COLLEGE OF ENGINEERING

MDPvis: An Interactive Visualization for Testing Markov Decision Processes

Sean McGregor, Hailey Buckingham, Rachel Houtman, Claire Montgomery, Ronald Metoyer, and Thomas Dietterich

How did I get here?

- **2010:** Started with simulator building and optimization
- **2010 to Present:** Solve problems with slow and buggy software from foresters
- **2014:** Develop MDP visualizations for foresters
- **Today:** We also need tools for MDPs

McGregor, S., Buckingham, H., Dietterich, T. G., Houtman, R., Montgomery, C., & Metoyer, R. (2015). Facilitating Testing and Debugging of Markov Decision Processes with Interactive Visualization. IEEE Symposium on Visual Languages and Human-Centric Computing.

Motivation

- Many **sequential decision making problems** combine **complex models** to optimize on Monte Carlo rollouts

McGregor, S., Buckingham, H., Dietterich, T. G., Houtman, R., Montgomery, C., & Metoyer, R. (2015). Facilitating Testing and Debugging of Markov Decision Processes with Interactive Visualization. IEEE Symposium on Visual Languages and Human-Centric Computing.

Motivation

- Many **sequential decision making problems** combine **complex models** to optimize on Monte Carlo rollouts
- Models and MDP specification may be **misspecified or poorly implemented**

McGregor, S., Buckingham, H., Dietterich, T. G., Houtman, R., Montgomery, C., & Metoyer, R. (2015). Facilitating Testing and Debugging of Markov Decision Processes with Interactive Visualization. IEEE Symposium on Visual Languages and Human-Centric Computing.

Motivation

- Many **sequential decision making problems** combine **complex models** to optimize on Monte Carlo rollouts
- Models and MDP specification may be **misspecified or poorly implemented**
- **Want: better systems** for understanding MDPs and testing for bugs

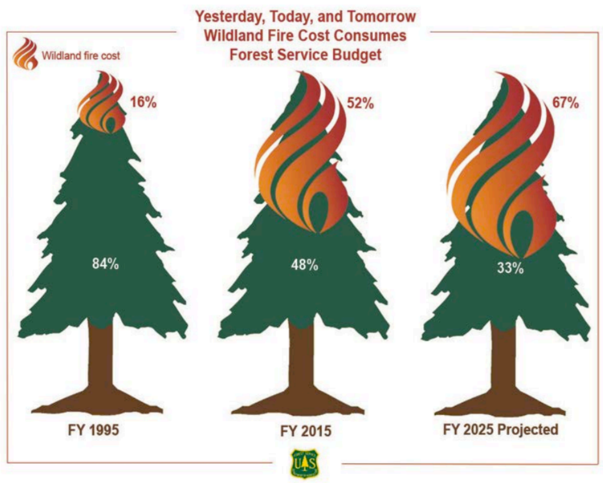
McGregor, S., Buckingham, H., Dietterich, T. G., Houtman, R., Montgomery, C., & Metoyer, R. (2015). Facilitating Testing and Debugging of Markov Decision Processes with Interactive Visualization. IEEE Symposium on Visual Languages and Human-Centric Computing.

Outline

1. **Wildfire Suppression MDP Example**
   - Basic Introduction
   - Testing
2. MDPvis
   - Design
   - Testing Examples
   - MDPvis Use Case Study
   - Integrating Your Domain or Optimizer
3. Concluding

Motivating Domain of Wildfire

Starting in 1935, the United States adopted the "**10 AM policy**"



We need a more nuanced approach.

Houtman, R. M., Montgomery, C. A., Gagnon, A. R., Calkin, D. E., Dietterich, T. G., McGregor, S., & Crowley, M. (2013). Allowing a Wildfire to Burn: Estimating the Effect on Future Fire Suppression Costs. International Journal of Wildland Fire, 22(7), 871–882.

http://www.fs.fed.us/sites/default/files/2015-Fire-Budget-Report.pdf

Modeling Wildfire

S	All the possible configurations of trees/ignitions
P_0	A snapshot of the current forest, with a random fire
A	Suppress or let-burn
$R(s, a)$	Timber harvest, Suppression Expense
$\gamma \in (0, 1)$	0.96 (Forest Service Standard)
P	Several Simulators
$\pi(s) \to a$	Suppress all fires

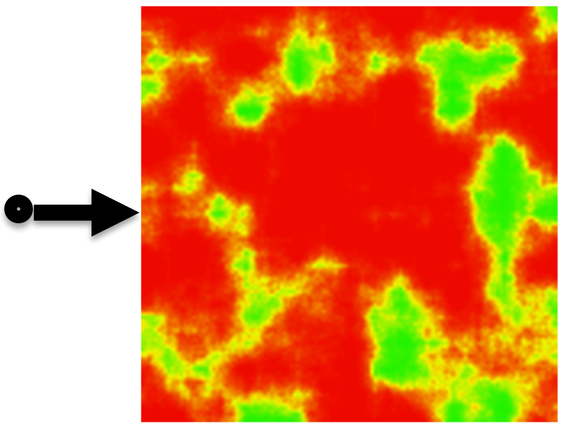
Represents a challenging and general class of MDPs

- High Dimensional States
- Large State Space
- Integrates Several Simulators

P_0 •

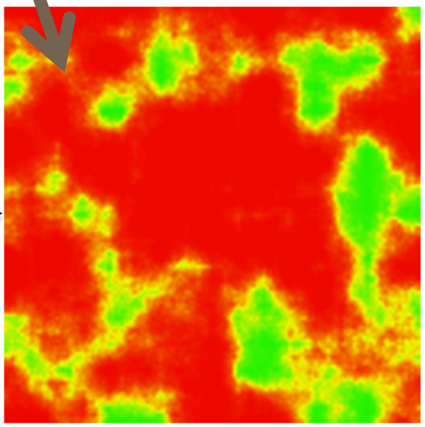
Start with Today's Landscape

P_0



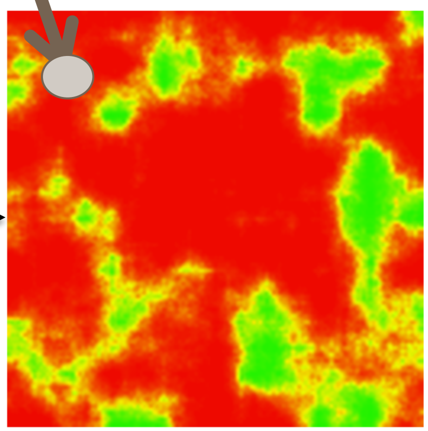
Generate an ignition and weather

P_0

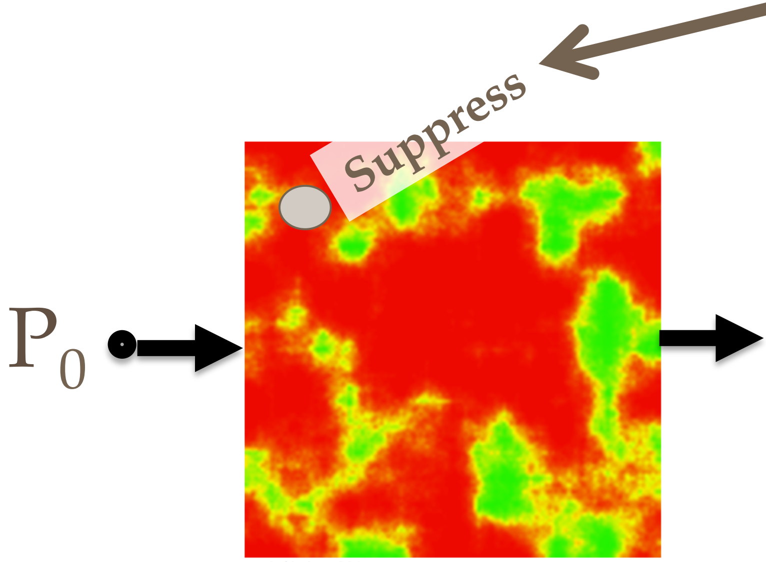


Generate an ignition and weather

P_0

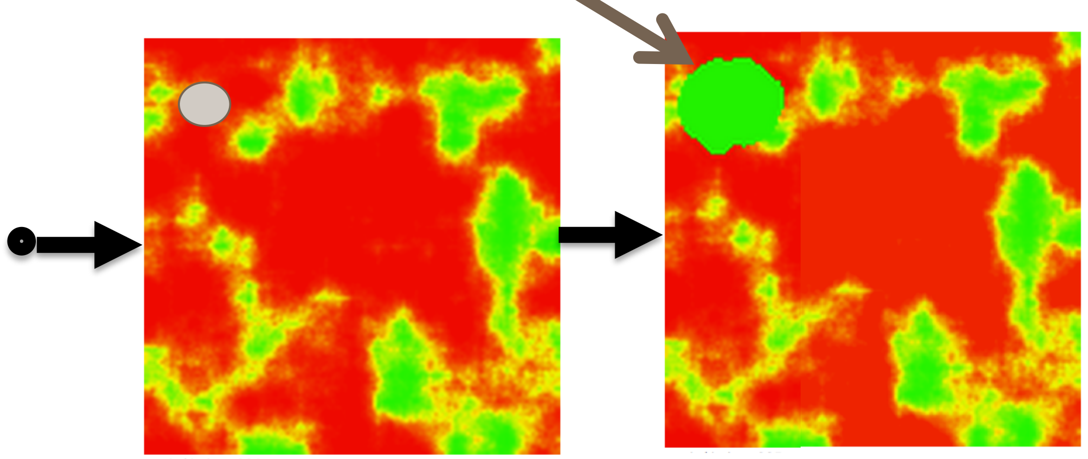


Select an Action



Update Vegetation for Wildfire

P_0

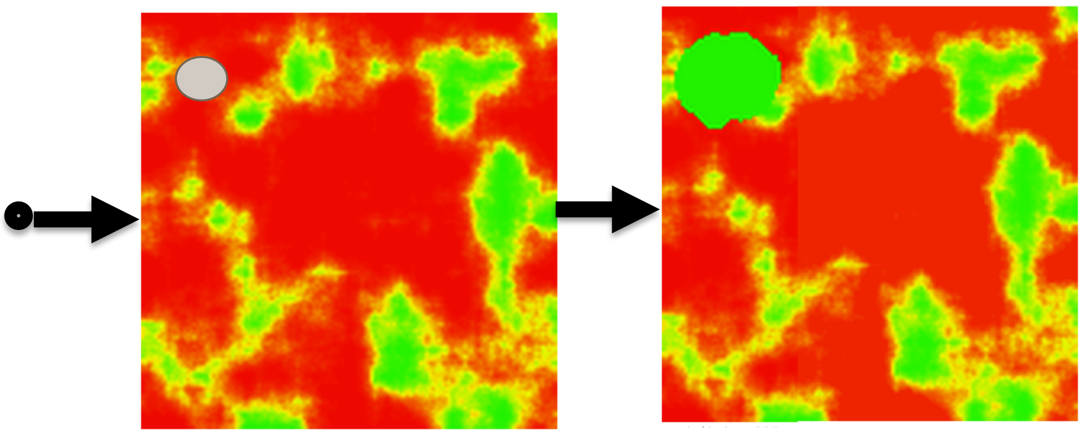


Fire Suppression Effort

Fire Suppression Costs

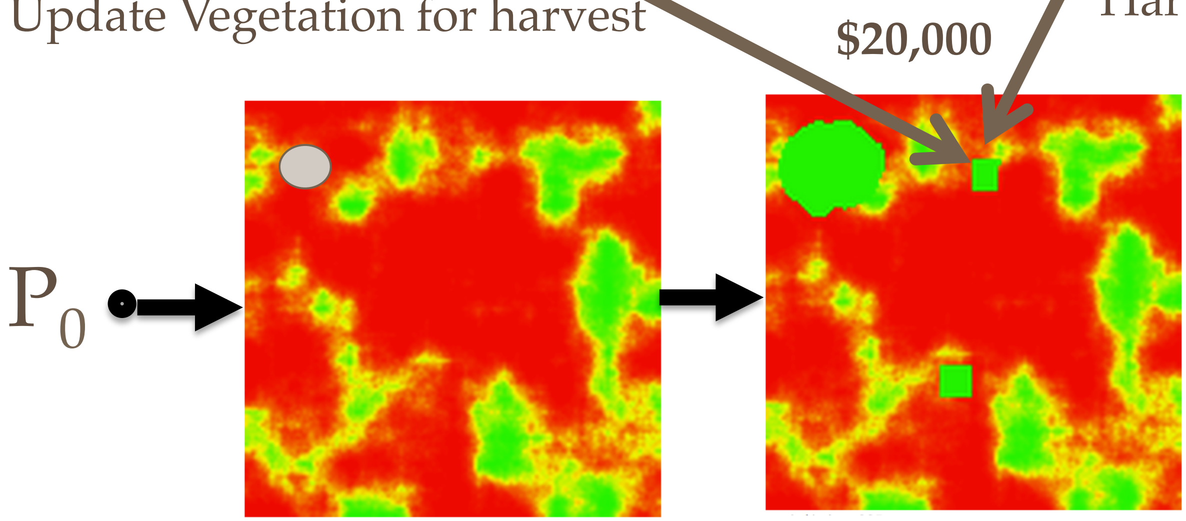
$(95,000)

P_0



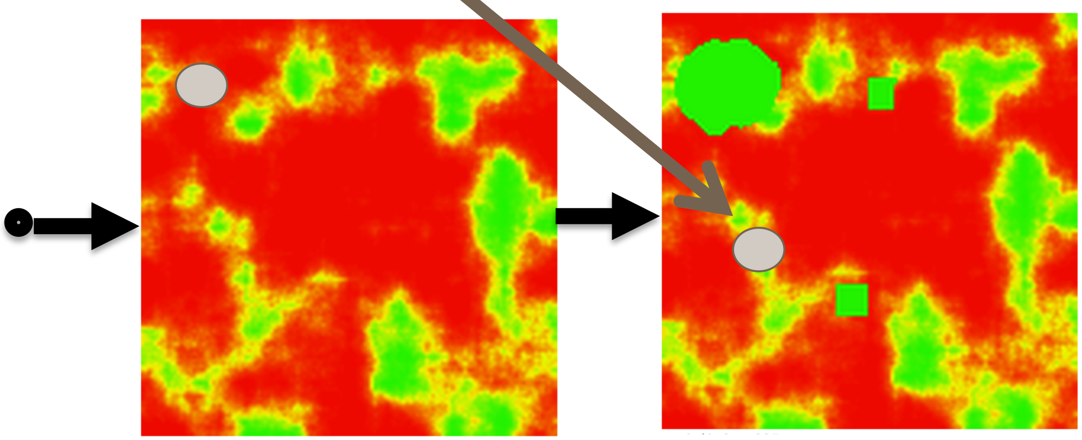
Update Vegetation for harvest

Harvest Revenue



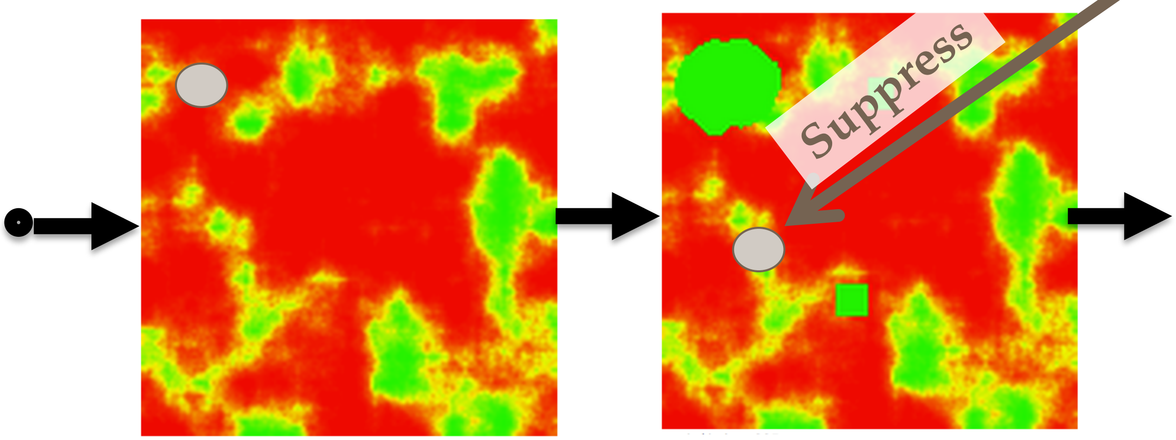
Generate an ignition and weather

P_0

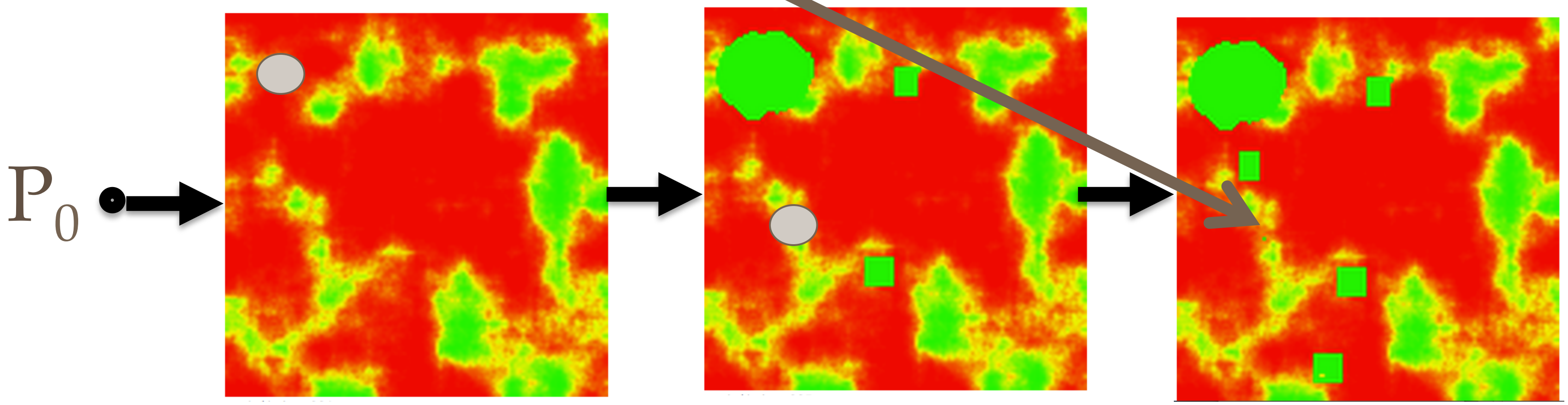


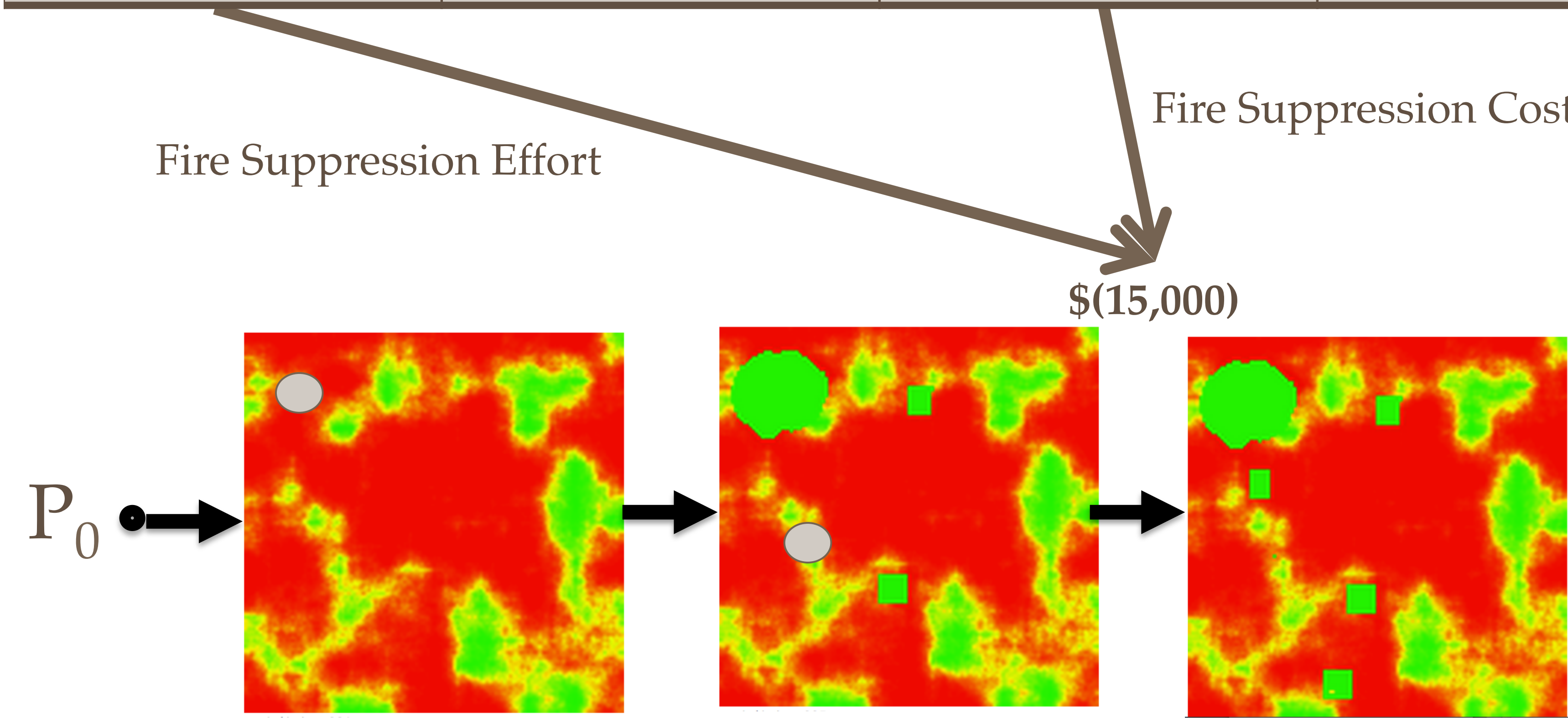
Select an Action

P_0



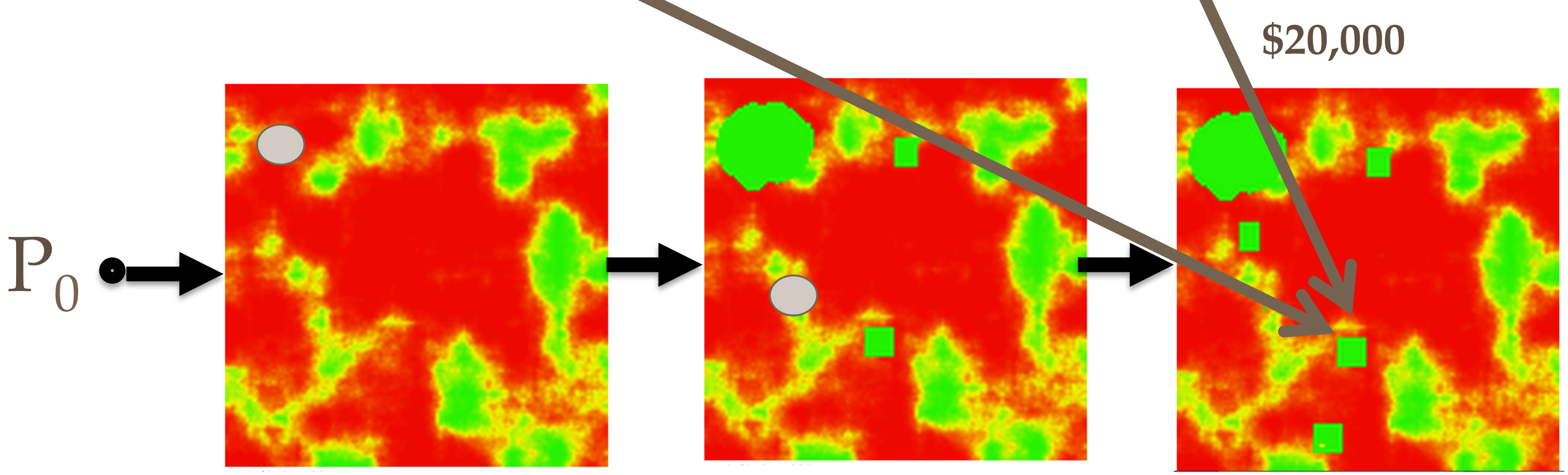
Update Vegetation for Wildfire



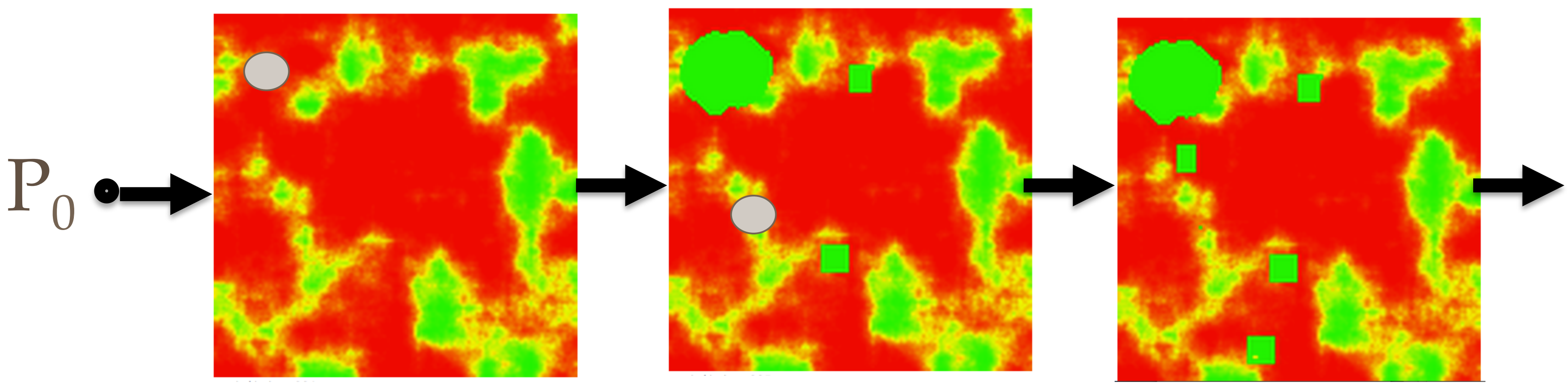

Fire Suppression Costs
Fire Suppression Effort
$(15,000)
P0

Update Vegetation for Harvest

Harvest Revenue

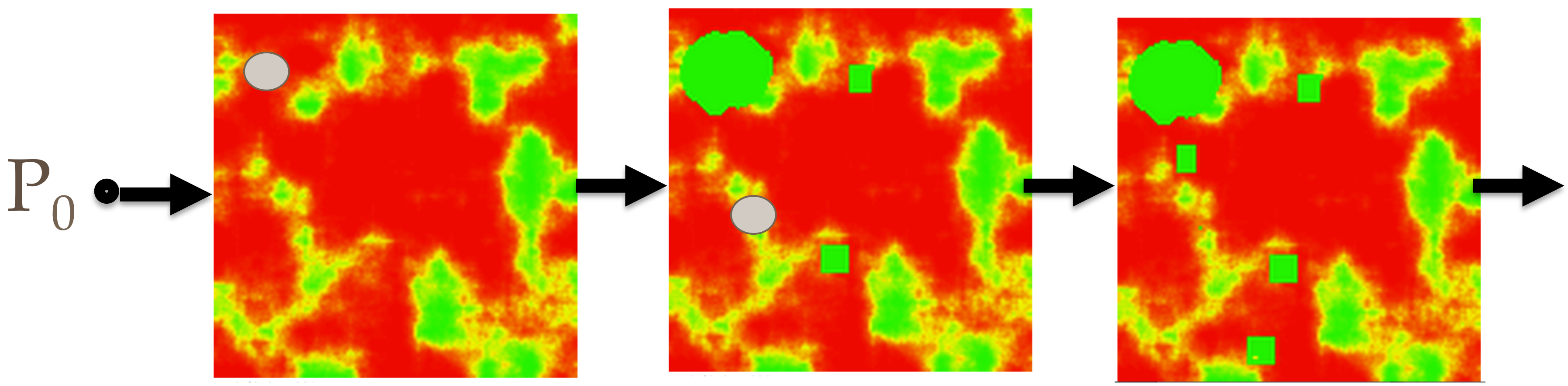


(Continue Until Reaching the Horizon)



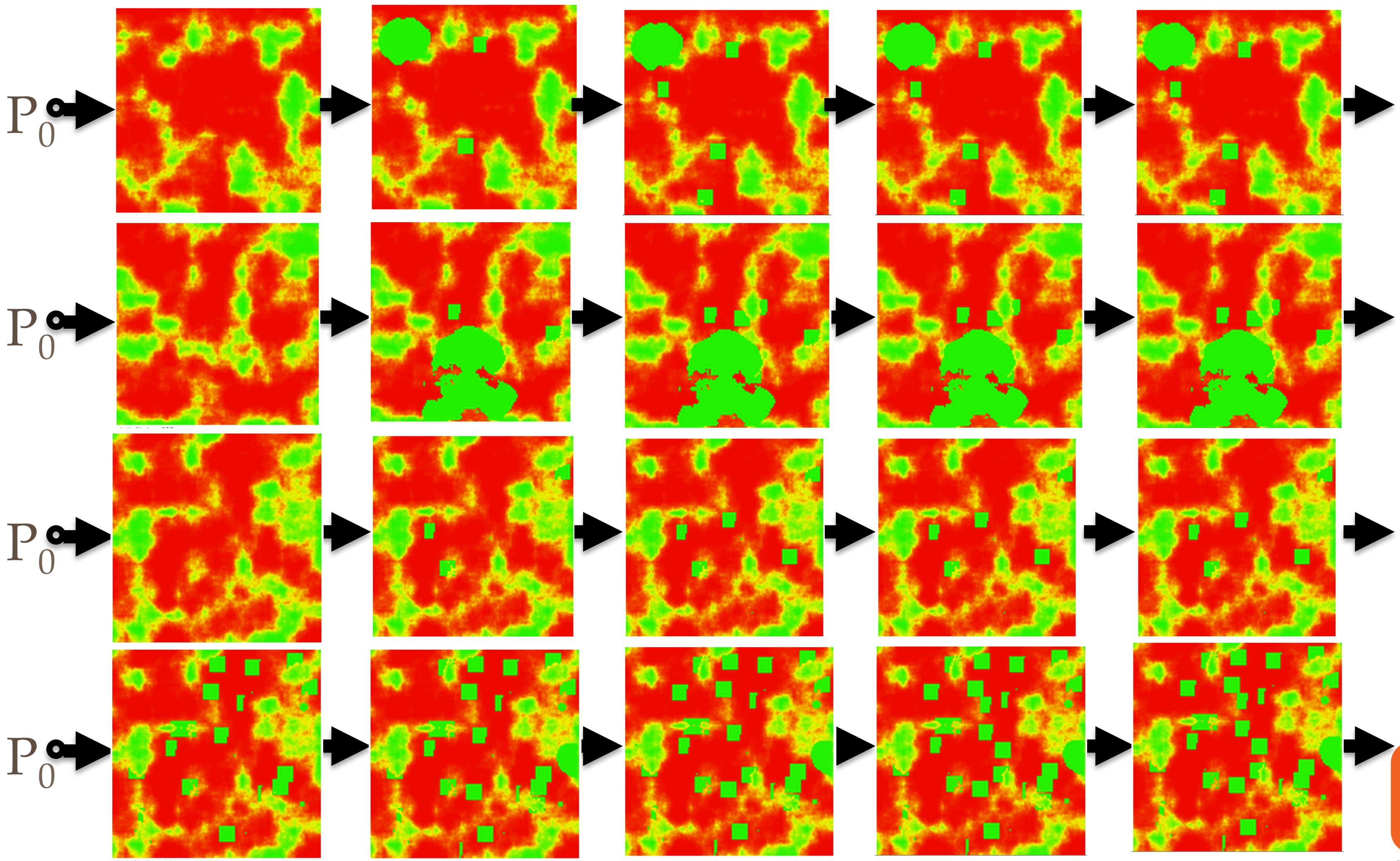
Oregon State University

A High Dimensional Probabilistic Time Series

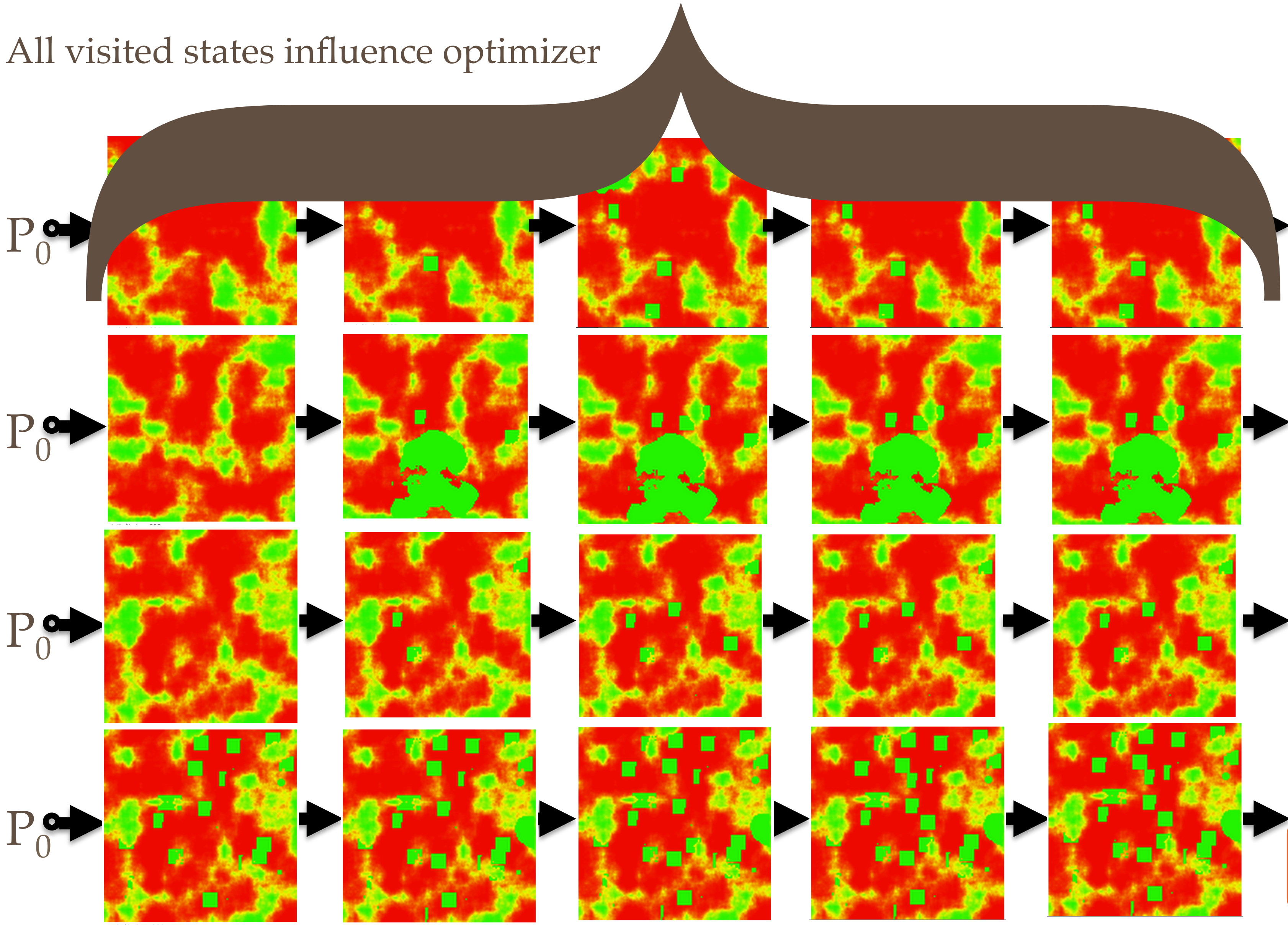


…And this is just one of many!

Monte Carlo Rollouts

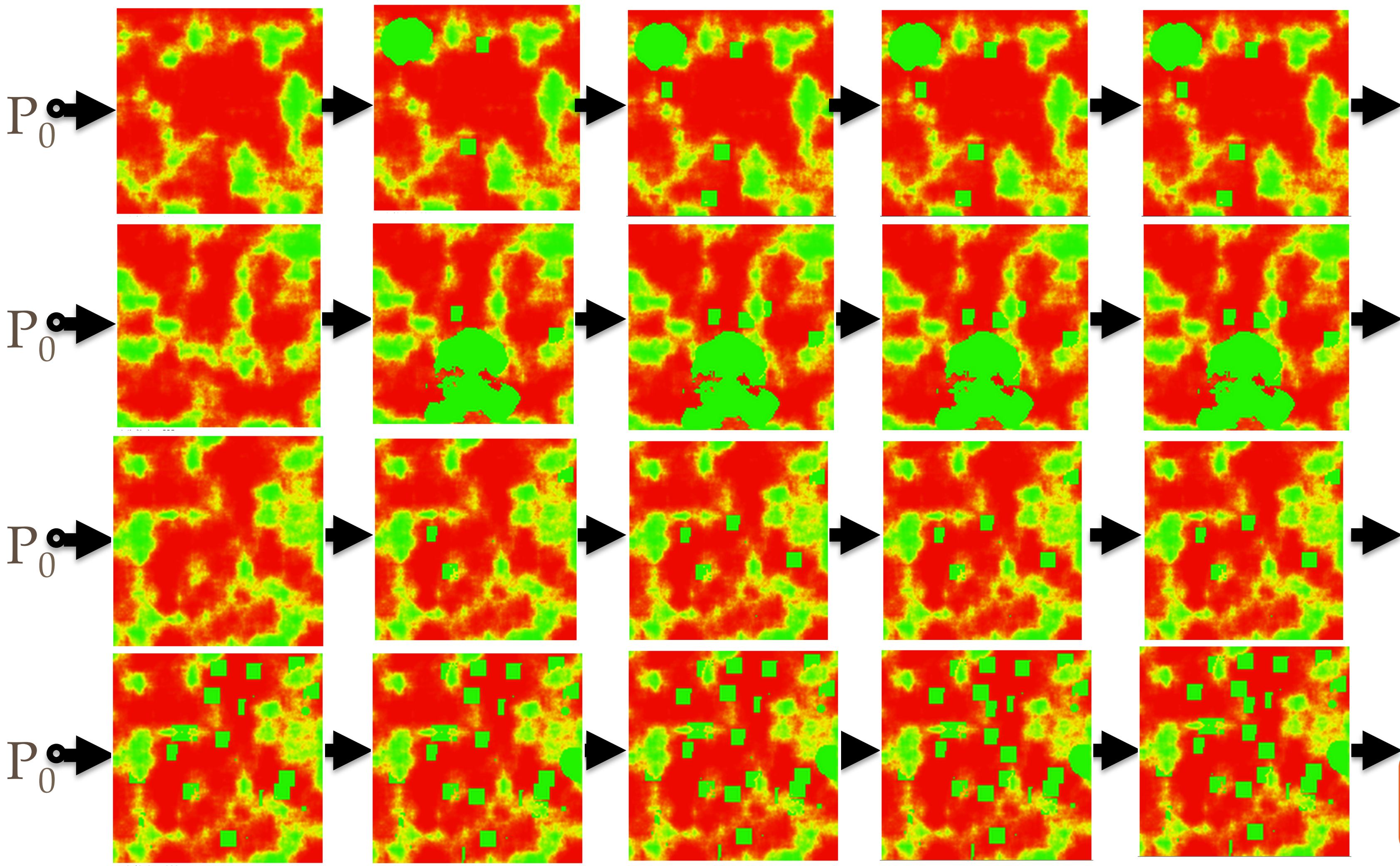


All visited states influence optimizer



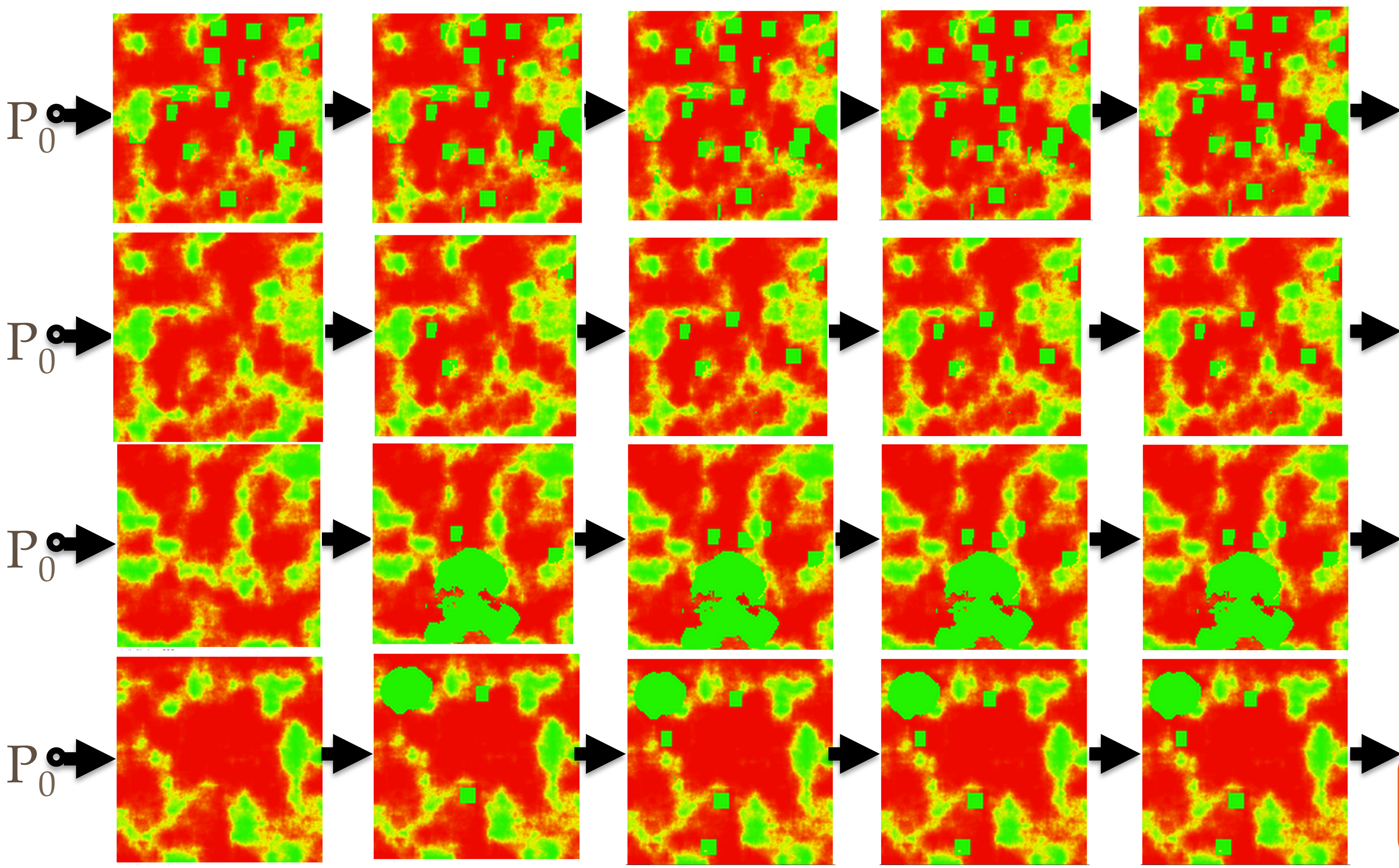
Simulators | Optimizer | Rewards | Policy

Update Policy



OSU Oregon State UNIVERSITY

The Rollout Distribution Changes!



Oregon State UNIVERSITY

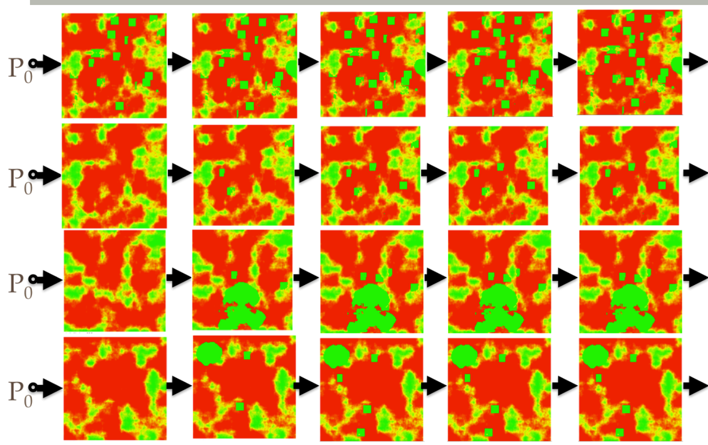
MDP Testing Challenges

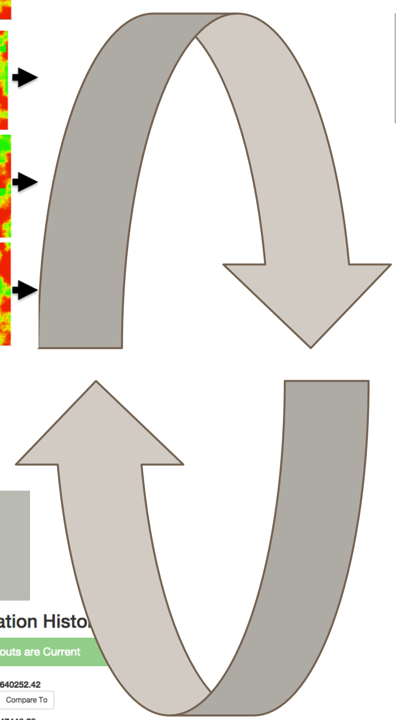
- Bugs are **probabilistically** expressed in a **high dimensional temporal dataset.**
- The **dataset changes** with changes to parameters.
- The **optimizer sees more of the state and policy space** than the user.

Testing requires exploring rollouts and parameters

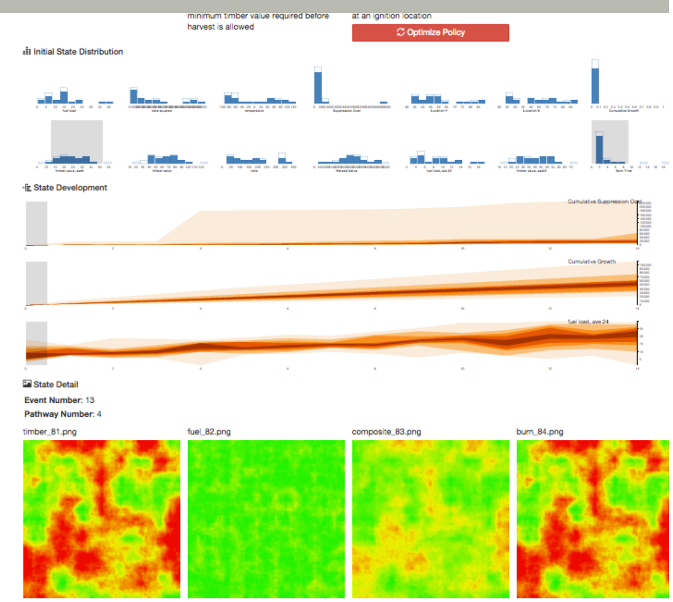
Testing and Debugging Process

1. Generate Rollouts

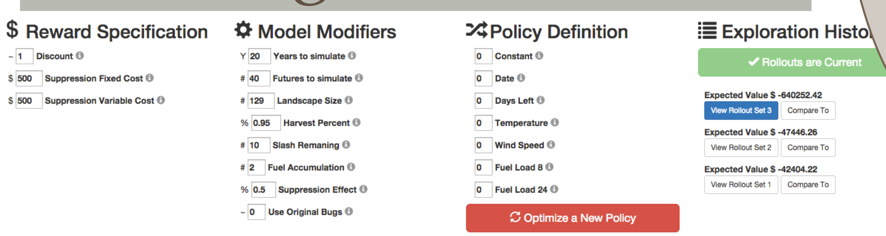




2. Visualize the data



3. Change Parameters

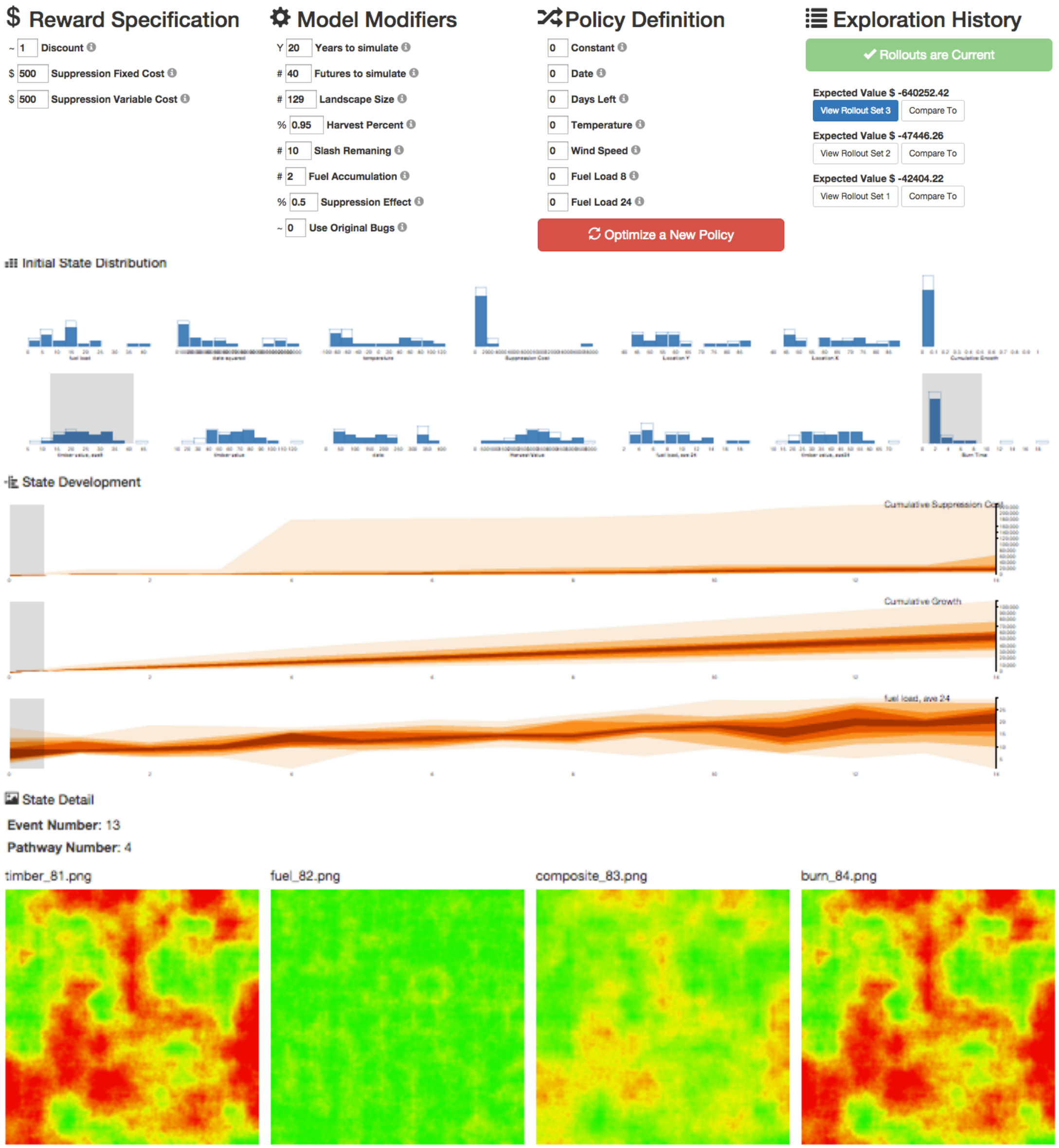


OSU Oregon State UNIVERSITY

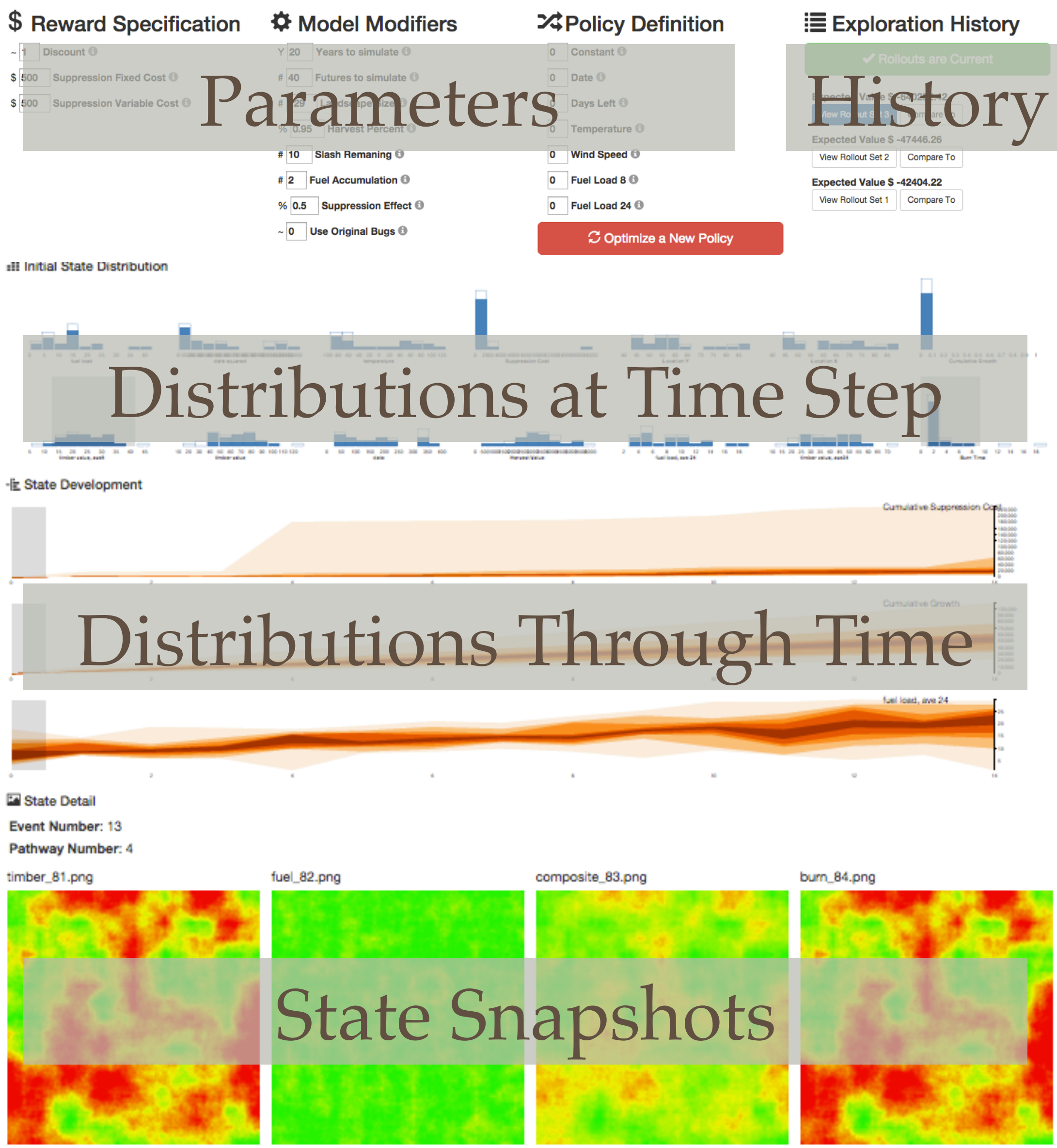
Outline

1. Wildfire Suppression MDP Example
   - Basic Introduction
   - Testing
2. MDPvis
   - **Design**
   - Testing Examples
   - MDPvis Use Case Study
   - Integrating Your Domain or Optimizer
3. Concluding

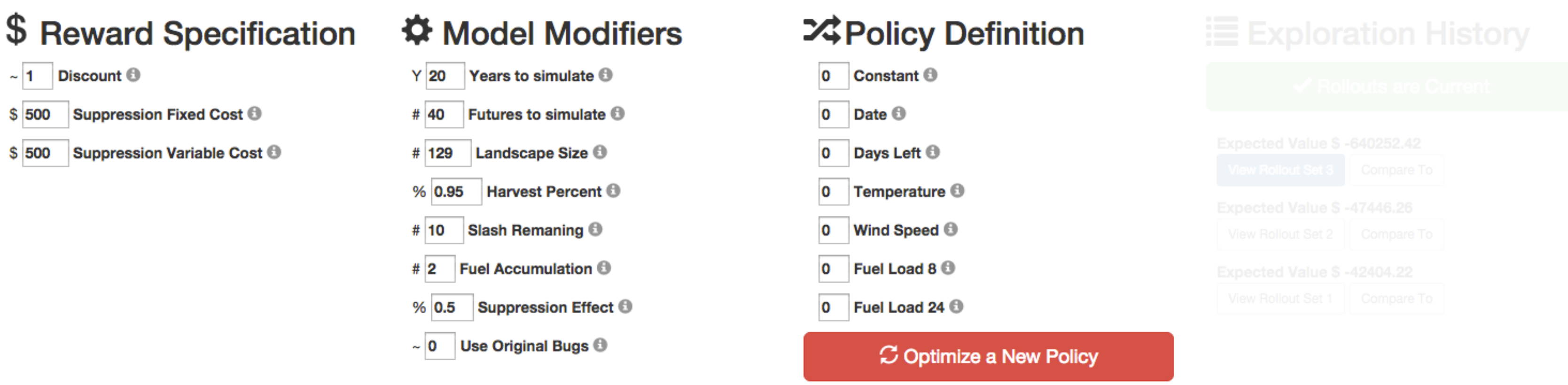
Introducing MDPvis



What are the elements of the MDPvis design?

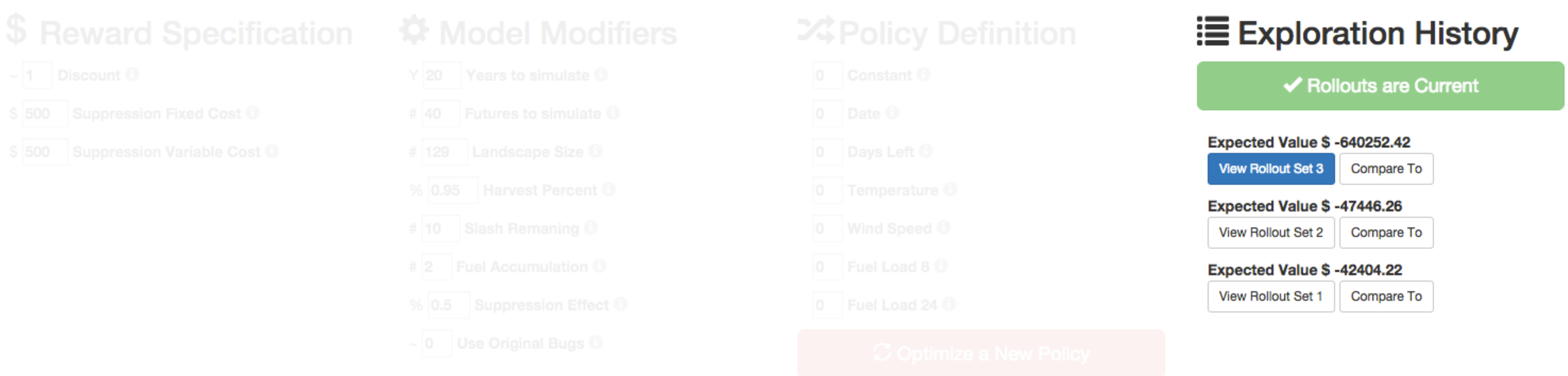


Parameter Areas



Oregon State UNIVERSITY

History Area

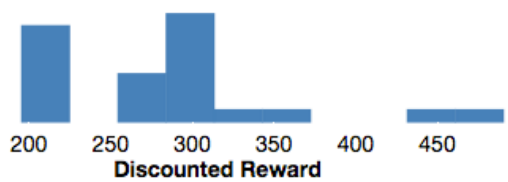


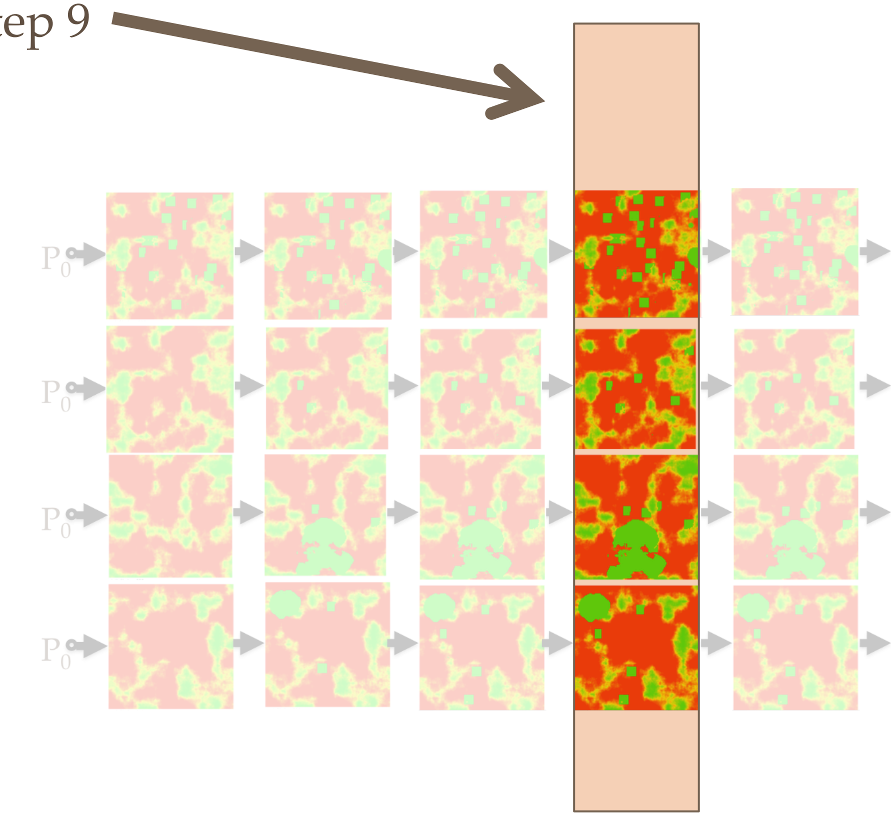
Oregon State UNIVERSITY

Visualization Areas

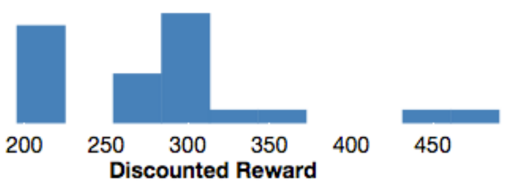
State Variable Distributions for a Fixed Time Step

Time step 9

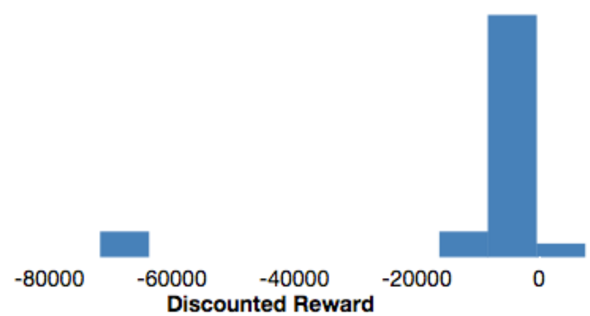




State Variable Distributions for a Fixed Time Step



π_1: Let-Burn

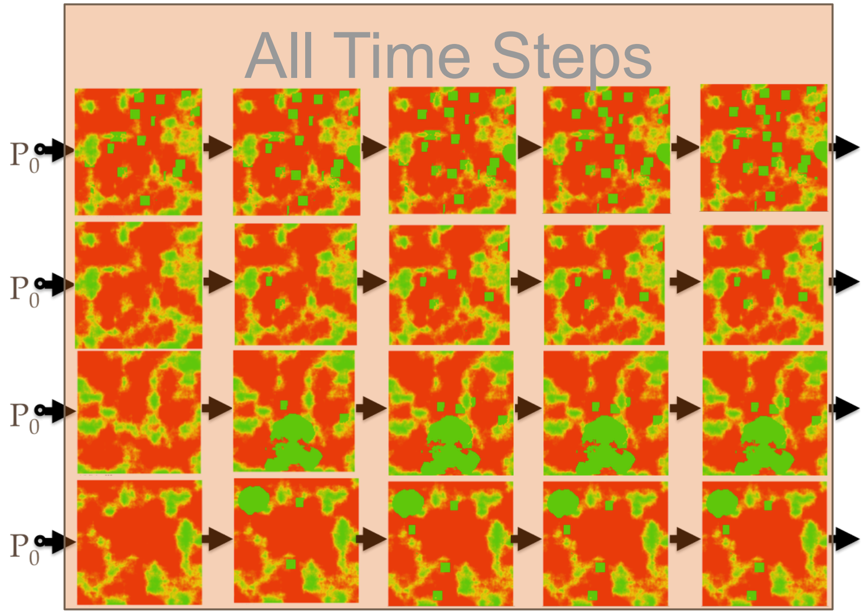


π_2: Suppress-All

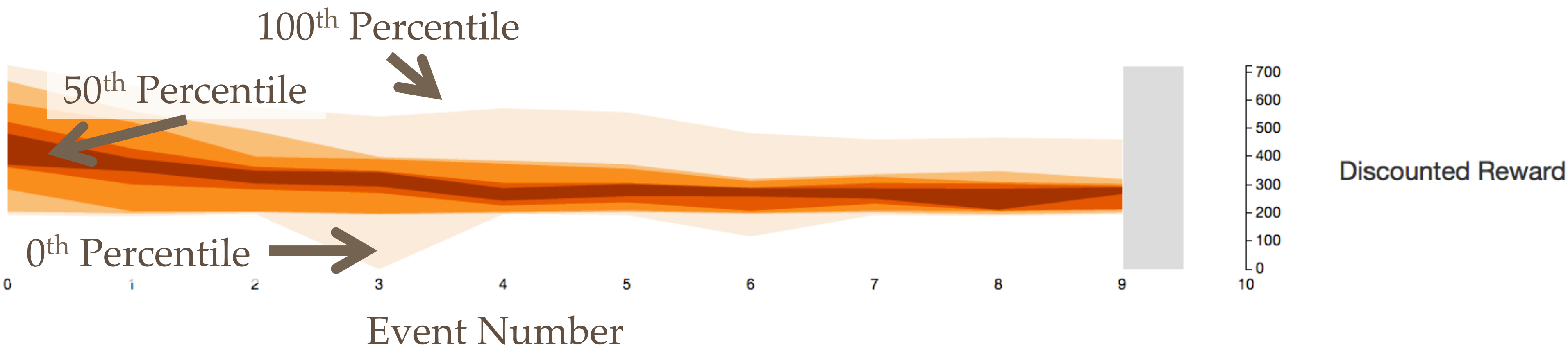
Comparison

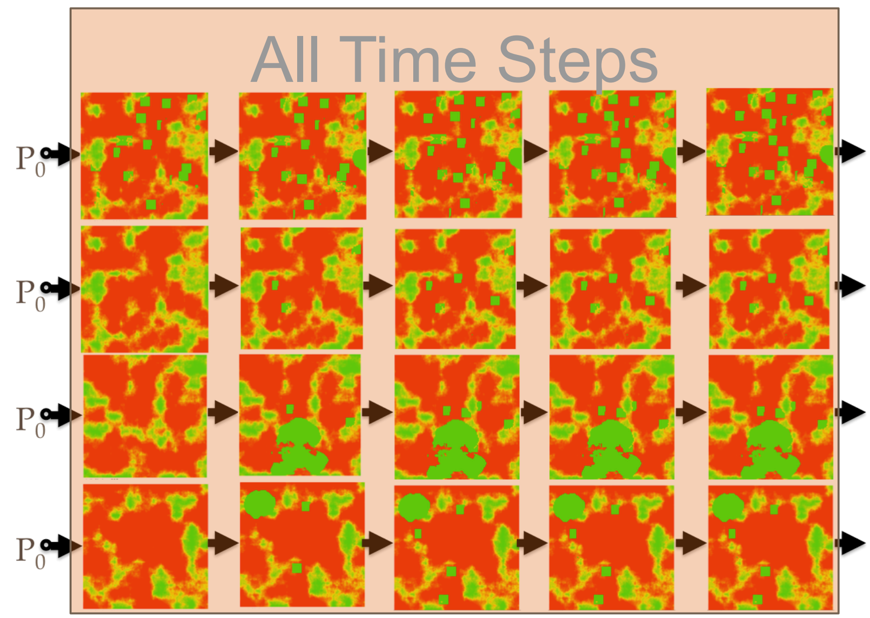
$\pi_1 - \pi_2$

State Variable Distributions through Time

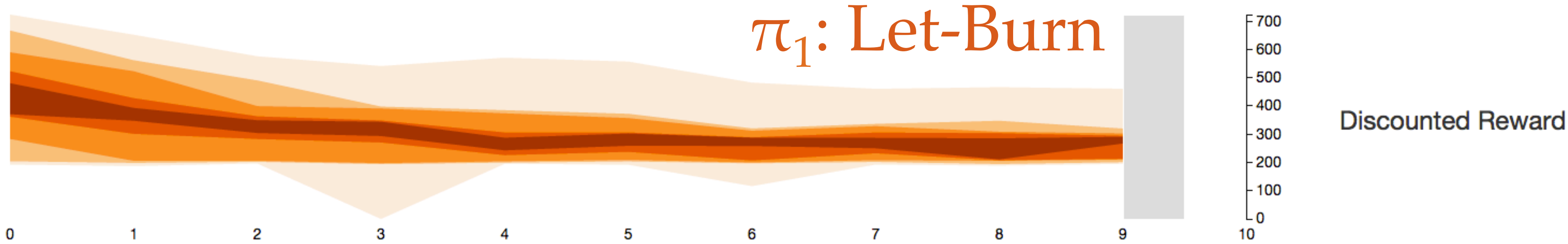


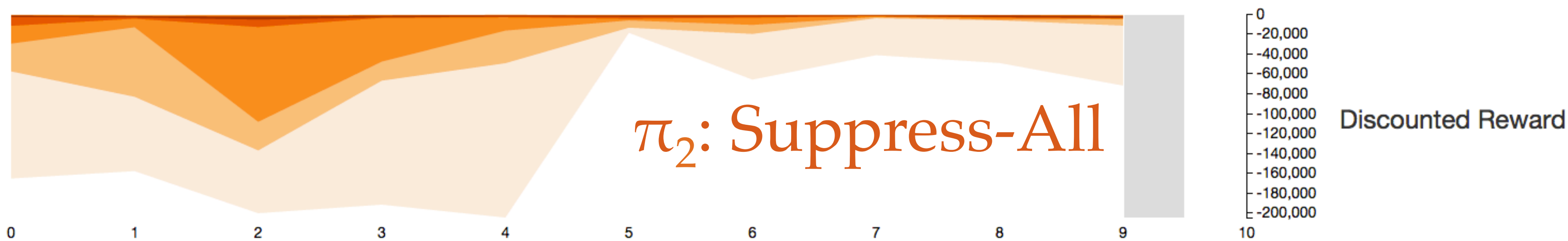
State Variable Distributions through Time

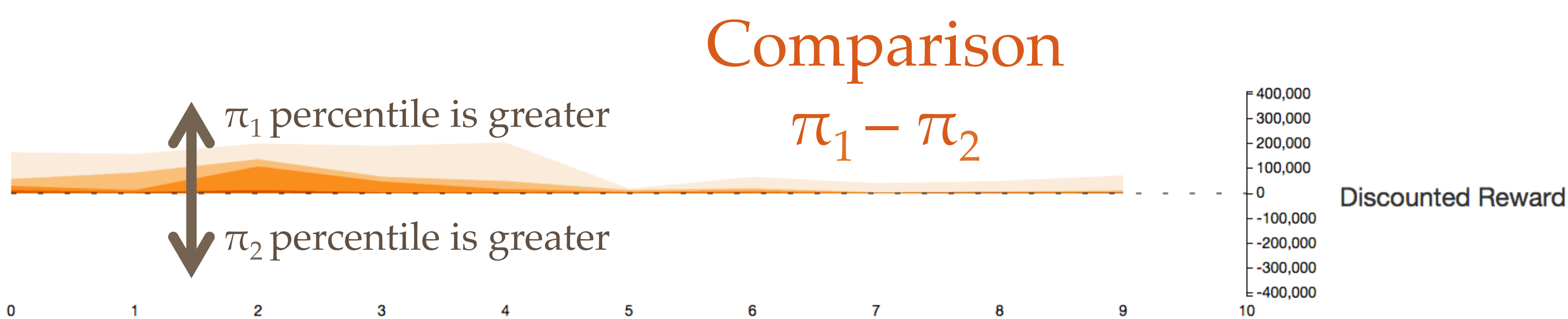




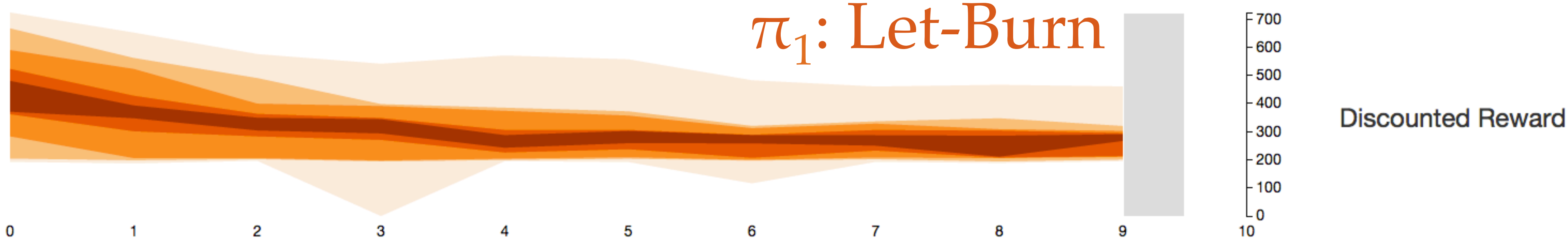
State Variable Distributions through Time

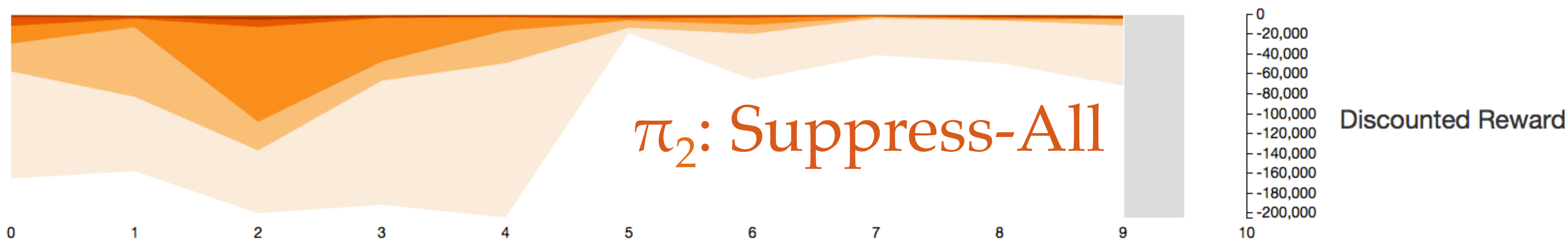


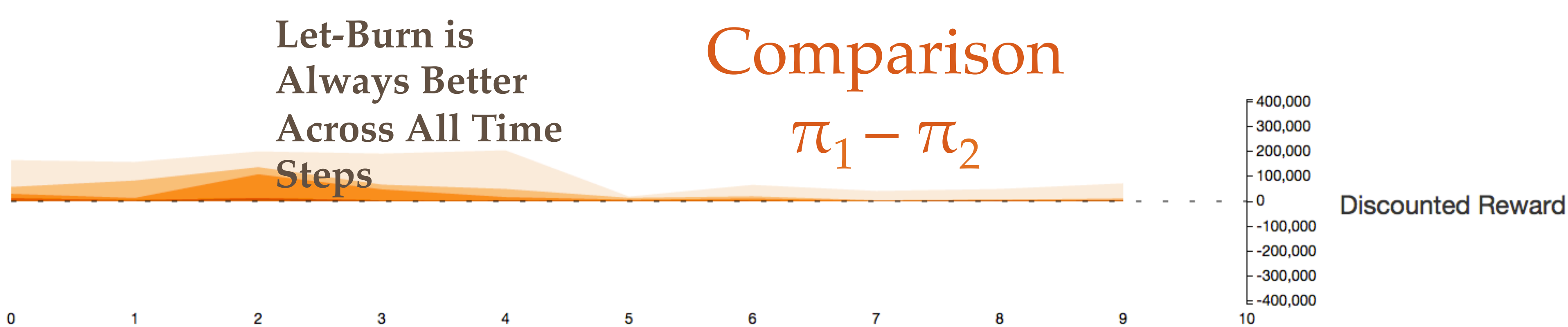




State Variable Distributions through Time

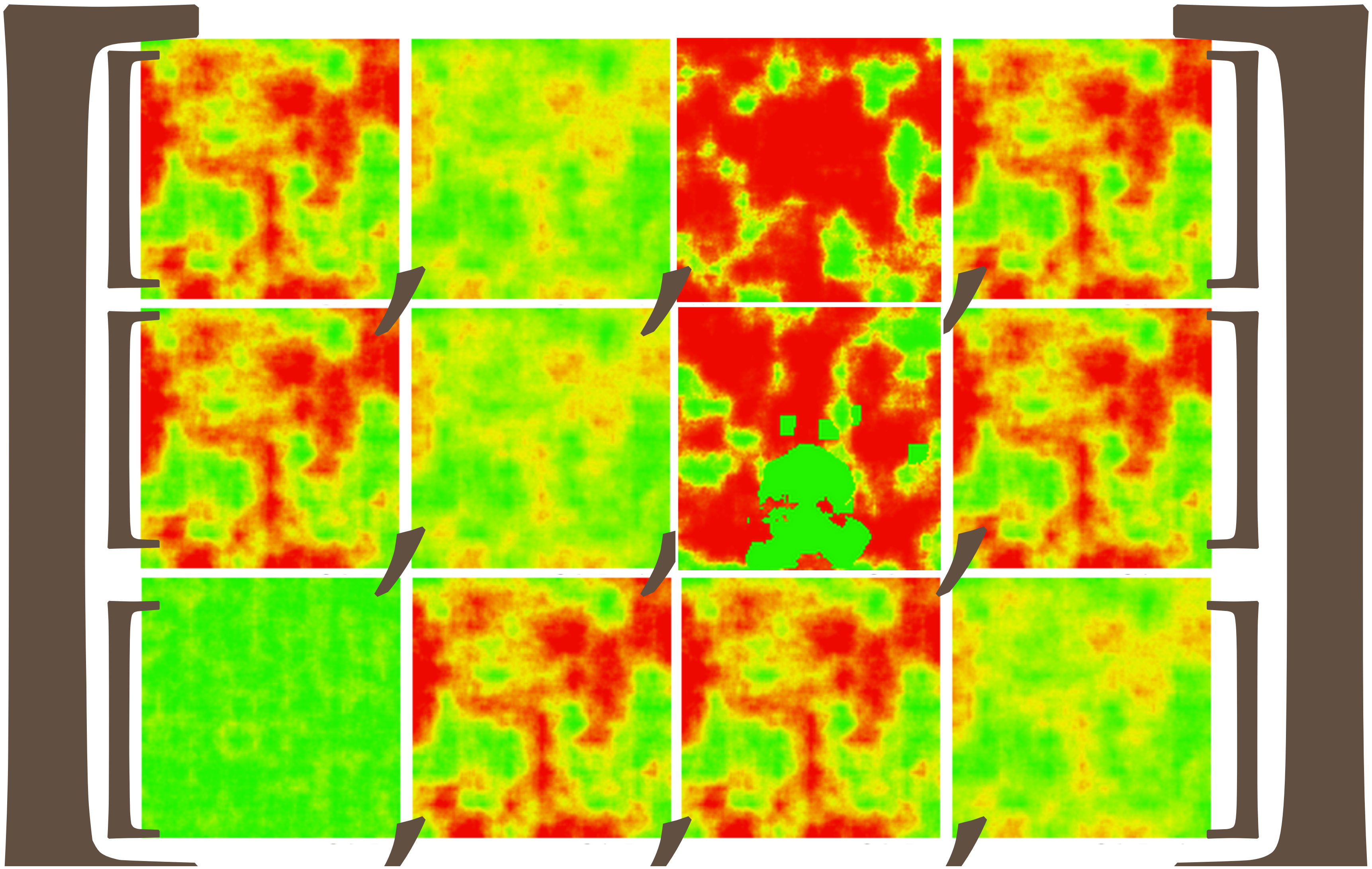






State details

Allow MDP Simulator to Generate State Visualizations



Outline

1. Wildfire Suppression MDP Example
   - Basic Introduction
   - Testing
2. MDPvis
   - Design
   - **Testing Examples**
   - MDPvis Use Case Study
   - Integrating Your Domain or Optimizer
3. Concluding

Parameter Space Analysis (PSA)

"[PSA] *is the systematic variation of model input parameters, generating outputs for each combination of parameters, and investigating the relation between parameter settings and corresponding outputs.*"

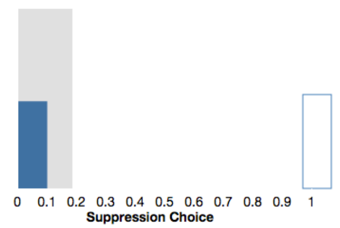
Categories
Sensitivity
Optimization
Outliers
Partition
Uncertainty
Fitting

Sedlmair, M., Heinzl, C., Bruckner, S., Piringer, H., & Möller, T. (2014). Visual parameter space analysis: A conceptual framework. Visualization and Computer Graphics, IEEE Transactions on, 20(12).

Is the suppression decision sensitive to the date of the fire?

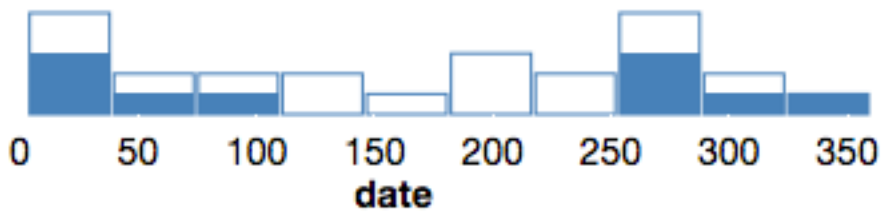
Interaction

1. Select states where fire is allowed to burn



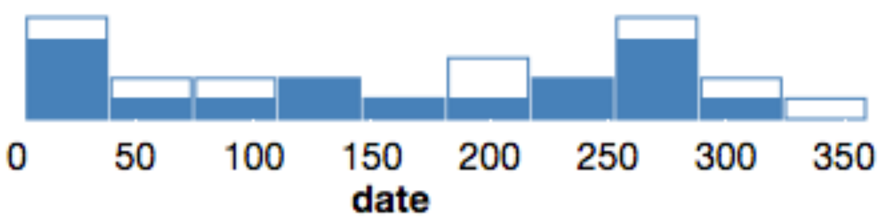
Expectation

2. Date is a determinant of suppression choice



Buggy Result

3. Date does not determine suppression choice



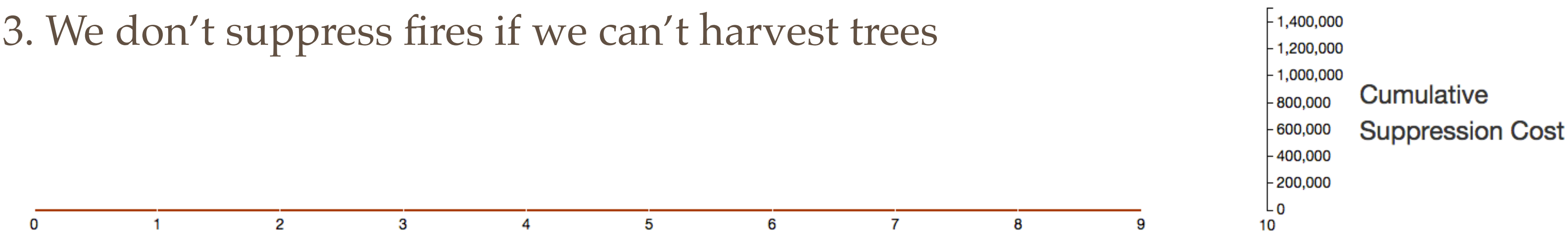
Is the optimization sensitive to the reward signal?

Interaction

1. Zero-out harvest rewards % 0 Harvest Percent
2. Re-optimize and generate rollouts Optimize a New Policy

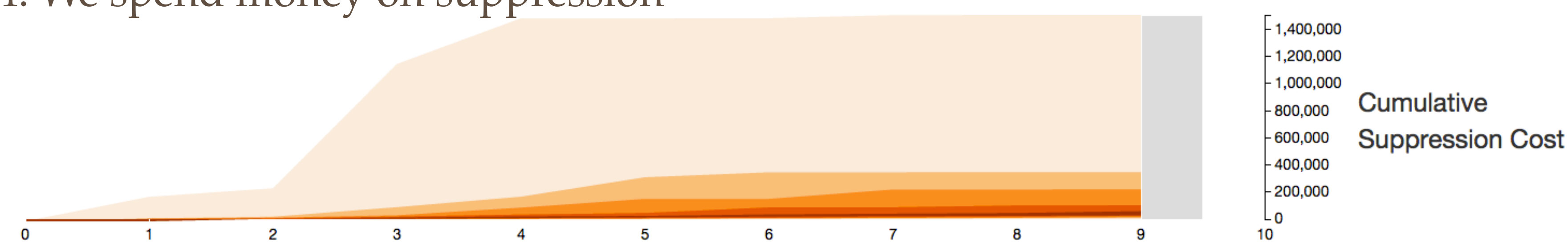
Expectation

3. We don't suppress fires if we can't harvest trees



Buggy Result

4. We spend money on suppression



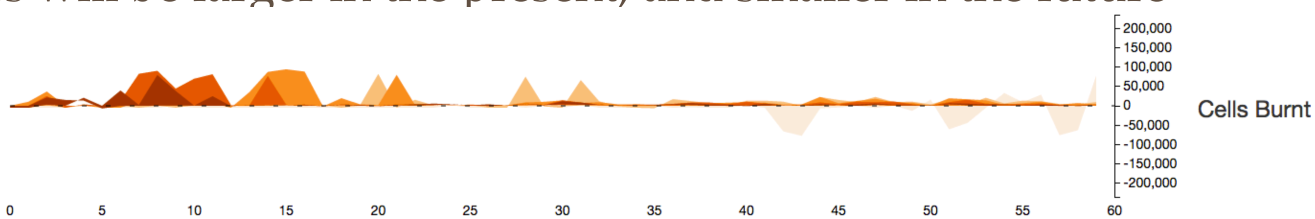
Does the let-burn policy have bigger initial fires and smaller subsequent fires?

Interaction

1. Generate suppress-all rollouts
2. Generate let-burn-all rollouts
3. Click the "compare rollouts" button

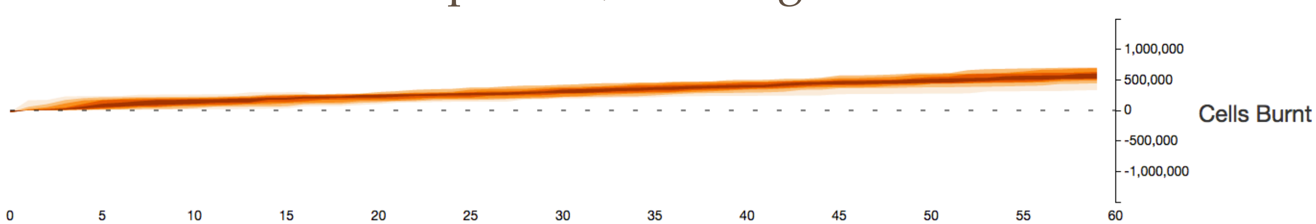
Expectation

4. Fires will be larger in the present, and smaller in the future



Buggy Result

5. Fires are the same in the present, and larger in the future



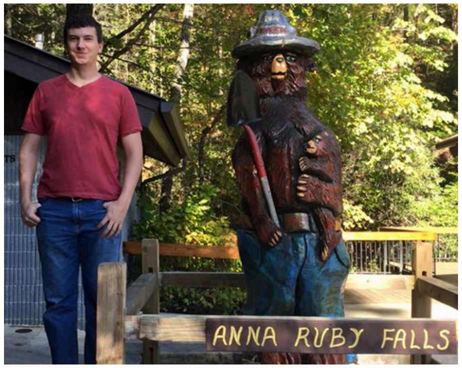
Outline

1. Wildfire Suppression MDP Example
   - Basic Introduction
   - Testing
2. MDPvis
   - Design
   - Testing Examples
   - **MDPvis Use Case Study**
   - Integrating Your Domain or Optimizer
3. Concluding

Use Case Study of MDPvis

We tested a new wildfire policy domain

Wildfire Optimization Expert
(Faculty Research Assistant)



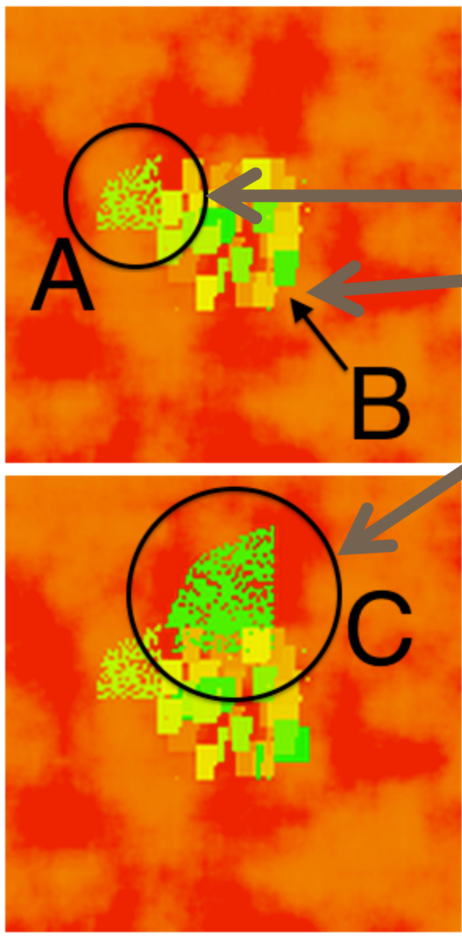
Visualization Developer
(Ph.D. Student in Computer Science)

New Fire Domain Developer
(Ph.D. Student in Forestry)

We found numerous bugs

Oregon State UNIVERSITY

Evaluation of MDPvis



Viewed Largest Fires in Rollouts

Second Largest Fire

Harvest Areas

Largest Fire

Fires are not spreading east!

Hidden except in most extreme fire by harvests

Evaluation of MDPvis

Interaction

1. Compare rollouts from two policies

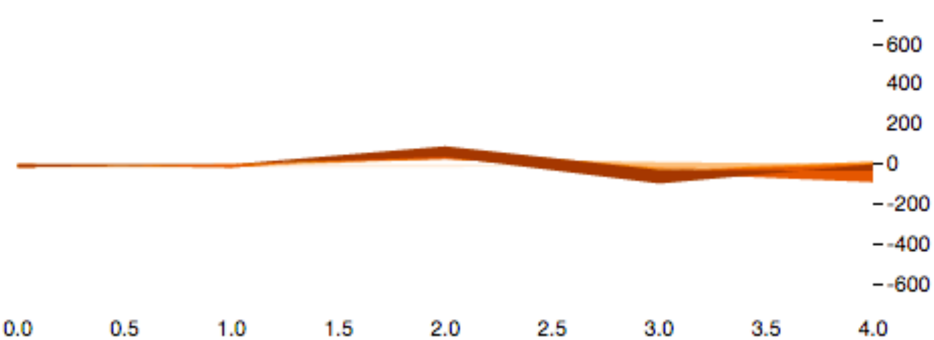
Expectation

2. Fire dates do not change between policies

Ignition Date

Buggy Result

3. Policies choose the weather



Ignition Date

Outline

1. Wildfire Suppression MDP Example
   - Basic Introduction
   - Testing
2. MDPvis
   - Design
   - Testing Examples
   - MDPvis Use Case Study
   - **Integrating Your Domain or Optimizer**
3. Concluding

Oregon State UNIVERSITY

Integrating MDPvis

4 HTTP Requests

1. */initialize*
2. */rollouts*
3. */optimize* (optional)
4. */state* (optional)

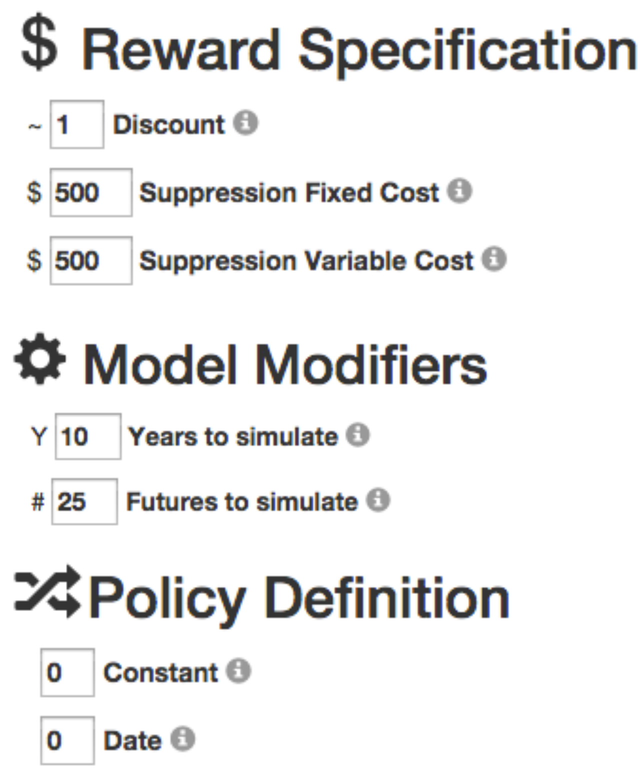
Oregon State UNIVERSITY

Integrating MDPvis

/initialize

```
"reward": [
        {"name": "Discount",
         "description":"The per-year discount",
         "current_value": 1, "max": 1, "min": 0, "units": "~"},
        {"name": "Suppression Fixed Cost",
         "description":"cost per day of suppression",
         "current_value": 500, "max": 999999, "min": 0, "units": "$"}
        ],
"transition": [
         {"name": "Years to simulate",
          "description": "how far to look into the future",
          "current_value": 10, "max": 150, "min": 0, "units": "Y"},
         {"name": "Futures to simulate",
          "description": "how many stochastic futures to generate",
          "current_value": 25, "max": 1000, "min": 0, "units": "#"}
         ],
"policy": [
        {"name": "Constant",
         "description":"for the intercept",
         "current_value": 0, "max": 10, "min":-10, "units": ""},
        {"name": "Date",
         "description":"for each day of the year",
         "current_value": 0, "max": 10, "min":-10, "units": ""}

       ]
```



Integrating MDPvis

/rollouts

```python
def rollouts(query):
    rollouts = []
    for rollout_number in range(0,200):
        rollout = getRollouts(rollout_number, query)
        formatted_rollout = formatRollout(rollout)
        rollouts.append(formatted_rollout)
    return rollouts
```

/optimize

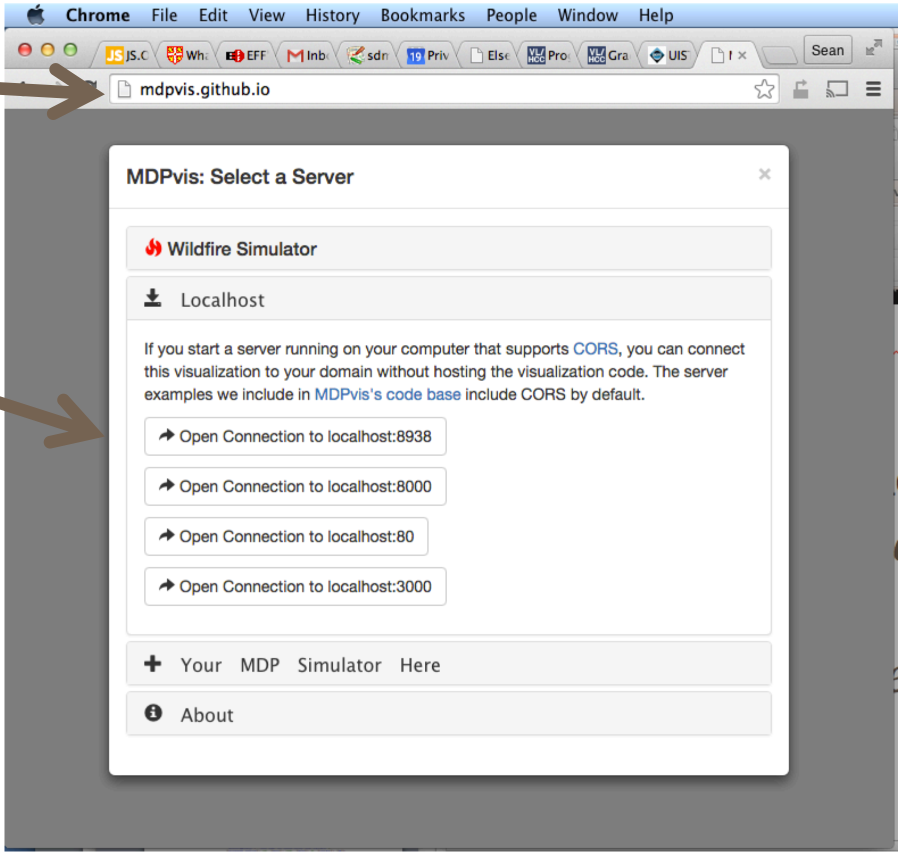
```python
def optimize(query):
    updated_parameters = optimize(query)
    return updated_parameters
```

/state

```python
def state(query):
    image_urls = getImages(query["rollout_number"], query)
    return image_urls
```

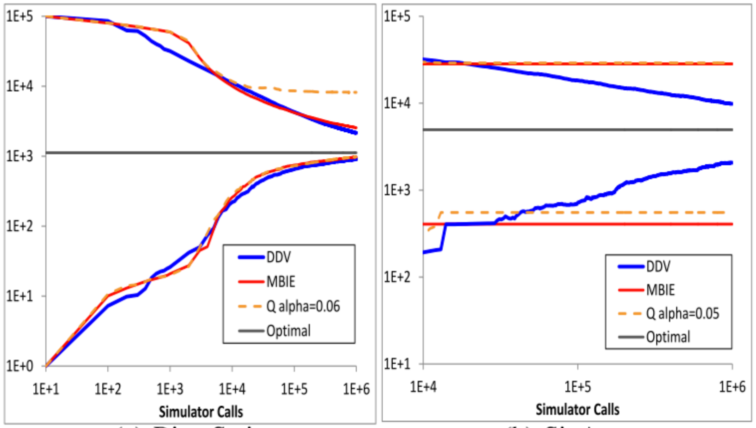
Integrating MDPvis

Connect the **remotely hosted** visualization to your **locally hosted** simulator and optimizer



Oregon State UNIVERSITY

Conclusion



(a) RiverSwim (b) SixArms

Figure 1: Learning curves for MBIE, Q-learning, and DDV as measured by confidence bounds on $V(s_0)$

Dietterich, T., Taleghan, M., & Crowley, M. (2013). PAC Optimal Planning for Invasive Species Management: Improved Exploration for Reinforcement Learning from Simulator-Defined MDPs. Twenty-Seventh AAAI Conference on Artificial Intelligence. Retrieved from http://www.aaai.org/ocs/index.php/AAAI/AAAI13/paper/download/6478/6850

Oregon State UNIVERSITY

Interactive Demo

MDPVis.github.io

* Not robust to many *simultaneous* requests

Oregon State UNIVERSITY

Thanks

- **Advisor:** Thomas Dietterich
- **Research Group:** Ronald Metoyer, Claire Montgomery, Rachel Houtman, Mark Crowley, Hailey Buckingham
- **Funder:** National Science Foundation

MDPVis.github.io

This material is based upon work supported by the National Science Foundation under Grant No. 1331932.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

OSU
Oregon State
UNIVERSITY

End.

Questions?

MDPVis.github.io

Contact
Email: VLHCC@SeanBMcGregor.com
Twitter: @SeanMcGregor

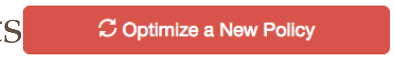
Oregon State UNIVERSITY

Outline

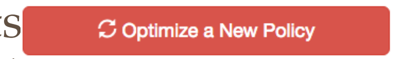
1. Wildfire Suppression MDP Example
   - Basic Introduction
   - Testing
2. MDPvis
   - Design
   - Integrating Your Domain or Optimizer
   - Testing Examples
   - MDPvis Use Case Study
3. **Concluding**

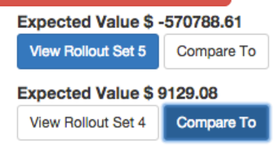
How consistent is the policy for small changes to the model?

Interaction

1. Optimize and generate rollouts 

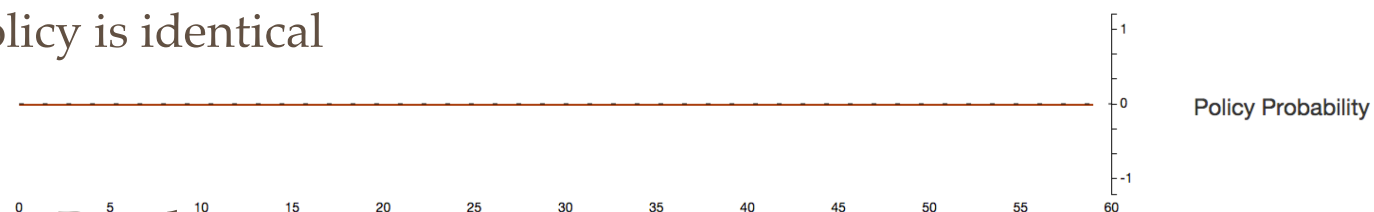
2. Add air tankers to the model


3. Optimize and generate rollouts 

4. Click the "Compare Rollouts" button


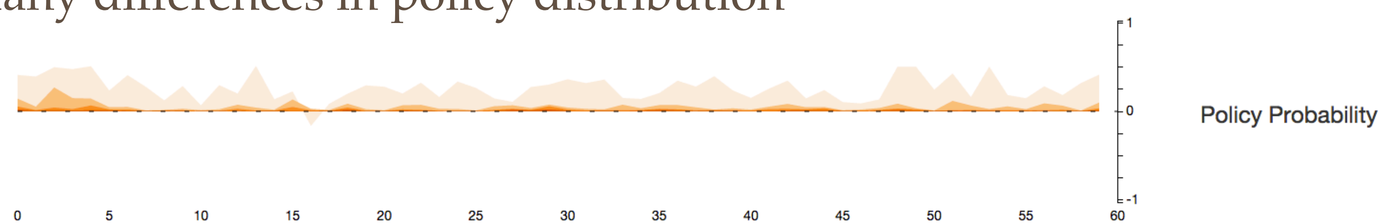
Expectation

5. Policy is identical



Buggy Result

6. Many differences in policy distribution



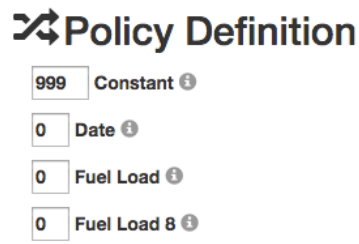
Does the growth rate match the historical dataset?

Pre-Process

1. Add a variable for the growth percentile within the historic data

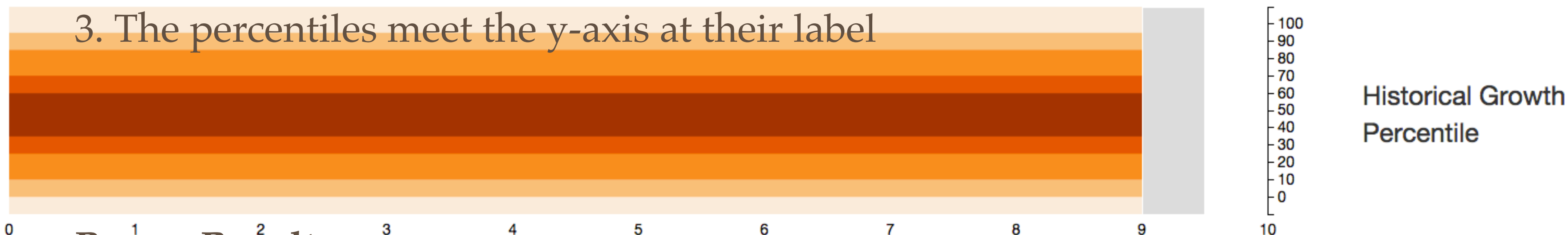
Interaction

2. Assign the policy to the historical policy (suppress all)



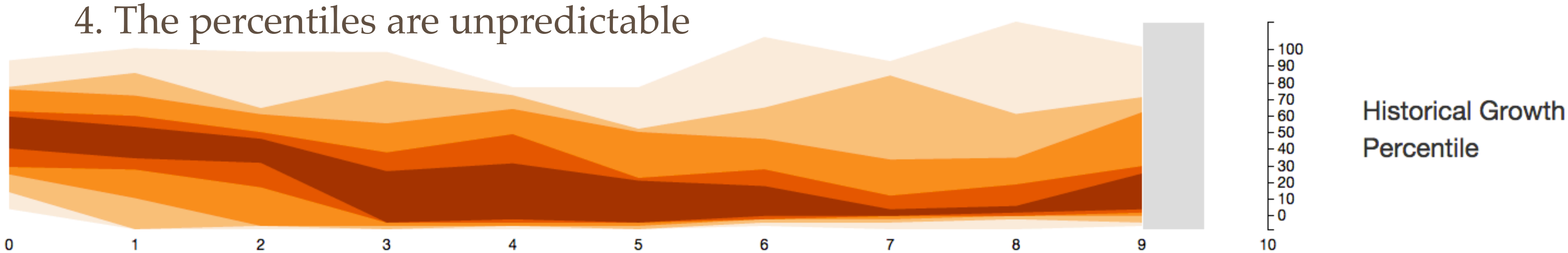
Expectation

3. The percentiles meet the y-axis at their label



Buggy Result

4. The percentiles are unpredictable



MDPvis Value Proposition

1. **Build understanding** of how policy performs
2. **Explore distributions and filter** to interesting rollouts
3. Easy integration of **your custom visualizations**
4. **Shorter experiment/analysis cycle** by connecting tools directly to implementation
5. Parameterizations are **shareable**
6. **Simple integration** with existing domains