Learning from Less: SINDy Surrogates in RL

Aniket Dixit[1] Muhammad Ibrahim Khan[2] Faizan Ahmed[3] James Brusey[4]

[1]Coventry University, United Kingdom

ICLR 2025

Outline

Motivation

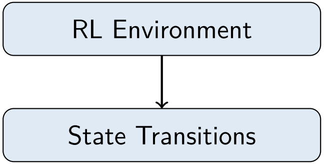
- Reinforcement Learning (RL) requires extensive environmental interactions
- Challenges:
  - High computational costs
  - Safety concerns in real-world applications
  - Limited sample efficiency
- Our approach: Develop efficient surrogate environments using SINDy
- SINDy = Sparse Identification of Nonlinear Dynamics

Contributions

1. Novel framework for creating SINDy-based surrogate environments
2. Significant data reduction while maintaining high fidelity
3. Demonstrated effectiveness on OpenAI Gym environments
    - Mountain Car
    - Lunar Lander
4. 20-35% reduction in computational costs

Data Collection

- Used pre-trained RL models to collect state transitions
- **Mountain Car**: 75 transitions (SAC agent)
- **Lunar Lander**: 1000 transitions (PPO agent)
- ϵ-greedy policy ($\epsilon = 0.2$) for exploration/exploitation balance
- Recorded (s_t, a_t, s_{t+1}) tuples



SINDy Model Development

Four-stage process:

1. **Initial Model Construction**
   - Fitted basic models using Sequential Thresholded Least Squares (STLSQ)
2. **Residual Analysis**
   - Added nonlinear terms based on prediction errors
   - Trigonometric terms for Mountain Car
   - Polynomial terms for Lunar Lander
3. **Parameter Optimization**
   - Grid search across threshold and regularization values
4. **Cross-Validation**
   - Final selection based on minimal MSE

SINDy-Driven Surrogate Environment

- Original physics engine replaced by SINDy model
- Preserves key characteristics of original implementations
- State transition function:

$$s_{t+1} = f_{SINDy}(s_t, a_t) \tag{1}$$

Algorithm 1 SD-RL Algorithm

// **Data Collection**
for $i = 1$ to N **do**
  Initialize environment
  for $t = 1$ to max timesteps **do**
    Select action a_t using ϵ-greedy
    Execute a_t, observe s_{t+1}, r_t
    Store (s_t, a_t, s_{t+1}, r_t) in D
  end for
end for
// **Train SINDy Model**
Train SINDy model using D
// **Train RL Agent in Surrogate**
Train RL agent using surrogate

SINDy Model Performance

Environment/Component	MSE	Correlation
Mountain Car		
Position	7.21×10^{-4}	0.999
Velocity	3.11×10^{-6}	0.997
Lunar Lander		
x/y-position	$1.42 \times 10^{-6}/9.64 \times 10^{-6}$	1.000/1.000
x/y-velocity	$1.58 \times 10^{-5}/3.38 \times 10^{-5}$	1.000/0.999
angle/angular vel.	$1.03 \times 10^{-5}/1.24 \times 10^{-4}$	0.999/0.989

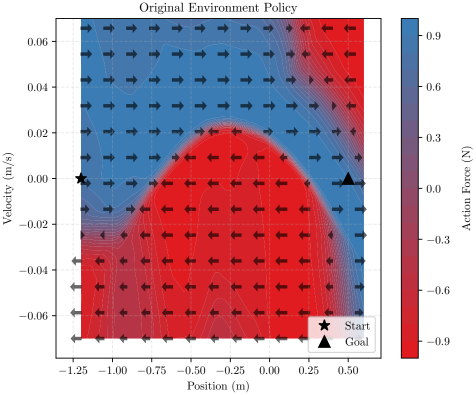
Library Function Impact

Library Function	MC MSE	LL MSE
Polynomials only	5.32×10^{-3}	1.58×10^{-4}
With trigonometric	3.11×10^{-6}	1.62×10^{-4}
With rational terms	4.15×10^{-4}	1.87×10^{-4}

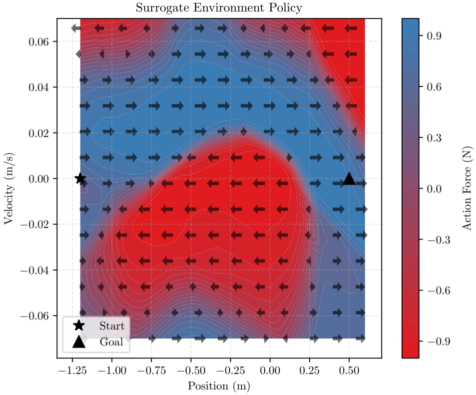
- **Trigonometric functions** crucial for Mountain Car
- **Polynomial terms** sufficient for Lunar Lander
- Adding rational terms did not improve performance

Policy Learning Results - Mountain Car

Original Environment



Surrogate Environment



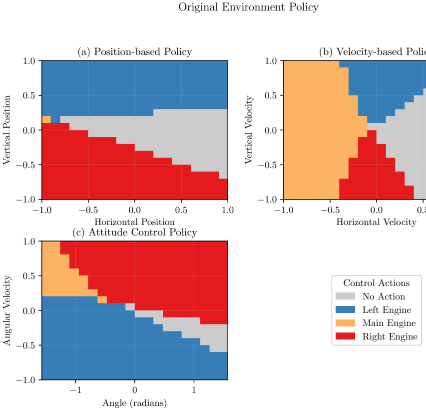
Policy Learning Results - Mountain Car (Key Findings)

Key Findings:

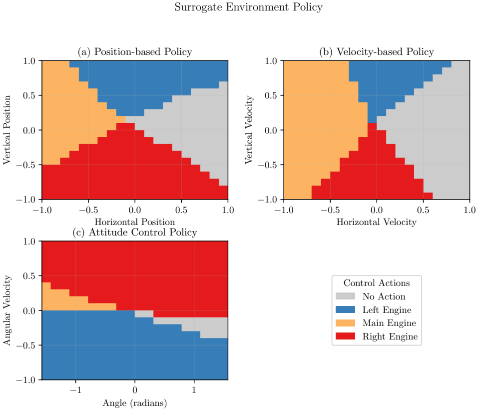
- Identical momentum building in valley regions (blue)
- Similar oscillatory behavior in middle regions
- Consistent stabilization near goal (red)

Policy Learning Results - Lunar Lander

Original Environment



Surrogate Environment



Policy Learning Results - Lunar Lander (Key Findings)

Key Findings:

- Similar engine activation patterns across dimensions
- Slight tactical differences in descent control
- No compromise to landing performance

Computational Efficiency

Environment	Original	SINDy	Reduction
Mountain Car	100,000	65,075	35%
Lunar Lander	1,000,000	801,000	20%

Table: Training steps comparison

- Minimal data collection requirements
  - 75 state transitions (Mountain Car)
  - 1,000 state transitions (Lunar Lander)
- Superior accuracy vs. neural networks
  - 95% less computational resources
  - Better MSE: 3.11×10^{-6} vs. 4.45×10^{-6}

Conclusion & Future Work

Key Results

- Exceptional fidelity (correlations > 0.99) with minimal data
- Critical importance of appropriate library functions
- Near-identical policies confirm preservation of essential dynamics
- 20-35% computational efficiency gains with interpretability advantages

Future Directions

- Scale to higher-dimensional state spaces
- Test generalization capability to different initial conditions
- Develop hybrid approaches combining SINDy with other techniques
- Validate on physical systems for real-world applications

Thank you!

Questions?

Contact: dixita4@uni.coventry.ac.uk