Inferno Tactics

1st Shaurya Mathur
Dept. of Computer Science & Engineering
University at Buffalo
Buffalo, USA
smathur4@buffalo.edu

2nd Shreyas Bellary Manjunath
Dept. of Computer Science & Engineering
University at Buffalo
Buffalo, USA
sbellary@buffalo.edu

Abstract—In this project, Inferno Tactics, we explore the integration of Reinforcement Learning (RL) with Deep Learning (DL) to develop a proactive system for wildfire prediction and verification using satellite imagery and spatio-temporal data. The system is structured into two interconnected modules. The first module leverages a deep learning model trained on historical wildfire patterns and Earth Engine satellite data to forecast potential wildfire events — predicting both the likely date and geographic location of ignition. Building on this, the second module introduces a reinforcement learning agent that interacts with a dynamic environment of evolving satellite data. The agent's objective is to validate the predicted wildfire events by actively selecting and analyzing recent image sequences, maximizing verification accuracy through feedback-driven exploration. This multi-stage architecture harnesses the predictive power of DL and the decision-making strengths of RL to create a more robust and intelligent wildfire monitoring pipeline. While still under active development, preliminary results suggest that the RL-enhanced validation significantly improves the system's ability to filter out false positives and deliver timely alerts for disaster management.

Index Terms—wildfire, gym, deep learning, reinforcement learning

I. Introduction

Wildfires are increasingly becoming a global threat, with rising temperatures and shifting climate patterns contributing to their frequency and intensity. These fires not only devastate ecosystems but also endanger human lives and infrastructure. Traditional methods of wildfire detection and response often rely on environmental monitoring and sensor-based data; however, these approaches are largely reactive and struggle to provide timely and strategic interventions over vast, unmonitored regions.

Inferno Tactics explores a novel approach that integrates Deep Learning (DL) with Reinforcement Learning (RL) to both predict and plan for wildfire events. The DL component analyzes satellite imagery and historical wildfire data to forecast the time and location of potential wildfire outbreaks. In parallel, the RL component simulates a dynamic environment where wildfires may occur, allowing an intelligent agent to learn optimal mitigation strategies through trial-and-error interactions. These strategies can involve decisions such as resource deployment, early evacuation planning, or the prioritization of firebreak construction.

By modeling wildfire scenarios as a Markov Decision Process (MDP), the RL agent learns to make sequential decisions that aim to minimize long-term damage. This dual-system framework—combining predictive modeling with adaptive planning—seeks to enable a more proactive and automated wildfire management system.

This report presents the current progress of the project, including the design and development of the initial models, challenges faced during implementation, and the roadmap for future enhancements. The objective is to demonstrate the feasibility of combining DL and RL for improving wildfire prediction, verification, and strategic response planning.

II. Background and Motivation

A. Wildfires and the Need for Intelligent Response

Wildfires are becoming increasingly destructive due to climate change, land-use patterns, and prolonged dry seasons. Traditional wildfire response mechanisms, while valuable, often rely on reactive strategies that lack foresight and scalability. In the face of rapidly spreading fires, there's a growing need for proactive and intelligent systems that not only predict the likelihood of wildfires but also simulate and learn effective mitigation responses in real-time.

B. Simulating Wildfire Behavior with Real-World Data

To study wildfire dynamics in a realistic setting, we have developed an interactive simulation environment using React.js and three.js. This simulation serves as a visual and behavioral model of wildfire propagation, incorporating real-world geographic data to enhance its accuracy.

Specifically, we use Google Earth Engine APIs to fetch terrain elevation and land cover data, which are then integrated into the simulation. These attributes—along with wind speed and direction—play a crucial role in shaping how fire spreads. By grounding the simulation in actual Earth data, we aim to mimic plausible wildfire scenarios that are both visually intuitive and spatially accurate.

C. Modeling Firefighting Agents via Reinforcement Learning

At the core of our project is a Reinforcement Learning (RL) agent designed to develop and optimize wildfire mitigation strategies. We connect our React-based frontend to a custom Python-based Gym environment via WebSockets, enabling real-time interaction between the simulation and the RL backend.

The initial agent we have developed simulates a helitack unit—a helicopter-based firefighting strategy. Using the Stable Baselines3 library, we train the agent to explore and learn optimal policies to suppress fires efficiently across different terrain and wind conditions. The agent's goal is to minimize the spread and damage caused by the fire over time, adapting its actions based on changing environmental dynamics.

D. Motivation for Hierarchical Multi-Agent Planning

While the helitack strategy is effective in certain scenarios, real-world wildfire mitigation often involves a combination of approaches, including fire trucks, aerial water drops, and firebreak construction. To simulate such complexity, we plan to extend our system using Hierarchical Reinforcement Learning (HRL).

In this framework, high-level policies will determine which mitigation strategy (e.g., helitack, fire truck dispatch) to deploy, while lower-level agents execute specialized tasks based on the chosen strategy. This modular architecture enables coordinated planning, resource allocation, and scalable learning across different fire response techniques. Our motivation stems from building a generalized, adaptable simulation tool that can train agents capable of multimodal disaster response. By modeling realistic wildfire environments and training agents through trial-and-error, we aim to develop an intelligent system that can assist human decision-makers in planning efficient, timely, and context-aware fire mitigation strategies.

III. Preliminary Work

Significant progress has been made in laying the foundation of the Inferno Tactics system. Our work spans the development of a 3D client simulation, the construction of a custom reinforcement learning (RL) backend, and the integration of real-world data sources to ensure a realistic training and evaluation environment. This section outlines the key components of our implementation so far.

A. Client-Side Simulation Environment (React + Three.js)

We developed an interactive simulation environment using React.js in combination with Three.js to visualize wildfire behavior in a 3D space. This browser-based frontend serves as a real-time visualization tool where:

- Wildfires spread dynamically across a simulated terrain.
- Land cover types (e.g., forests, grasslands, barren regions) influence the spread rate and fire intensity.
- Environmental conditions such as elevation and wind direction are incorporated into the fire propagation logic.

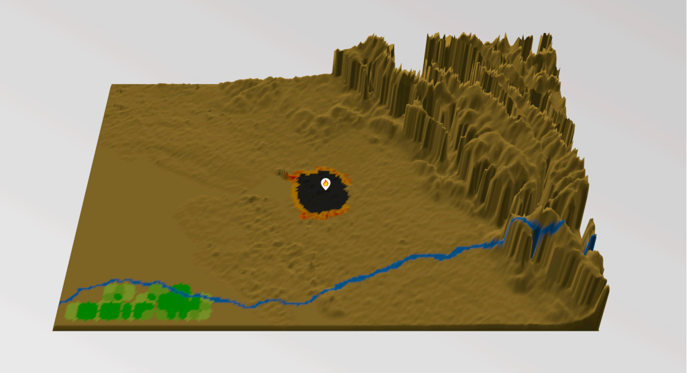


Fig. 1. Wildfire Simulation Environment.

The simulation is designed to support interaction with RL agents via live updates, reflecting both wildfire progression and agent interventions.

B. Backend Architecture (Python, Gym, WebSockets)

To interface with the simulation, we developed a custom OpenAI Gym environment in Python that models the wildfire spread logic and exposes a standardized RL interface.

- The Gym environment is synchronized with the React frontend using WebSockets, allowing real-time bidirectional communication.
- Fire dynamics in the Gym environment replicate the frontend behavior to maintain consistency across training and visualization.
- The environment supports multi-step episodes where the agent can interact with the fire over a series of timesteps, receiving observations and rewards accordingly.

C. RL Agent Development (Stable Baselines3)

We have trained a baseline agent using the Stable Baselines3 library. The agent simulates a helitack (helicopter-based) firefighting strategy. Key aspects of this setup include:

- The agent observes a spatial grid representing the fire map, land cover types, and wind patterns.
- Actions consist of moving to a particular coordinate and applying suppression in that region.
- Rewards are defined based on the amount of fire suppressed, area saved, and resource efficiency.
- The current policy is trained using PPO (Proximal Policy Optimization), with experiments underway to test other algorithms like DQN and A2C.

D. Real-World Data Integration (Earth Engine SDK)

To enhance realism and generalization, we integrated data from Google Earth Engine:

- Elevation data is used to simulate terrain effects on fire spread.

- Land cover maps (e.g., MODIS, ESA WorldCover) help define fuel loads and flammability regions.
- This geospatial data is preprocessed and embedded into the simulation environment for both visual rendering and RL input representation.

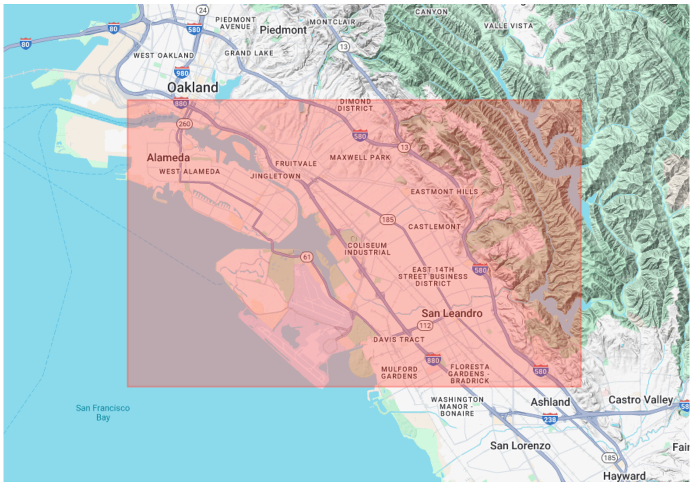


Fig. 2. Terrain View from EarthEngine.

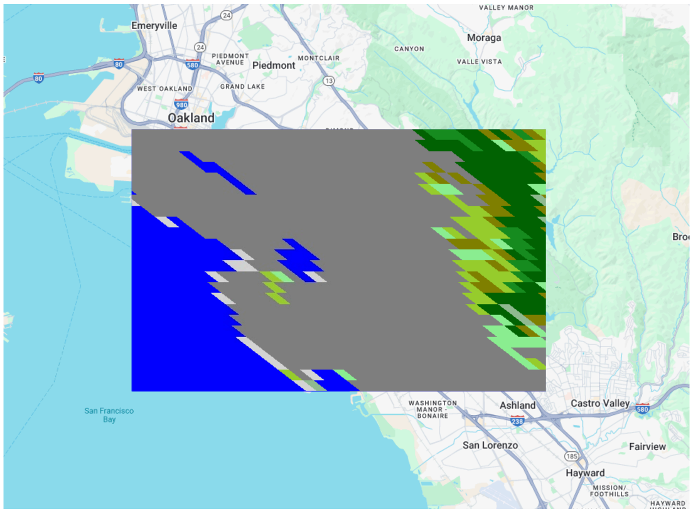


Fig. 3. Land Cover from EarthEngine.

This integration allows our simulation to reflect actual wildfire-prone locations, enabling future transfer learning from simulated to real-world geographies.

E. Current Capabilities

At this stage, our system is capable of:

- Predicting and simulating the spread of wildfire on realistic terrain.
- Allowing a trained RL agent to interact with the fire and attempt suppression.
- Visualizing both fire progression and agent interventions in real time.

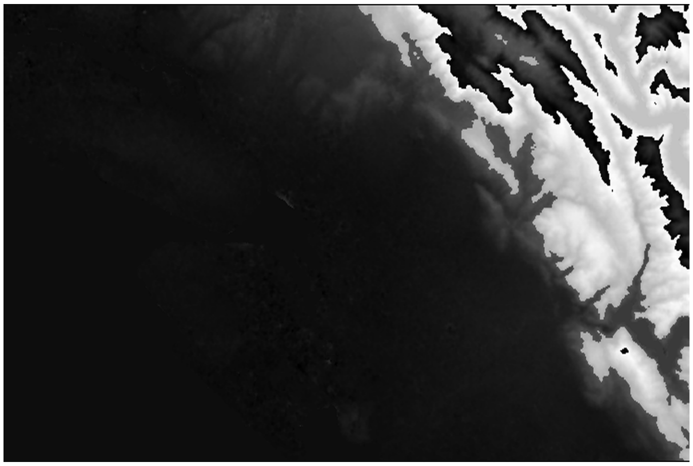


Fig. 4. Heightmap of terrain depicting land elevations.

- Supporting expansion to additional agent types and hierarchical planning.

Our preliminary work provides a strong technical foundation to explore advanced strategies such as hierarchical reinforcement learning and multi-agent coordination in future stages.

IV. Future Work

While the current stage of the project has successfully established the foundational architecture and demonstrated promising initial results, several key enhancements are planned to extend the system's capabilities, realism, and impact. These future steps are outlined below.

A. Hierarchical Reinforcement Learning

Our current system models a single firefighting strategy (helitack) through a single-agent RL approach. We aim to expand this to a multi-tiered control system using Hierarchical Reinforcement Learning (HRL). In this framework:

- A high-level policy will choose between multiple mitigation strategies, such as deploying fire trucks, helitacks, or constructing firebreaks.
- Low-level policies (sub-agents) will handle the specific execution of each strategy.
- This approach will allow the agent to operate at both strategic and tactical levels, improving overall adaptability and efficiency.

B. Multi-Agent Coordination

In real-world wildfire mitigation, coordination between multiple response units is essential. As a next step, we plan to implement a multi-agent system where:

- Each agent represents a distinct firefighting unit (e.g., ground crew, aerial team).
- Agents may have overlapping but not identical observation spaces and action sets.
- Coordination and communication protocols will be explored using frameworks like MADDPG (Multi-Agent Deep Deterministic Policy Gradient).

C. Policy Transfer to Real Geographies

Currently, our training environments are based on simulated data generated from real-world maps. As the models mature, we intend to:

- Fine-tune trained agents on historical wildfire case studies (e.g., California, Australia).
- Leverage transfer learning techniques to adapt policies across varying terrain and climate conditions.
- Validate agent performance using satellite data and fire event records.

D. Real-Time Decision Support Tool

A major goal of this project is to build a decision-support interface that can assist emergency response teams. Future development includes:

- Deploying the simulation as a cloud-based tool with real-time updates.
- Allowing human operators to interact with the system and adjust RL agent recommendations.
- Incorporating user feedback to dynamically re-train or fine-tune models.

E. Explainability and Trust in RL Decisions

As we deploy RL in high-stakes decision-making scenarios, building trust in the agent's choices is critical. Planned work includes:

- Developing visualization techniques to explain agent decisions.
- Analyzing action trajectories and counterfactuals to understand failure cases.
- Using interpretable RL models or post-hoc explainability tools to aid human oversight.

F. Integration with Government and NGO Systems

Long-term, we envision this system being seamlessly integrated with wildfire management platforms used by:

- Government agencies (e.g., CAL FIRE, NASA FIRMS) for real-time fire spread insights and mitigation planning.
- NGOs involved in environmental protection and disaster response for prioritizing interventions and resource deployment.
- Research organizations focused on climate-resilient infrastructure and AI-powered disaster forecasting.

As a final output, the system generates a concise, human-readable one-page report using a fine-tuned large language model (LLM), summarizing the fire's behavior, estimated spread, and recommended actions. This empowers field officers and policy makers to make fast, informed decisions without needing to interpret complex geospatial data.
By continuing to refine and scale this pipeline, we aim to offer a robust, intelligent, and accessible decision-support framework to help tackle one of the world's most urgent environmental threats.

V. Bonus: Project Management Tool

For this project, we utilized Trello as our project management tool to ensure structured progress tracking and clear communication. The project was divided into several milestones including data exploration, proposal drafting, basic and advanced model implementations, training, testing, and final presentation preparations.

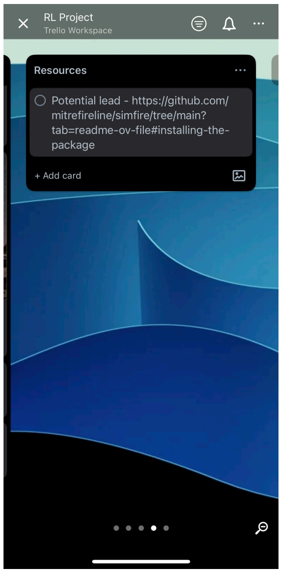
Since our collaboration spanned both the reinforcement learning (RL) and deep learning (DL) parts of the project, both Trello boards were linked. Weekly tasks were clearly documented, enabling effective contribution tracking and synchronization between team members.

Key milestones included:

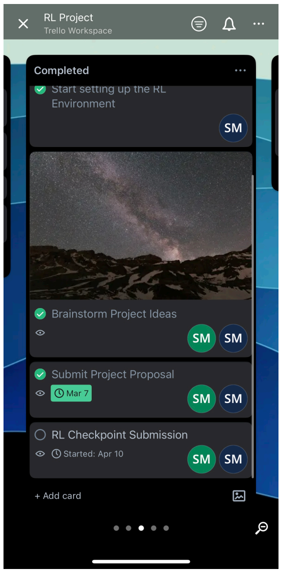
- Project initialization
- Data gathering and preprocessing
- Initial model development
- Advanced model enhancements
- Training and validation of models
- Evaluation and final adjustments

A. Project Management Screenshots

Below are screenshots showcasing our Trello-based project management efforts:



○ Potential lead - https://github.com/mitrefireline/simfire/tree/main?tab=readme-ov-file#installing-the-package

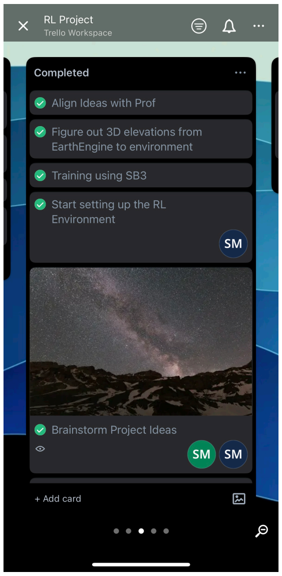


✓ Start setting up the RL Environment

✓ Brainstorm Project Ideas

✓ Submit Project Proposal
Mar 7

○ RL Checkpoint Submission
Started: Apr 10

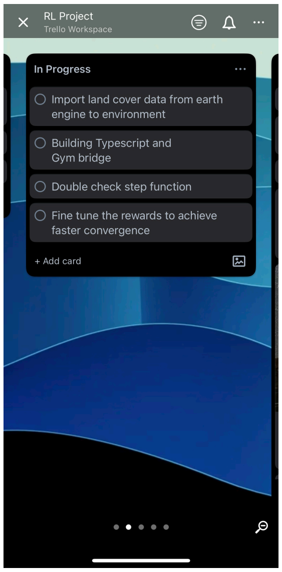


RL Project
Trello Workspace

Completed

- ✓ Align Ideas with Prof
- ✓ Figure out 3D elevations from EarthEngine to environment
- ✓ Training using SB3
- ✓ Start setting up the RL Environment SM
- ✓ Brainstorm Project Ideas SM SM

+ Add card

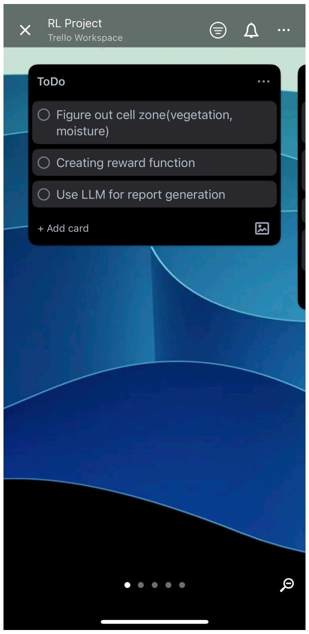


RL Project
Trello Workspace

In Progress

- ○ Import land cover data from earth engine to environment
- ○ Building Typescript and Gym bridge
- ○ Double check step function
- ○ Fine tune the rewards to achieve faster convergence

+ Add card



RL Project
Trello Workspace

ToDo

- ○ Figure out cell zone(vegetation, moisture)
- ○ Creating reward function
- ○ Use LLM for report generation

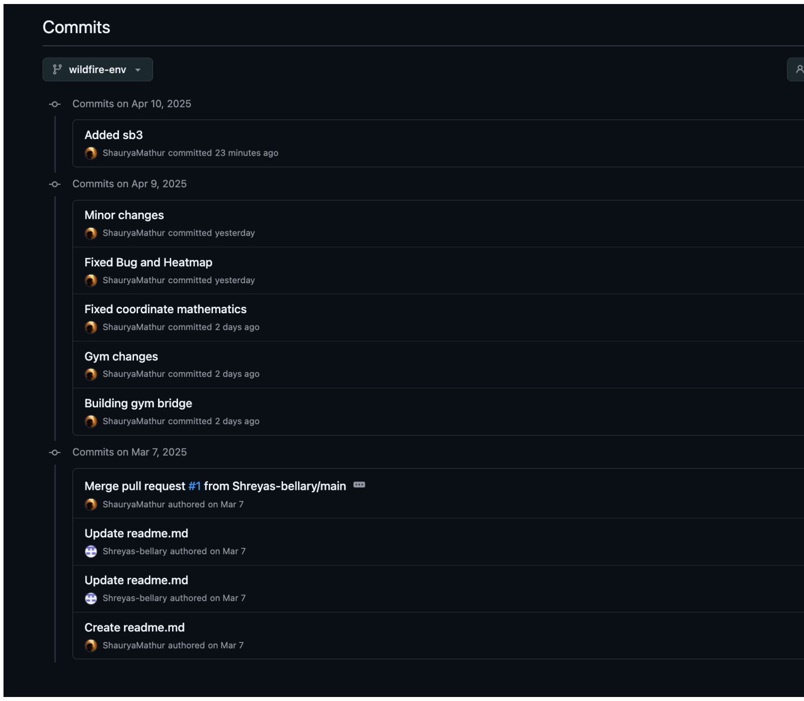
+ Add card

Trello Board URL:
https://trello.com/b/pBNt9VwL/rl-project

GitHub Repository:
https://github.com/ShauryaMathur/inferno-tactix/tree/wildfire-env

B. GitHub Commit History

Below is a snapshot of our git commit history demonstrating ongoing collaboration and iterative development:



Commits

wildfire-env

Commits on Apr 10, 2025

Added sb3
ShauryaMathur committed 23 minutes ago

Commits on Apr 9, 2025

Minor changes
ShauryaMathur committed yesterday

Fixed Bug and Heatmap
ShauryaMathur committed yesterday

Fixed coordinate mathematics
ShauryaMathur committed 2 days ago

Gym changes
ShauryaMathur committed 2 days ago

Building gym bridge
ShauryaMathur committed 2 days ago

Commits on Mar 7, 2025

Merge pull request #1 from Shreyas-bellary/main
ShauryaMathur authored on Mar 7

Update readme.md
Shreyas-bellary authored on Mar 7

Update readme.md
Shreyas-bellary authored on Mar 7

Create readme.md
ShauryaMathur authored on Mar 7

Fig. 5. git commit history

References

[1] https://github.com/amanbasu/wildfire-detection.
[2] https://code.earthengine.google.com/
[3] https://wildfire.concord.org/
[4] https://github.com/concord-consortium/wildfire-model/tree/master
[5] U.S. Geological Survey, Landsat 8 (L8) Data Users Handbook, U.S. Geological Survey, 2019. [Online]. Available: https://landsat.usgs.gov/landsat-8
[6] Y. Zhao, M. Chen, and S. Kumar, “A Comprehensive Review of Deep Learning Techniques for Wildfire Detection in Satellite Images,” IEEE Access, vol. 9, pp. 10523–10539, 2021.
[7] M. Pereira and G. H. A., “Active Fire Detection in Landsat-8 Imagery: A Large-Scale Dataset,” GitHub, 2023. [Online]. Available: https://github.com/pereira-gha/activefire