of trajectories increases. This shift underscores the estimator's capacity to effectively leverage the data available, ultimately leading to more accurate and reliable estimates in practice.

5.2.2.3 Model and Guided Importance Sampling Combining Estimator

The Model and Guided Importance Sampling Combining Estimator (MAGIC) represents a sophisticated approach to enhancing the efficiency and accuracy of the estimation process within the realm of reinforcement learning. Specifically, MAGIC leverages the benefits of the Bayesian Importance Sampling (BIM) estimator in conjunction with the Weighted Distributional Risk (WDR) estimator, which serves as the importance sampling estimator. This innovative combination merges purely model-based estimates with those derived from the guided importance sampling algorithm, WDR, creating a more robust framework for off-policy evaluation.

To understand the technical intricacies of MAGIC, one must pay close attention to the definitions of $IS^j(D)$ and $AM^j(D)$, as these foundational components can vary significantly across different implementations of the estimator. The off-policy j-step return is defined as follows:

$$g^j(D) := \underbrace{\sum_{i=1}^{n}\sum_{t=0}^{j}\gamma^t\omega_t^i R_i^{H_i}}_{(a)} + \underbrace{\sum_{i=1}^{n}\gamma^{j+1}\omega_j^i\hat{v}^{\pi_e}(S_{j+1}^{H_i})}_{(b)} \tag{5.17}$$

$$\underbrace{-\sum_{i=1}^{n}\sum_{t=0}^{j}\gamma^t(\omega_t^i\hat{q}^{\pi_e}(S_t^{H_i},A_t^{H_i})-\omega_{t-1}^i\hat{v}^{\pi_e}(S_t^{H_i}))}_{(c)}, \tag{5.18}$$

where the term (a) represents the contributions from the model-based estimations, while (b) and (c) pertain to the control variates that blend the information from both the model-based and WDR estimators. Additionally, $\hat{\Omega}$ denotes the sample covariance matrix, which is derived from estimates of Ω based on n trajectories collected in the dataset D.

One of the primary challenges in this estimation process is accurately estimating the bias vector b_n. This complexity arises from the strong dependence of the bias vector on the important but often unknown value $v(\pi_e)$, which corresponds to the expected return of the target policy. The estimates generated by the AM method exhibit a high bias that cannot be overlooked, while the WDR estimator often conflates this bias with the high variance inherent in other importance sampling estimates, leading to potential inaccuracies.

In scenarios where the number of trajectories is limited, the variance of these estimates tends to overshadow the bias, resulting in a high mean square error (MSE). Consequently, it becomes essential to adopt a strategy for estimating b_n that is initially conservative but progressively refines as the number of trajectories increases. To quantify this, we define $CI(g^{\infty}(D), \delta)$ as a $1 - \delta$ confidence interval for the expected value of the random variable $g^{\infty}(D) = WDR(D)$. As the number of trajectories n increases, this confidence interval converges toward $g^{\infty}(D)$, which, in turn, approaches the true value of $v(\pi_e)$.

Therefore, we can estimate $b_n(j)$, the bias of the off-policy return at the j-th step, using a 10% confidence interval. The resulting estimate for $b_n(j)$ is expressed as follows:

$$b_n(j) = CI(g^{\infty}(D), 0.1) \tag{5.19}$$

This formulation underscores the critical importance of continual refinement in the estimation process, which ultimately leads to enhanced accuracy and reliability in the evaluation of off-policy returns within the MAGIC framework. The significance of this refinement cannot be overstated, as it plays a pivotal role in ensuring that the estimates generated by the MAGIC algorithm are not only consistent but also robust against various forms of bias that can arise in reinforcement learning scenarios. By meticulously refining the estimation procedures, MAGIC aims to address and mitigate the adverse effects of bias and variance, which are common challenges faced by many reinforcement learning algorithms. Such improvements not only enhance the quality of the estimates but also contribute to the overall performance of the reinforcement learning systems that utilize these methodologies.

$$\hat{b}_n(j) := dist(g^{\mathcal{J}_j}, CI(g^{\infty}(D), 0.5)), \tag{5.20}$$

At the heart of this refinement process is the mathematical component represented by the expression $dist(y, \mathcal{Z})$, which quantifies the distance between a given point y in the real number space $\mathbb{R}$ and a subset $\mathcal{Z}$ contained within $\mathbb{R}$. The specific formulation $dist(y, \mathcal{Z}) = \min_{z \in \mathcal{Z}} |y - z|$ serves as a critical tool for evaluating the closeness of estimates to their true values, thereby playing an essential role in the effectiveness of the MAGIC estimator.

Moreover, the MAGIC estimator is recognized as a strongly consistent estimator of $v(\pi_e)$. This assertion holds true under the assumption that Weighted Direct Returns (WDR) are incorporated as one of the off-policy returns, as well as the underlying assumptions that support the proof of strong consistency for WDR. For those interested in a deeper understanding of these theoretical underpinnings, we encourage consultation of [85], particularly Appendix H, which contains detailed mathematical proofs and discussions that elucidate these concepts further. This exploration not only reinforces the theoretical foundation of MAGIC but also highlights its practical implications in the field of reinforcement learning.

5.2.2.4 The ModelFail Domain

The ModelFail domain was meticulously designed to illustrate how behavior policies can lead to experience trajectories that fail to converge to the true Markov Decision Process (MDP). This phenomenon can occur in several ways, one of which is the reliance on function approximation. When a model leverages function approximation, it may not possess the capacity to accurately represent the true MDP, leading to flawed decision-making. Additionally, the presence of partial observability in the system can further complicate matters. This limitation is especially prevalent in numerous real-world applications across various fields, including robotics, finance, healthcare, and more. Such partial observability can hinder an agent's ability to make informed decisions, as it lacks complete information about its current state, which significantly impacts the accuracy and effectiveness of its outcomes.

To elaborate, the specific MDP utilized in the ModelFail experiment is depicted in Fig 5.4. The MDP comprises three standard states along with a terminal absorbing state, but critically, the agent does not possess the capability to discern its exact location within these states. Instead, the agent only has access to a single observation, which limits its decision-making faculties. The journey begins in the left-most state, where the agent is presented with two distinct actions to choose from. The first action consistently propels the agent to the upper state, while the second action invariably leads it to the lower state. Notably, in both scenarios, the agent does not receive any reward, which could have a profound impact on its decision-making process, as it may struggle to evaluate the outcomes of its choices.

At time $t = 1$, the agent finds itself situated in either the upper or lower state, yet it remains unable to differentiate between these states or the initial state it started from. Faced with a critical decision, the agent must choose between two available actions. Both actions lead to the terminal absorbing state, but the rewards differ based on the state the agent was in: if it was in the upper state, it receives a reward of $R_1 = 1$, whereas if it was in the lower state, the reward is $R_1 = -1$. The horizon is defined as $L = 2$, establishing a limited timeframe for the agent's decision-making process. The behavior policy is configured to select action a_1 with a probability of approximately 0.88 and action a_2 with a probability of approximately 0.12. These probabilities were determined arbitrarily through the application of weights of 1 and -1 utilizing softmax action selection, and notably, they were not optimized for performance. Conversely, the evaluation policy operates in a contrasting manner, selecting action a_1 with a probability of approximately 0.12 and action a_2 with a probability of approximately 0.88. This creates a dynamic and challenging decision-making environment for the agent, illustrating the complexities associated with learning and optimizing behavior policies in partially observable and functionally approximated MDPs.

When we consider the task of modeling a Markov Decision Process (MDP) using only the observations generated by executing a behavior policy, we encounter a complex scenario. This situation involves creating an infinite number of trajectories based on the limited information we can gather, specifically, observing just a single state throughout the entire process. It's crucial to recognize that this restrictive

observation can lead to significant challenges in accurately capturing the dynamics of the MDP.

To begin our analysis, let us delve into the transition dynamics of this scenario. We find that, remarkably, half of the time, the actions taken by the agent lead to a transition back to this single observed state. In contrast, the other half of the time, the agent transitions to an absorbing state, which signifies the termination of the episode. This binary outcome illustrates the inherent uncertainty and complexity involved in modeling the MDP. The fact that the agent has only one observable state complicates our understanding of the potential consequences of various actions.

Next, we must scrutinize the rewards associated with these actions. In this model, we see that half of the time, the agent receives no reward whatsoever. Furthermore, there exists a probability of $0.88/2$ that the agent will receive a reward of 1, while there's also a probability of $0.12/2$ that the agent will receive a negative reward of -1. This distribution of rewards is particularly noteworthy because it appears to be entirely uncorrelated with the specific actions chosen by the agent. For instance, non-zero rewards may manifest at time $t = 1$, but the action A1 does not influence either the rewards or the subsequent state transitions that follow. This lack of correlation indicates that from the perspective of the model, the actions taken do not have any meaningful impact on the resulting state transitions or the rewards.

As a direct consequence of this situation, every policy evaluated within this model is considered equally effective. Each policy yields an expected return of 0.38, despite the fact that, in reality, an optimal policy would yield an expected return of 0.5, while a pessimal policy would produce a significantly lower expected return of -0.5. This discrepancy highlights the inadequacies of the model when it is based on such limited observational data.

To enhance the validity of this model, we provided it with the true horizon, $L = 2$. This adjustment allows the model to predict that the expected rewards, R_t, will be zero for any time t that is greater than or equal to 2. This setup showcases the inherent challenges and limitations that arise when attempting to model MDPs based solely on restricted observations, without taking into account the complete context of the decision-making environment. The reliance on limited data can lead to misguided conclusions about the effectiveness of different policies, ultimately impeding the ability to make informed decisions based on the model's predictions.

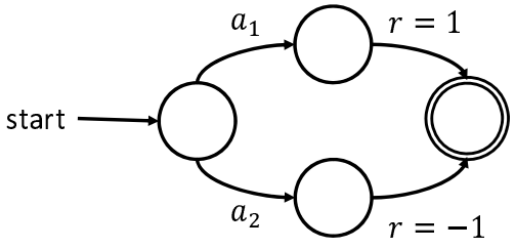


Fig. 5.4: Example ModelFail MDP [85].

5.2.2.5 Experiments

Experiments are conducted to compare the performance of the Weighted Doubly Robust (WDR) estimator against several previous estimators, including Importance Sampling (IS), Partial Doubly Importance Sampling (PDIS), Weighted Importance Sampling (WIS), Cumulative Weighted Partial Doubly Importance Sampling (CWPDIS), Doubly Robust (DR), and the Adaptive Method (AM). This comparative analysis is crucial for understanding the advantages and limitations of each estimator in various contexts. The experiments utilize three distinct domains: ModelFail, ModelWin, and a gridworld setting. Each of these domains is characterized by a finite horizon and employs a discount factor of $\gamma = 1.0$.

ModelFail represents a scenario where the model encounters various failure states, allowing us to assess how different estimators handle adverse outcomes. This is crucial for understanding the robustness and resilience of different algorithms when faced with challenges that could lead to suboptimal performance. The conditions under which these failures occur can vary widely, encompassing issues such as incorrect predictions, unexpected changes in the environment, or limitations in the model's ability to generalize from training data to real-world situations. By systematically analyzing these failure states, we can identify which estimators are more adept at managing uncertainty and instability in their predictions.

Conversely, ModelWin is designed to simulate successful transitions and outcomes, providing a contrasting backdrop for performance evaluation. In this environment, the focus shifts to understanding how various estimators capitalize on favorable conditions to optimize their performance. This duality between ModelFail and ModelWin creates a holistic framework for examining the strengths and weaknesses of each estimator under different circumstances. The gridworld setting serves as a versatile environment where agents navigate a defined space with specific rewards and penalties, allowing for dynamic interactions that can be influenced by the choice of estimator. This setting not only facilitates the exploration of decision-making strategies but also enables a clearer comparison of how different models perform in varying situations.

To provide a comprehensive understanding of our findings, we will first describe each of the experimental domains in detail, highlighting their unique characteristics and the rationale behind their selection. Each scenario is carefully crafted to illuminate specific aspects of model performance, ensuring that our analysis covers a broad spectrum of potential real-world applications. Following this, we will outline the experimental setup, detailing the methodologies, parameter settings, and evaluation criteria used in our analysis. This will include a discussion of the statistical techniques employed to ensure the validity and reliability of our results. Ultimately, we will present the empirical results obtained from our analyses, offering insights into the relative performance of the WDR estimator compared to the other methods. This structured approach ensures that our findings are not only clear but also actionable for future research and practical applications in the field. By distilling our insights into a format that is both accessible and informative, we aim to con-

tribute meaningfully to ongoing discussions in the domain of model evaluation and optimization.

The ModelWin Domain

The ModelWin domain was meticulously designed to ensure that the approximate model of the Markov Decision Process (MDP) converges rapidly to the true MDP. In contrast to this, importance sampling-based methods, such as the Doubly Robust (DR) and Weighted Doubly Robust (WDR) approaches, are expected to exhibit high variance throughout their operational phases. As discussed thoroughly in Section 6, both DR and WDR demonstrate a remarkable tendency to reduce to a straightforward model-based approach when the approximate MDP aligns closely with the true MDP. This is particularly evident in scenarios characterized by deterministic state transitions and rewards. However, to mitigate the risk of oversimplification arising from this tendency, we introduced stochastic state transitions within the ModelWin domain. These probabilistic transitions are crucial because they ensure that the control variable employed by DR and WDR does not perfectly counterbalance the Policy Density Importance Sampling (PDIS) term. This dynamic maintains a necessary level of complexity in the decision-making process, which in turn enhances the overall robustness of the model.

The ModelWin MDP is visually represented in Fig. **??**. In stark contrast to the ModelFail domain, the ModelWin domain provides the agent with direct access to the true underlying states of the MDP. This particular MDP consists of three distinct states, in addition to a terminal absorbing state that is not depicted in the figure. The agent consistently begins its journey in state s_1, where it faces a critical decision between two potential actions. The first action, labeled as a_1, allows the agent to transition to state s_2 with a probability of 0.4 and to state s_3 with a probability of 0.6. Conversely, the second action, a_2, produces the inverse effect: facilitating a transition to state s_2 with a probability of 0.6 and to state s_3 with a probability of 0.4. Upon successfully transitioning to state s_2, the agent receives a reward of 1, while a transition to state s_3 results in a penalty of -1.

In both states s_2 and s_3, the agent is presented with two available actions; however, both actions invariably yield a reward of zero and a deterministic transition back to the initial state, s_1. The horizon for this model is deliberately set to $L = 20$, ensuring that S_{20} remains infinite. This infinity presents a unique challenge for the agent, compelling it to navigate through its decision-making process with care and strategy. The design of this model encourages the agent to explore various strategies while reinforcing the complexities inherent in decision-making under conditions of uncertainty. Ultimately, this intricate setup facilitates a more robust evaluation of the different estimators' performances, thereby advancing our understanding of their effectiveness in real-world applications. This investigation not only contributes to the field of reinforcement learning but also provides valuable insights into the practical challenges faced by agents operating within stochastic environments.

The behavior and evaluation policies both select actions uniformly at random in states s_2 and s_3. This uniform selection indicates a lack of preference or bias in

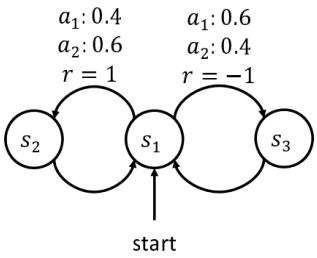


Fig. 5.5: Example ModelWin MDP [85].

these particular states, which can be beneficial in scenarios where exploration of all possible actions is essential for learning and improving decision-making strategies. However, when we turn our attention to state s_1, we observe a distinct divergence in the behavior policy, which exhibits a notable preference for action a_1. In this state, action a_1 is chosen with a probability of approximately 0.73, whereas action $a2$ is selected with a lower probability of around 0.27. This skewed preference suggests that the behavior policy has identified action a_1 as more favorable, possibly due to prior experience or knowledge regarding the potential rewards associated with that action.

Conversely, the evaluation policy operates in stark contrast to the behavior policy. In state s_1, the evaluation policy selects action a_1 with a probability of approximately 0.27, while action $a2$ is chosen with a significantly higher probability of about 0.73. This inversion in selection probabilities emphasizes the critical role of evaluation in understanding the consequences of actions, particularly when there is a need to assess the effectiveness of different strategies. The specific probabilities we observe in these policies are the result of implementing softmax action selection, which assigns weights of 1 to action a_1 and 0 to action a_2. This weight assignment effectively skews the selection process, allowing for a nuanced approach to decision-making that balances exploration and exploitation.

To illustrate the implications of these probabilities further, we can draw upon examples from related domains, such as the ModelFail and ModelWin scenarios. In the ModelWin domain, we provided the approximate model with the true horizon of the Markov Decision Process (MDP), which was set at $L = 20$. This strategic choice ensures that the model's predictions regarding the reward function, denoted as R_t, are zero for all time steps t that are greater than or equal to 20. This alignment is crucial as it allows the model to accurately understand the relationship between actions and future rewards within the MDP framework. By carefully considering the implications of both probabilities and horizons, we can enhance the modeling and evaluation of decision-making processes, ultimately leading to improved outcomes in complex environments. Such a detailed approach aids in refining our understanding of how different policies can influence behavior and decision-making in various states, thereby fostering better strategies for action selection.

The Gridworld Domain

The third domain that the experiments utilized was the gridworld domain, which was specifically developed for the purpose of evaluating OFFPE (Off-Policy Policy Evaluation) algorithms. This gridworld consists of a structured $4x4$ layout, featuring four distinct actions available to the agent, and operates under a fixed horizon of $L = 100$. In this controlled environment, the agent navigates through the grid, seeking to optimize its actions to maximize the cumulative reward over the course of its interactions. The environment is characterized by deterministic transition and reward functions, providing a consistent backdrop for assessing various OFFPE methods under well-defined conditions. Such deterministic characteristics ensure that the same sequence of actions leads to predictable outcomes, thus allowing for a rigorous evaluation of the performance of different algorithms.

The original research on this gridworld proposed five distinct policies that can effectively serve as both behavior and evaluation policies. These policies were crafted to explore various strategies and decision-making processes that the agent could adopt while interacting with the grid. This flexibility in policy design is crucial, as it allows researchers to test the robustness and adaptability of OFFPE algorithms in response to different behavioral patterns.

While the gridworld was designed for robust OFFPE evaluations, it was not originally intended to accommodate newer methodologies like Direct Reinforcement (DR) and Weighted Direct Reinforcement (WDR), which were introduced in subsequent studies. This limitation arises from the deterministic nature of the state-transition and reward functions; when the model is accurately specified, AM (Actor-Mimic), DR, and WDR will exhibit similar performance levels. However, the introduction of DR and WDR methodologies raised questions about their applicability in this deterministic context, prompting further investigation into their strengths and weaknesses compared to traditional approaches.

Moreover, the simplicity of the gridworld serves as both an advantage and a limitation. While it allows for clear insights into the functioning of OFFPE algorithms, it may not capture the complexities and nuances found in real-world scenarios. This raises important considerations for researchers looking to generalize findings from the gridworld to more complex environments, where stochastic elements and dynamic transitions could significantly alter algorithm performance. Overall, the gridworld remains a foundational tool in the exploration of OFFPE methodologies, serving as a stepping stone toward more elaborate and realistic testing environments.

To address these limitations, a series of experiments were conducted utilizing two distinct variants of a gridworld to better understand the effects of horizon perception on agent performance. In the first variant, which we refer to as Gridworld-TH, the approximate model was provided with the correct horizon, set at $L = 100$. This allowed the agent to operate with accurate knowledge of its trajectory length, facilitating optimal decision-making and enabling it to effectively navigate toward rewards without the complications that arise from uncertainty.

Conversely, the second variant, termed Gridworld-FH, introduced a level of partial observability by supplying the model with an incorrect horizon of $L = 101$. This

seemingly minor adjustment has considerable ramifications for the value predictions made by the model, especially as the agent approaches the end of a trajectory. In Gridworld-FH, the model inaccurately forecasts when rewards will inevitably diminish to zero, leading to potential miscalculations in the agent's strategy. The agent may continue to pursue actions that are no longer beneficial, resulting in a degradation of performance as it misinterprets the remaining opportunities for reward accumulation.

To differentiate these setups clearly, we have established the nomenclature: Gridworld-TH for the gridworld in which the agent is endowed with the true horizon, and Gridworld-FH for the scenario with the false horizon. The implications of these two configurations are significant and multifaceted, as they provide a foundation for a comprehensive analysis of how variations in horizon perception can profoundly affect the performance of Optimal Finite Horizon Policy Evaluation (OFFPE) methods in practical applications.

By conducting these experiments, we aim to delineate the nuances of agent behavior in response to different horizon perceptions. This exploration not only enhances our understanding of the underlying dynamics at play within gridworld tasks but also offers critical insights into the broader implications for reinforcement learning applications. Ultimately, these findings may inform the design of more robust algorithms that can better handle uncertainty in horizon estimations, thereby improving performance in complex environments where accurate state information is not always guaranteed.

Experiment Settings

For each domain, we generated a set of n trajectories, where n varied across multiple predefined values to assess how the number of trajectories influenced the performance of different Off-Policy Evaluation (OFFPE) methods. In this process, we meticulously computed the sample mean squared error (MSE) between the predictions made by the various OFFPE algorithms and the actual performance of the evaluation policy. To determine the true performance of the evaluation policy, we relied on an extensive number of on-policy Monte-Carlo rollouts, which provided a high level of accuracy and reliability in our comparisons. The robustness of our approach is underscored by the fact that for each specified value of n and for each OFFPE algorithm, we conducted a total of 128 separate experimental trials. This repetition allowed us to mitigate any anomalies or outliers in the data, ensuring that the results we reported were representative of typical performance under each condition.

To provide a comprehensive overview of the performance of the different algorithms, we calculated the average sample mean squared error across the 128 trials for every combination of n and OFFPE method. This approach not only allowed us to quantify the performance of each algorithm but also facilitated meaningful comparisons among them. For the estimation of $b_n(j)$, we utilized the tighter of the two statistical techniques: the percentile bootstrap confidence interval and the methodology outlined in [86], alongside Chernoff-Hoeffding's inequality. This rig-

orous statistical framework ensured that our confidence intervals were both accurate and reliable.

All resulting plots were meticulously designed to include standard error bars, which serve to illustrate the variability inherent in the results. By employing logarithmic scales for both the horizontal and vertical axes, we aimed to enhance the clarity and interpretability of the data presented. This methodological rigor not only strengthens our findings but also allows for deeper insights into the performance dynamics of the various OFFPE algorithms under investigation. Through this comprehensive analysis, we hope to contribute valuable knowledge to the field of Off-Policy Evaluation, offering guidance for future research and practical applications.

Perhaps surprisingly, determining a fair comparison between the various OFFPE algorithms proves to be quite challenging, primarily due to the inherent differences in their methodologies and the assumptions they make about the data. Clearly, Importance Sampling (IS), Per-Decision Importance Sampling (PDIS), Weighted Importance Sampling (WIS), and Conditional Weighted Per-Decision Importance Sampling (CWPDIS) should utilize all of the trajectories in the dataset D, as these algorithms do not necessitate an approximate model to operate effectively. The definitions and theoretical foundations of these four importance sampling techniques can be found in [80], which provides a comprehensive overview of their mechanisms and applications.

On the other hand, the Approximate Model (AM) method is designed to leverage the entirety of the data available in order to construct its approximate model effectively and improve the robustness of its predictions. However, this leads to a pertinent question: how should the available data be appropriately partitioned when applying the Double Robust (DR), Weighted Double Robust (WDR), and the MAGIC estimators? Proper data partitioning is crucial, as it can significantly influence the performance and reliability of these estimators. In this context, we propose at least three reasonable approaches to address this issue.

First, one could consider a stratified sampling approach, which involves dividing the dataset into distinct strata based on relevant characteristics before applying the estimators. This can help ensure that each estimator has access to a representative subset of the data. Second, a cross-validation technique may be employed, allowing for multiple training and testing iterations to better assess the performance of the estimators in a controlled manner. Lastly, one could explore a bootstrapping method, which would enable the creation of multiple resampled datasets, providing a robust means of evaluating the variability and stability of the performance metrics across different configurations. Each of these approaches offers unique advantages and can contribute to a more nuanced understanding of the comparative effectiveness of the various OFFPE algorithms.

- It is essential for DR, WDR, and MAGIC to be supplied with additional trajectories that are not accessible to IS, PDIS, WIS, and CWPDIS. These additional trajectories should be strategically employed to construct an approximate model that can enhance the overall performance of these methods. By utilizing these extra trajectories, we can create a more robust framework that closely resembles a scenario

where prior domain knowledge is leveraged. This knowledge may not necessarily be limited to trajectories but can encompass various forms of data and insights that can significantly contribute to building an effective approximate model. It is important to highlight that IS, PDIS, WIS, and CWPDIS do not incorporate such prior knowledge into their frameworks, which can limit their effectiveness in certain contexts. The additional trajectories serve as a crucial resource that enriches the modeling process, allowing for a more nuanced understanding of the underlying dynamics.

- An alternative approach worth considering is that DR, WDR, and MAGIC should utilize the complete dataset D to construct an approximate model. Following this initial construction, they would then reuse this same dataset to compute their estimates. While this method may seem reasonable and potentially beneficial for enhancing the estimates, it is critical to recognize that reusing data could potentially invalidate our theoretical guarantees. The implications of this method must be carefully considered, especially in terms of how it affects the validity of the results obtained. Nevertheless, based on empirical observations, we have noted that this strategy allows DR, WDR, and MAGIC to perform at their optimal levels. The empirical evidence suggests that, despite the theoretical concerns, the practical outcomes of this approach may be favorable in many real-world applications.
- Lastly, we propose that DR, WDR, and MAGIC should implement a systematic partitioning of the dataset D into two distinct subsets. The first subset should be specifically designated for constructing the approximate model, while the second subset should be reserved for computing the estimates of DR, WDR, and MAGIC using the previously constructed approximate model. This deliberate partitioning strategy could serve to ensure that the estimates generated remain unbiased, thereby facilitating a fair and equitable evaluation of the various OFFPE methods. By adopting this partitioning approach, we can enhance the reliability of our experimental conclusions and ensure that the insights drawn from our studies are grounded in robust methodologies. This structured approach not only mitigates potential biases but also allows for a clearer interpretation of the performance differences among the methods under consideration.

Since there is not necessarily a singularly "correct" answer to the question of which method of performing experiments is optimal, we have chosen to present our results using both the second and third approaches to provide a comprehensive view. This dual approach not only enriches our findings but also allows for a more nuanced understanding of the experimental outcomes. For each domain considered in our study, the "full-data" variant employs the second approach, which utilizes the entirety of the available data to draw conclusions. In contrast, the "half-data" variant employs the third approach, wherein the dataset D is partitioned into two sets of equal size for analysis. This methodological divergence is crucial, as it helps to highlight the potential variability and robustness of our results under different experimental conditions.

Given that all the domains we focus on possess finite state and action sets, we apply a straightforward maximum-likelihood approximate model to our evaluations. This model serves as a fundamental framework for understanding how different states and actions interrelate within our experimental setup. Specifically, we predict that the probability of transitioning from state s to state s', given action a, is calculated by taking the number of times this particular transition was observed and dividing it by the total number of times action a was executed in state s. This approach ensures that our estimations of transition probabilities are grounded in empirical evidence, thus enhancing the reliability of our conclusions.

In situations where dataset D contains no recorded instances of action a being taken in state s, we adopt the assumption that taking action a in state s invariably leads to a transition into the terminal absorbing state. This assumption is critical, as it allows us to account for scenarios where data is sparse or non-existent, thereby ensuring that our model remains functional even in less-than-ideal circumstances. By incorporating these varied methodologies and assumptions, we strive to present a comprehensive analysis that not only addresses the complexities of the domains under investigation but also provides valuable insights for future research endeavors. This multifaceted approach underscores the importance of flexibility and adaptability in experimental design, ultimately contributing to a deeper understanding of the systems we study.

Experiment Results

In this section, we present a comprehensive set of empirical results derived from four well-established importance sampling methods: standard importance sampling (IS), per-decision importance sampling (PDIS), weighted importance sampling (WIS), and consistent weighted per-decision importance sampling (CWPDIS). Each of these methods has been rigorously tested and validated in various contexts, providing a robust foundation for our findings. In addition to these traditional approaches, we also include results from guided importance sampling techniques, which encompass doubly robust importance sampling (DR) and weighted doubly robust importance sampling (WDR). These guided methods are particularly noteworthy, as they aim to enhance the efficiency and accuracy of the sampling process by leveraging additional information about the underlying decision-making framework. Moreover, we introduce results from the purely model-based method known as approximate modeling (AM), which utilizes a distinct approach that relies heavily on the underlying model dynamics rather than sampling methods.

To facilitate a clear understanding of the results shared in this appendix, we have ensured that the legend utilized by all the plots is clearly provided in Fig. **??**. This legend serves as a crucial reference point, enabling readers to easily interpret the various results presented across different methods. By offering this structured overview, we aim to enhance the accessibility and clarity of our empirical findings, allowing for a more thorough analysis and understanding of the comparative performance of these importance sampling techniques.

Fig. 5.6: The legend used by all plots of experiment results [85].

ModelFail Results

In Fig. 5.7, we reproduce this experiment in a full-data setting, which allows for a comprehensive evaluation of the various importance sampling methods employed. This setting is particularly advantageous as it furnishes a richer dataset, enabling us to draw more nuanced conclusions regarding the efficacy of these sampling techniques. Within this context, the weighted importance sampling methods, specifically Weighted Importance Sampling (WIS) and Control Weighted Partial Data Importance Sampling (CWPDIS), appear to be obscured by the curve representing the performance of Weighted Doubly Robust (WDR) estimators. This overshadowing suggests that while WIS and CWPDIS have their merits, they may not perform as effectively as the WDR method in certain scenarios. Conversely, the unweighted importance sampling methods, including Importance Sampling (IS) and Partial Data Importance Sampling (PDIS), are similarly obscured by the curve associated with the Doubly Robust (DR) estimators. This indicates that the DR estimators maintain a strong performance relative to their unweighted counterparts, providing a competitive edge in this full-data context.

A notable observation is that WDR significantly outperforms the Augmented Model (AM) by orders of magnitude, demonstrating its superiority and suggesting that the methodologies employed in WDR are particularly effective in capturing the underlying data distribution. Furthermore, it surpasses the performance of the DR method by approximately an order of magnitude as well, reinforcing the idea that WDR represents a robust choice for estimation tasks in complex data environments.

Additionally, it is important to note that despite the inaccuracy of the approximate model utilized in this setting, which implies that the control variates employed by both DR and WDR may not be optimal, the performance of the DR and WDR estimators does not diminish below that of PDIS and CWPDIS, respectively. This suggests a robustness in their performance even with less-than-ideal models, indicating that these methodologies are not only resilient but also capable of yielding reliable estimates across various conditions. This robustness is critical in practical applications where model inaccuracies are often unavoidable, thus highlighting the relevance of WDR and DR methods in real-world scenarios. Overall, these findings emphasize the importance of selecting appropriate sampling methods based on the specific characteristics of the data and the modeling context.

Moving on to Fig. 5.8, we replicate this experiment in a half-data setting, which introduces a different dynamic and context for analyzing the methods employed. In this scenario, the available data is effectively halved, creating a unique challenge for the various approaches being tested. Since the AM (Approximate Model) method does

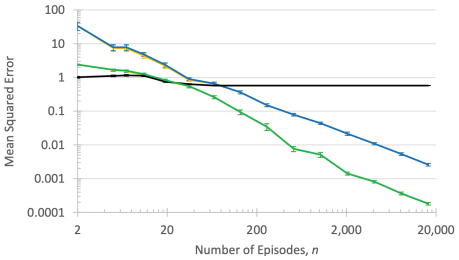


Fig. 5.7: ModelFail, full-data [85].

not leverage any data for importance sampling, its performance remains identical in both the half-data and full-data scenarios. This feature underscores the AM method's inherent robustness and stability when confronted with reduced data availability.

Similarly, the methods of IS (Importance Sampling), PDIS (Propensity-weighted Direct Importance Sampling), WIS (Weighted Importance Sampling), and CWPDIS (Covariate-weighted Propensity-weighted Direct Importance Sampling) do not utilize an approximate model. Instead, they rely on the full dataset to maintain their performance metrics, resulting in consistent outcomes across both experimental settings. This characteristic highlights the reliance of these methods on comprehensive data for accurate estimations and assessments.

However, the situation differs significantly for DR (Doubly Robust) and WDR (Weighted Doubly Robust) methods. These approaches utilize half of the available data to construct their approximate model while employing the remaining half to compute their respective estimates. Consequently, this division of data can result in a potentially inferior approximate model for both DR and WDR. As such, this limitation leads to a slight upward shift in their performance curves as they adapt to the constraints imposed by the reduced dataset. It is noteworthy that while their performance may be slightly diminished, the overarching trends remain clear: WDR consistently outperforms AM by significant margins, demonstrating its superior ability to leverage available data effectively. Similarly, DR also exhibits a performance advantage over AM, albeit by a smaller order of magnitude.

This consistency in performance across different data settings emphasizes the reliability and adaptability of the WDR and DR approaches in various scenarios. It suggests that, even when faced with limited data, these methods can still provide valuable insights and robust results, making them essential tools in the toolkit of data-driven decision-making. As researchers and practitioners continue to explore different methodologies, the resilience of WDR and DR under varying conditions serves as a testament to their potential in real-world applications.

Fig. **??** provides a comprehensive depiction of the results obtained from implementing both importance sampling and guided importance sampling methods, alongside the approximate model estimator, within the context of the ModelWin experimental setup under a full-data scenario. The analysis presented in this figure allows for a detailed comparison of the various methodologies employed in the

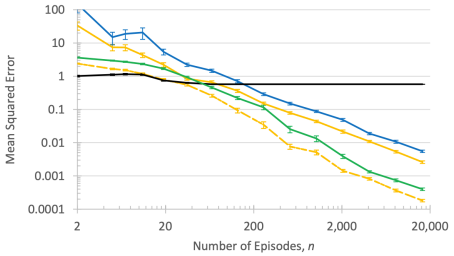


Fig. 5.8: ModelFail, half-data [85].

study, highlighting not only their respective performance metrics but also the underlying principles that dictate their effectiveness. Notably, the Adaptive Method (AM) demonstrates an approximately order of magnitude lower mean squared error (MSE) compared to all other methodologies examined, including the Weighted Dynamic Regression (WDR) approach. This substantial performance advantage underscores the robustness and efficiency of the AM in processing and analyzing data in this specific context.

Furthermore, the results indicate that the AM's capability to dynamically adjust to the data characteristics significantly enhances its predictive power. This p erformance edge served as a key motivator for our innovative strategy of combining AM with WDR through the use of Bayesian Inference Methods (BIM). By integrating these two methodologies, we aim to harness the strengths of both approaches, potentially leading to even lower error rates and improved model accuracy. The synergy created between AM and WDR is anticipated to offer a more nuanced understanding of the data, while also accommodating the complexities often encountered in practical applications. This exploration into the combination of methodologies represents a promising avenue for future research and development in statistical modeling and data analysis.

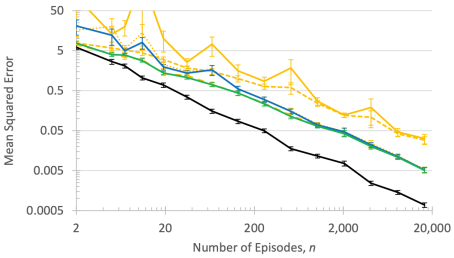


Fig. 5.9: ModelWin, full-data [85].

In Fig. **??**, we replicate this experimental analysis in a half-data setting, which offers additional insights into the performance dynamics of the methods under consideration. This half-data scenario allows us to examine how the reduction in

available data impacts the efficacy of the different methodologies employed. Consistent with our findings from the ModelWin setup, we observe that the reduction in available data adversely affects the performance of both the Doubly Robust (DR) and Weighted Doubly Robust (WDR) methods. Particularly noteworthy is the observation that when the number of trajectories is limited, the impact on performance appears to be more pronounced for the DR methodology in comparison to the WDR approach. This suggests a potential robustness in the WDR method that may be advantageous in data-scarce environments.

However, it is essential to consider that this observation may be influenced by the presence of noise within the data. As evidenced by the substantial standard error bars associated with the DR curve, particularly when the sample size (n) is small, it becomes increasingly evident that the variability in results can skew our understanding of the true performance of the method. The high standard errors indicate that the estimates derived from the small sample are less reliable, highlighting the necessity for cautious interpretation of the results. Overall, this analysis underscores the critical importance of data quantity and quality in evaluating the performance of algorithmic approaches in statistical modeling and predictive analytics.

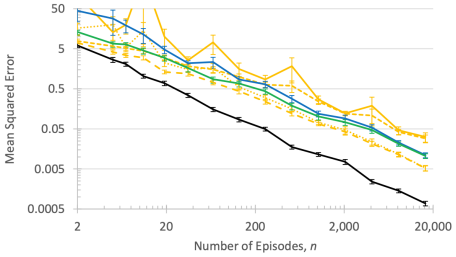


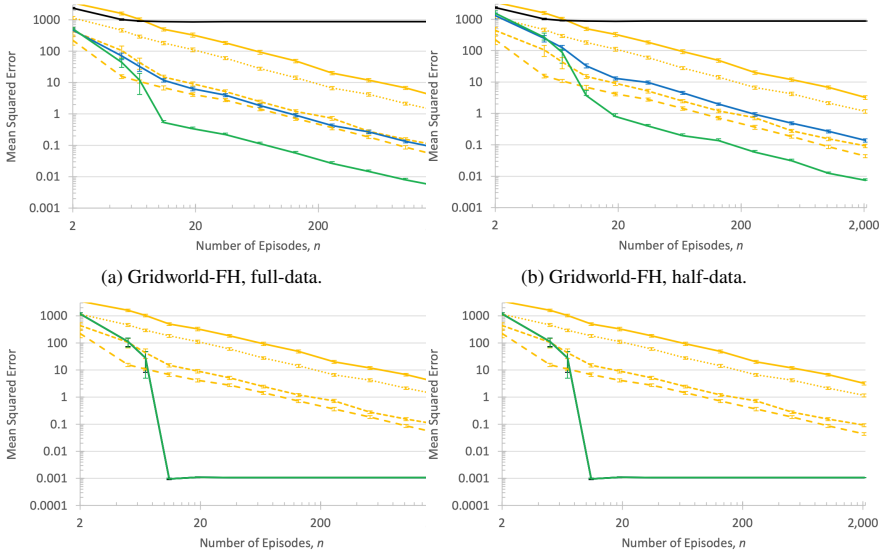
Fig. 5.10: ModelWin, half-data [85].

Fig. **??** provides a detailed depiction of the results obtained from utilizing the fourth gridworld policy, denoted as π_4, functioning as the behavior policy, alongside the fifth policy, π_5, which serves as the evaluation policy for the Gridworld-FH domain under a full-data setting. It is noteworthy that the Weighted Doubly Robust (WDR) method significantly outperforms all other competing methods by at least an order of magnitude, showcasing its effectiveness and reliability in this context. This substantial performance advantage can be attributed to the way WDR integrates both off-policy data and model-based estimates, thus effectively mitigating biases that often arise in reinforcement learning scenarios.

In Fig. **??**, we replicate this experiment within a half-data setting. As anticipated, there is minimal change observed in the overall trends of the results, with the exception that both the Doubly Robust (DR) and WDR curves exhibit an upward shift. This upward shift indicates a modest improvement in estimation accuracy, likely due to the varying data distribution when only half of the original data is utilized. Remarkably, WDR continues to be the top-performing estimator, maintaining its

superiority by approximately an order of magnitude over others, thereby reinforcing the robustness of its approach even when faced with limited data.

Subsequently, we performed a reproduction of Fig. 5.11a and Fig. 5.11b, this time focusing on the Gridworld-TH scenario as opposed to Gridworld-FH. The results are illustrated in Fig. 5.11d and Fig. **??**, respectively. It is essential to highlight that when the true horizon is accurately provided, the Adaptive Method (AM) demonstrates exceptional performance. In the full-data setting, both DR and WDR align closely with the curve for AM, which is logical given that the transition function and reward function in this context are deterministic. The deterministic nature of these functions means that the outcomes are predictable, leading to a higher degree of alignment between the methods. Consequently, the way we constructed our approximate model leads both methods to converge precisely to AM. However, in the half-data setting, DR and WDR lag slightly behind AM's curve, primarily due to the limitation of utilizing only half as much data. This reduction in available data can introduce variability and uncertainty, impacting the accuracy of the estimations. Overall, these findings emphasize the importance of data availability in reinforcement learning and highlight the robustness of WDR in various scenarios.



(a) Gridworld-FH, full-data.

(b) Gridworld-FH, half-data.

(c) Gridworld-TH, full-data.

(d) Gridworld-TH, half-data.

Fig. 5.11: The Gridworld Domain; π_4 behavior policy, π_5 evaluation policy

Furthermore, we replicated these four figures employing the first gridworld policy, denoted as π_1, as the behavior policy and the second policy, π_2, as the evaluation policy. This experimental setup allowed us to explore the dynamics of the OFFPE (Off-Policy Evaluation) framework under different conditions. In contrast to the deterministic nature exhibited by policies π_4 and π_5, which consistently generate

longer trajectories, the policies π_1 and π_2 are significantly less deterministic. This inherent randomness in their action selection leads to the production of notably shorter trajectories, which can have important implications for the learning and evaluation processes.

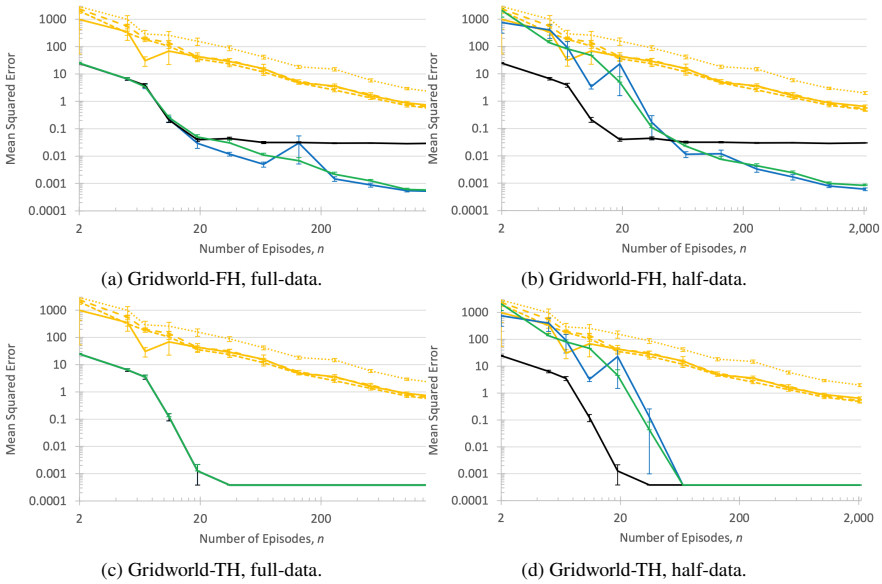
Notably, the behavior policy, π_1, selects actions in a uniformly random manner. This characteristic creates a distinctly different setting for the OFFPE evaluation, as the randomness introduces various exploration-exploitation trade-offs that are not present in the deterministic policies. The outcomes from this alternative configuration are presented in Fig. **??**, which illustrates the performance metrics associated with these policies.

In this particular example, both the Direct Reinforcement (DR) and Weighted Direct Reinforcement (WDR) methods perform comparably well. They show significant improvement over traditional importance sampling algorithms such as Importance Sampling (IS), Per Decision Importance Sampling (PDIS), Weighted Importance Sampling (WIS), and Continuous Weighted Per Decision Importance Sampling (CWPDIS). Moreover, they achieve marginally better results than the Average Model (AM) when ample data is available for training and evaluation.

Additionally, when provided with the true horizon, both DR and WDR converge towards the performance of the AM method, which highlights the robustness and adaptability of these techniques in various data scenarios. This convergence indicates that even in environments characterized by high variability and uncertainty, DR and WDR can effectively leverage the available information to enhance performance, thus demonstrating their potential utility in real-world applications where data may be limited or non-stationary. Overall, these findings emphasize the importance of exploring different policies and evaluation methods to truly understand the capabilities and limitations of off-policy evaluation frameworks.

The key takeaways from these comprehensive experiments indicate that the Weighted Dynamic Reweighting (WDR) method tends to outperform other importance sampling estimators. This includes traditional methods such as Importance Sampling (IS), Per-Decision Importance Sampling (PDIS), Weighted Importance Sampling (WIS), and Conditional Weighted Per-Decision Importance Sampling (CWPDIS), as well as the guided importance sampling technique known as Doubly Robust (DR). Notably, none of these competing methods managed to achieve mean squared errors that were within an order of magnitude of WDR's performance across all of our diverse experimental conditions. This significant disparity highlights the robustness and effectiveness of WDR as a guided importance sampling method, showcasing its potential in various applications where accurate estimations are critical.

However, it is crucial to note that WDR did not consistently emerge as the superior approach in every experimental setting. For instance, in the ModelFail setting, the Adaptive Method (AM) outperformed WDR by a considerable order of magnitude. Such observations underscore the importance of context when evaluating the effectiveness of these methods. Similar insights have been documented in previous studies; for instance, in the experiments conducted, AM consistently outperformed DR. However, the researchers did not compare their results to WDR since it had



(a) Gridworld-FH, full-data.

(b) Gridworld-FH, half-data.

(c) Gridworld-TH, full-data.

(d) Gridworld-TH, half-data.

Fig. 5.12: The Gridworld Domain; π_1 behavior policy, π_2 evaluation policy

not yet been introduced at that time. This notable gap in the literature served as a key motivation for our introduction of the Blended Importance Sampling (BIM) estimator, which is designed specifically to harness the strengths of both WDR and AM.

It is also important to recognize that under certain conditions—specifically, when the transition function and reward function are deterministic and there is an absence of partial observability, such as in the gridworld experiments utilizing the true horizon—both DR and WDR can effectively degenerate to AM based on the construction of our approximate model. While this degeneration is not inherently detrimental, it does suggest that importance sampling methods may not be necessary under these specific conditions. However, it is worth noting that this degeneration would not occur if the approximate model employed function approximation techniques.

Lastly, it is noteworthy that both DR and WDR exhibited superior performance in the full-data setting compared to the half-data setting. This finding implies that in practical applications, it is advisable to utilize all available data to create an approximate model and to compute the DR and WDR estimates. Although this approach may be seen as a violation of the assumptions underlying our theoretical guarantees, it does not imply that methods like MAGIC will not remain strong, consistent estimators for the given application context. In fact, the adaptability of these methods to the complexities of real-world data further illustrates their viability in various scenarios, making them invaluable tools for researchers and practitioners alike.

5.2.3 Simulation-Based Evaluation

Simulation-based evaluation is an essential technique in the field of Reinforcement Learning (RL) that involves the creation and interaction with simulated environments to comprehensively assess the performance of RL agents. This innovative approach offers several significant advantages over direct evaluation methods conducted in real-world settings. These advantages include:

- Safety : By utilizing simulations, researchers can avoid potential risks or damages that could arise during real-world experiments, ensuring the well-being of both the agents and the environment. For instance, in applications involving autonomous vehicles or robotic systems, conducting tests in a simulated environment can prevent accidents that might occur if the same experiments were carried out in real life. This is particularly crucial in high-stakes domains such as healthcare, where the implications of failures can be dire.
- Control : Simulated environments allow for precise manipulation of parameters and conditions, enabling researchers to systematically study their effects on agent performance and decision-making processes. By controlling variables such as reward structures, environmental dynamics, or the presence of obstacles, researchers can gain deeper insights into how these factors influence the learning and behavior of RL agents. This level of control is often unattainable in real-world scenarios due to the complexity and unpredictability of natural systems.
- Efficiency : Simulation-based evaluation accelerates the experimentation process, allowing for rapid data gathering and analysis, which is invaluable in optimizing and refining RL algorithms. In a simulated setting, thousands of episodes can be run in a fraction of the time it would take to conduct a single real-world experiment. This efficiency not only speeds up the research cycle but also enables the exploration of a vast parameter space, facilitating the identification of optimal strategies and solutions.
- Flexibility : This approach facilitates the exploration of a wide range of scenarios and counterfactuals, enabling researchers to test the robustness of their agents in various hypothetical situations and improve their adaptability. By simulating different environmental conditions, unexpected events, or even changes in the rules of engagement, researchers can assess how well an RL agent can generalize its learning and adapt to new challenges. This flexibility is particularly important in dynamic environments where the circumstances may change rapidly, requiring agents to be resilient and capable of adjusting their strategies on the fly.

In conclusion, simulation-based evaluation stands out as a powerful tool in the domain of reinforcement learning (RL). By prioritizing safety, control, efficiency, and flexibility, it not only enhances our understanding of agent behavior but also accelerates the development and deployment of sophisticated RL systems across various applications. This innovative method allows researchers and practitioners to push the boundaries of what is possible within the field of artificial intelligence, fostering continuous innovation while minimizing the inherent risks associated with real-world testing.

The significance of simulation-based evaluation lies in its ability to create controlled environments where RL agents can be rigorously trained and tested. This approach ensures that agents can learn from a wide range of scenarios and challenges without the high stakes often associated with real-world implementation. By utilizing simulations, researchers can explore various strategies, parameters, and learning algorithms, gaining insights that might otherwise remain hidden if confined to conventional testing methods. Ultimately, the use of simulations ensures that RL agents can be thoroughly vetted before being applied in real-world scenarios, paving the way for safer and more effective AI solutions.

Moreover, simulation-based evaluation is particularly advantageous for assessing agents that have been trained through the process of transfer learning. This method involves leveraging the knowledge acquired from training on an offline dataset, which is instrumental in enhancing performance in a new, yet similar, environment. Transfer learning is vital in scenarios where the computational cost of training from scratch would be prohibitively expensive or where data acquisition is challenging. In many situations, particularly when experimentation could be costly or impractical, real environments are often not accessible for direct testing. Thus, simulations provide an invaluable alternative for researchers to evaluate the efficacy and robustness of their algorithms.

Furthermore, as the complexity of environments increases, the need for sophisticated simulation tools becomes more pronounced. Advancements in graphics, computational power, and algorithm design have led to the creation of highly realistic and interactive simulations. These developments allow researchers to model intricate real-world dynamics more accurately, which in turn enhances the learning process for RL agents. Consequently, simulation-based evaluation plays an indispensable role in shaping the future of reinforcement learning, ensuring that it remains at the forefront of AI advancements.

The main components of simulation-based evaluation encompass several critical elements, including evaluation metrics, environment simulation, and agent interaction. These components work together to create a robust framework for assessing how well an agent performs in a simulated environment, which can closely mimic real-world scenarios.

Environment simulation typically consists of two essential processes: model construction and environment parameterization. Model construction involves designing a detailed digital representation of the real-world environment, capturing its dynamics, states, actions, and associated rewards. This step is crucial, as it lays the groundwork for how the agent will interact with the simulated environment. In some cases, model construction may be optional if evaluation models can be directly derived from real physical models or training models related to the reinforcement learning (RL) problems being investigated. However, even when using existing models, careful consideration must be given to ensure they accurately reflect the complexities of the environment.

On the other hand, environment parameterization is responsible for establishing the properties and rules that govern the environment. This includes defining the boundaries within which the agent operates, the types of actions available to it,

and the reward structure that incentivizes specific behaviors. One of the primary algorithm properties that must be rigorously tested is its robustness to environment initialization. If the performance of the agent is significantly influenced by the specific state in which the simulation begins, it is generally safe to conclude that the learning algorithm lacks robustness. This indicates that the agent's ability to generalize its learning across various scenarios may be limited, which is a crucial aspect to address in order to enhance the effectiveness and reliability of the simulation-based evaluation process.

Agent interaction involves several critical components, including policy execution, environment updates, and the collection of experiences for evaluation purposes. The RL agent's policy plays a pivotal role, as it determines the actions the agent takes based on the current state of the environment. Once the agent performs its actions, these actions lead to transitions between states and generate reward signals that indicate the success or failure of those actions. The feedback loop created by this interaction is essential for the agent's learning.

To facilitate effective learning, the agent meticulously stores its interactions comprising state, action, reward, and next state—allowing for thorough evaluation. This evaluation process is typically conducted at regular intervals to ensure the agent's learning and performance are continuously assessed and improved. Moreover, the insights gained from these evaluations can be invaluable, helping to refine both the agent's learning algorithms and the simulation environment itself. Ultimately, by focusing on the intricate interplay between environment simulation and agent interaction, developers can create more sophisticated and capable agents that perform well across a variety of tasks and conditions. This iterative process of evaluation and refinement is key to advancing the field of reinforcement learning and its practical applications.

Simulation-based evaluation methods frequently utilize synthetic environments within environment simulation to rigorously test the learned policies or trained agents. Synthetic environments are intricately designed to develop and create highly advanced simulated scenarios that closely mimic the complexities and nuances of the real-world environment. These meticulously crafted simulations serve a dual purpose: they enable a comprehensive evaluation of the agent's performance and behavior while also providing invaluable insights into how well the agent functions under a diverse array of conditions and scenarios. Such evaluations are critical, as they ultimately enhance the agent's overall effectiveness and adaptability in dynamic situations.

There exists a wide range of reinforcement learning (RL) software that offers synthetic environments tailored for various purposes, making them instrumental in RL performance evaluation. Notable examples include Android Env [88] and MuJoCo [87], both of which are widely utilized for simulating physical systems and robotic tasks. These platforms allow researchers to create intricate simulations that reflect real-world physics, thereby providing agents with the opportunity to learn and adapt in environments that closely resemble their actual operating conditions. Additionally, OpenAI [14] and TFAgents [26] provide robust capabilities for developing customized environments, alongside sophisticated implementations

of numerous popular environments. These tools enable researchers to tailor simulations to their specific research needs, facilitating a more nuanced analysis of agent behaviors across different contexts.

5.2.4 Benchmarking

Benchmarking is an indispensable component of RL evaluation, providing a standardized framework to compare the performance of different algorithms and agents in an effective manner. By establishing common evaluation criteria and utilizing diverse datasets, benchmarking allows researchers and practitioners to systematically assess the progress of RL research over time. This process is crucial for identifying specific areas where improvements are needed, ensuring that advancements in the field are both measurable and reproducible. Furthermore, benchmarking fosters innovation by creating a competitive landscape where new algorithms can be tested against established ones, driving further developments in RL applications across various domains, such as robotics, finance, healthcare, and autonomous systems. This structured approach not only enhances collaboration among researchers but also accelerates the pace of discovery and application of reinforcement learning technologies in real-world scenarios.

The key benefits of benchmarking in the realm of reinforcement learning (RL) are numerous and significant. These advantages include a standardized environment, which ensures consistency in testing conditions and allows for fair comparisons between different algorithms. Additionally, standardized evaluation metrics are rucial, as they provide objective criteria for assessing the performance of various models. Baseline algorithms, which are typically incorporated into mainstream RL software, serve as reference points, enabling researchers to gauge the effectiveness of their new approaches against well-established methods. Moreover, sophisticated RL software offers flexible environment setups, allowing for tailored configurations that can accommodate a wide range of experimental requirements. These factors collectively streamline the process of performance comparison with existing popular algorithms, fostering a more structured and efficient research environment.

In the current landscape of reinforcement learning, multiple popular benchmark suites are available to researchers and practitioners alike. However, it is essential to note that, owing to the present status of RL development, the environments available in these suites are frequently simple or simplified models of real-world scenarios. This limitation highlights the ongoing need for more complex and representative environments that can truly challenge RL algorithms and reflect the intricacies of real-world applications.

- **OpenAI Gym Benchmarking**: Among the most recognized benchmarking suites is the OpenAI Gym Benchmarking framework. This comprehensive collection includes a diverse array of standardized benchmarks specifically designed for evaluating performance across various RL tasks. By providing a consistent plat-

form, OpenAI Gym facilitates meaningful comparisons and aids researchers in developing more effective RL algorithms.

- **DeepMind Control Suite**: Another noteworthy suite is the DeepMind Control Suite Benchmark, which is tailored specifically for continuous control tasks. This advanced benchmark offers a range of challenging environments that enable researchers to assess algorithmic performance in dynamic settings, pushing the boundaries of what RL algorithms can achieve.
- **Roboschool**: Roboschool stands out as a sophisticated physics-based simulator developed for benchmarking robotic control algorithms. It provides realistic environments that allow for rigorous testing and evaluation of robotic learning approaches, ensuring that algorithms can be effectively trained and assessed in conditions that closely mimic real-world challenges.
- **Meta-World**: Lastly, the Meta-World benchmark is specialized for meta-reinforcement learning, placing significant emphasis on the importance of generalization and adaptation across different tasks. This focus is crucial for developing intelligent agents capable of learning efficiently in diverse scenarios, ultimately advancing the field of RL and its applications.

5.3 Online Performance Evaluation

Online performance evaluation of RL systems plays a crucial role in assessing an agent's performance as it interacts with its environment in real-time. This dynamic evaluation approach offers valuable insights into the agent's behavior, learning progress, and overall effectiveness in achieving designated goals. Key metrics utilized in this evaluation process include reward, success rate, episode length, and learning curve. Each of these metrics serves to quantify the agent's ability to learn and adapt over time.

Several techniques for online evaluation are frequently employed, such as real-time monitoring, A/B testing, and human evaluation. Real-time monitoring allows for the continuous tracking of an agent's performance during training, enabling adjustments to the learning process as necessary. A/B testing serves to compare the real-time performance of different RL agents or algorithms, helping to identify which approaches yield superior results. Human evaluation, on the other hand, involves expert assessments of the agent's performance, providing qualitative feedback that can enhance the learning process.

However, online performance evaluation of RL systems comes with its own set of unique challenges. The exploration-exploitation trade-off is a prominent concern; RL agents must balance the need to explore new actions with the desire to exploit known successful actions. While online evaluation can shed light on how effectively an agent manages this balance, ensuring that the exploration-exploitation dynamics align between training and testing phases is paramount for obtaining an accurate assessment.

Moreover, online evaluation facilitates immediate feedback on the agent's performance, allowing for timely adjustments to the learning process. However, this responsiveness often incurs the drawback of delayed agent behaviors in the primary procedure, which can complicate the evaluation process. Lastly, it is important to note that online evaluation can be computationally expensive, particularly in complex environments or large-scale RL problems, potentially limiting its practicality in certain scenarios. Balancing these various factors is essential for optimizing the evaluation process and ensuring effective training outcomes.

5.3.1 Real-time Monitoring

Real-time monitoring is an indispensable component of Reinforcement Learning (RL) that entails the continuous tracking and analysis of an agent's performance as it engages with its environment. This ongoing observation is crucial because it allows for timely interventions, thereby facilitating necessary adjustments to the learning process. Moreover, real-time monitoring plays a pivotal role in identifying potential issues at an early stage, which can significantly enhance the efficiency and effectiveness of the training process.

In essence, real-time monitoring consists of five fundamental components: performance metrics, visualization, logging, alerts, and debugging tools. Performance metrics are vital for tracking essential indicators of the agent's learning journey, including reward, success rate, episode length, and learning curve. These metrics offer insights into how well the agent is progressing toward its goals. Visualization complements this by employing various graphical representations—such as plots, graphs, and animations—to deliver a clear and intuitive understanding of the agent's behavior and decision-making processes. This visual feedback is key for both researchers and practitioners, as it can reveal patterns and anomalies that may not be immediately obvious through raw data alone.

Logging serves a critical function by systematically recording relevant data points throughout the training process. This data can be invaluable for later analysis and debugging, allowing researchers to trace back through the agent's actions and decisions. Alerts are also an important feature; they enable the setting up of notifications for significant events or deviations from expected behavior, ensuring that any anomalies are promptly addressed. Meanwhile, debugging tools are utilized to inspect the agent's state, actions, and rewards at various points in time, which can aid in diagnosing issues and refining the learning algorithm.

Common techniques employed in real-time monitoring include TensorBoard, custom dashboards, logging frameworks, and remote access capabilities. TensorBoard is a widely-used visualization tool that helps track various performance metrics and offers visualizations of neural network models. Custom dashboards can be created using libraries like Plotly or Matplotlib, enabling the visualization of specific metrics tailored to particular needs or research questions. Logging frameworks, such as Python's built-in logging module or specialized RL logging libraries, provide

robust solutions for recording data for future analysis. Furthermore, remote access to the training environment is increasingly being established, allowing practitioners to monitor the agent's performance from virtually anywhere, thereby enhancing flexibility and responsiveness during the training process. Overall, the integration of these techniques significantly enriches the monitoring capabilities in RL, paving the way for more informed decision-making and streamlined training workflows.

Real-time monitoring has emerged as a critical component in the landscape of online performance evaluation methods, particularly within the realm of reinforcement learning (RL). This approach offers a myriad of benefits that can significantly enhance the efficiency and effectiveness of learning algorithms. One of the foremost advantages is early detection of issues. By continuously monitoring the performance of the RL agent, one can swiftly identify problems such as divergence, instability, or suboptimal performance before they escalate into more significant challenges that could hinder the learning process. This proactive approach allows for timely interventions, which can prevent prolonged periods of inefficient learning.

Another notable benefit is improved learning efficiency. With access to real-time feedback, researchers and practitioners can make necessary adjustments to the learning process as it unfolds. This adaptability leads to faster convergence rates and ultimately better performance of the RL agent. By analyzing real-time data, it becomes possible to fine-tune hyperparameters, modify reward structures, or even alter the learning environment to facilitate optimal learning conditions.

Furthermore, real-time monitoring fosters enhanced understanding of the agent's behavior and decision-making processes. By observing how the agent interacts with its environment in real-time, developers can gain insights into its strategies, strengths, and weaknesses. This understanding is crucial for refining the agent's algorithms and improving its overall performance.

Facilitated debugging is another significant benefit of real-time monitoring. By employing monitoring tools effectively, one can quickly pinpoint and resolve issues in the agent's performance. This capability not only streamlines the debugging process but also reduces the time and effort required to identify the root causes of performance anomalies.

To implement a robust real-time monitoring module, it is essential to select relevant metrics that are specifically tailored to the reinforcement learning task at hand. Not all metrics are equally informative; therefore, a careful selection process is vital. Additionally, the visualization of these metrics should be designed to ensure clarity and ease of interpretation, allowing users to comprehend performance trends at a glance.

Configuring alerts for critical events or deviations from expected behavior is also imperative. This ensures that any abnormal performance is promptly flagged, enabling quick diagnosis and remediation. Moreover, logging important data in a manner that is informative and accessible facilitates thorough debugging and analysis. Logged data serves as a valuable feedback loop that can be utilized for ongoing performance improvements.

Finally, leveraging debugging tools that allow for human intervention and investigation is invaluable. These tools can provide insights into the real-time performance

of the system, enabling a more hands-on approach to inspecting the agent's state and behavior. By integrating these strategies into the real-time monitoring framework, one can significantly enhance the overall performance evaluation and learning process of reinforcement learning agents.

5.3.2 A/B Testing

A/B testing, also known as split testing, is a powerful technique extensively utilized in the field of reinforcement learning (RL) to compare and evaluate the performance of different RL agents or algorithms. By simultaneously running multiple agents within the same environment, researchers and practitioners can systematically assess their effectiveness and efficacy in tackling specific tasks or challenges. This method provides invaluable insights and helps in identifying the most effective approach for any given application, thus paving the way for more efficient and optimized solutions.

The process of A/B testing in reinforcement learning typically encompasses five critical steps: defining the experiment, setting up the environment, running the agents, collecting data, and analyzing the results. The first step, defining the experiment, is crucial as it establishes a clear and concise framework for the entire testing process. This includes specifying the primary objectives of the experiment, detailing the various RL algorithms or agents that will be compared, and determining the performance metrics that will be utilized to evaluate each agent's effectiveness.

Once the experiment is defined, the next step involves setting up the environment. A controlled and consistent environment is essential to ensure that both agents can interact and learn under the same conditions. This consistency is vital for maintaining the integrity of the comparison, as variations in the environment could lead to skewed results. After the environment is established, the agents are run, allowing them to train and interact with the environment for a predetermined duration. This phase is critical, as it provides the agents with the opportunity to learn from their experiences and adapt their strategies accordingly.

During or following the testing phase, data collection becomes a focal point. Performance metrics such as reward, success rate, episode length, and learning curve are gathered meticulously. This data serves as the foundation for the next step: analyzing the results. In this phase, statistical analysis techniques are employed to compare the performance of the agents based on the collected data, helping to determine whether the observed differences are statistically significant.

The final step involves iterating on the experiment. Based on the results obtained, modifications can be made to the agents or algorithms, and the testing process can be repeated to refine the approach further. This iterative nature of A/B testing is highly beneficial, as it allows for continuous improvement and optimization of the RL agents.

Through the implementation of A/B testing, practitioners can accurately identify the most effective RL algorithm or agent for specific tasks. This method offers a

level of performance evaluation that is often unattainable through offline assessments, which are inherently limited due to their restricted access to real-time, online data that is crucial for serving customers effectively in live environments. Beyond mere performance evaluation, A/B testing significantly mitigates the risk associated with deploying underperforming agents, as it allows for the thorough evaluation of multiple options before making a final decision.

Moreover, A/B testing empowers informed decision-making regarding the design and implementation of RL agents, backed by empirical evidence gathered during the testing process. The insights gained from iterative A/B testing not only accelerate the development process but also provide clarity on what strategies and methodologies yield the best results. By effectively leveraging A/B testing, organizations can glean valuable insights into the performance of their RL agents, enabling data-driven decisions that optimize both their development and deployment in real-world scenarios. Ultimately, this leads to more robust and effective reinforcement learning solutions that align with organizational goals and user needs.

Obtaining effective performance evaluation from A/B testing presents a variety of challenges and considerations that must be carefully navigated to ensure reliable results. Firstly, it is crucial to ensure that the differences in performance observed between the agents are statistically significant. This can be accomplished through methods such as hypothesis testing or the calculation of confidence intervals, which provide a structured framework for assessing the validity of the observed differences. The significance of the results is paramount, as it determines whether the observed performance variations are due to the interventions being tested or merely the result of random chance.

In addition to establishing statistical significance, it is also essential to determine the appropriate duration for the experiment. The required length of the A/B test can vary widely based on factors such as the complexity of the task being evaluated and the desired level of confidence in the results. Longer experiments may be necessary for more complex tasks, where variations in performance may take longer to manifest and stabilize. This consideration is vital because running an experiment for too short a period can lead to inconclusive or misleading results.

Furthermore, A/B testing must be conducted long enough to gather sufficient data, ensuring that the comparisons made are both reliable and accurate. Randomization plays a critical role in this process, as it helps to mitigate biases that could skew the results. By randomly assigning agents to different environments or starting conditions, researchers can minimize the influence of external factors that may otherwise confound the results.

Control over influential variables is also an important component of effective performance evaluation. A/B testing should include protocols for controlling variables that could potentially impact performance metrics, such as random seeds, hardware differences, and environmental conditions. By systematically managing these variables, researchers can enhance the quality of the metrics being evaluated.

Lastly, if the agents are being evaluated in a real-world setting, the design and implementation of the A/B testing framework must take into account relevant ethical implications and potential risks. This includes considering the potential impact on

end-users and ensuring that the testing process does not inadvertently harm any individuals or communities involved. By addressing these challenges and considerations, researchers can conduct A/B tests that yield meaningful insights into performance and guide future decision-making effectively.

5.3.3 Testing Interleaving

Testing interleaving in reinforcement learning is a vital process that entails evaluating the effectiveness of this technique in enhancing the agent's performance and its ability to generalize across various tasks. This approach is particularly significant in scenarios where agents are required to adapt to changing environments or diverse challenges. Interleaving, which involves alternating between different tasks during training, aims to improve learning efficiency and outcomes. While some of the testing interleaving techniques share similarities with A/B testing, it is crucial to recognize that the primary objectives of these two online performance evaluation methods differ significantly. A/B testing typically focuses on comparing two distinct versions of a system or method to determine which performs better, while interleaving emphasizes optimizing learning and performance across multiple tasks.

To elucidate the effectiveness of interleaving in reinforcement learning, we can identify four common approaches to testing this technique:

- Direct Comparison : This method involves a straightforward assessment where:
  - Control Group : A baseline agent is trained without any interleaving on the same tasks or environments, serving as a reference point.
  - Experimental Group : An agent is trained with interleaving, also operating within the same tasks or environments.
  - Performance Metrics : The performance of both agents is compared using relevant metrics such as total reward accumulated, success rates across tasks, episode lengths, and learning curves, providing a comprehensive understanding of the impact of interleaving.
- Generalization Testing : This approach focuses on evaluating the agent's adaptability:
  - Novel Tasks : The agent's ability to excel in new tasks that were not part of its training set is examined.
  - Transfer Learning : The effectiveness of transferring knowledge from previously learned tasks to unfamiliar tasks is assessed.
  - Domain Randomization : This involves testing the robustness of the agent by introducing variations in the environment through random adjustments to task parameters or conditions.
- Ablation Studies : These studies investigate the specific elements contributing to performance outcomes:

  - Varying Interleaving Frequency : Different frequencies of task switching are experimented with to discover the optimal interleaving rate for performance.
  - Task Similarity : The impact of the similarity between tasks on the effectiveness of interleaving is evaluated.
  - Curriculum Learning : The performance of interleaving is compared against curriculum learning, where tasks are introduced gradually in increasing levels of difficulty, providing insights into their comparative advantages.

- Visualization : This technique involves graphical representation to gain insights:

  - Learning Curves : Learning curves for both the interleaved and non-interleaved agents are plotted to visually compare their convergence rates over time.
  - Policy Visualization : The learned policies of the agents are visualized to understand how the interleaving process influences their structure and generalization capabilities.
  - Task Representation : An analysis is conducted to observe how interleaving impacts the agent's representation of different tasks, providing a deeper understanding of the learning dynamics at play.

Through these testing approaches, researchers and practitioners can gain valuable insights into the effectiveness of interleaving in reinforcement learning, guiding the development of more robust and efficient learning algorithms.

To design and implement Testing Interleaving effectively and efficiently, several important considerations must be taken into account: statistical significance, experimental design, data collection, and ethical considerations. Firstly, we need to ensure that the observed differences in performance between the interleaved and non-interleaved agents are statistically significant. This involves employing robust statistical methods such as hypothesis testing and confidence intervals, which are frequently used to assess significance and determine whether the observed effects are likely due to chance or represent true differences in performance.

Moreover, the experiments should be meticulously designed to control for various variables that could potentially affect performance outcomes, including random seeds or hardware differences. It is crucial to create a balanced experimental framework that minimizes bias and ensures that all conditions are comparable. This might involve randomizing the assignment of agents to different conditions, ensuring that environmental factors remain constant, and carefully documenting all aspects of the experimental setup.

Furthermore, the volume and quality of data collected are also paramount. A sufficiently large dataset is essential to ensure reliable and accurate comparisons between the various testing conditions. Data sufficiency not only increases the power of statistical tests but also enhances the credibility of the findings. It is advisable to conduct a power analysis beforehand to determine the sample size needed to detect meaningful differences.

Finally, ethical considerations become increasingly critical when the agents are being evaluated in a real-world setting. In such cases, it is essential to weigh the ethical implications and potential risks associated with the testing process, especially

for large-scale or long-term implementations. This may include considering the impact on users and stakeholders, ensuring transparency, and safeguarding the rights and well-being of those affected by the agents' actions.

By conducting thorough testing and analysis, we can gain invaluable insights into the effectiveness of interleaving in reinforcement learning applications. This enables researchers and practitioners to make informed decisions about its implementation, ultimately leading to more effective algorithms and improved performance in real-world applications. In summary, a well-rounded approach that includes statistical rigor, careful experimental design, comprehensive data collection, and ethical mindfulness will facilitate the successful implementation of Testing Interleaving and contribute to advancements in the field.

5.4 Performance Emulator

A performance simulator or emulator plays a crucial role in the field of performance evaluation, particularly in the context of reinforcement learning (RL). These sophisticated tools create virtual environments that closely mimic real-world scenarios, allowing researchers and developers to analyze and assess the behavior of RL agents without engaging in potentially risky or costly real-world interactions. This capability is especially beneficial in situations where the real environment is either too intricate, expensive, or hazardous to experiment with directly. For instance, in applications such as autonomous driving, aerospace exploration, or robotic surgery, the stakes can be incredibly high, making simulators an invaluable resource for testing and validation.

Unlike offline performance evaluation methods that may generate trajectories solely for the purpose of testing or evaluation, the design and implementation of performance emulators often demand a higher level of complexity and effort. These emulators are typically utilized for online performance evaluation, meaning they can provide immediate feedback and insights as the agent interacts with the simulated environment, thus facilitating a more dynamic learning process.

The key components that constitute a performance emulator include the environment model, agent simulator, and performance metrics. The environment model serves as a comprehensive representation of the real-world environment, encapsulating its dynamics, states, actions, and rewards. This model is essential for accurately recreating the conditions under which the RL agent will operate. The agent simulator, on the other hand, functions as a platform that enables the execution of the RL agent's policy within the confines of the simulated environment, effectively allowing the agent to learn and adapt in real-time. Additionally, performance metrics are critical, as they provide a standardized set of measurements to evaluate the agent's performance, including but not limited to reward accumulation, success rate, episode length, and learning curve.

In general, performance emulators can be categorized into three distinct types: physics-based emulators, rule-based emulators, and data-driven emulators. Physics-

based emulators leverage advanced physics engines to replicate the dynamics of real-world objects and environments accurately. Rule-based emulators, in contrast, rely on predefined rules or models to simulate environmental behavior, offering a more abstracted approach. Lastly, data-driven emulators utilize historical data or machine learning techniques to learn and predict the dynamics of the environment, enabling a more adaptive and responsive simulation experience. Each type of emulator has its unique strengths and applications, making them invaluable tools in the realm of performance evaluation and reinforcement learning.

The advantages of utilizing a performance emulator over traditional offline and online performance evaluation methods are numerous, encompassing aspects such as cost-effectiveness, safety, controllability, and scalability. To begin with, emulators can dramatically reduce the costs associated with experimentation. This is achieved by eliminating the necessity for physical hardware or real-world resources, which can be prohibitively expensive. Instead of investing in costly equipment or facilities, researchers can leverage virtual environments to conduct experiments, thereby maximizing their research budgets.

Moreover, performance emulators provide a critical safety advantage. They enable the testing of agents in dangerous or risky scenarios without the potential for harm to humans or damage to expensive equipment. In the context of robotics, for example, engineers can simulate environments that involve hazardous materials or unpredictable conditions, allowing for extensive testing without real-world consequences. This level of safety ensures that the development process can proceed without the associated risks that traditional methods may entail.

Another significant benefit of performance emulators is the precise control they offer over experimental environments. This level of control facilitates the isolation of variables, making it much easier to conduct controlled experiments and draw meaningful conclusions from the data generated. By adjusting specific parameters within the emulator, researchers can observe how changes affect the performance of their agents, leading to a more thorough understanding of the underlying dynamics involved.

Scalability is yet another key advantage of using performance emulators. These tools can be designed to simulate large-scale environments or to accommodate multiple agents simultaneously, something that is often impractical when relying on offline performance evaluation methods. Additionally, utilizing online performance evaluation methods in such scenarios can result in significant resource consumption, making it both cost-inefficient and logistically challenging.

Performance emulators find their applications across various domains within reinforcement learning, but they are particularly beneficial in areas where accurate simulations of the reinforcement learning system are feasible and significantly more cost-effective than conducting experiments in real-world systems. Notable examples of these application areas include robotics, autonomous vehicles, game AI, and healthcare.

In the robotics industry, for instance, simulation environments are employed to replicate the behavior of robots in diverse settings, allowing for thorough testing of their control algorithms and safety measures. Autonomous vehicle testing, on the

other hand, leverages emulators to evaluate the performance of self-driving cars in a multitude of driving scenarios, all while eliminating the risk to human lives. The gaming industry also benefits from these tools, as game AI emulators are utilized to develop and test AI agents within simulated environments, ensuring that they can effectively interact with players and adapt to dynamic game conditions.

In the healthcare sector, performance emulators simulate medical procedures or patient interactions to train AI agents for various healthcare tasks. By using these emulators, researchers and engineers can efficiently evaluate the performance of their agents in a controlled and scalable environment. This accelerates the development of intelligent systems, ultimately leading to innovations that can enhance safety, efficiency, and effectiveness across multiple fields. As a result, the deployment of performance emulators represents a transformative approach in the realm of reinforcement learning, paving the way for advancements that were previously constrained by the limitations of traditional evaluation methods.

Fig. 5.13 presents a comprehensive overview of example performance simulation results derived from various game emulators. In this analysis, each game showcases a diverse array of learning rates, all generated from random initializations. This variation highlights the complexity and unpredictability involved in performance evaluations. Conducting such evaluations through traditional offline and online performance assessment methods would be not only impractical but also, at best, cost-inefficient when considering the extensive resources required. In contrast, utilizing simulators or emulators for performance evaluation offers significant advantages. These platforms enable the acquisition of more smoothed and stable performance metrics, owing to their inherent flexibility in parameter adjustments. This flexibility allows researchers and developers to fine-tune various aspects of the simulation environment, leading to more reliable outcomes. Furthermore, the cost-efficiency of simulators can facilitate broader testing scenarios, ultimately providing deeper insights into system performance while conserving valuable resources.

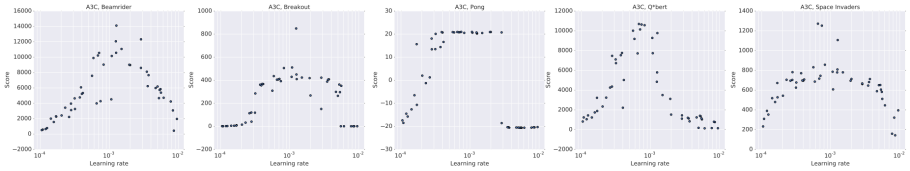


Fig. 5.13: Sample plots of scores obtained by A3C through game emulators of five games (Beamrider, Breakout, Pong, Q*bert, Space Invaders) for 50 different learning rates and random initialization. The results demonstrate the robustness of A3C to different learning rates and initial random weights [57].

5.5 Challenges and Best Practices

Performance evaluation in reinforcement learning (RL) is a complex task that presents numerous challenges. Below, we will elaborate on three critical challenges that researchers and practitioners often encounter when assessing the performance of RL agents.

- Non-Stationarity: One of the most significant challenges in performance evaluation is the non-stationarity of RL environments. Non-stationarity implies that the dynamics of the environment can evolve over time, which may be due to various factors such as changes in the underlying system, the introduction of new agents, or even shifts in user behavior. This variability can lead to inconsistencies in evaluating an agent's performance, as what may have been an effective strategy at one point in time may no longer yield the same results later. Consequently, evaluating an agent's performance becomes increasingly complex, necessitating adaptive evaluation techniques that can account for these dynamic changes.
- Stochasticity: Another fundamental challenge is the stochastic nature of many RL environments. In stochastic settings, the results of actions taken by an agent can vary widely, even when the same action is repeated under identical conditions. This randomness makes it inherently difficult to compare the performance of different agents, as an agent's success may be heavily influenced by chance rather than skill. To overcome this challenge, researchers often employ statistical methods to aggregate results over multiple episodes or trials, allowing for a more reliable assessment of an agent's true performance capabilities.
- Multiple Objectives: In certain applications, RL agents may need to juggle multiple objectives simultaneously, such as maximizing rewards while minimizing risks or resource consumption. This multiplicity of goals complicates the evaluation process, as it becomes challenging to define a single performance metric that adequately captures the agent's effectiveness across all objectives. The trade-offs between objectives can also lead to conflicting strategies, making the evaluation of agent performance even more nuanced.

To effectively conduct performance evaluation, several best practices should be adhered to.

- Experiment Design: It is crucial to design experiments meticulously to ensure that they are fair and unbiased. This includes controlling for extraneous variables and ensuring a representative sampling of conditions.
- Baseline Comparison: Comparing the performance of the RL agent to a baseline, such as a random policy or a simple heuristic, provides a point of reference that can help contextualize the agent's performance. This comparison allows for a clearer understanding of how well the agent performs relative to basic strategies.
- Statistical Significance: Employing statistical tests is essential to determine whether the observed differences in performance are statistically significant. This step helps ensure that conclusions drawn from the evaluation are not merely the result of random fluctuations.

- Visualization: Visual tools are invaluable for understanding the agent's behavior over time. By employing graphs and charts, researchers can identify trends, patterns, and areas for potential improvement, making the evaluation process more intuitive.
- Ablation Studies: Conducting ablation studies enables researchers to evaluate the impact of different components of the RL algorithm systematically. By removing or altering specific elements, one can assess how these changes affect overall performance, leading to a deeper understanding of the algorithm's strengths and weaknesses.

In conclusion, addressing the challenges of non-stationarity, stochasticity, and multiple objectives is crucial for effective performance evaluation in reinforcement learning. By adhering to best practices, researchers can obtain meaningful insights that guide the development of more robust and effective RL agents.

5.6 System-wise Performance Evaluation

System-wise reinforcement learning (RL) evaluation encompasses a variety of critical components that play a pivotal role in ensuring the robustness and effectiveness of RL systems. These components include performance evaluation, recovery testing, security evaluation, stress testing, and deployment testing. Each of these elements contributes to a comprehensive assessment of how well an RL system operates in realistic environments, where various challenges and constraints are present.

Performance evaluation focuses primarily on assessing the run-time performance of RL systems, examining how well they function under different operating conditions, such as varying levels of resource availability, environmental complexities, and user interactions. This aspect is crucial because it provides insights into the agent's efficiency, responsiveness, and overall capability to achieve its objectives in dynamic settings. Performance evaluation not only looks at the accuracy of the agent's decisions but also considers factors such as latency, throughput, and resource consumption. By analyzing these metrics, developers can identify potential bottlenecks and areas for improvement.

Recovery testing is equally important, as it verifies the RL system's ability to recover from failures or disruptions. This testing ensures that the system can maintain resilience in its operation, even in adverse conditions such as hardware malfunctions or unexpected input scenarios. A robust recovery mechanism is essential for maintaining user trust and system reliability, particularly in critical applications where downtime can have significant consequences.

Security evaluation examines the effectiveness of the system's protection mechanisms. This includes assessing its ability to prevent unauthorized access, penetration, or manipulation by malicious entities. Given the increasing prevalence of cyber threats, ensuring the integrity and security of RL systems is paramount. A thorough security evaluation can help identify vulnerabilities and reinforce the system's defenses against potential attacks.

Stress testing plays a vital role in evaluating how well the system copes with abnormal resource demands. This evaluation is particularly relevant when the number of active agents becomes significantly large, which can lead to potential performance bottlenecks and impact overall system functionality. By simulating extreme conditions, developers can better understand the limits of their systems and implement necessary optimizations.

Deployment testing assesses the system's capability to upgrade its software and hardware resources, ensuring that it remains adaptable to changing requirements over time. As technology evolves and user needs shift, the ability to seamlessly integrate updates is critical for maintaining system relevance and effectiveness.

While all these aspects are integral to a comprehensive system-wise evaluation of RL systems, it is beneficial to narrow our focus to emphasize system-wise performance evaluation within the context of RL foundations. Performance evaluation involves a thorough and systematic assessment of the RL agent's effectiveness within a larger system architecture. This evaluation considers integration with existing systems, scalability to accommodate growth, and the myriad challenges associated with real-world deployment scenarios. Such a meticulous evaluation approach is vital for gaining meaningful insights into the practical implications of RL technology. It ensures that RL agents can be effectively utilized in real-world scenarios, where conditions tend to be more complex and variable than those encountered in controlled environments. By concentrating on performance evaluation, we can better understand how to improve RL systems to meet the demands of practical applications, thereby maximizing their effectiveness and reliability. Ultimately, a well-rounded evaluation framework that prioritizes performance will help bridge the gap between theoretical research and practical application, driving advancements in the field of reinforcement learning.

Key considerations for system-wise performance evaluation encompass a range of critical factors that can influence the overall success of a reinforcement learning (RL) deployment. These factors include not only the technical aspects of integration and scalability but also the inherent challenges that come with real-world applications. When deploying RL systems, organizations must navigate the complexities of integrating these advanced algorithms into existing infrastructures, ensuring that they can scale efficiently as demands grow. Moreover, the safety and security concerns associated with these deployments must be meticulously addressed to preemptively identify any risks tied to the RL agent's behavior. This includes potential vulnerabilities that could be exploited by malicious actors or unintended consequences arising from the agent's decision-making processes.

Ethical implications also play a significant role in the evaluation process; understanding issues such as bias and discrimination is essential in ensuring that the technology is applied responsibly. Deploying an RL agent without considering the potential for biased outcomes can lead to significant social and economic repercussions. Consequently, organizations must incorporate fairness and transparency into their evaluation frameworks to mitigate these risks. Lastly, a thorough cost-benefit analysis is key to determining whether the advantages of deploying the RL agent justify the associated costs and resource investments. This analysis should encom-

pass not only direct costs but also indirect costs related to potential ethical and reputational risks.

In summary, a holistic approach to evaluating RL agents within the framework of larger systems is necessary to ensure their effective application in real-world settings. This multi-faceted evaluation process aids in identifying potential pitfalls and optimizing the design and implementation of RL solutions for various applications.

In addition to the common performance evaluation metrics of RL algorithms, system-wide evaluation emphasizes unique metrics that may not be readily accessible through stand-alone or algorithm-specific performance evaluations. These distinct metrics are crucial in understanding the holistic impact of the RL agent on the overall system. The main metrics for system-wise evaluation encompass several key aspects: incremental system performance, cost-effectiveness, user experience, reliability, and adaptability. By focusing on these unique metrics, organizations can gain deeper insights into how the RL agent interacts with other components of the system, thereby fostering a more comprehensive understanding of its role and effectiveness in achieving desired outcomes. This broader perspective is essential for continuous improvement and innovation in the deployment of RL technologies.

- Incremental System Performance: This metric assesses how the overall performance of the system improves with the implementation of the Reinforcement Learning (RL) agent. It examines not only the efficiency gains but also how the agent contributes to achieving the system's objectives. A thorough analysis of this metric involves measuring various performance indicators before and after the RL agent's deployment. These indicators may include response times, throughput rates, and accuracy of outcomes. By quantifying these improvements, stakeholders can better understand the tangible benefits of integrating an RL agent into their existing system. Furthermore, it is essential to consider the long-term effects of the RL agent's learning capabilities, as it can continually refine its actions based on feedback, potentially leading to sustained performance enhancements over time.
- Cost-Effectiveness: This evaluation criterion focuses on whether deploying and maintaining the RL agent is cost-effective. It explores the balance between the financial investment required and the benefits derived from enhanced performance. A comprehensive cost analysis should account for initial development and training costs, ongoing maintenance expenses, and any necessary infrastructure changes. Additionally, it is important to weigh these costs against the operational savings generated by improvements in efficiency and productivity. By conducting a thorough cost-benefit analysis, organizations can make informed decisions about the viability of implementing an RL agent, ensuring that the investment aligns with their overall strategic goals and financial constraints.
- User Experience: An essential aspect of any system is its interaction with users. This metric investigates how the presence of the RL agent influences user satisfaction, engagement, and overall experience, considering both positive and negative impacts. User experience can be assessed through qualitative methods such as surveys and interviews, as well as quantitative measures like user retention rates and task completion times. Understanding how users interact with the RL agent

is crucial, as their acceptance and satisfaction can significantly affect the system's success. Additionally, identifying potential areas of friction or misunderstanding can inform further refinements to enhance the user experience and ensure that the RL agent serves its intended purpose effectively.

- Reliability: This metric evaluates how dependable the RL agent is under real-world conditions, considering factors such as consistency of performance and the ability to function correctly over time. Reliability encompasses the agent's capacity to handle various scenarios without failure, which is essential for maintaining user trust and system integrity. Testing the RL agent under diverse conditions, including varying workloads and unforeseen disruptions, can provide insights into its robustness. Establishing reliability benchmarks and continuously monitoring performance can help identify potential issues early, allowing for timely interventions to maintain optimal functionality.
- Adaptability: Finally, adaptability examines the RL agent's capacity to adjust to changes in the environment, including shifting user needs or evolving operational conditions. This flexibility is vital for ensuring long-term success in dynamic settings, thus enhancing the system's robustness. An adaptable RL agent can learn from new data and modify its strategies accordingly, which is particularly important in industries where demands and technologies are constantly evolving. By fostering a culture of continuous learning and improvement, organizations can leverage the adaptability of their RL agents to stay competitive and responsive to market trends, ultimately driving sustained growth and innovation.

System-wise evaluation of reinforcement learning (RL) systems faces a myriad of real-world challenges that significantly complicate the assessment of their performance and effectiveness. One of the primary difficulties arises from the intricate interdependence of various components within complex RL systems. Each component often interacts with others in unpredictable and non-linear ways, resulting in emergent behaviors that are not only difficult to anticipate but also challenging to assess accurately. This inherent complexity is a major reason why traditional evaluation methods, which are often designed to test the performance of RL algorithms in isolation, may not be adequate. These basic evaluation techniques may fail to capture the multifaceted interactions and dynamics involved in real-world applications, leading to misleading conclusions about a system's performance.

Furthermore, real-world environments introduce a plethora of constraints and variables that are frequently absent in controlled simulated settings. Such discrepancies can lead to significant differences between performance evaluations conducted in simulated environments and those undertaken in actual RL systems deployed in the field. For instance, factors such as noise, variability in user behavior, and unforeseen external influences can dramatically alter the performance of an RL system, making it imperative to account for these elements during evaluation. This gap between simulation and reality emphasizes the need for more sophisticated and robust evaluation frameworks that can accommodate the complexities of real-world scenarios.

Additionally, ethical considerations in reinforcement learning evaluation can be exceptionally intricate and require thorough and nuanced attention. In particular,

the evaluation of customer-facing RL systems often necessitates handling sensitive private data related to customers' behaviors and preferences. As a result, privacy preservation emerges as a paramount ethical concern, demanding the implementation of stringent data protection measures and ethical frameworks to ensure that customer data is managed responsibly and transparently. Addressing these ethical challenges is not merely an afterthought but a fundamental aspect of advancing the field of RL. Ensuring that RL systems are evaluated and deployed in a manner that respects privacy and promotes ethical standards is crucial for fostering public trust and acceptance of these technologies, ultimately paving the way for their responsible application in real-world scenarios. By tackling these multifaceted challenges, researchers and practitioners can make significant strides in enhancing the reliability and ethical integrity of reinforcement learning systems.

To design and implement effective and efficient system-wise evaluation procedures, best practices include adopting a holistic approach that considers all aspects of the system, conducting real-world testing that simulates actual usage scenarios, performing comprehensive ethical assessments to ensure that the system adheres to moral standards, engaging in continuous monitoring to track performance over time, and actively seeking user feedback to gather insights that can inform necessary improvements. By integrating these practices, organizations can ensure that their evaluation processes are robust, responsive, and aligned with the needs of users while also maintaining high standards of accountability and transparency.

- Holistic Approach: It is essential to consider the entire system, encompassing hardware, software, and human factors. This comprehensive viewpoint ensures that all components interact seamlessly, leading to a better understanding of the system's overall performance. By taking a holistic approach, developers can identify potential bottlenecks or inefficiencies that may arise from the interaction between different components. For instance, hardware limitations can significantly impact the performance of software algorithms, while human factors, such as user interaction and experience, can influence the effectiveness of the system. Therefore, understanding how these elements interrelate is crucial for optimizing design and functionality. Moreover, this approach encourages interdisciplinary collaboration, integrating insights from various fields such as engineering, cognitive science, and user experience design, which can lead to innovative solutions and improved outcomes.
- Real-World Testing: Testing the reinforcement learning (RL) agent in real-world conditions is crucial for accurately assessing its performance. This type of testing helps identify strengths and weaknesses that may not be evident in controlled environments, providing valuable insights into how the agent will behave when deployed. Real-world testing simulates the complexities and unpredictabilities of actual scenarios, enabling developers to evaluate the agent's decision-making capabilities and adaptability. Additionally, it provides an opportunity to observe how the agent interacts with other systems and users, revealing potential issues that could arise in practice. By gathering data from these tests, organizations can

refine the RL agent, enhancing its robustness and reliability, thereby ensuring that it meets the necessary performance standards before widespread deployment.

- Ethical Assessment: Conducting an ethical assessment is vital to identify and address potential risks associated with the RL agent. This process helps ensure that the system aligns with ethical standards and addresses concerns such as bias, privacy, and fairness. By evaluating these factors, organizations can mitigate risks that could lead to negative societal impacts or harm to users. Ethical assessments should involve a diverse team to consider various perspectives and values, ensuring a comprehensive understanding of the implications of the RL agent's actions. Furthermore, engaging with stakeholders, including users and community representatives, can provide insights into ethical concerns that may not have been initially considered, fostering a more inclusive approach to system development.
- Continuous Monitoring: Ongoing monitoring of the RL agent's performance over time allows organizations to identify and address emerging issues proactively. By tracking key performance indicators, organizations can make timely adjustments to improve the agent's effectiveness and reliability. This continuous feedback loop is essential for ensuring that the RL agent adapts to changing environments and user needs. Moreover, monitoring can help detect drift in the agent's performance, which might indicate the need for retraining or algorithm adjustments. By implementing robust monitoring systems, organizations can maintain high standards of performance and accountability, ensuring that the RL agent continues to operate effectively in dynamic conditions.
- User Feedback: Gathering feedback from users is critical for understanding the RL agent's impact on their experience. Engaging with users helps uncover insights that can lead to enhancements in the system, ensuring it meets their needs and expectations effectively. User feedback can take various forms, including surveys, interviews, and usability tests, providing rich qualitative and quantitative data. Analyzing this feedback allows organizations to identify trends and common pain points, facilitating targeted improvements. Furthermore, involving users in the development process fosters a sense of ownership and trust, which can enhance user satisfaction and promote long-term adoption of the RL agent. Ultimately, prioritizing user feedback is a vital step in creating a responsive and user-centered system that delivers meaningful value.

By diligently following these comprehensive guidelines, organizations can effectively evaluate the performance of reinforcement learning (RL) agents within the broader context of their complex systems. This thorough evaluation process ultimately leads to more informed and strategic decisions regarding the deployment and ongoing optimization of these advanced agents. By taking into account various performance metrics and contextual factors, organizations can ensure that they are not only tracking the agents' immediate outputs but also understanding their long-term impact on system efficiency and effectiveness. This multi-faceted approach fosters a deeper understanding of how these agents operate and interact with various elements of the system, including user behavior, environmental variables, and other dynamic components. Such insights are crucial in ensuring the success of RL

agents in real-world applications, where adaptability and responsiveness are key to achieving desired outcomes. Additionally, this systematic evaluation can help identify potential areas for improvement and innovation, ultimately driving continuous enhancement of the agent's capabilities and the overall system performance.