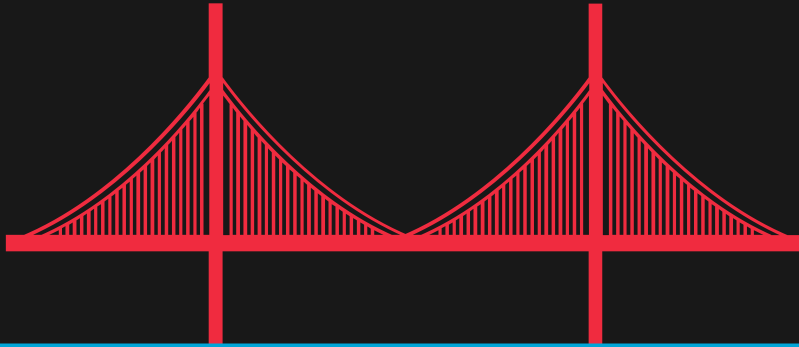
StructuRL Bridge Maintenance

Leveraging RL



Punna Chowdhury, Jane Slagle, and Diana Krmzian

CS 138 - Reinforcement Learning

Instructor - Yash Shukla

December 16, 2024

Table of contents

01

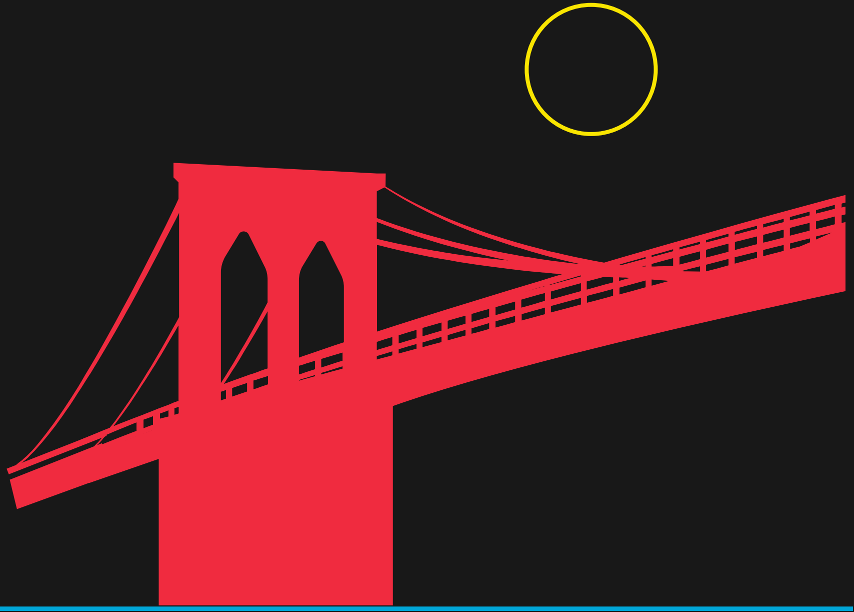
Problem

Challenges facing bridge infrastructure

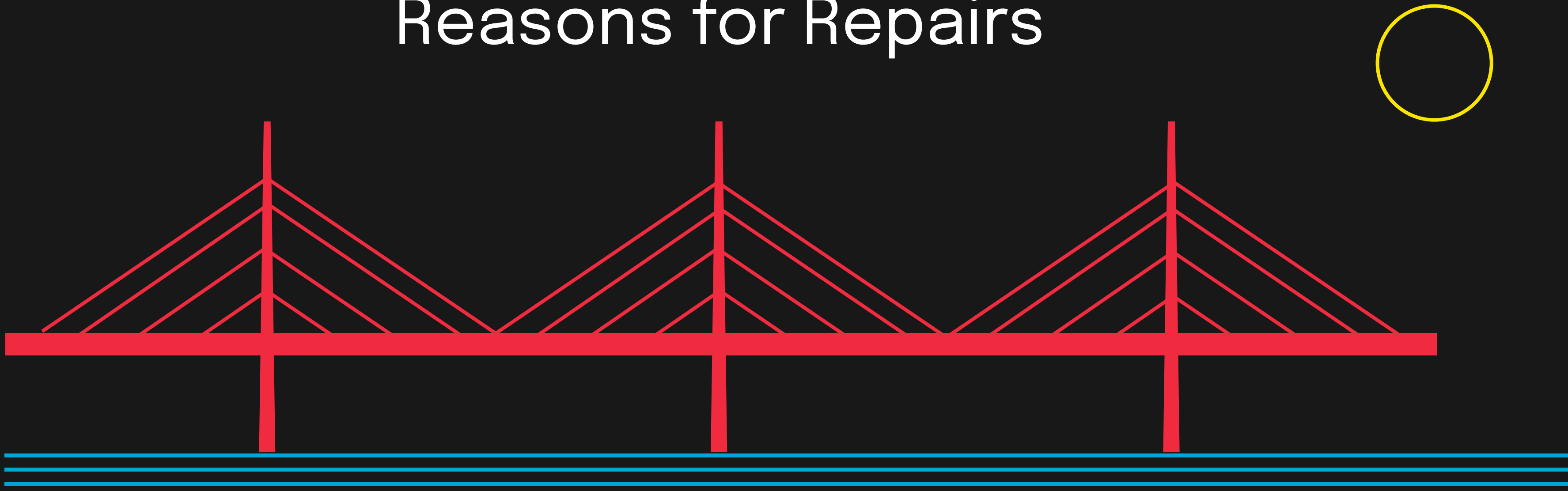
American Road & Transportation Builders Association

221,800

of America's **623,147 bridges** need repair, which span over **6,100 miles**



Reasons for Repairs



Aging, Usage & Deterioration

Maintenance & Intervention

Inspection and Monitoring Accuracy

Remarks

“state of bridges in the U.S. is not good,
and we’re losing the
battle”

- William Ibbs,
a civil engineering professor at
the University of California Berkeley

02

Solution

Implementing RL with SMDP, Deep SARSA and Deep Q-learning

Inspiration for Solution

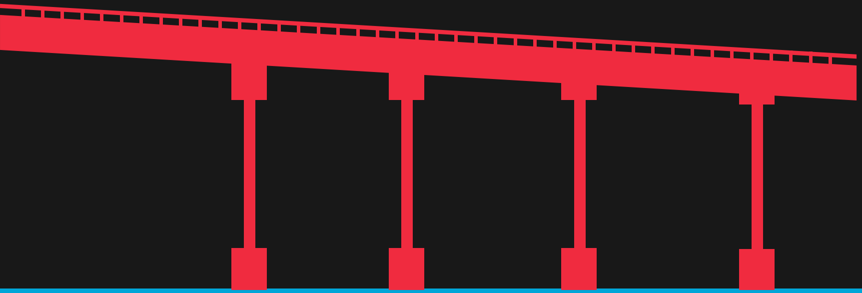
Research Paper on Bridge Maintenance

Hierarchical reinforcement learning for transportation infrastructure maintenance planning

by Zachary Hamida and James-A. Goulet

Key Considerations of Paper

- Adaptive decision-making
- Resource optimization
- Policy for long-term planning and cost savings



Project solution

Decision-Making

Simulate various bridge maintenance tasks that mirror real-world scenarios, addressing pragmatic considerations in decision-making such as budget constraints and the tracking of improvement progress

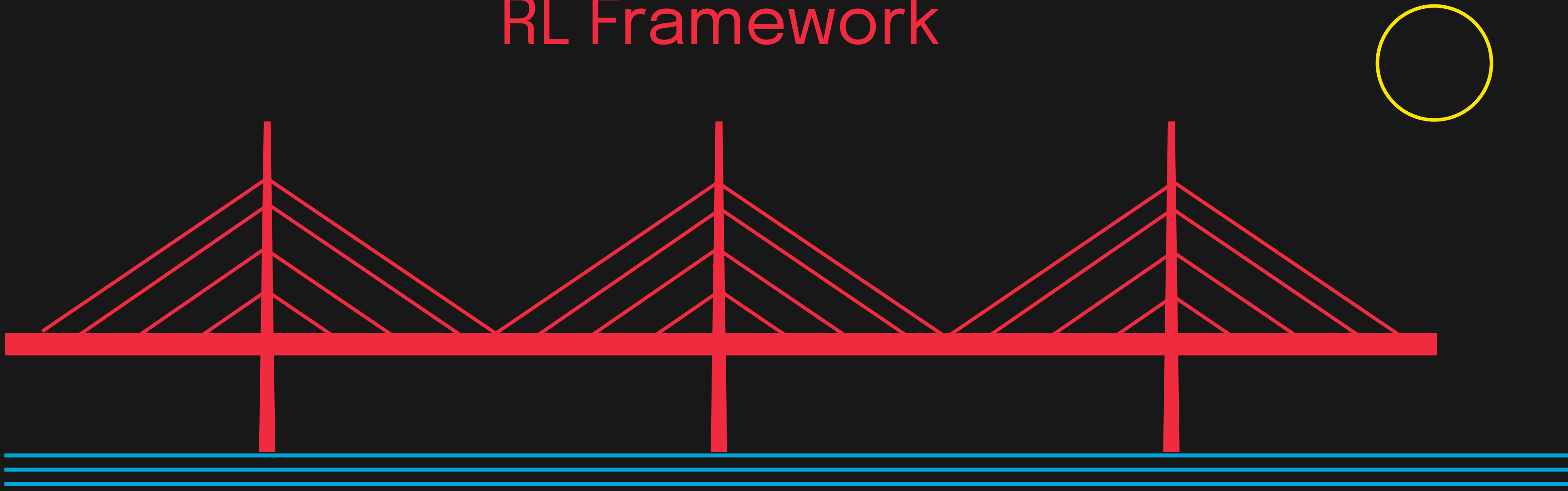
Reward for Prioritization

Assign rewards based on the bridge's current condition, its improvement over time, and effective budget management to prioritize both maintenance and long-term improvement

Cost-Efficient Model

Apply penalties for actions that worsen bridge condition or exceed the budget, and rewards for improvements within budget, ensuring a cost-efficient solution

RL Framework



Semi-MDP

Deep SARSA

Deep Q-Learning

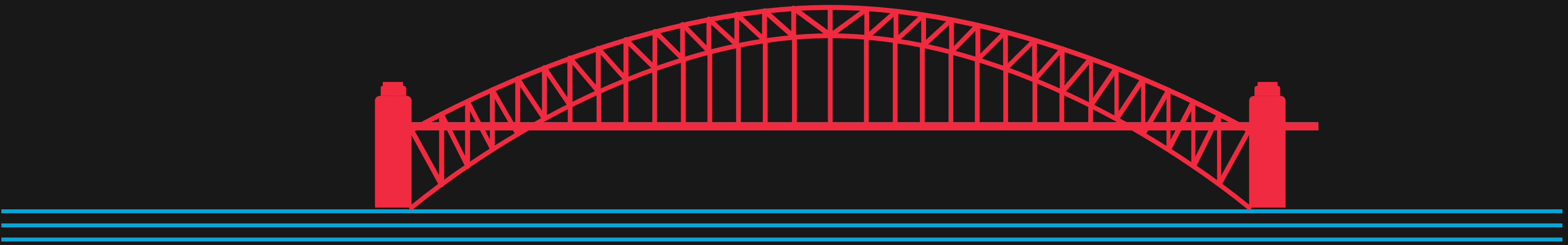
03

Environment

Existing working environment

Inspiration for Environment

Hierarchical reinforcement learning for transportation infrastructure maintenance planning GitHub: InfraPlanner environment



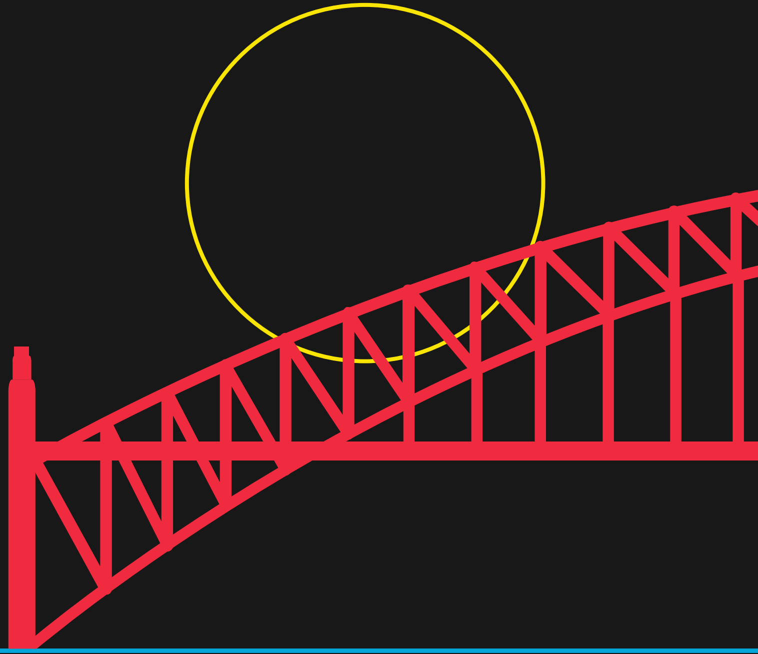
Key Aspects of InfraPlanner Environment:

- Action space
- Action costs
- Budget limitations

Custom Implementation of InfraPlanner Environment

A simulation environment for bridge infrastructure maintenance on one bridge over a 100 year period

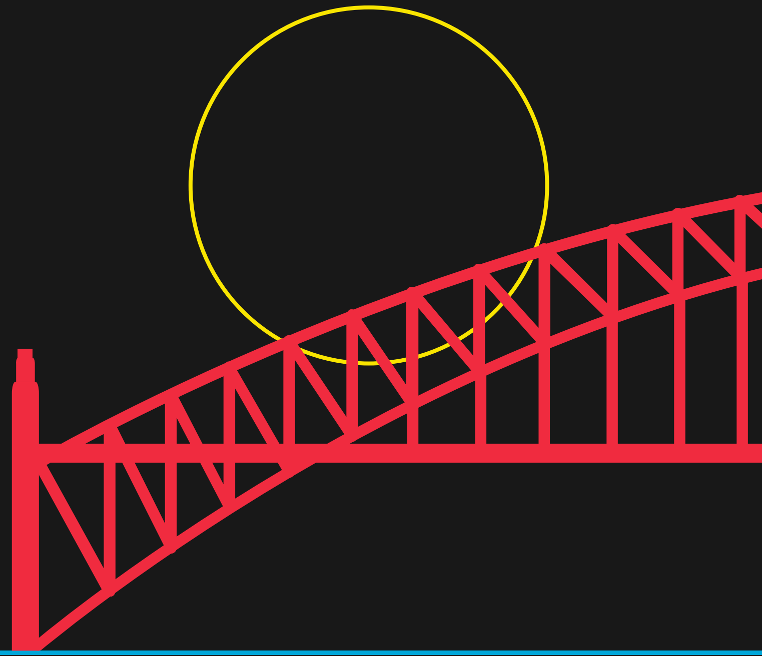
- Incorporate budget constraints
- Cost assigned to each action
- Determination of reward:
    - Bridge condition improvement over time
    - Budget management over time



State Space

Condition of the bridge

- integer value between 0-100
  - 0 = completely deteriorated & unsafe
  - 100 = perfect condition
- initialized as 40



Action Space

3 possible actions related to bridge maintenance tasks

- do nothing - neglect maintenance
  cost of 0, worsens condition by 1%
- maintenance
  cost of 2, improves condition by 1%
- replace
  fixed cost of 5, sets condition to 100

Reward Function

Based on:

(1) current condition of bridge

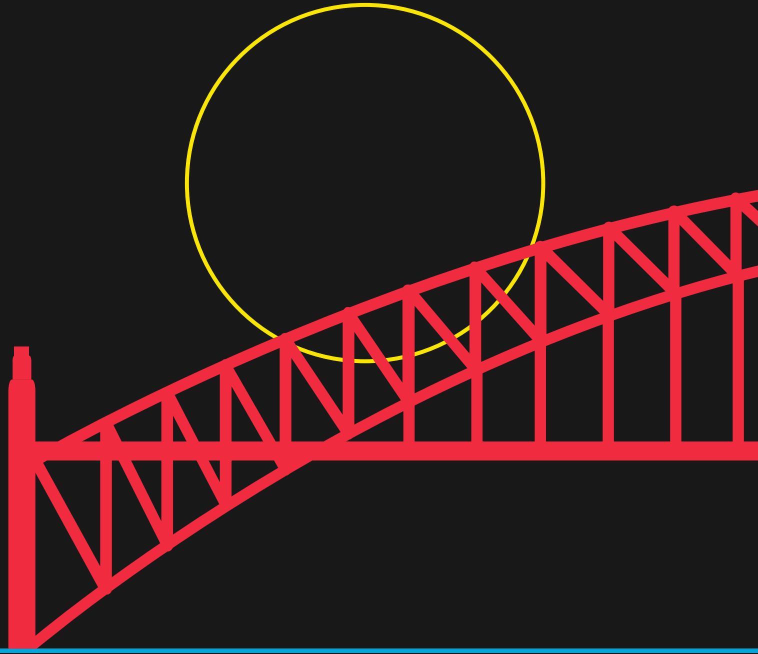
80 & over: incentivize +10
20 & below: penalize -10

(2) improvement from previous condition

incentivize +3

(3) budget

exceeds: penalize -5
else: incentivize +2



04

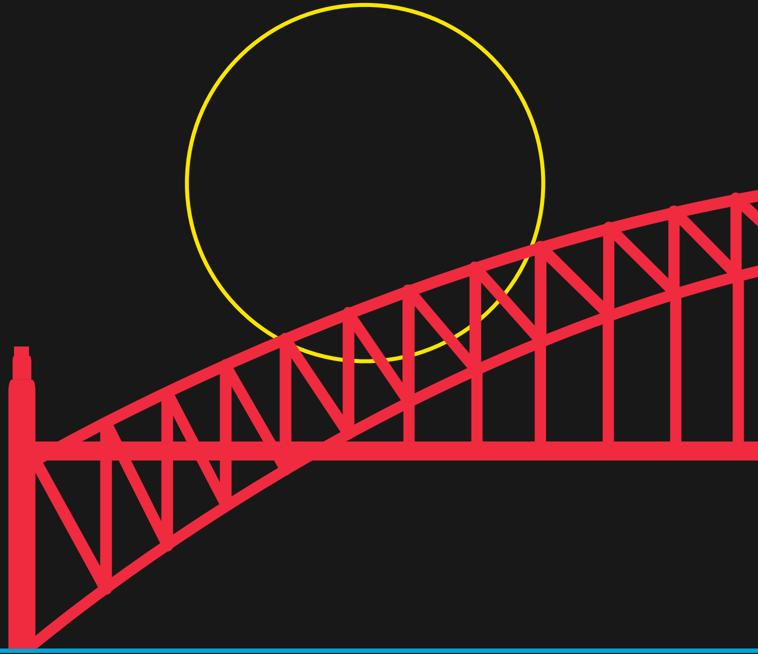
SMDP

Application of SMDP RL approach and results

Semi-Markov decision processes (SMDP)

- Model sequential decision making
- Handle actions over time intervals with varying duration
- Accommodate temporal flexibility

All leads to more realistic representation



Methodology

StructuRL

Combination of Techniques

Q learning - agent observes results, interacts with environment, gets feedback in form of rewards

SMDP - *variable action duration length*

Policy for Action Section

Epsilon Greedy Policy

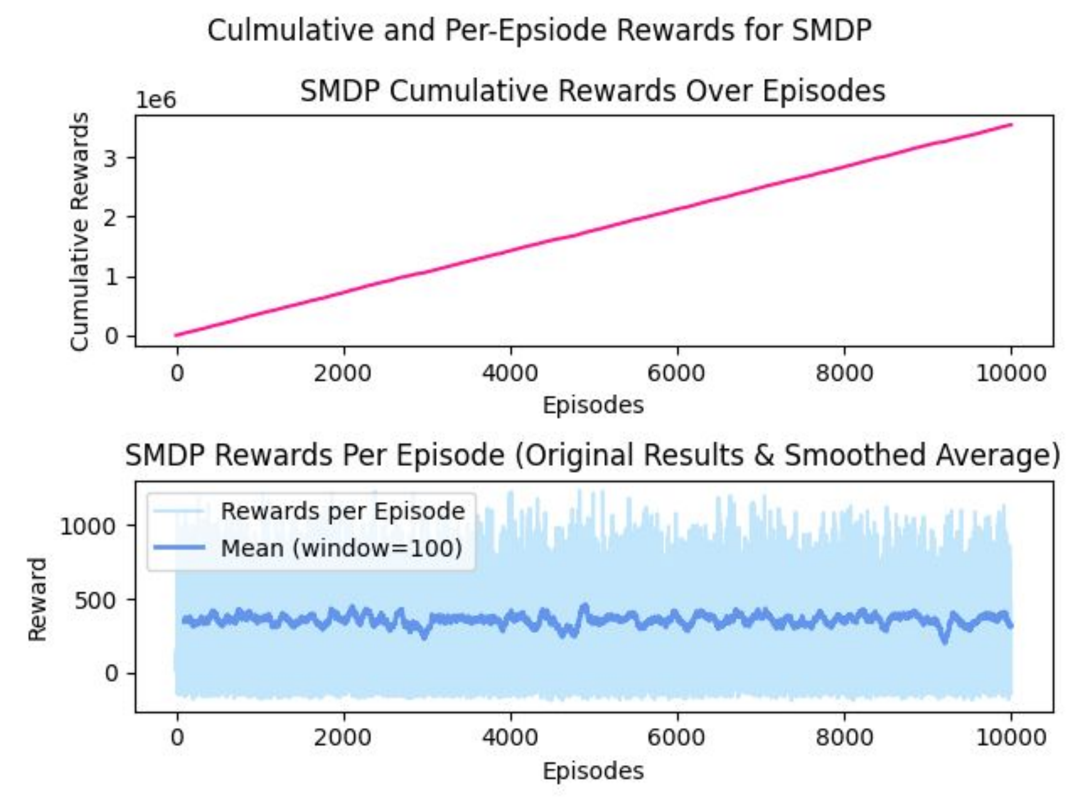
Variable Time Interval Durations

In env step & agent Q value updates, incorporates variable time interval durations, decided from a set range randomly, for each action to best simulate real-world scenarios

Training Goal

Uses Q learning scaled by the variable action duration lengths to update Q-values, balancing immediate and future rewards

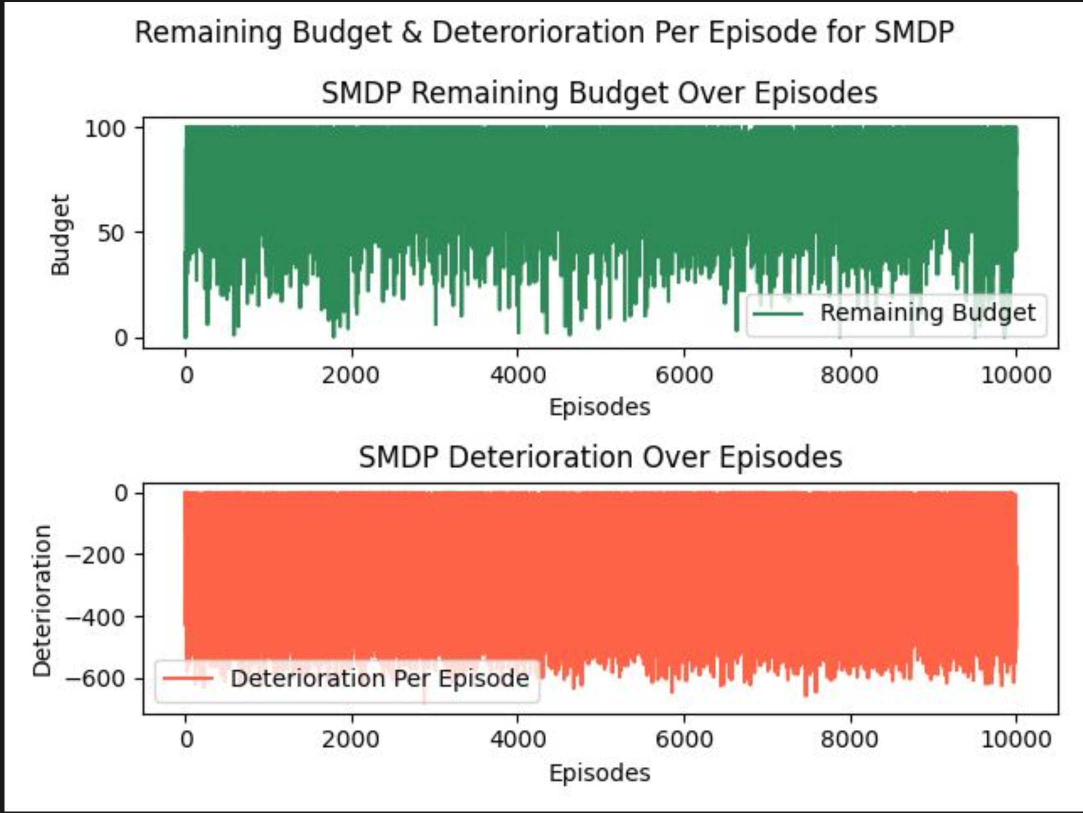
SMDP: Results over 10,000 episodes



Average Cumulative Reward:

354.49

SMDP: Results over 10,000 episodes



Total Budget Spent:

31.0%

Bridge Condition improved:

75.0%

SMDP Results over 10,000 episodes

- Improved condition by **75.0%** from 40 to 85
- Spent **31.0%** of total budget
- Cost efficiency (ratio of average cumulative reward to total cost spent) = **11.44**
- Actions taken:
  - do nothing: 93.23% of time
  - maintenance: 3.35% of time
  - replace: 3.42% of time

→ prioritized waiting out situations where bridge conditions were sufficient enough to deter maintenance interventions

05

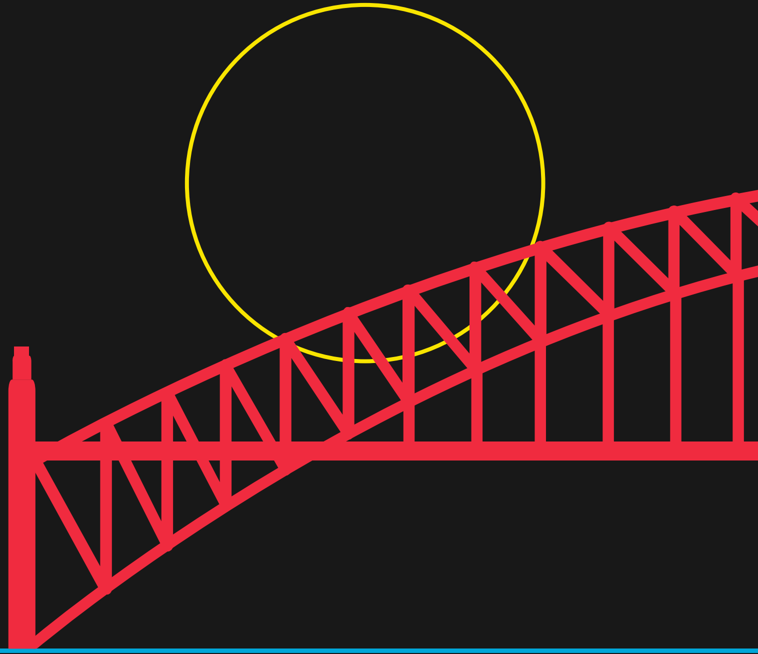
Deep SARSA

Application of Deep SARSA RL
approach and results

Deep SARSA

- Combination of Deep Learning and SARSA (State-Action-Reward-State-Action)
- Update Q-values using the SARSA rule
- Uses neural networks to approximate Q-values in continuous state spaces

*Leads to a stable and adaptable **on-policy** learning approach for complex tasks*



Methodology

StructuRL

Combination of Techniques

Deep learning - neural network for Q-values state-action pairs
SARSA - updates Q-values

Policy for Action Section

Epsilon-Greedy Policy - balances exploration and exploitation during training, converging at optimal policies

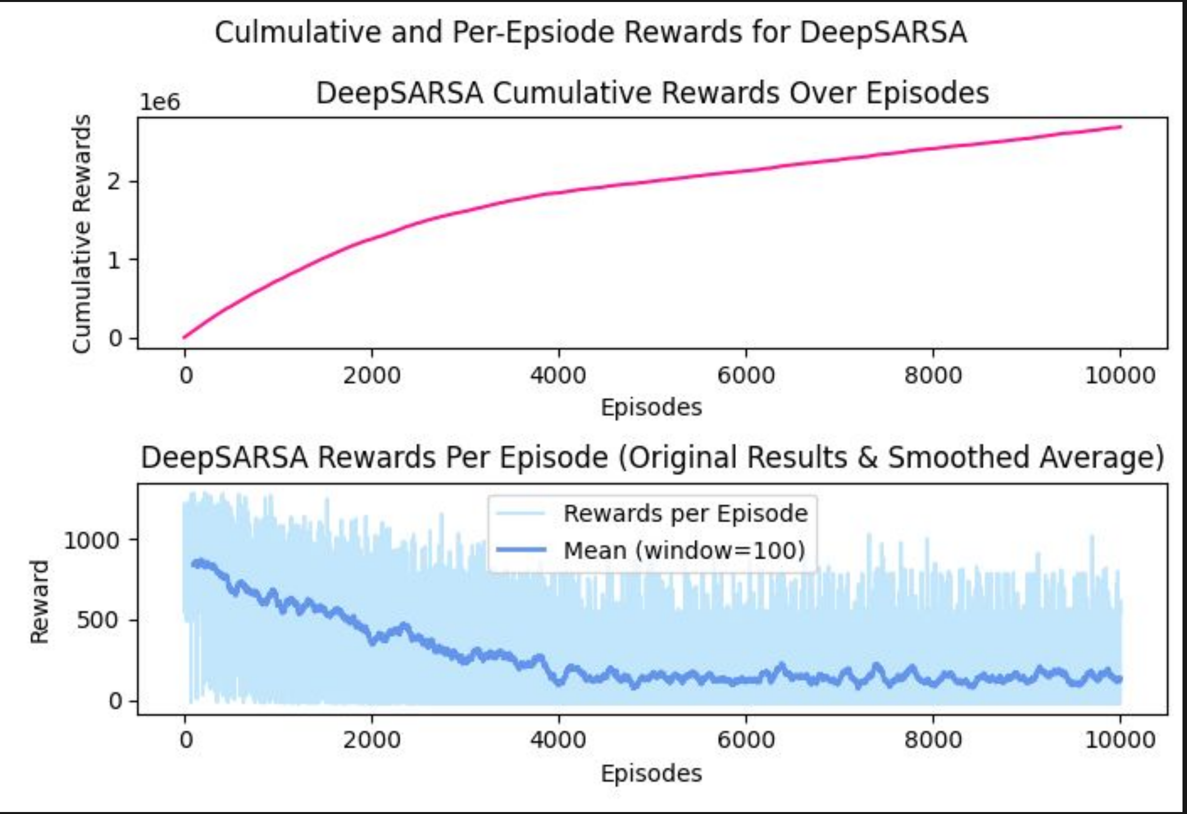
Continuous State Spaces

Leverages neural networks to handle continuous state spaces, with Q-values updating based on variable time intervals

Training Goal

Uses TD learning with the SARSA rule to update Q-values, balancing immediate and future rewards

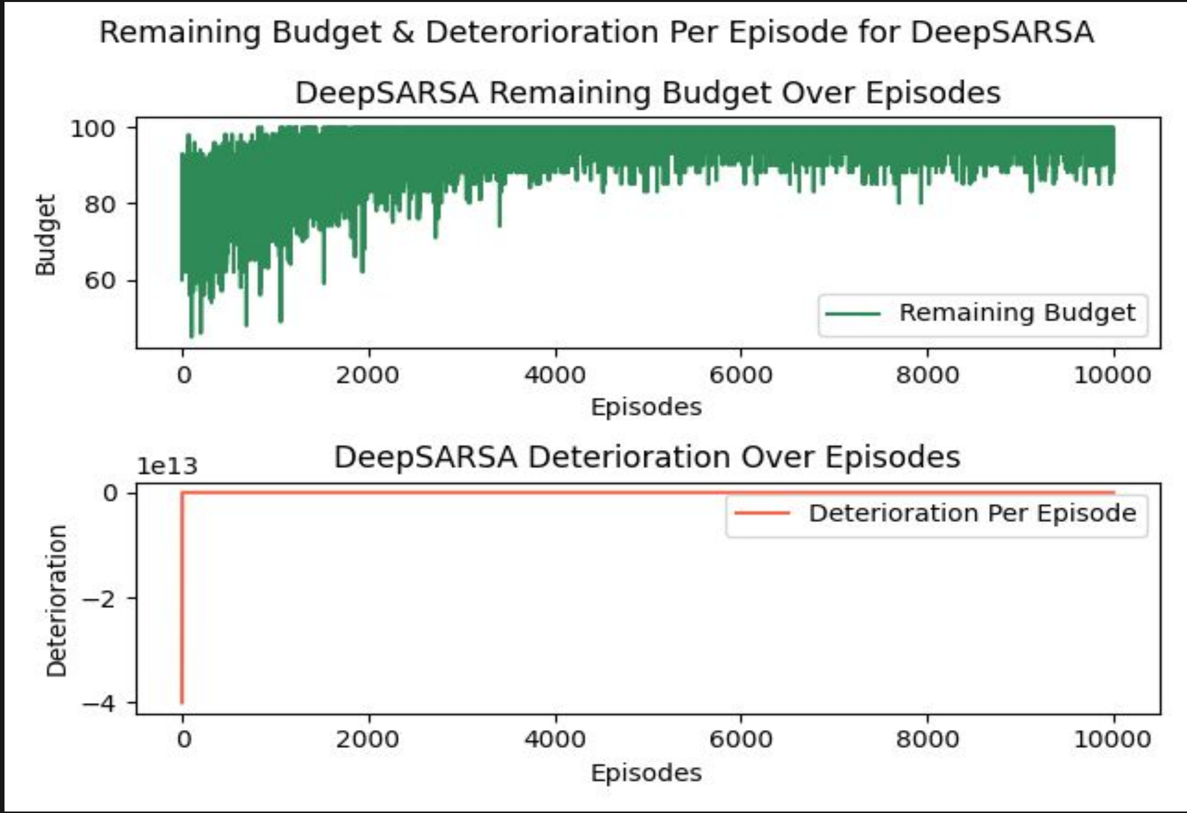
Deep SARSA: Results over 10,000 episodes



Average Cumulative Reward:

268.28

Deep SARSA: Results over 10,000 episodes



Total Budget Spent:

5.00%

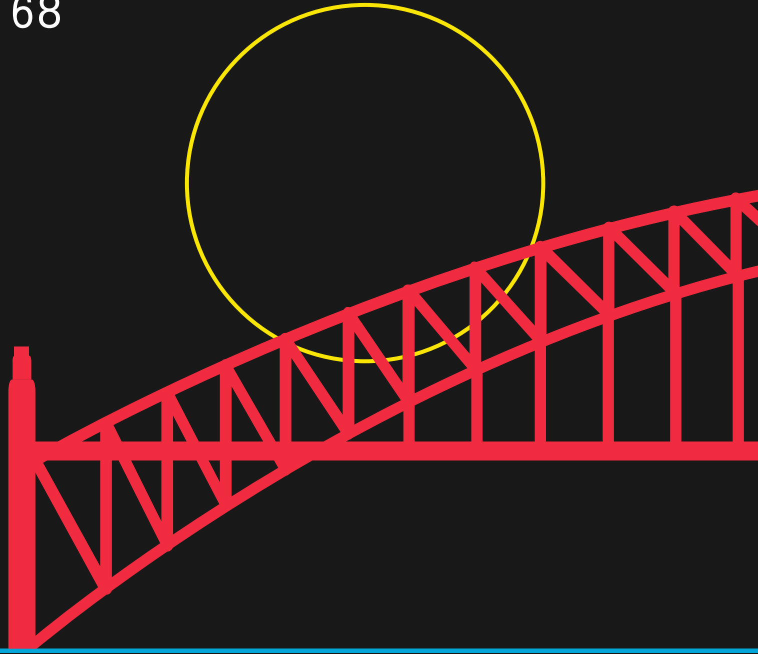
Bridge Condition improved:

46.67%

Deep SARSA Results over 10,000 episodes

- Improved condition by **46.67%** from 40 to 68
- Spent **5%** of total budget
- **53.66** cost efficiency
- Actions taken:
  - do nothing: 98.44% of time
  - maintenance: 0.76% of time
  - replace: 0.80% of time

→ prioritized waiting out situations where bridge conditions were sufficient enough to deter maintenance interventions



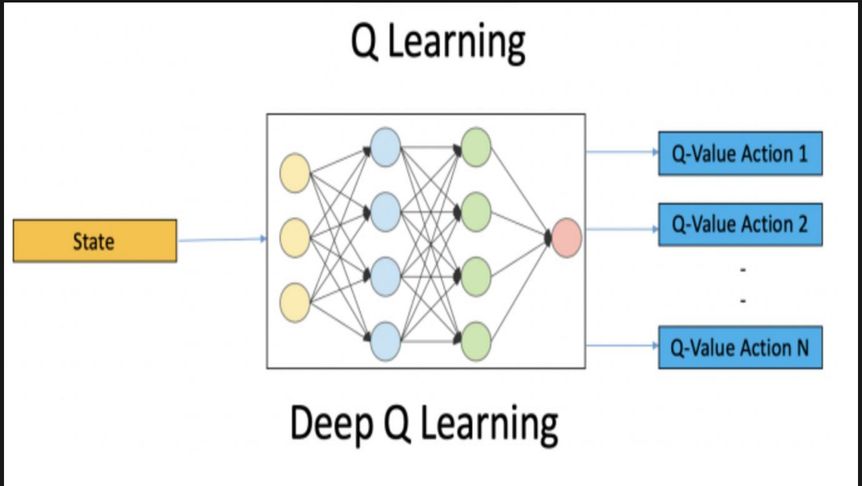
06

Deep Q-Learning

Application of Deep
Q-learning RL approach and
results

Deep QL

- Addresses complexity of large state-action spaces
- Handles Continuous Learning
- Better Performance with limited data
- Provides scalable alternative to tabular Q-learning



Makes capable of generalizing from experiences, efficiently solving problems in complex environments → like bridge maintenance

Methodology

StructuRL

Combination of Techniques

Use of neural networks to approximate Q-values instead of using the traditional Q-tables. It allows learning in large-state action spaces where Q-learning fails

Policy for Action Section

Epsilon-Greedy Policy - balances exploration and exploitation and also helps the agent explore efficiently while improving decisions over time

Training Process

Leverages neural networks to handle continuous state spaces, with Q-values updating based on variable time intervals

Scalability and Efficiency

DQL is capable of handling large state-action spaces. By using fixed time intervals,the training process stays stable to implement ensuring consistent outcomes

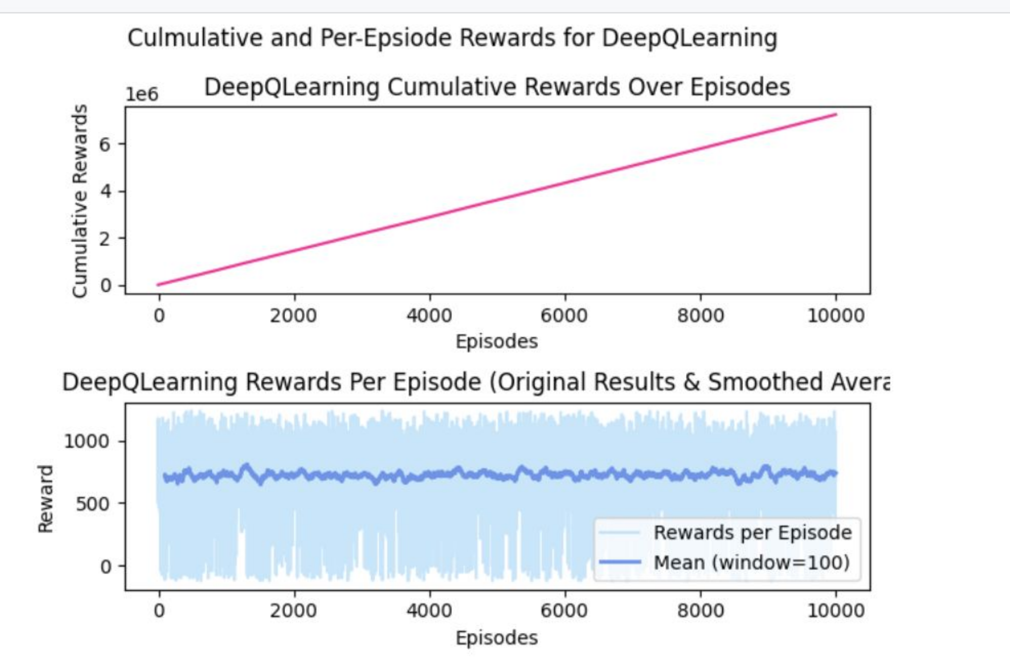
Deep QL: Results over 10,000 episodes

Cumulative Rewards:

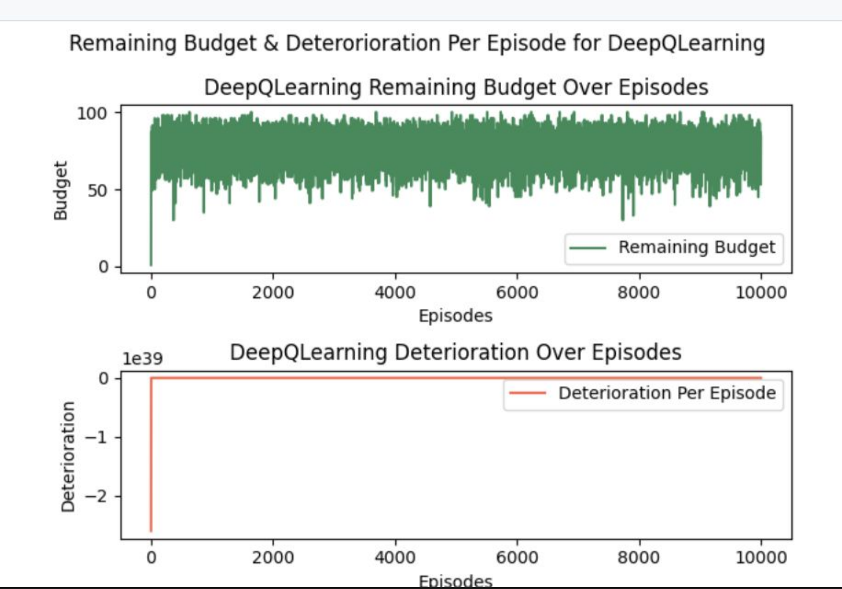
- Steadily increases over the 10,000 episodes showing consistent learning and improvement by the agent
- The linearity shows that the agent is successfully maximizing rewards over time

Rewards Per Episode:

- Individual rewards fluctuate across episodes but stabilize over time.
- The smoothed average(window=100), is consistent showing the agent's performance is steady as it learns the environment



Deep QL: Results over 10,000 episodes



Remaining Budget Over Episodes

- The remaining budget fluctuates but stays high over the 10,000 episodes averaging between 50 and 100 units
- This shows us that the agent is managing the budget effectively and avoiding overspending, balancing actions like “do nothing”, “maintenance” and “replace”

Deterioration over Episodes

- It shows abnormal values
- Could suggest a possible issue in how deterioration is being recorded or calculated leading to unrealistic results

Deep QL Results over 10,000 episodes

- Improved condition by **28.33%** from 40 to 57
- Spent **16.0%** of total budget
- Cost efficiency = **45.2**
- Actions taken:
  - do nothing: 93.32% of time
  - maintenance: 3.35% of time
  - replace: 3.34% of time

→ prioritized waiting out situations where bridge conditions were sufficient enough to deter maintenance interventions

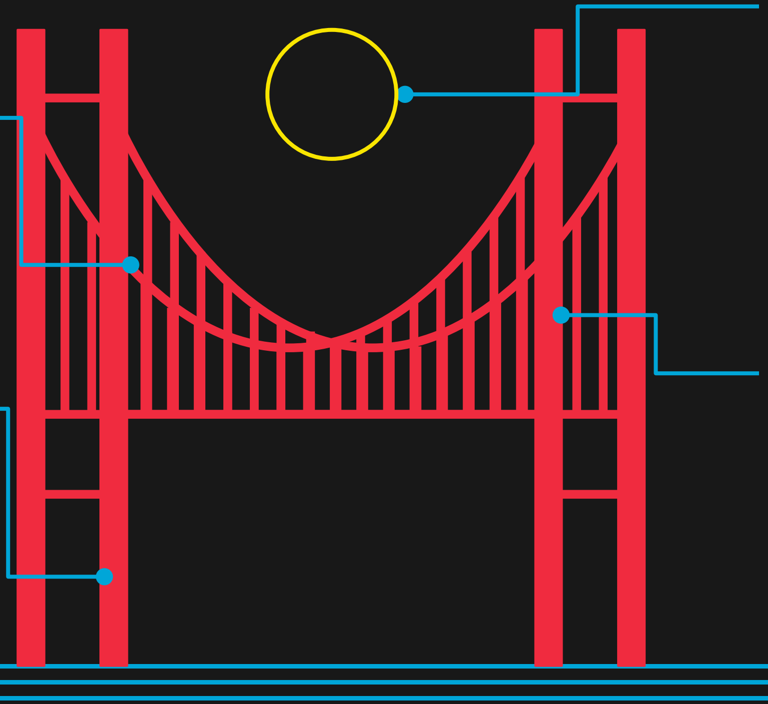
Improving our Algorithms

Training

Extend the training duration

Deterioration

Fix the deterioration bug



Environment Actions

Break down maintenance and replace actions into smaller tasks to make more realistic

Budget Management

Help the agent spend the budget more wisely while keeping the bridge in good condition

07

Key Takeaways

Comparison of algorithms and discuss how our approaches are beneficial in bridge maintenance

Algorithm Performance

Algorithm	Avg. Cumulative Reward	Cost Efficiency	Bridge Improvement (%)	Key Takeaways
SMDP	354.49	11.44	75.00%	Balances actions, performing best in maintaining bridge condition but inefficient in rewards and cost
Deep SARSA (on-policy)	268.28	53.66	46.67%	Balances condition and reward well, but heavily relies on do nothing, limiting improvements
Deep Q-Learning (off-policy)	723.23	45.20	28.33%	Maximizes rewards and efficiency with greedy updates, but struggles with long-term condition

Highlights the best performance in each section

Next steps & project extensions

Future improvements

Advanced Model

Integrate a hierarchical RL framework to prioritize bridge components (e.g., beams, pavement, slabs) based on urgency and condition, ensuring critical tasks are addressed first

Optimization Goals

1

Costs

Reduce costs and maximize budget use

2

Policies

Improve policies for dynamic environment

3

Scalability

Implement framework to real bridge systems

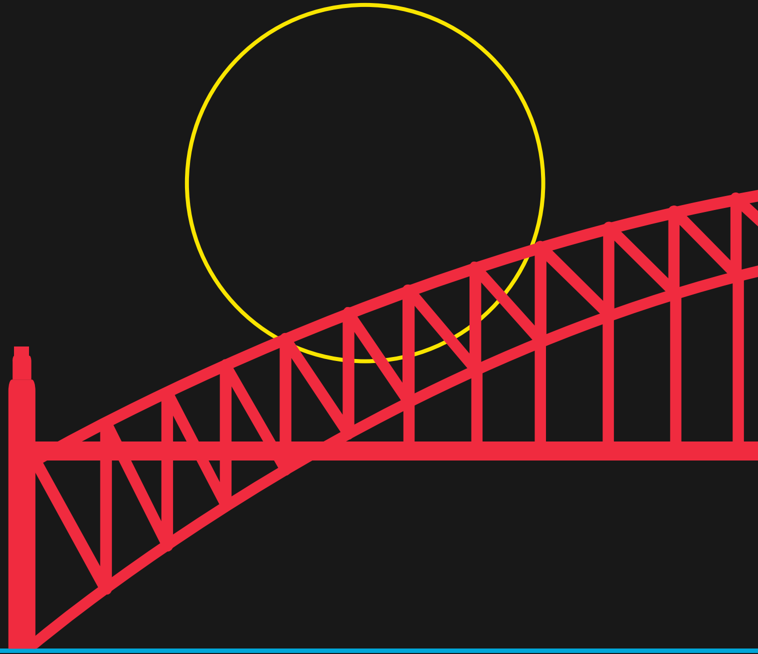
Objectives

- Save money: Plan repairs better to optimize budget usage and reduce the unnecessary costs

- Fixing Urgent issues: Address the urgent issues before they cause serious damage.

- Make easier decisions: use the automated tools to manage large bridge networks efficiently

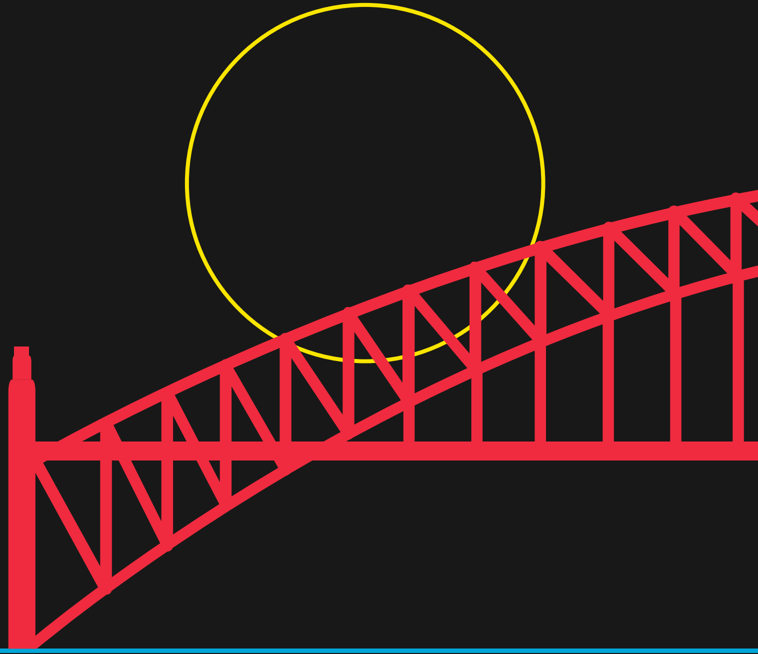
- Handle bigger systems: use methods that work well for large and complex tasks.



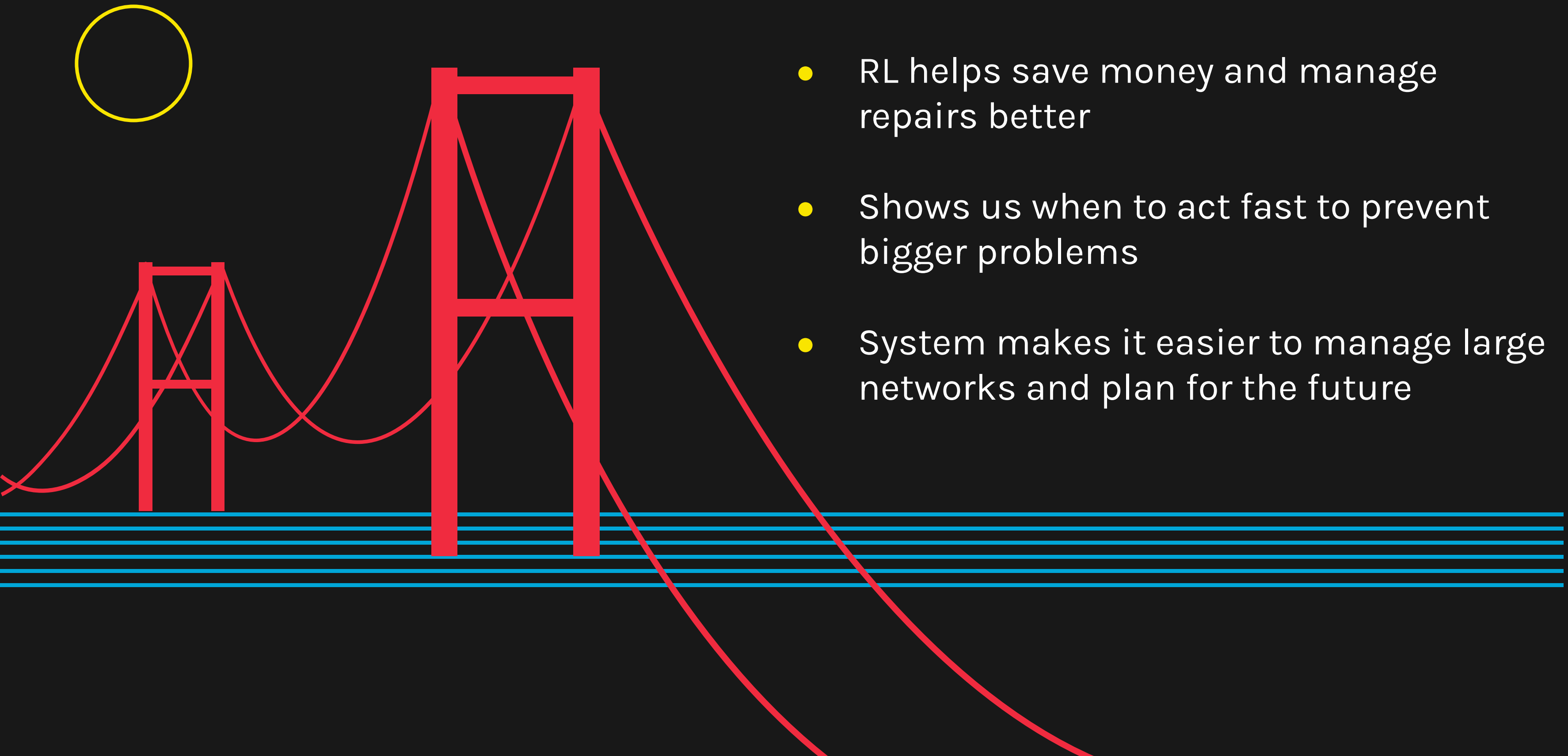
The Benefits

- Reduces disruptions by planning smarter repairs
- Provides clear, easy to understand policies
- Works well for large and complex systems
- Tracks damage to make fast decisions

RL helps prevent costly breakdowns, improves safety and makes sure everything works well by fixing the most important things in a timely manner.



Conclusion and Future Potential



- RL helps save money and manage repairs better
- Shows us when to act fast to prevent bigger problems
- System makes it easier to manage large networks and plan for the future

Thank you!