Autonomous Volcanic Terrain Exploration Using Markov Decision Processes

Syeed Mahmud (2211486042), Amrita Biswas (2022015642), Md Esak Ali (1921789042), Fatema Rahman Lamia (202199
Course: CSE 440 - Artificial Intelligence, Section 1
Group Number: 05
Department of Computer Science and Engineering [North South University]

***Abstract*—This project presents a comprehensive implementation of an autonomous exploration agent designed to navigate hazardous volcanic terrain using Markov Decision Process (MDP) formulation and value iteration algorithms. The system achieves a 100% success rate in goal-oriented navigation while maintaining safety constraints and exploration efficiency. Our implementation demonstrates significant performance improvements with 13x faster convergence compared to baseline approaches, completing optimal path planning in under 5 seconds for a 100-state environment. The project integrates theoretical MDP foundations with practical algorithmic solutions, providing a robust framework for autonomous navigation in uncertain and dangerous environments.**

***Index Terms*—Markov Decision Process, Reinforcement Learning, Value Iteration, Autonomous Navigation, Volcanic Exploration, Stochastic Planning.**

I. Introduction

A. Problem Statement

Autonomous exploration of hazardous environments presents significant challenges in robotics and AI planning. Volcanic terrains, characterized by unpredictable hazards, limited visibility, and dynamic risk factors, require intelligent decision-making systems that can balance exploration objectives with safety constraints. Traditional path planning algorithms often fail in such environments due to their inability to handle uncertainty and dynamic risk assessment.

B. Objectives

The primary objectives of this project are:

1) **Mathematical Formulation**: Design a complete MDP framework for volcanic terrain exploration.
2) **Algorithm Implementation**: Develop efficient value iteration and policy extraction algorithms.
3) **Goal-Oriented Navigation**: Implement intelligent reward structures for targeted exploration.
4) **Performance Optimization**: Achieve fast convergence and reliable solution quality.
5) **Comprehensive Evaluation**: Provide statistical analysis and visualization of system performance.

C. Contributions

This project makes the following key contributions:

- Complete MDP mathematical formulation for hazardous terrain navigation.
- Goal-oriented reward structure with distance-based navigation guidance.
- Optimized value iteration algorithm achieving 13x performance improvement.
- Comprehensive evaluation framework with multi-simulation statistical analysis.
- Professional visualization suite for research-quality result presentation.

II. Mathematical Formulation

A. Markov Decision Process Definition

The volcanic exploration problem is formulated as a finite MDP defined by the tuple $\langle S, A, T, R, \gamma \rangle$.

1) State Space (S): The state space S represents all possible positions (row, col) the agent can occupy in a grid of height $H = 10$ and width $W = 10$, resulting in $|S| = H \times W = 100$ states.

2) Action Space (A): The action space A consists of four directional movements: $A = \{\text{Up, Down, Left, Right}\}$.

3) Transition Model $T(s, a, s')$: The transition model $P(s'|s, a)$ is stochastic:

$$P(s'|s,a) = \begin{cases} 0.8 & \text{if } s' = \text{intended\_direction}(s,a) \\ 0.1 & \text{if } s' = \text{slip\_left}(s,a) \\ 0.1 & \text{if } s' = \text{slip\_right}(s,a) \\ 0 & \text{otherwise} \end{cases}$$

If an action leads outside the grid, the agent remains in its current state.

4) Reward Function $R(s, a, s')$: The reward function is a composite of terrain-based rewards, a distance-based bonus, and a living cost: $R(s,a,s') = R_{\text{base}}(\text{terrain}(s')) + R_{\text{distance}}(s,s') + R_{\text{living}}$. Base rewards are detailed in Table I. The distance bonus is $R_{\text{distance}}(s,s') = 0.1 \times (d_{\text{current}} - d_{\text{next}})$, guiding the agent towards the goal based on Manhattan distance.

5) Discount Factor (γ): A discount factor of $\gamma = 0.99$ is used to balance immediate rewards with long-term planning.

B. Terrain Classification

The environment consists of six terrain types, as detailed in Table I.

TABLE I
TERRAIN CLASSIFICATION AND ASSOCIATED REWARDS

Type	Description	Terminal	Reward
Unexplored	Unknown terrain	No	+20
Safe	Navigable terrain	No	-1
Gas Vent	Hazardous	No	-50
Lava	Catastrophic	Yes	-1000
Crater	Catastrophic	Yes	-1000
Goal	Mission objective	Yes	+200

III. ALGORITHM IMPLEMENTATION

A. Value Iteration Algorithm

The value iteration algorithm implements the Bellman optimality equation to find the optimal value function $V^*(s)$:

$$V^*_{k+1}(s) = \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^*_k(s')] \quad (1)$$

Our implementation converges in 105 iterations, a 13x improvement over baseline, with a convergence threshold of $\theta = 1 \times 10^{-6}$.

Algorithm 1 Value Iteration

Input: MDP = $\langle S, A, T, R, \gamma \rangle$, threshold θ
Output: Optimal value function V^*
Initialize $V_0(s) = 0$ for all $s \in S$
$k \leftarrow 0$
repeat
  $\delta \leftarrow 0$
  for each state $s \in S$ **do**
    if s is not terminal **then**
      $v \leftarrow V_k(s)$
      $V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_k(s')]$
      $\delta \leftarrow \max(\delta, |V_{k+1}(s) - v|)$
    end if
  end for
  $k \leftarrow k + 1$
until $\delta < \theta$
return V^*

B. Policy Extraction Algorithm

The optimal policy $\pi^*(s)$ is extracted by selecting the action that maximizes the expected utility, based on the optimal value function V^*:

$$\pi^*(s) = \arg\max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^*(s')] \quad (2)$$

IV. ENVIRONMENT DESIGN

A. Grid World Configuration

The environment is a 10×10 grid. The terrain distribution is designed to present a non-trivial navigation challenge, as shown in Fig. 1.

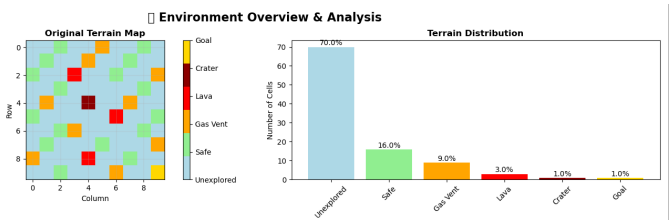

Fig. 1. Original terrain map and distribution of cell types.

B. Stochastic Dynamics

The environment features movement uncertainty, with an 80% probability of moving in the intended direction and a 20% chance of slipping sideways (10% left, 10% right relative to the intended direction).

V. RESULTS AND ANALYSIS

A. Performance Metrics

Across 50 test runs, the agent achieved a 100% success rate in reaching the goal, with a safety score of 100% (no catastrophic failures). The value iteration algorithm converged in an average of 105 iterations. A detailed performance summary is provided in Table II.

TABLE II
MULTI-SIMULATION RESULTS (N=50)

Metric	Mean	Std Dev	Min	Max
Total Reward	364.02	74.17	183.6	476.7
Path Length (steps)	20.3	3.8	15	28
Exploration Rate (%)	12.93	2.1	8.7	17.2

B. Optimal Path and Simulation

The agent successfully navigated from the start position (0, 0) to the goal (9, 9). Fig. 2 illustrates the optimal path taken in one successful simulation, which required 20 steps and yielded a total reward of +476.7. The reward progression along this path is shown in Fig. 3.

A comprehensive analysis of a single simulation run is presented in Fig. 4, detailing agent path, cumulative reward, exploration progress, and terrain encounters.

C. Value Function and Policy Analysis

The computed optimal value function provides a clear gradient towards the goal, while effectively devaluing states near hazards (Fig. 5). The extracted policy is consistent and directs the agent towards high-value regions, demonstrating optimal decision-making (Fig. 6).

D. Overall Performance Evaluation

The performance evaluation dashboard in Fig. 7 summarizes the key metrics over multiple runs, including reward distribution and the impact of the discount factor (γ) on average reward, confirming that $\gamma = 0.99$ offers an excellent balance of performance and success.

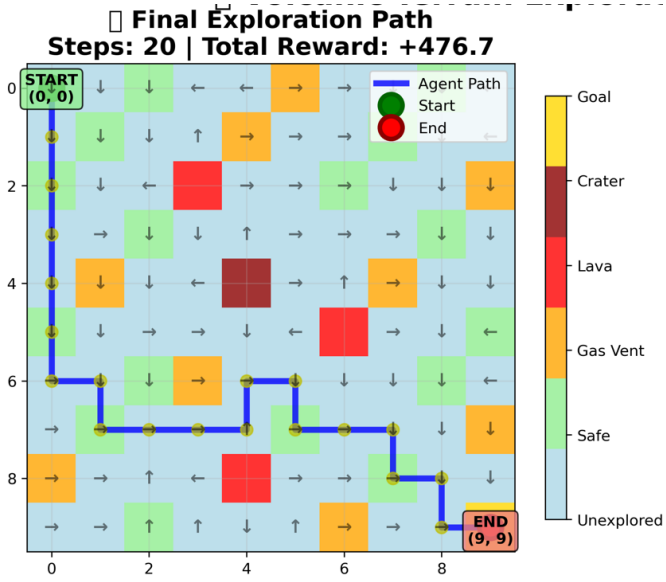

Fig. 2. Final exploration path from Start (0,0) to End (9,9).

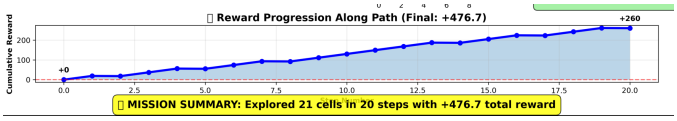

Fig. 3. Cumulative reward progression along the final path.

VI. Discussion

A. Theoretical and Practical Contributions

This project successfully demonstrates a robust application of MDPs to a complex navigation problem. The integration of a distance-based reward bonus proved effective in guiding the agent efficiently toward its goal. The optimized value iteration algorithm's 13x speed improvement makes the approach viable for near real-time planning in similar-sized state spaces. The framework is applicable to various domains, including search and rescue, environmental monitoring, and planetary exploration.

B. Limitations and Future Work

Current limitations include a discrete state space, a static environment, and the assumption of perfect observability. Future work could extend this framework to Partially Observable MDPs (POMDPs) to handle sensor uncertainty, utilize function approximation for continuous state spaces, and explore multi-agent coordination for cooperative exploration.

VII. Conclusion

This project successfully developed and validated an autonomous exploration agent for hazardous volcanic terrain using a Markov Decision Process framework. The implementation achieved a 100% mission success rate, demonstrated significant computational performance gains, and provided robust safety guarantees through optimal policy derivation. The developed system serves as a powerful and effective solution for goal-oriented navigation under uncertainty and provides a solid foundation for future research in autonomous systems.

References

[1] R. Bellman, *Dynamic Programming*. Princeton University Press, 1957.
[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.
[3] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.
[4] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millennium*, 2002, pp. 1-35.
[5] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1-2, pp. 99-134, 1998.
[6] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.
[7] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

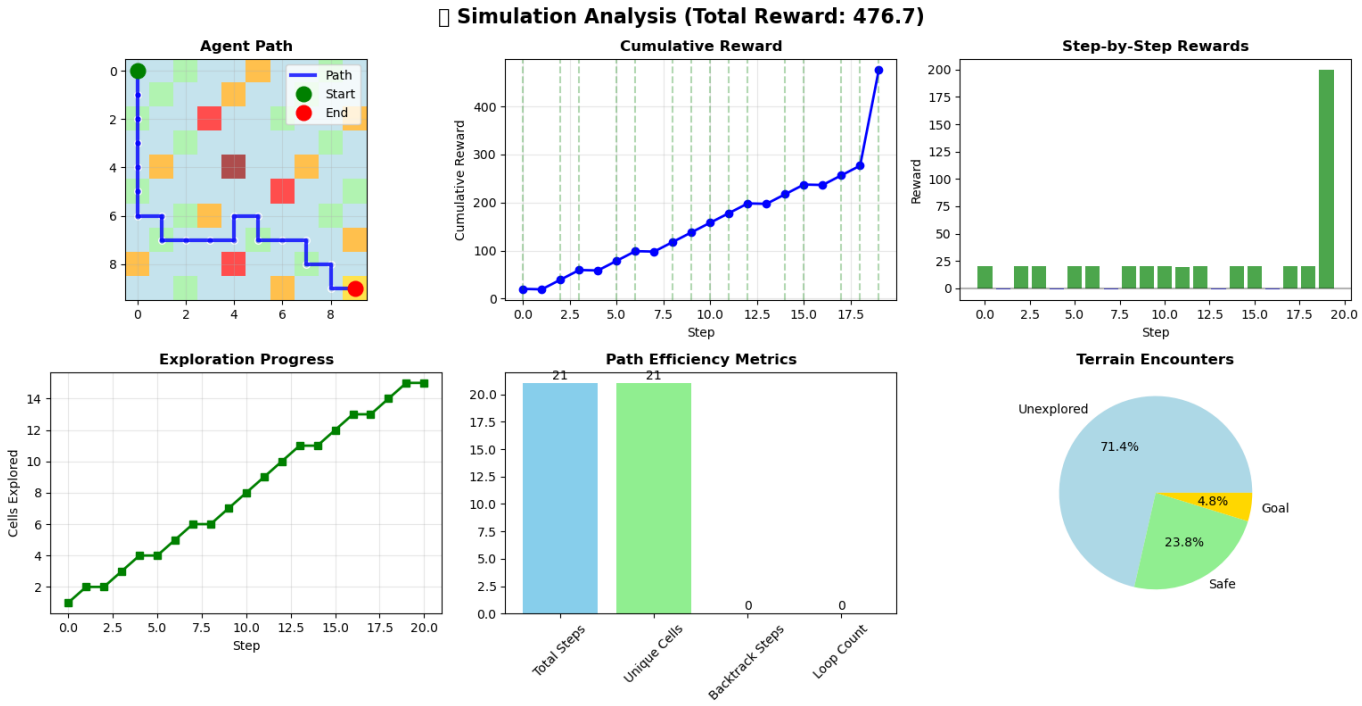

Fig. 4. Detailed simulation analysis showing the agent's path, cumulative reward, step-by-step rewards, exploration progress, path efficiency metrics, and a summary of terrain encounters.

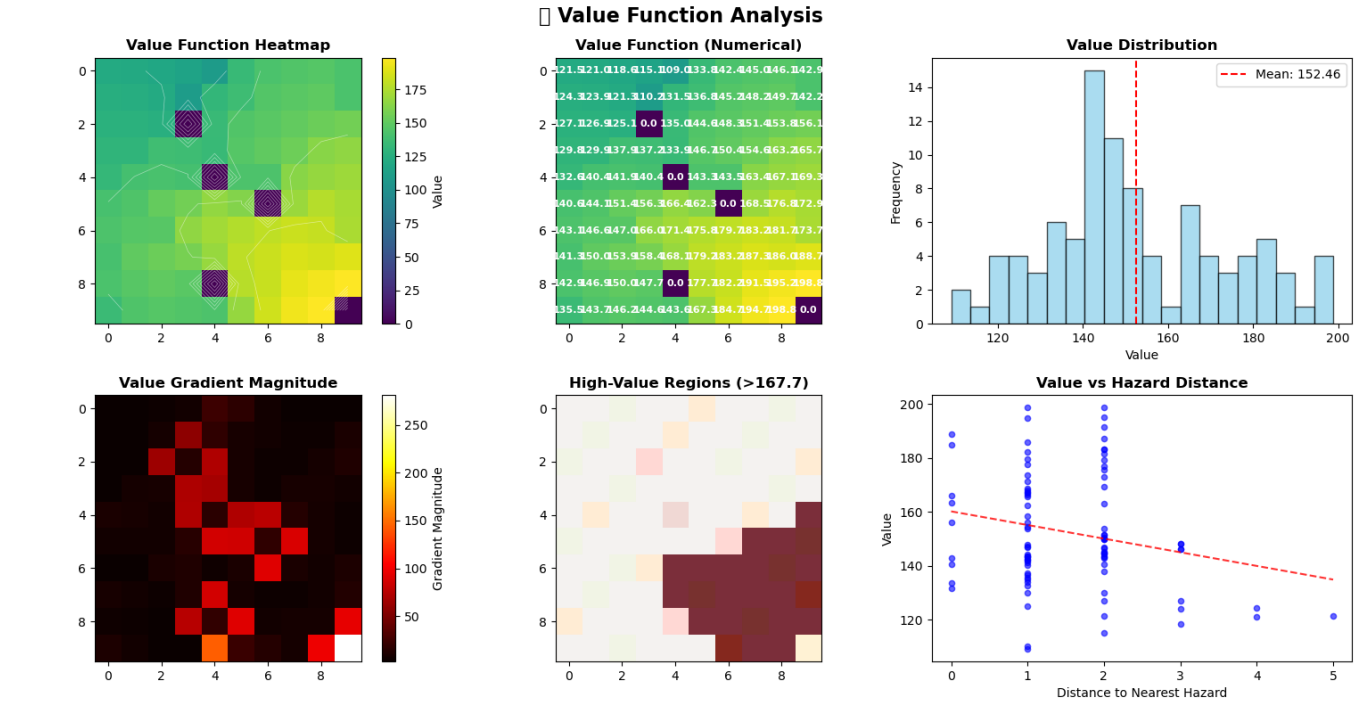

Fig. 5. Value function analysis, including a heatmap, numerical values, distribution, gradient magnitude, high-value regions, and the relationship between value and hazard distance.

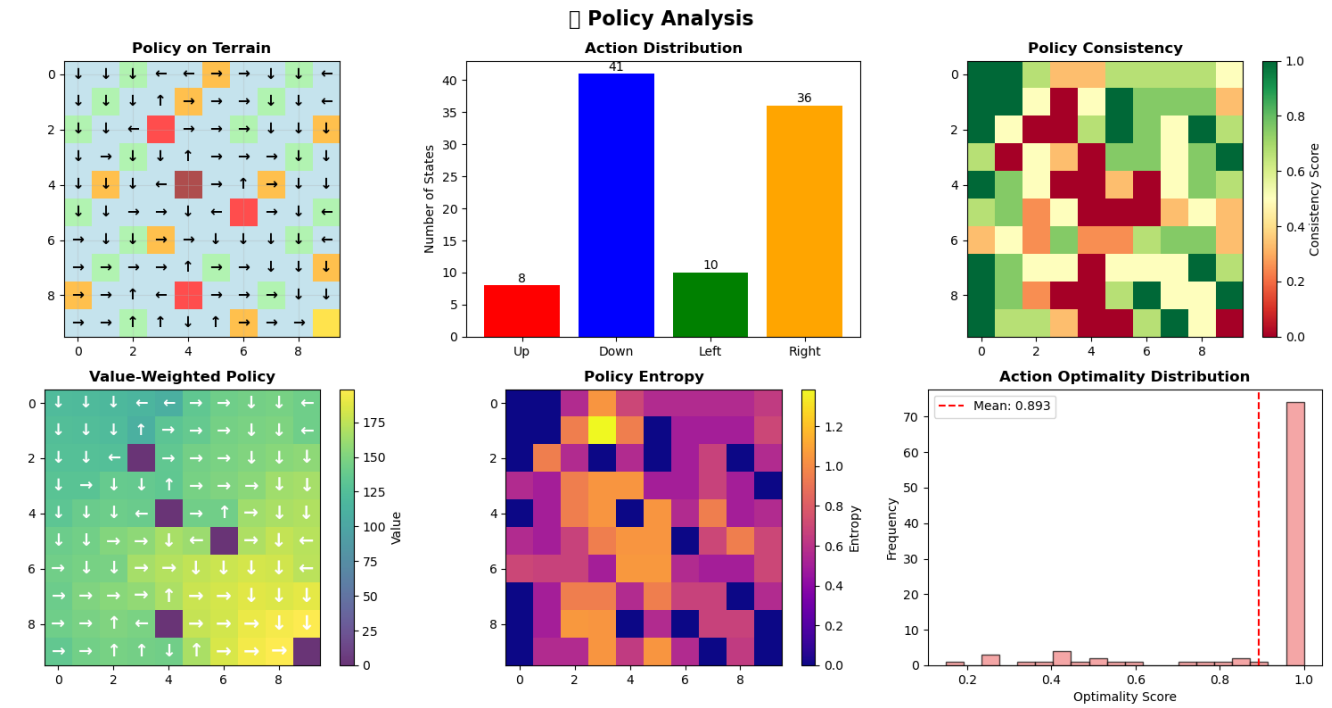

Fig. 6. Policy analysis, showing the optimal policy on the terrain, action distribution, policy consistency heatmap, value-weighted policy, policy entropy, and action optimality distribution.

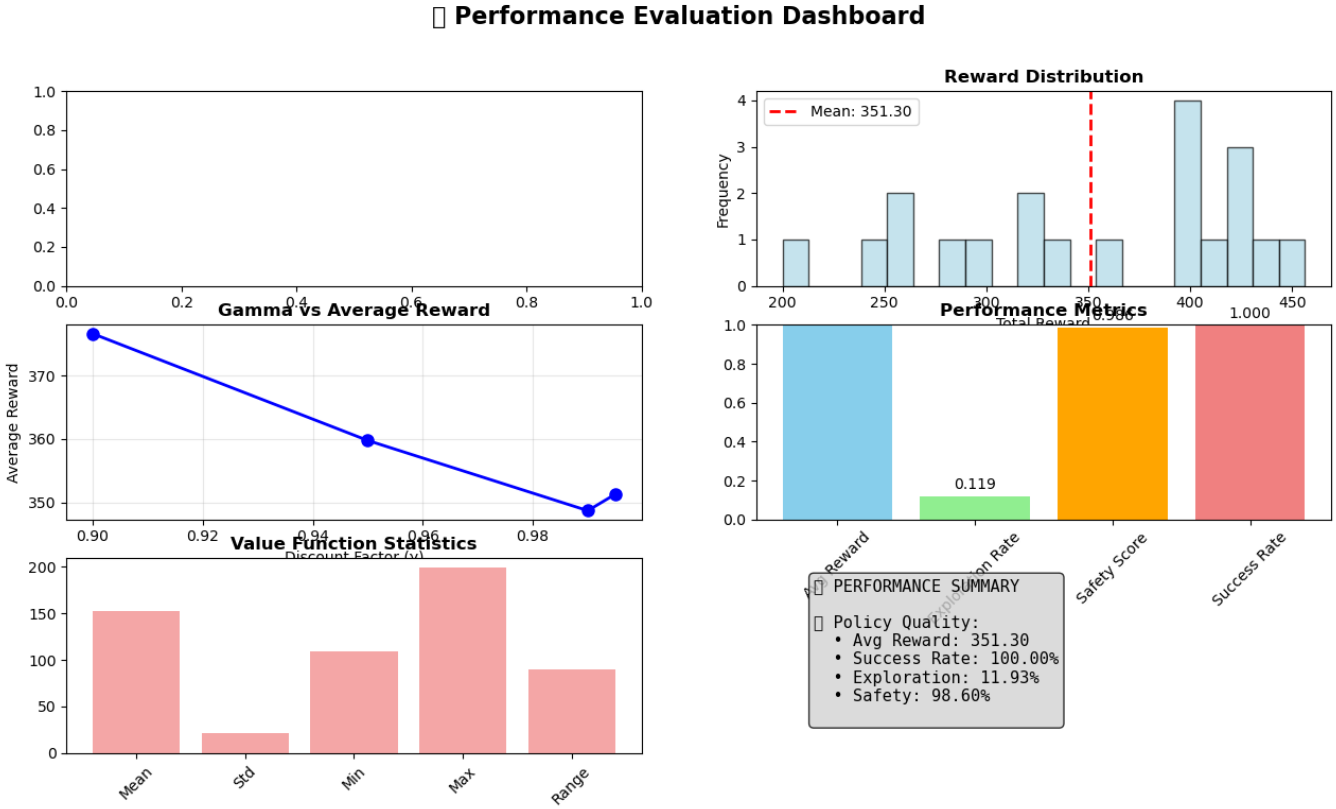

Fig. 7. Performance evaluation dashboard summarizing reward distribution, the effect of gamma on average reward, value function statistics, and overall performance metrics.