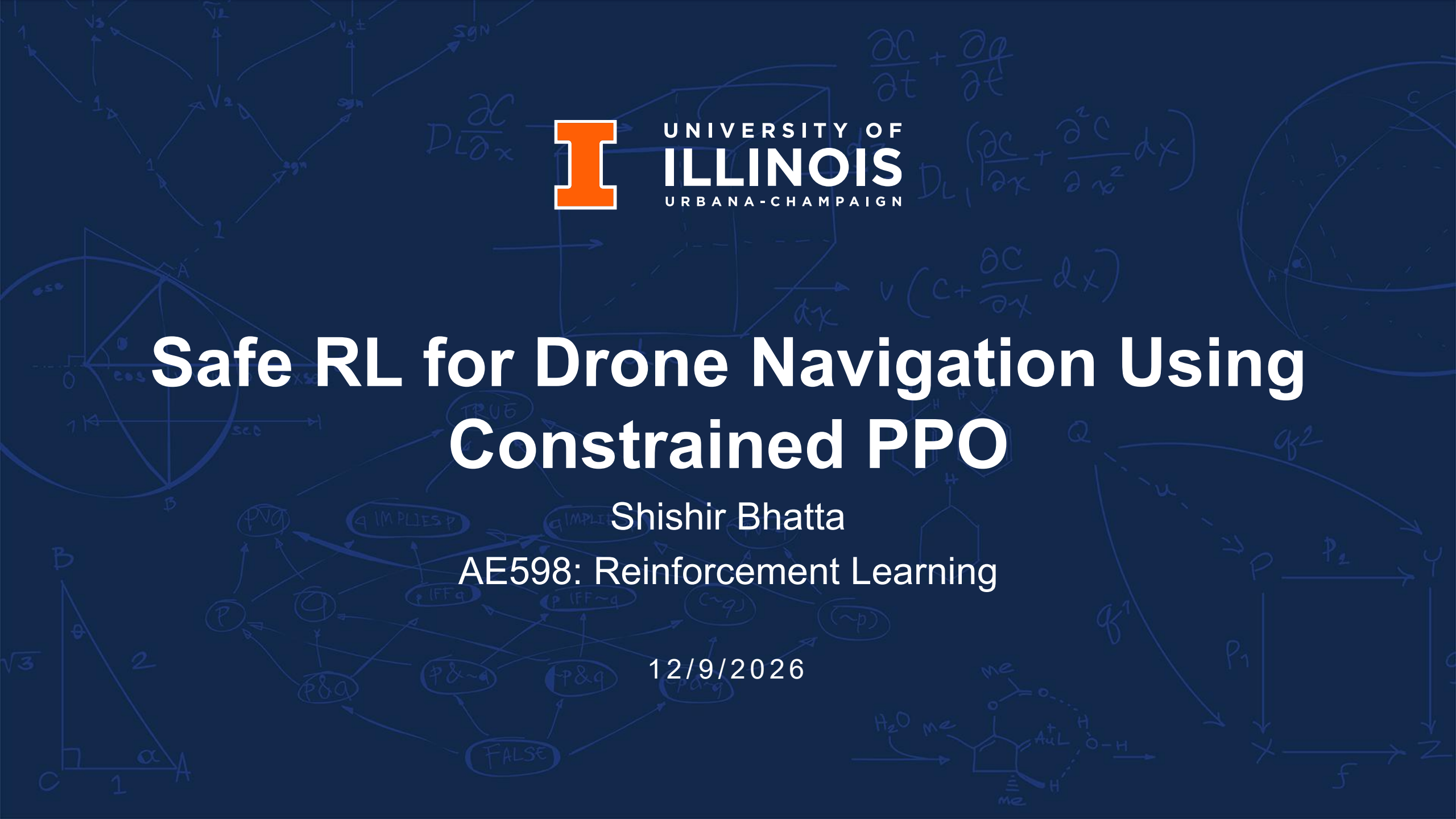


UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Safe RL for Drone Navigation Using Constrained PPO

Shishir Bhatta

AE598: Reinforcement Learning

12/9/2026

Drones require safe navigation

- Often operating in cluttered environments with obstacles
- RL can be unsafe during training
- Controllers must maximize performance **without violating safety limits**

Constrained PPO (CPPO)

- PPO maximizes expected return but has no safety guarantees
- Maximizes reward while subject to a constraint

$$\max_{\pi} \mathbb{E}[R] \text{ s.t. } \mathbb{E}[C] \leq d$$

where $C(s, a)$ is the cost of violating the safety constraints.

Lagrangian Formulation:

$$\mathcal{L}(\theta, \lambda) = \mathbb{E}\left[r_t(\theta)\widehat{A_t} - \lambda r_t(\theta)\widehat{A_t}\right]$$

- λ increases if the agent violates the safety constraints and vice versa

Environment

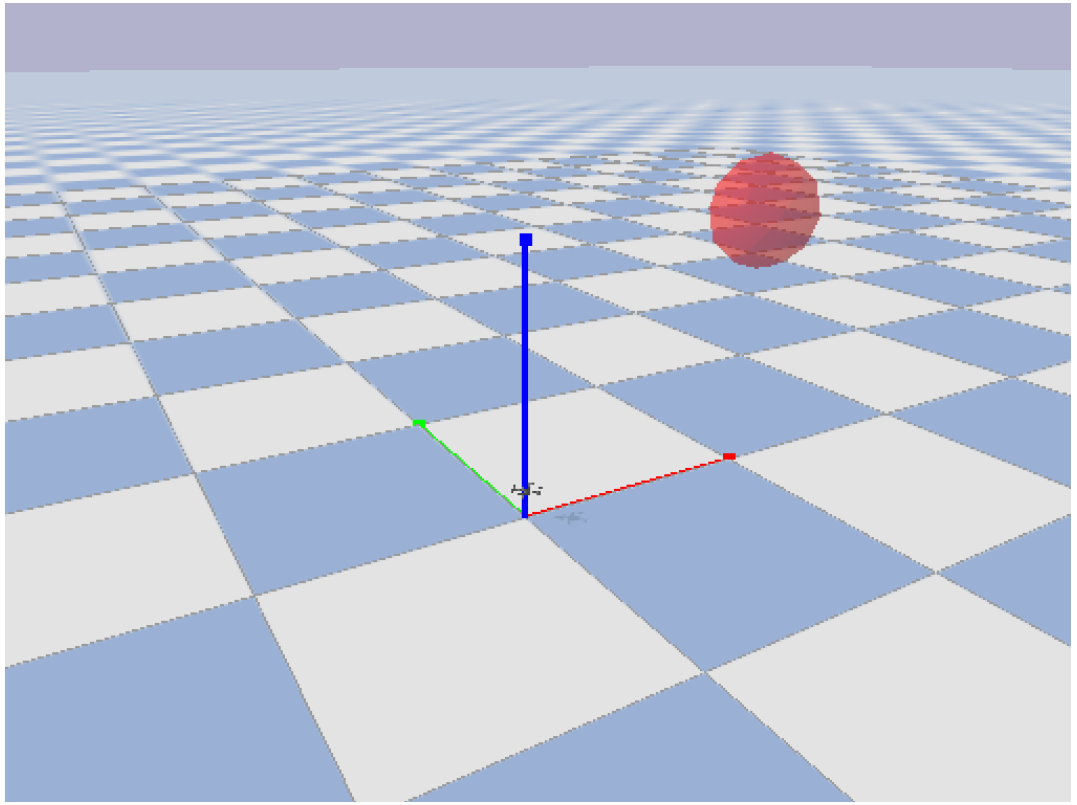
Simulator: gym-pybullet-drones

- 1 drone
- Static obstacles around the environment
- Observations: drone pos/vel + obstacle relative position
- Actions: 4 motor RPM commands
- **Constraints:**

If distance < 0.5 m -> cost = 1
Average cost per episode must be < threshold

Reward:

Distance reward: e^{-2d_t}
Distance penalty: $-0.1\, d_t$
Crash penalty: $-5 \cdot \mathbf{1}(p_{t,z} < 0.1)$
Out-of-range penalty: $-2 \cdot \mathbf{1}(d_t > 3)$



Training

Objective

PPO Objective: $\max_{\theta} \mathbb{E}\left[r_t(\theta)\widehat{A_t}\right]$

Constraint: $\mathbb{E}[\text{cost}] \leq C_{max}$

Architecture

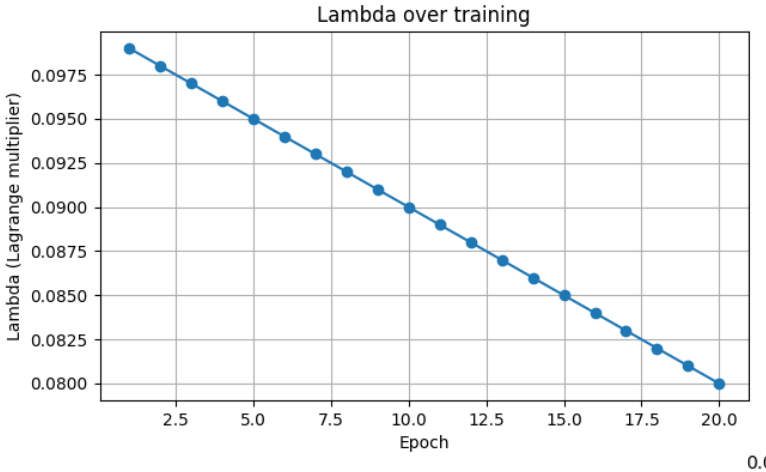
- 2 critic networks for reward and cost
- Gaussian Policy over RPM actions
- Safety cost: cost = 1 if drone is near the obstacle

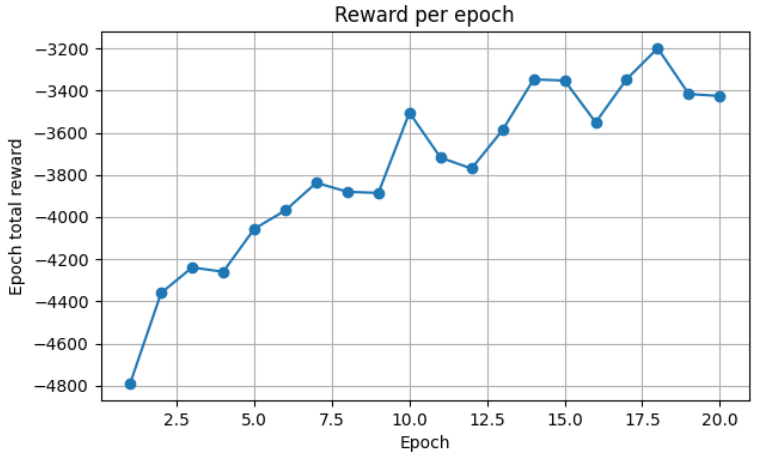
Loop

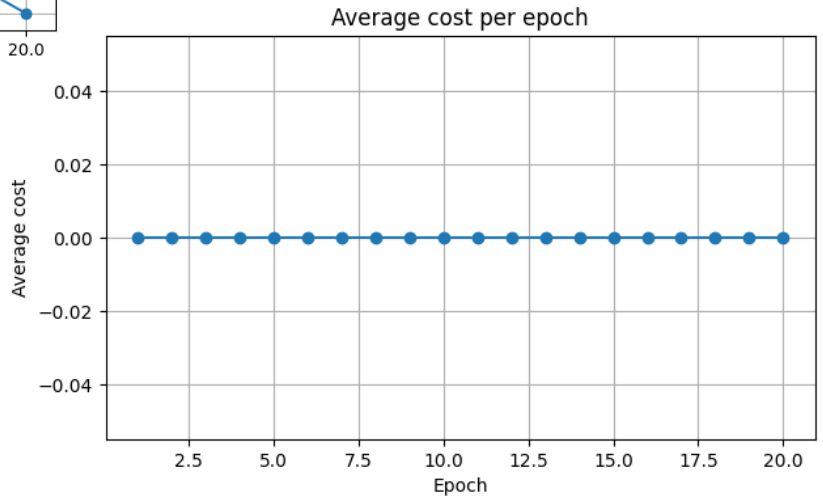
- Roll out trajectories -> observation, action, reward, cost, logprob
- Computer reward advantage and cost advantage
- Policy update using CPPO Lagrangian
- Update $\lambda \leftarrow \max(0, \lambda + \eta(\text{average cost} - C_{max})$

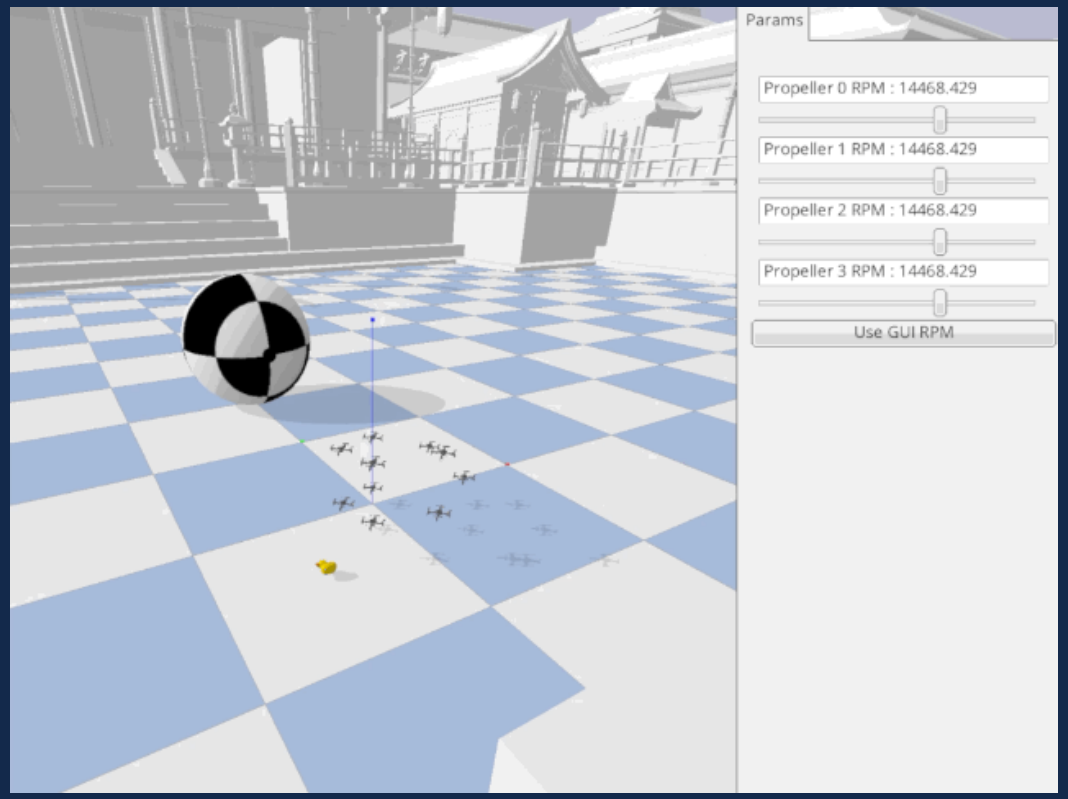
Results

- Average cost per epoch remains at 0, which means safety is never violated
- Average reward per time step was ~-1.75
- Generally making progress at each epoch
- Large dip after 20 epochs, model fails to recover
- Highly complex environment that needs more epochs to train on
- Parameters need tuning to get best result









CPPO successfully enforces safety by avoiding the obstacle.

Tuning parameters may make it perform better and reach the goal

Muti-agent systems can use this to safely naviaget their environments without crashing.

Further work would be comparing this against PPO and other algorithms to see how it fares.

Questions?