Learning Energy Efficient Jumping Strategies for Flexible-Legged Systems

MECC 2021

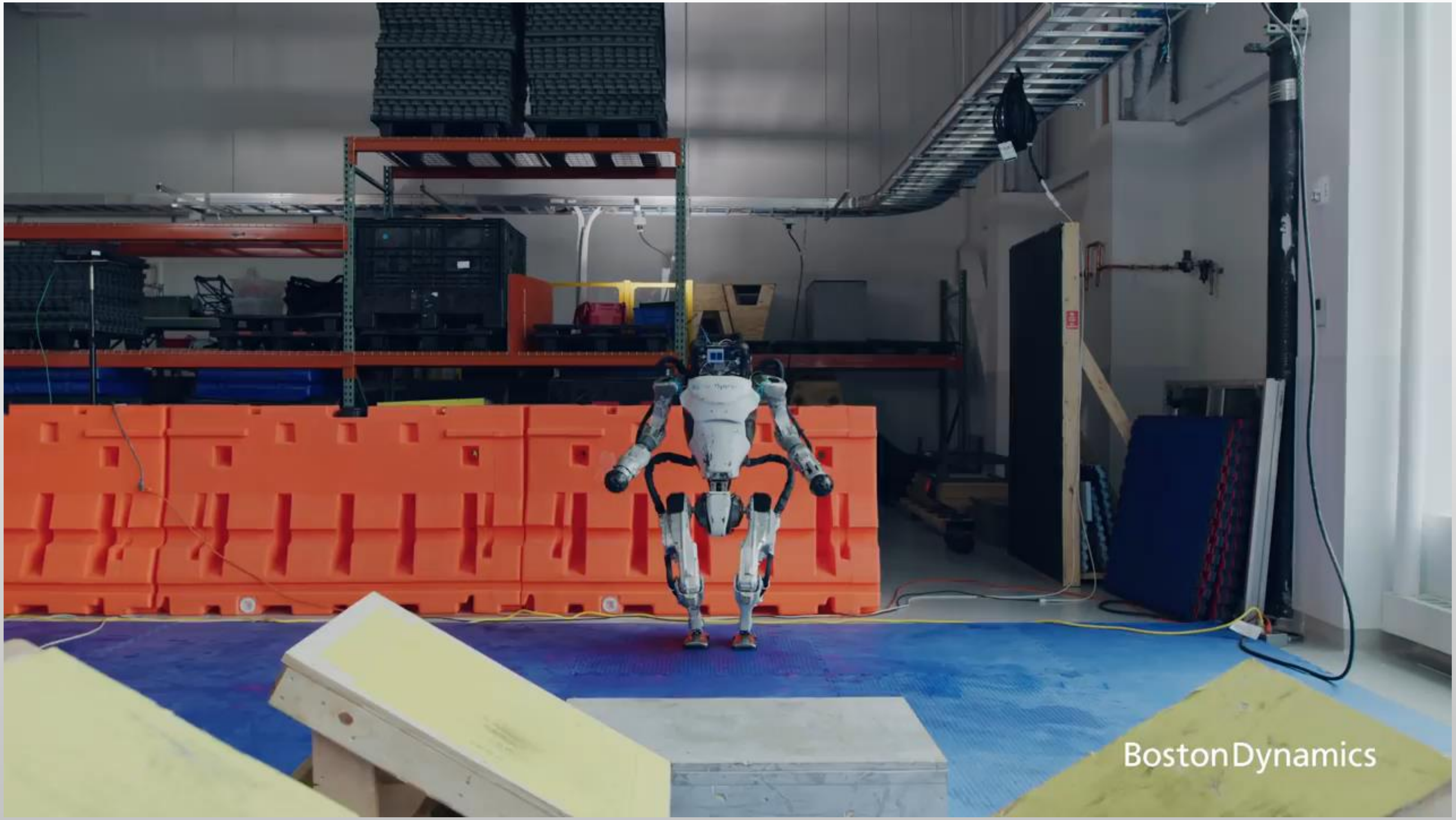
Andrew Albright, Joshua Vaughan

Dept. of Mechanical Engineering

Univ. of Louisiana at Lafayette

andrew.albright1@louisiana.edu, joshua.vaughan@louisiana.edu

Pros of Legged Robots



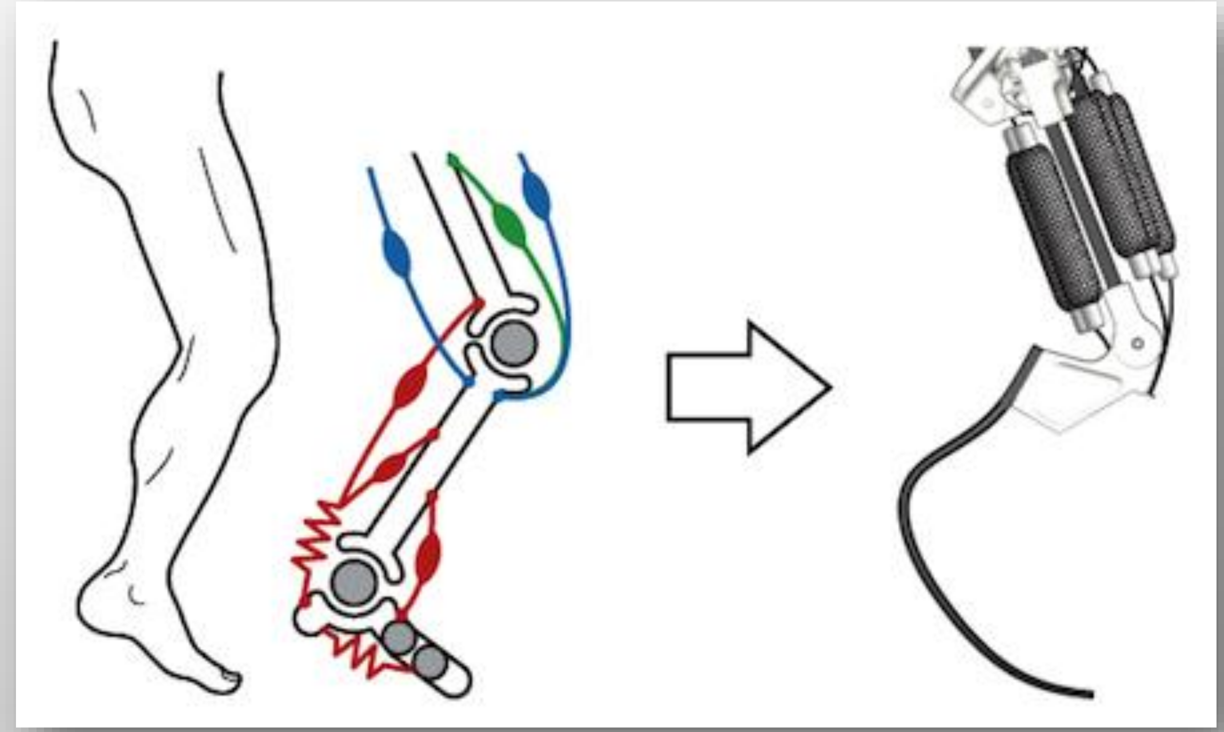
https://www.bostondynamics.com/atlas

Cons of Legged Robots

Watch the Atlas Robot take a spill running parkour #shorts - YouTube

Legged & Flexible Robots

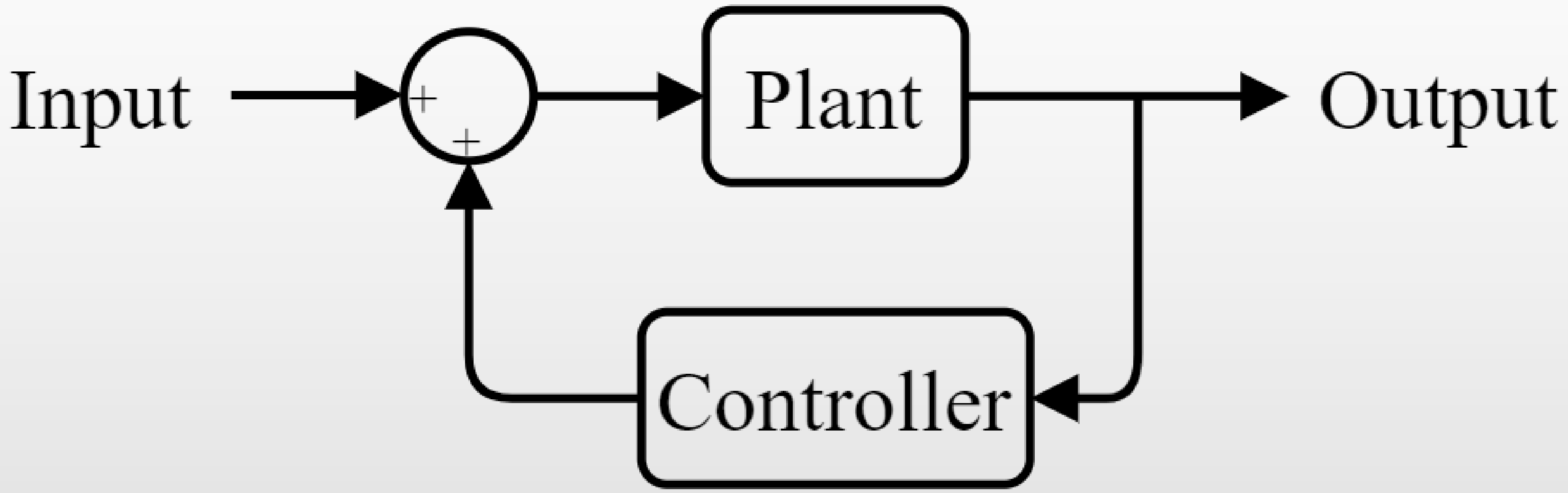
- Pros
  - Increases in efficiency
  - Better stability through adaptability
  - Higher performance
    - Running speed
    - Jumping height



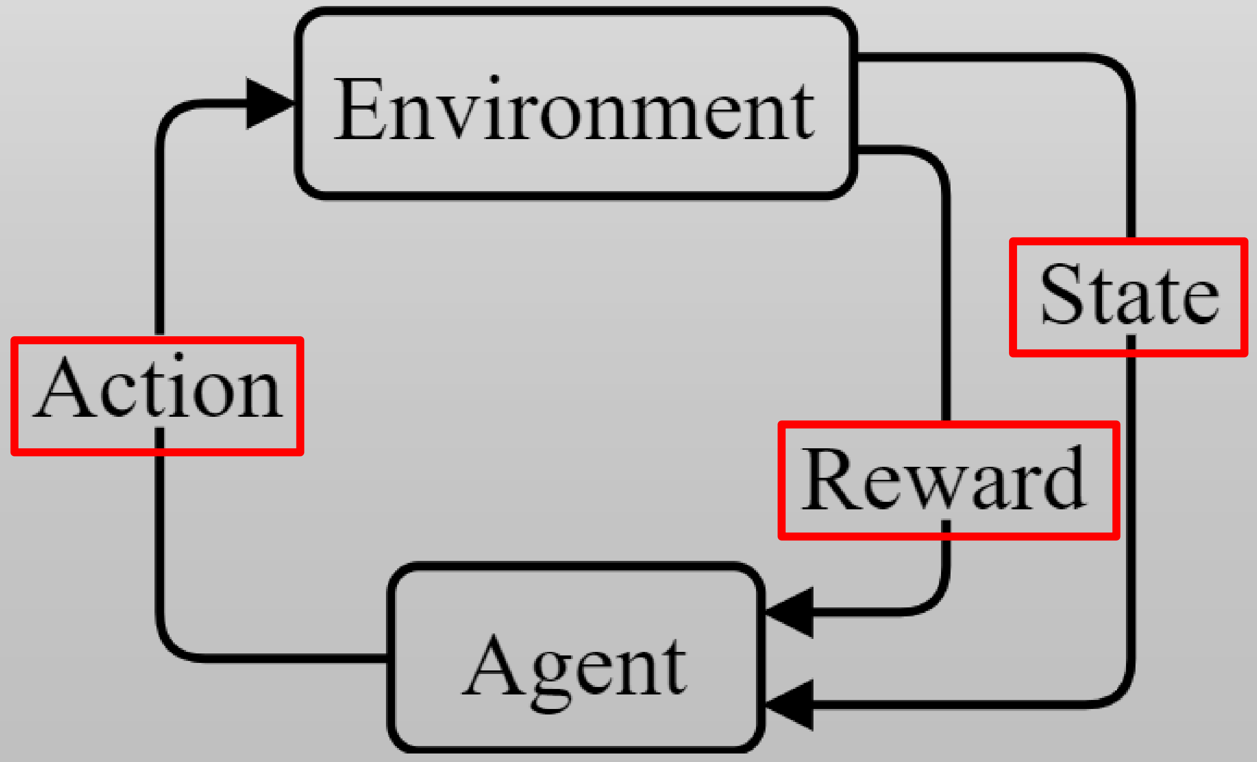
- Cons
  - Difficult to define models for due to non-linearities
  - Challenging to create controllers for

Control Methods

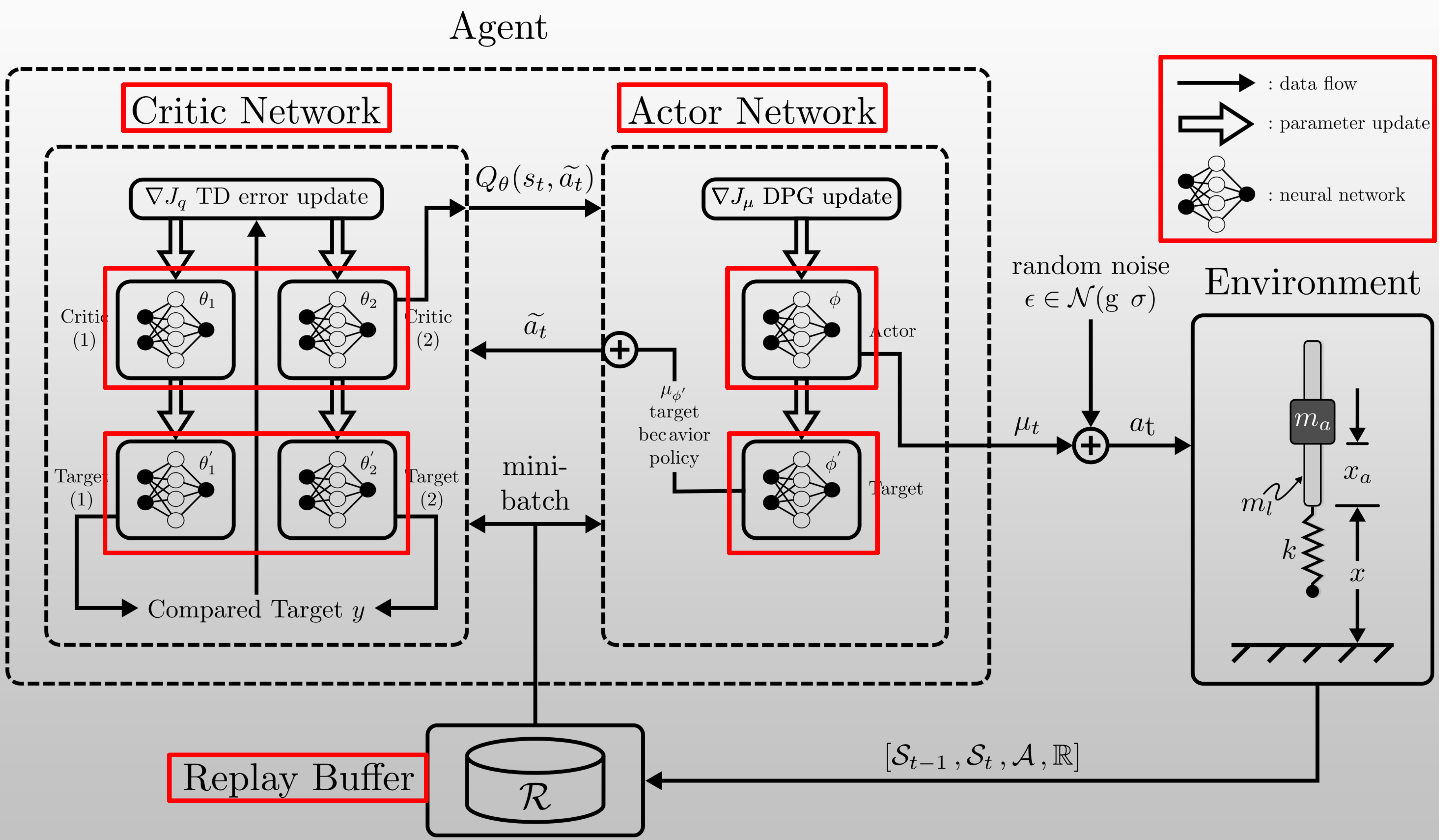
- Traditional Controls Method



- Reinforcement Learning Method

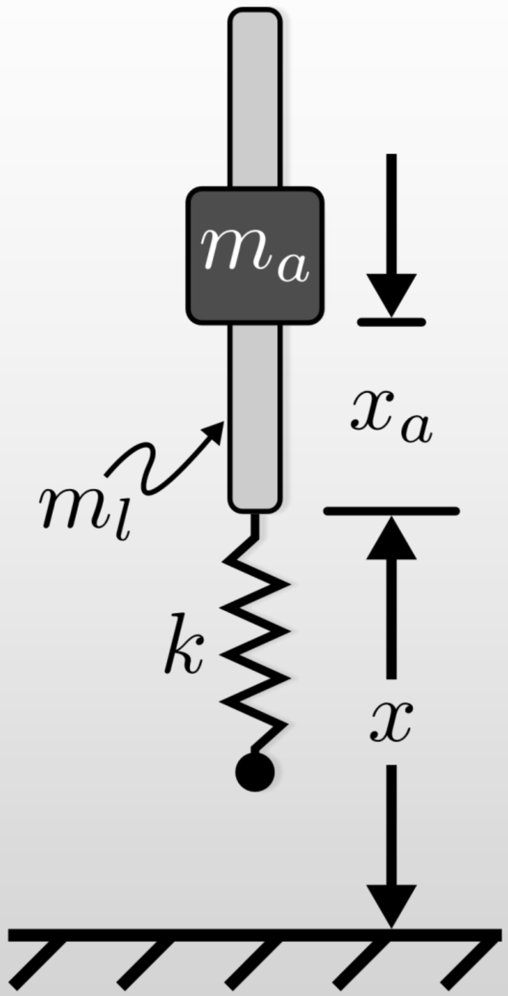


Algorithm Used – TD3



Structure of TD3 (Twin Delayed Deep Deterministic Policy Gradient) with... | Download Scientific Diagram (researchgate.net)

Flexible Jumping System

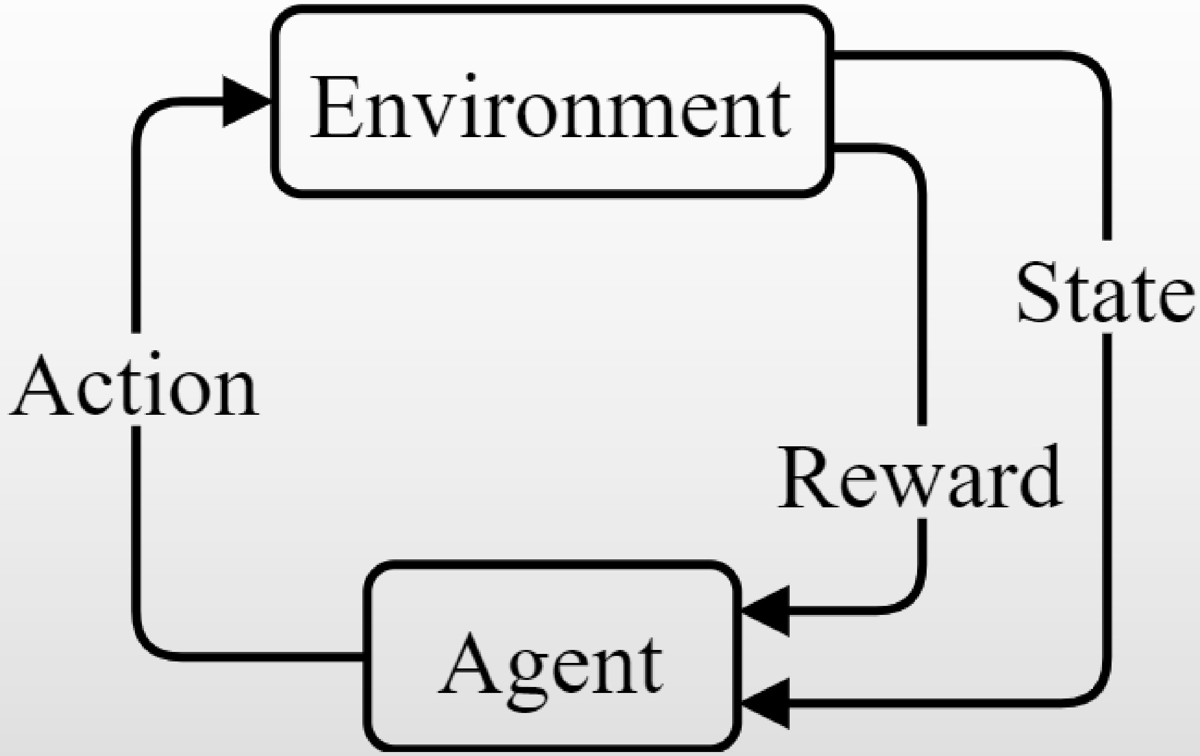


Model Parameter	Value
Mass of Leg, m_l	0.175 kg
Mass of Actuator, m_a	1.003 kg
Natural Frequency, ω_n	11.13 Hz
Spring Constant, k	200000 N/m
Actuator Stroke, $(x_a)_{\max}$	25 mm
Actuator Velocity, $(\dot{x}_a)_{\max}$	2.0 m/s
Actuator Acceleration, $(\ddot{x}_a)_{\max}$	10 m/s

$$\ddot{x} = \alpha\left(\frac{k}{m_t}x^3 + \frac{c}{m_t}\dot{x}\right) - \frac{m_a}{m_t}\ddot{x}_a - g$$

$$\alpha = \begin{cases} -1, & x \leq 0 \\ 0, & \text{otherwise} \end{cases}$$

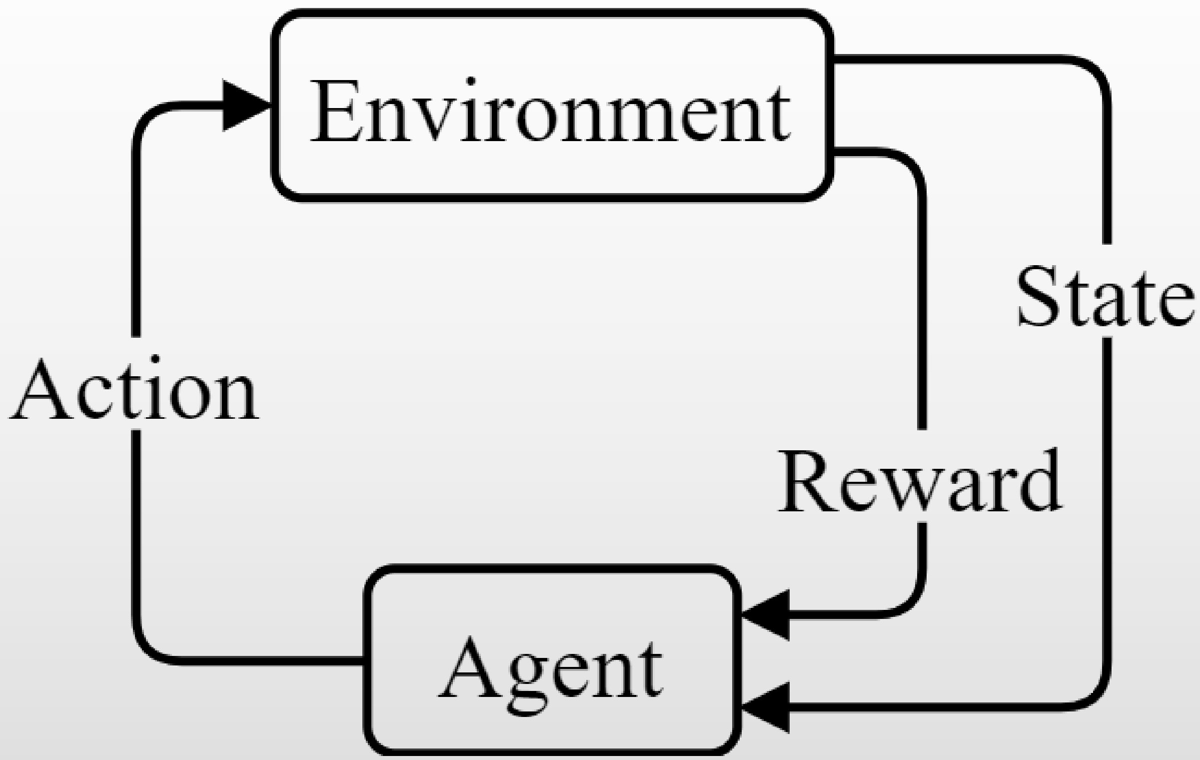
Environment Spaces



$$\mathcal{S} = [x_{a_t}, \dot{x}_{a_t}, x_t, \dot{x}_t]$$

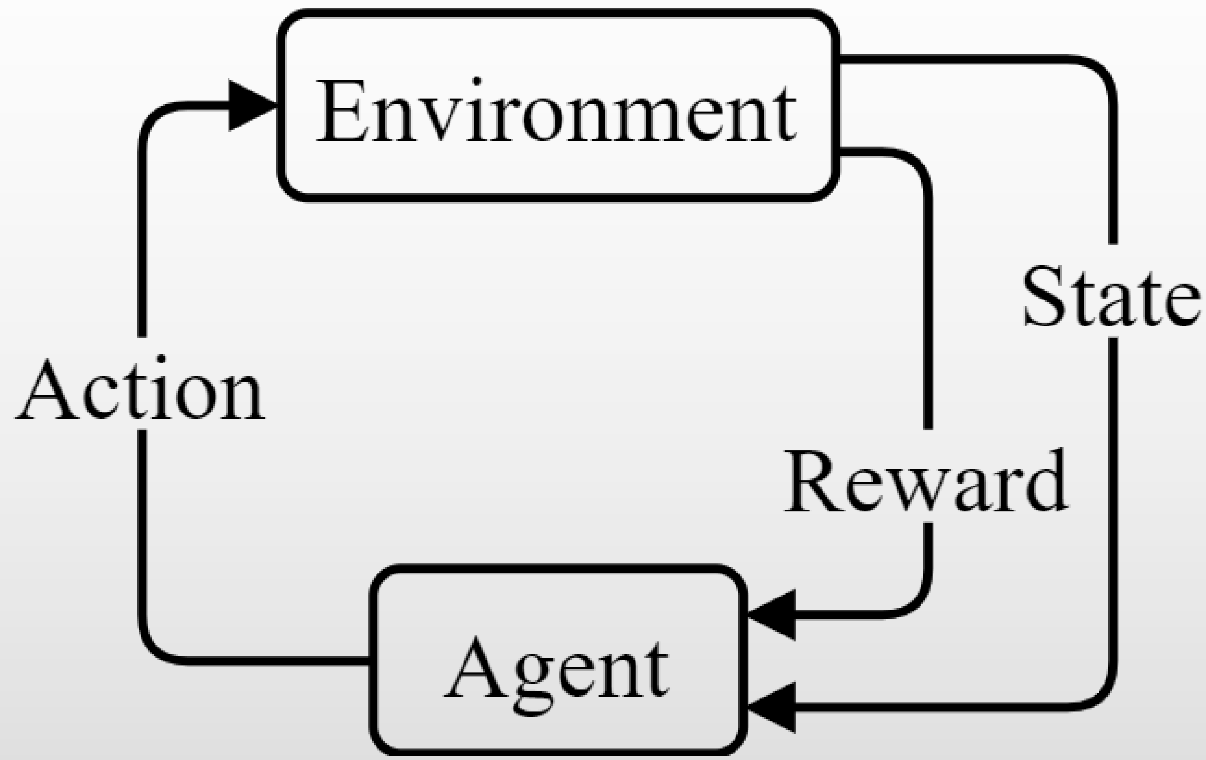
$$\mathcal{A} = [\ddot{x}_{a_t}]$$

Reward Jumping High



$$r = \frac{x_t - x_{min}}{x_{max} - x_{min}}$$

Reward Jumping Efficiently



$$r = \frac{e_t - e_{min}}{e_{max} - e_{min}}, where$$

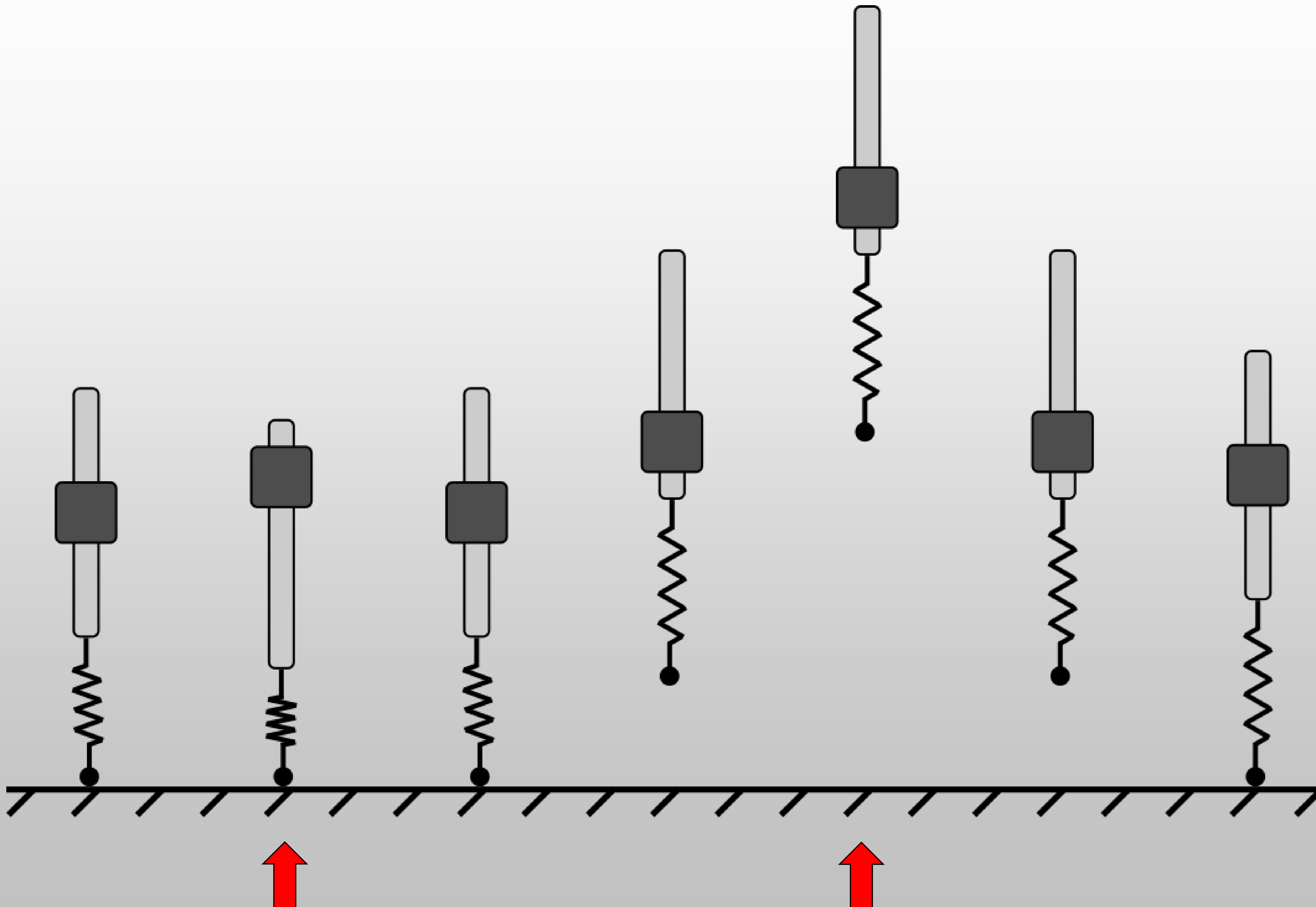
$$e_t = \frac{x_t}{\sum_{t=0}^{t} p_t}$$

$$p_t = m_a \ddot{x}_{a_t} \dot{x}_{a_t}$$

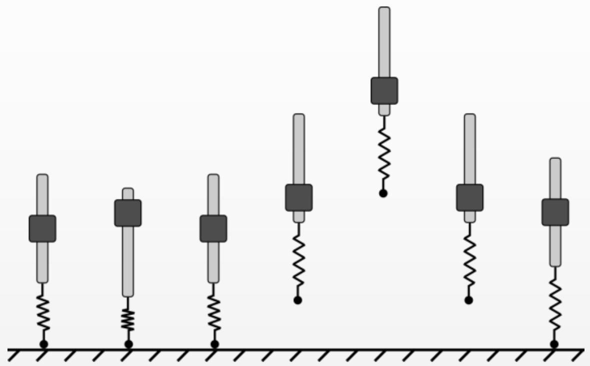
Training Schedule

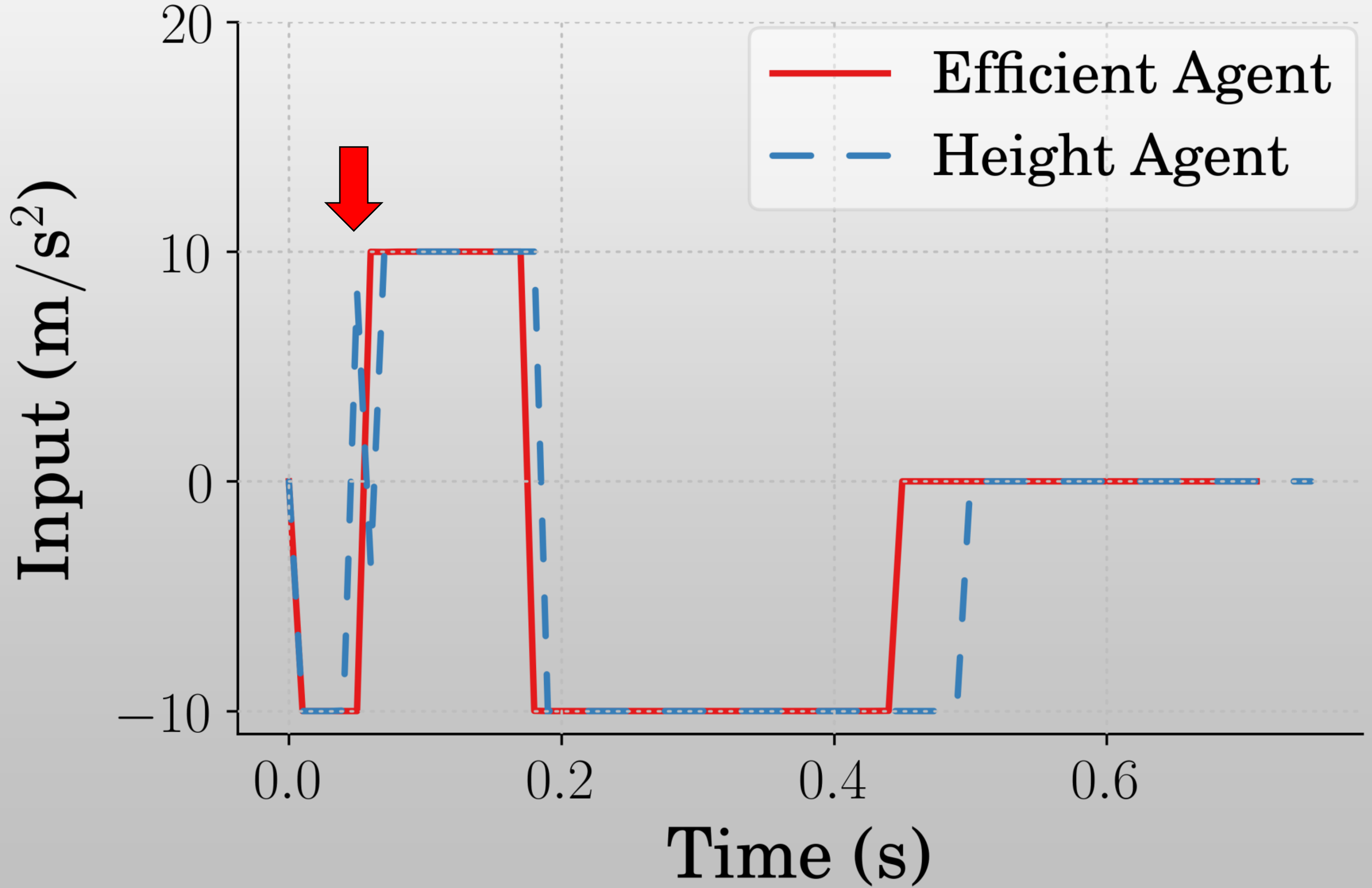
Task	Jump Type	# Agents	Network Seeds
Jump Hight	Stutter Jump	10	9, 16, 104, 107, 250, 676, 868, 878, 918, 947
	One Jump	10	
Conserve Power	Stutter Jump	10	
	One Jump	10	

Single Jump

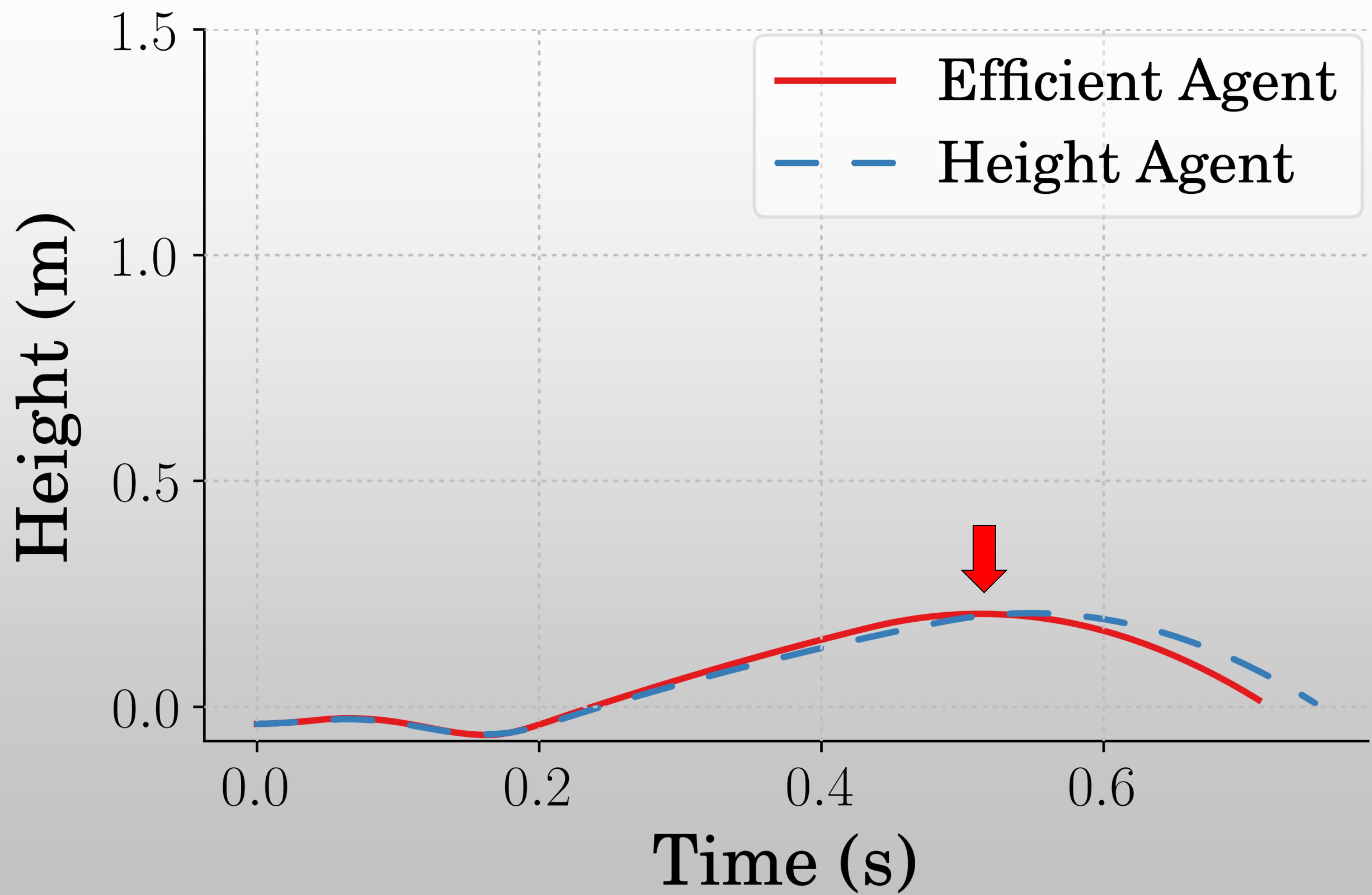


Single Jump Input

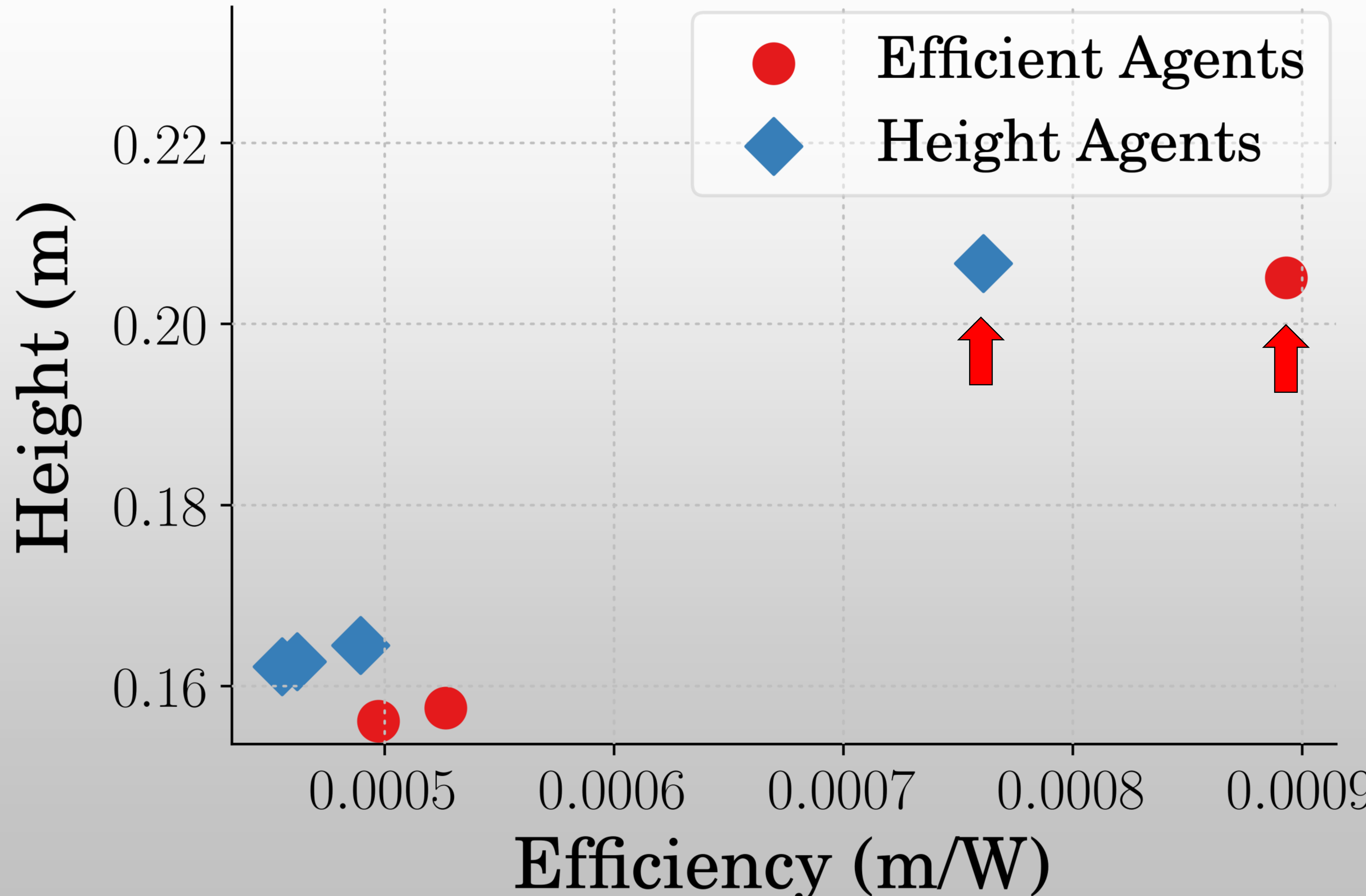




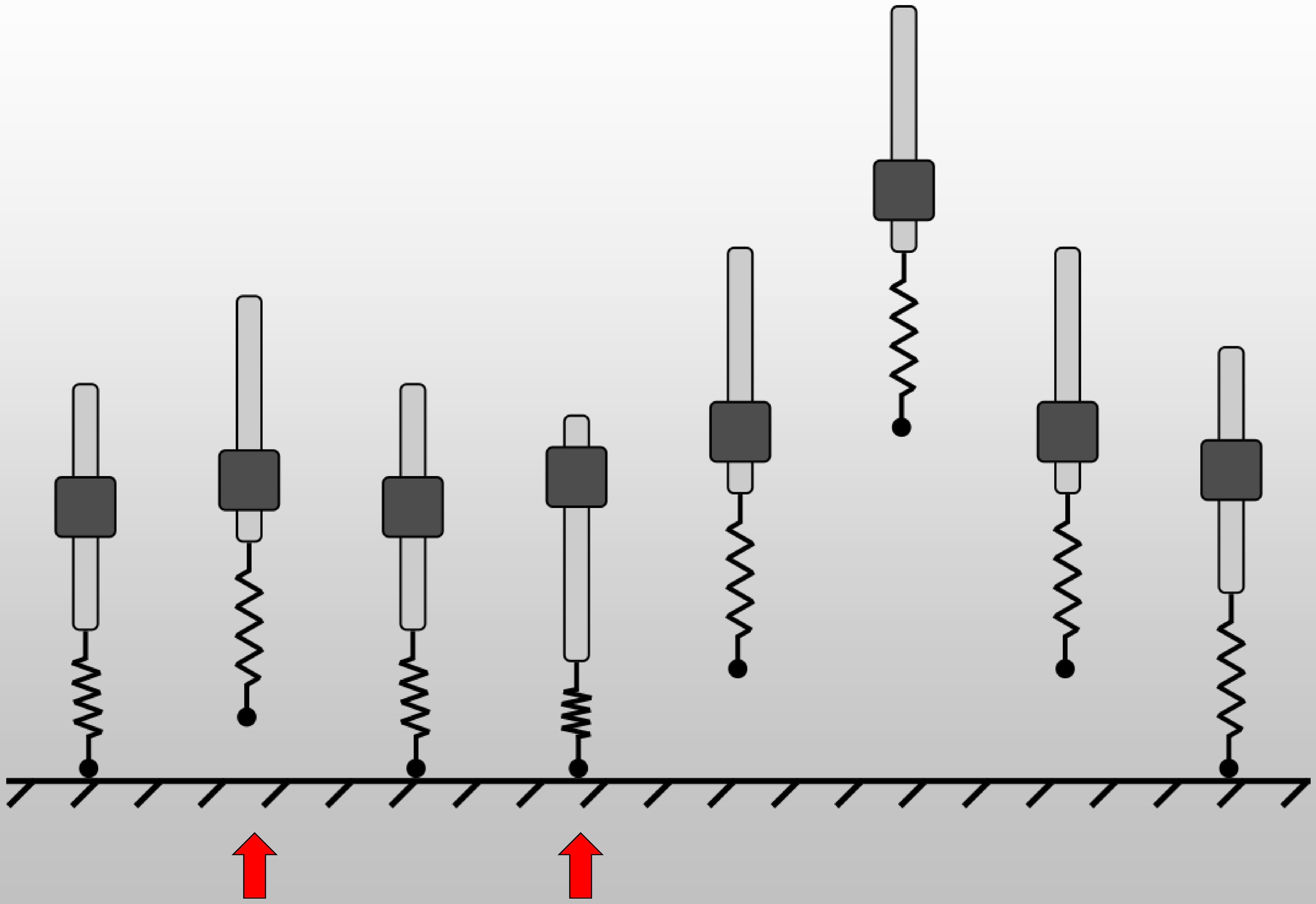
Single Jump Height Reached



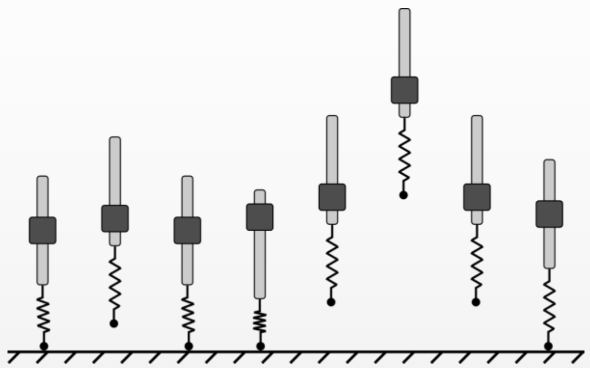
Single Jump Performance

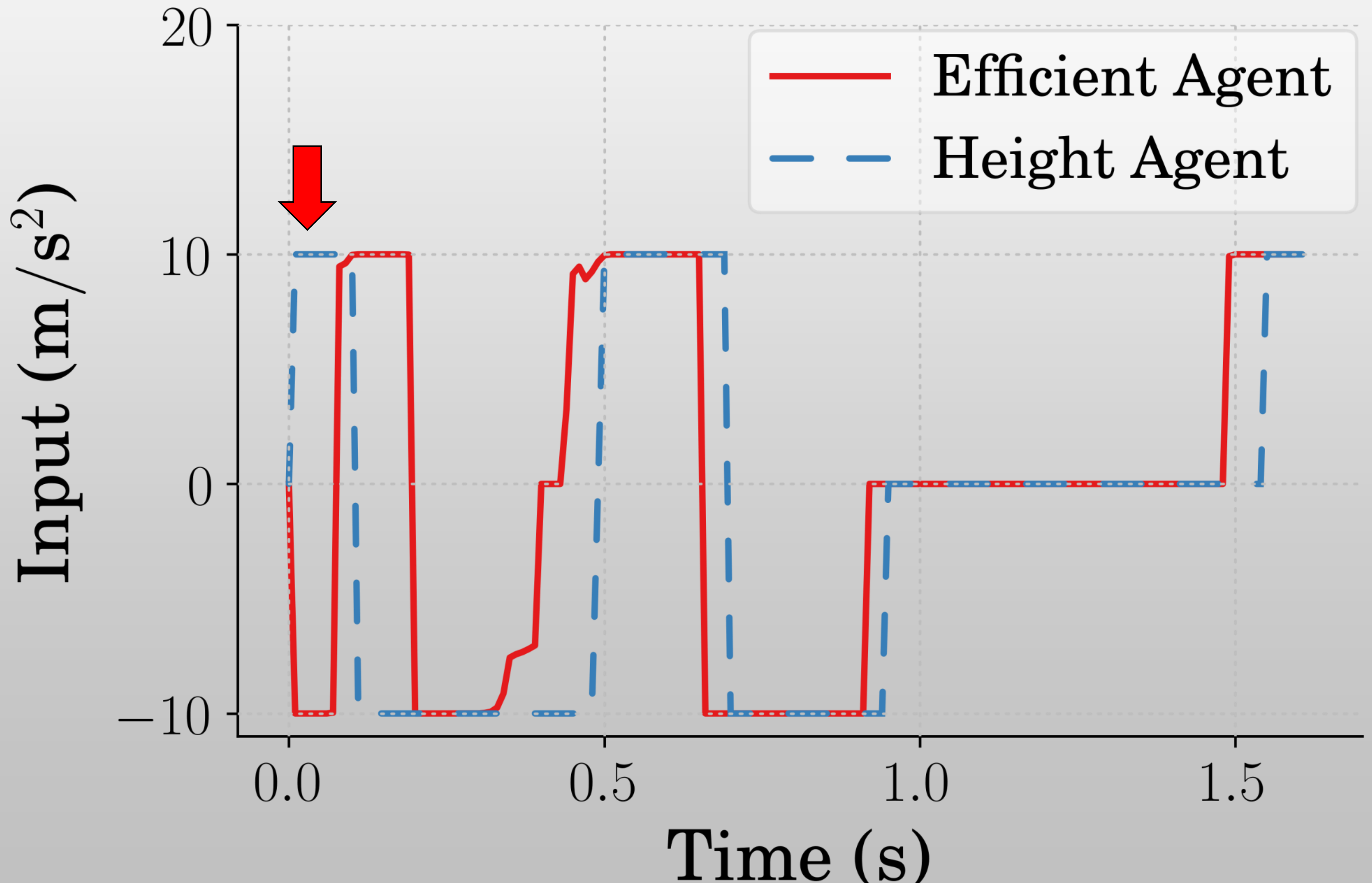


Stutter Jump

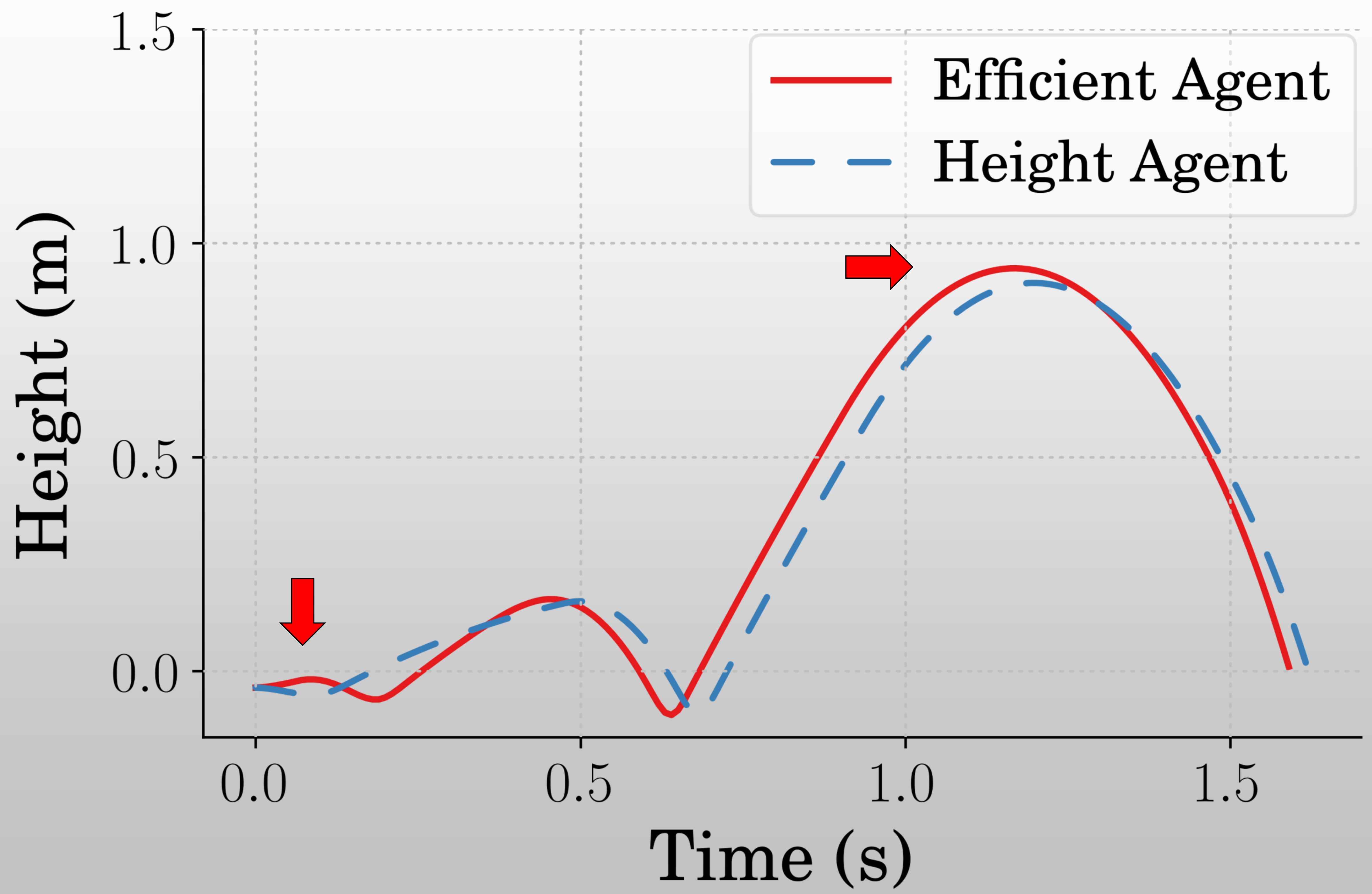


Stutter Jump Input

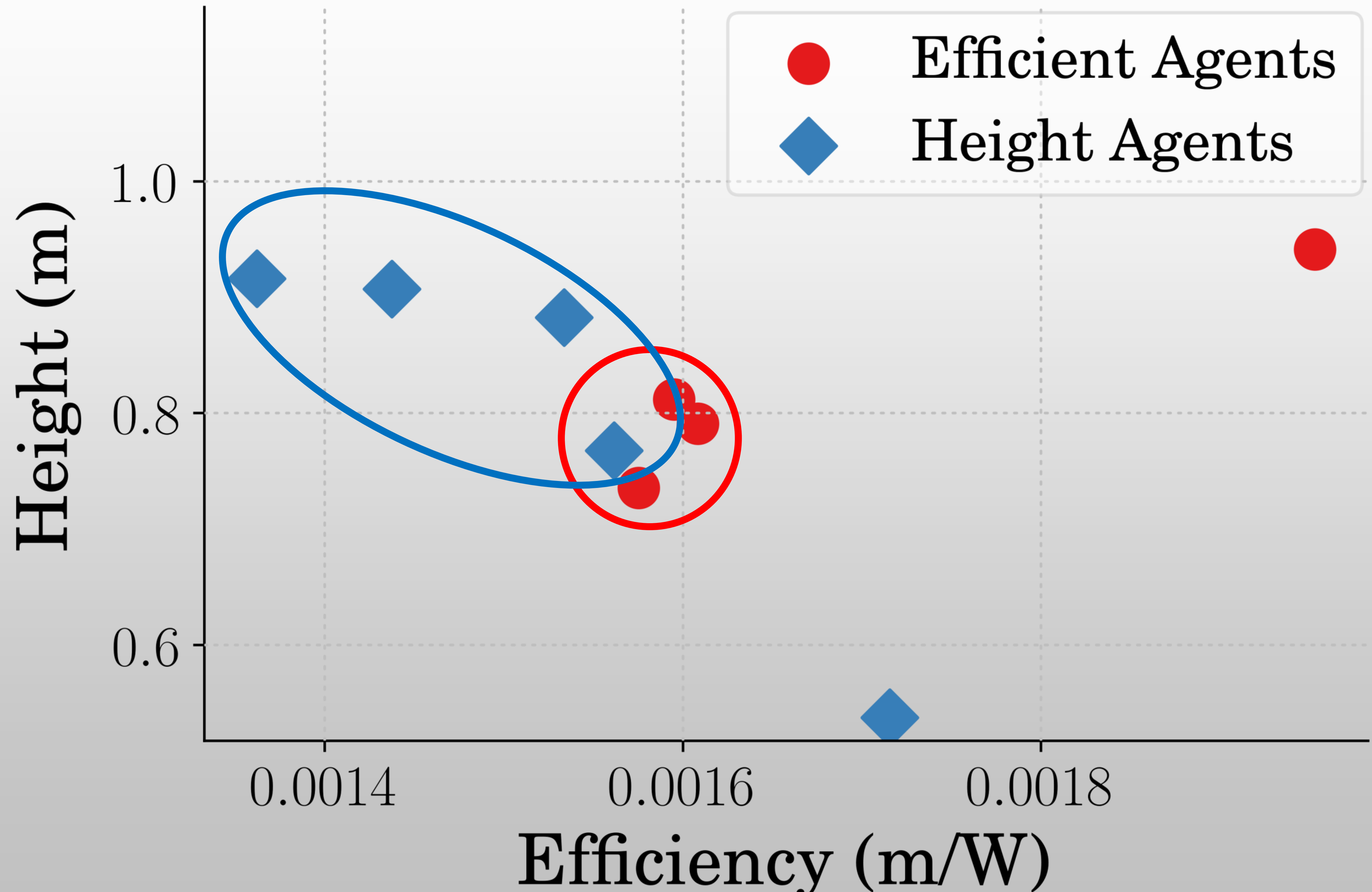




Stutter Jump Height



Stutter Jump Performance



Conclusion

- Reinforcement learning can be used to find control policies that are more efficient
- Agents can learn efficient control strategies where more complex control inputs are required
- Controllers can be learned that are both higher in performance and efficiency metrics

Thank you.