



UNIVERSITY OF
ILLINOIS
URBANA-CHAMPAIGN

Safe RL for Drone Racing: Exploring Time-Optimal Motion Planning with CBFs

AE 598 RL Final Project

Abhishek Pai

Motivation

- Autonomous drone racing requires pushing quadrotors to their physical limits (high speeds, agility)
- Traditional autonomy pipelines are often fragile for such environments
- End-to-end learning-based controllers are able to handle fast dynamics and unmodeled effects, but also show lack of safety guarantees and inconsistent success rates
- Gate traversal requires requires precise orientation, not just position

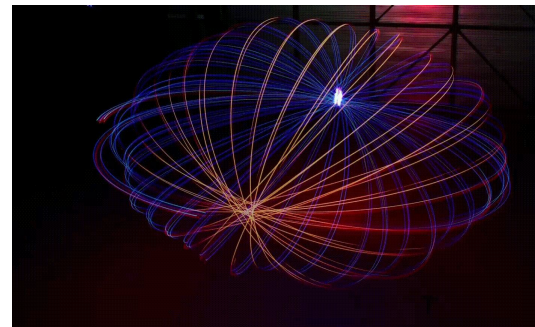


Fig 1: Agilicious by RPG labs, UZH

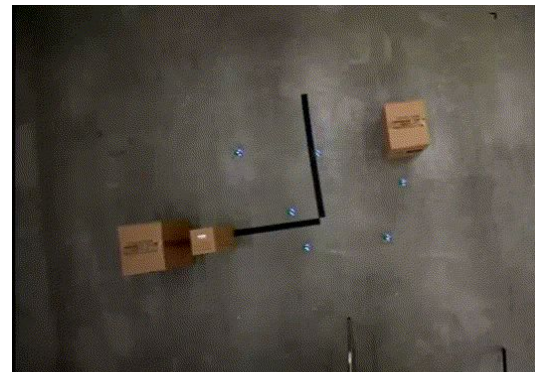


Fig 2: GCBF+, Chuchu Fan et al.

Contribution

- Reproduced a single agent version of Wang et al. “Dashing for the Golden Snitch: Multi-Drone Time-Optimal Motion Planning with Multi-Agent Reinforcement Learning” from ICRA 2025.
- Integrated a CBF that acts as a velocity-dependent virtual funnel for safe gate entry
- Modified existing reward functions to encourage smooth racing lines rather than greedy point-to-point flight

Problem formulation

- **MDP formulation:**
 - **Observations:** 2 primary components
$$\mathbf{o}_{gate} = [\mathbf{g}_1 - \mathbf{p}, \dots, \mathbf{g}_N - \mathbf{p}] \in \mathbb{R}^{3N}$$
 - **Target info** - Relative vector to the next N gate centers
 - **Ego state** - Full kinematic state vector $\mathbf{o}_{ego} = [\mathbf{p}, \mathbf{v}, \mathbf{R}(\mathbf{q}), \boldsymbol{\omega}] \in \mathbb{R}^{13}$
 - **Action:** Centralized thrust and body rates (CTBR) $\mathbf{a}_t = [f_{total}, \omega_x, \omega_y, \omega_z] \in \mathbb{R}^4$
- **Dynamics:** 6-DoF Non-linear Quadrotor Model $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$
- **Safety Constraints (CBF):** System must remain in a safe set defined by the barrier func.

Reward Shaping

$$r_t = r_{\text{prog}} + r_{\text{cmd}} + r_{\text{cbf}}$$

- **Target Reward:**

- Incentivizes advancement along the track centerline

$$r_{\text{prog}} = \underbrace{((\mathbf{p}_t - \mathbf{g}_{\text{prev}}) \cdot \mathbf{d}_{\text{seg}})}_{s(\mathbf{p}_t)} - \underbrace{((\mathbf{p}_{t-1} - \mathbf{g}_{\text{prev}}) \cdot \mathbf{d}_{\text{seg}})}_{s(\mathbf{p}_{t-1})}$$

- **Command Reward:**

- Penalize infeasible commands

$$r_{\text{cmd}} = -\lambda_{\text{rate}} \|\boldsymbol{\omega}_t\| - \lambda_{\text{act}} \|\mathbf{a}_t - \mathbf{a}_{t-1}\|^2$$

- **Safety Reward*:**

- "Safe Funnel" constraint that expands as distance from the gate increases.

$$h(\mathbf{x}) = (R_{\text{gate}} + \tan(\theta)d_n) - d_p$$

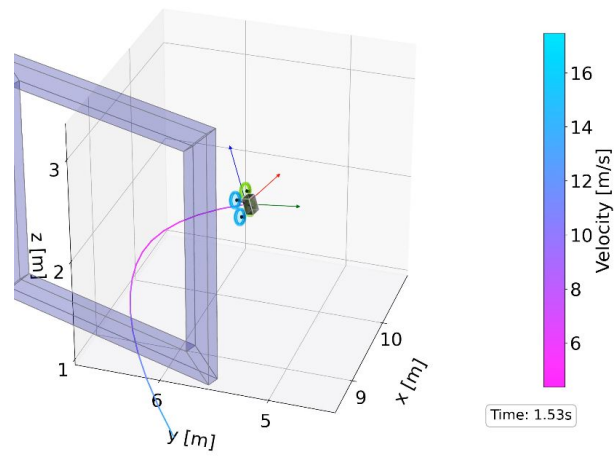
$$r_{\text{cbf}} = \begin{cases} -\beta|\dot{h} + \alpha h| & \text{if } \dot{h} + \alpha h < 0 \\ 0, & \text{otherwise} \end{cases}$$

- where R is radius of inscribed circle, dn and dp are long and lat dist.

*inspired from Song et al. Autonomous Drone Racing with Deep RL (2021)

Simulation Setup

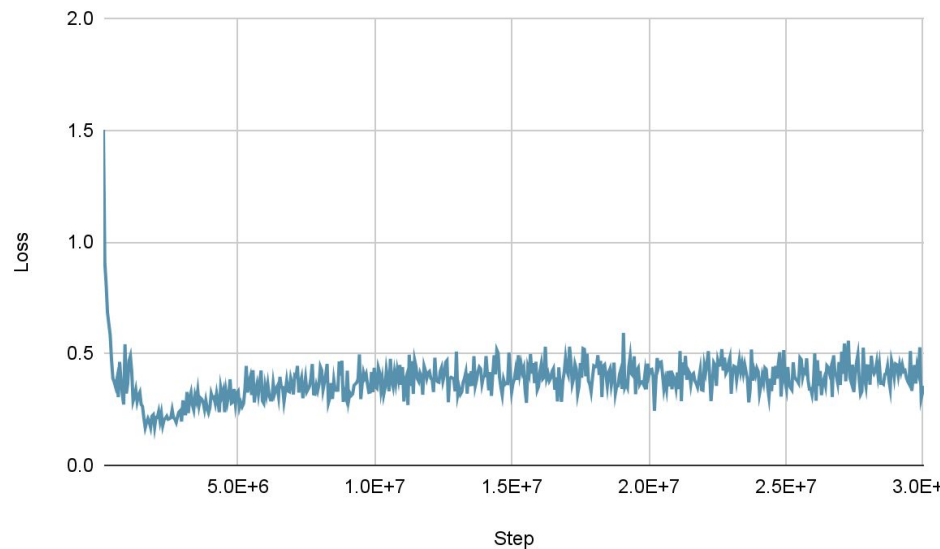
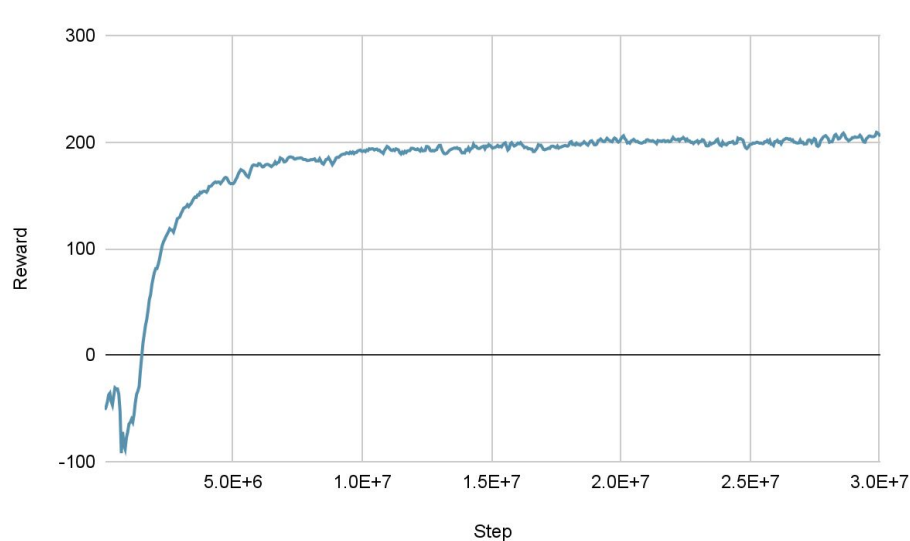
- **Environment:** gym-pybullet-drones + RaceUtils
- **Drone Model:** Standard racing quadrotor dyn parameters from Foehn et al. “Agilicious”, ETH Zurich
- **Algorithm:** SB3 PPO (Proximal Policy Optimization)
- **Track:** Fixed circuit with gates
- **Training hardware:** 14 core i7 + 16GB RAM
- **Training time:** ~5-6 hours



Simulation Setup

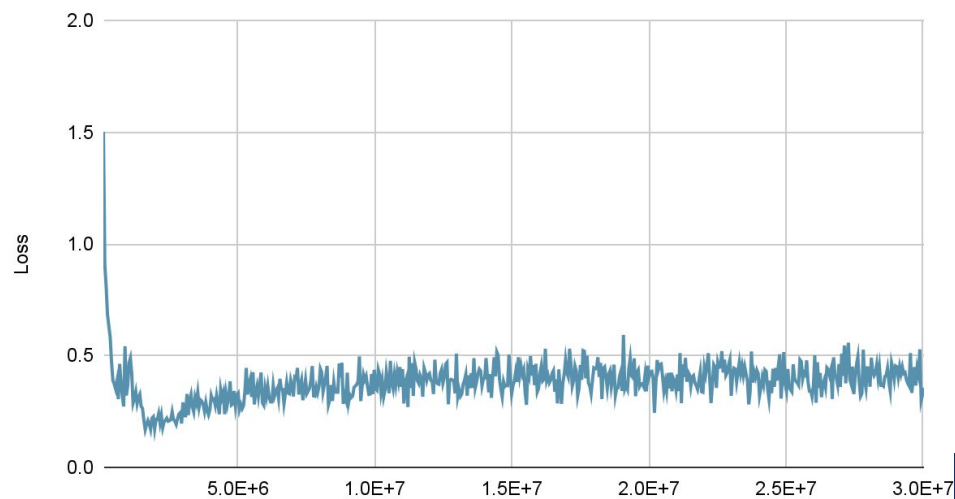
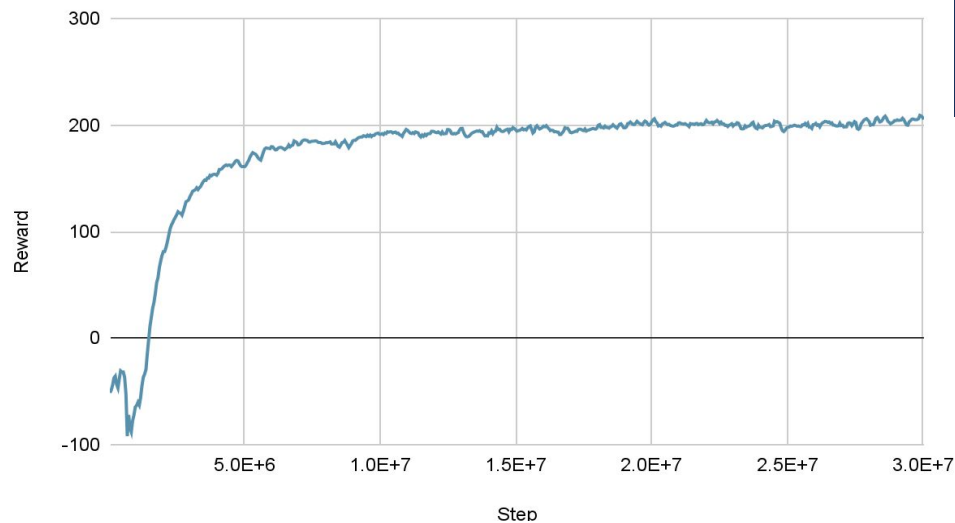
Parameter	Value	Description
Training Steps	30,000,000	Total timesteps across all environments
Parallel Envs	32	Number of vectorized environments
Activation Function	Tanh	Non-linear activation
Learning Rate	3e-4	0.0003
Horizon (n_steps)	2048	Steps per env per update
Batch Size	4096	Samples per minibatch
Number of Epochs	10	Optimization passes per rollout
Discount Factor (gamma)	0.99	Future reward weighting
GAE Lambda	0.95	Advantage estimation smoothing

Training Results

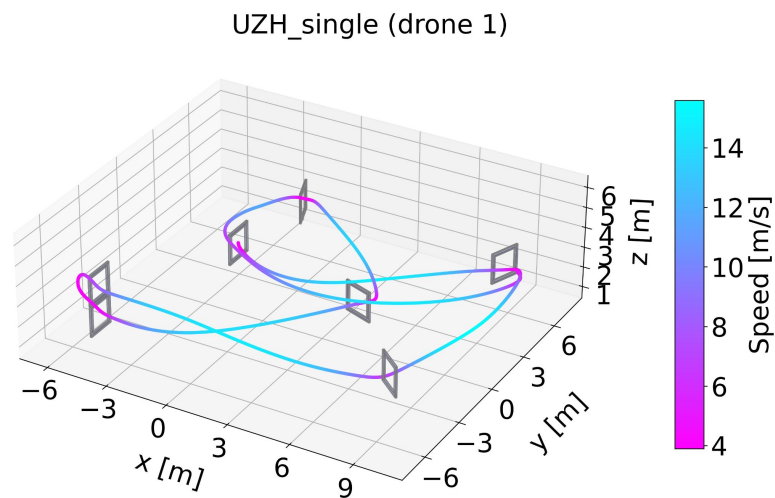
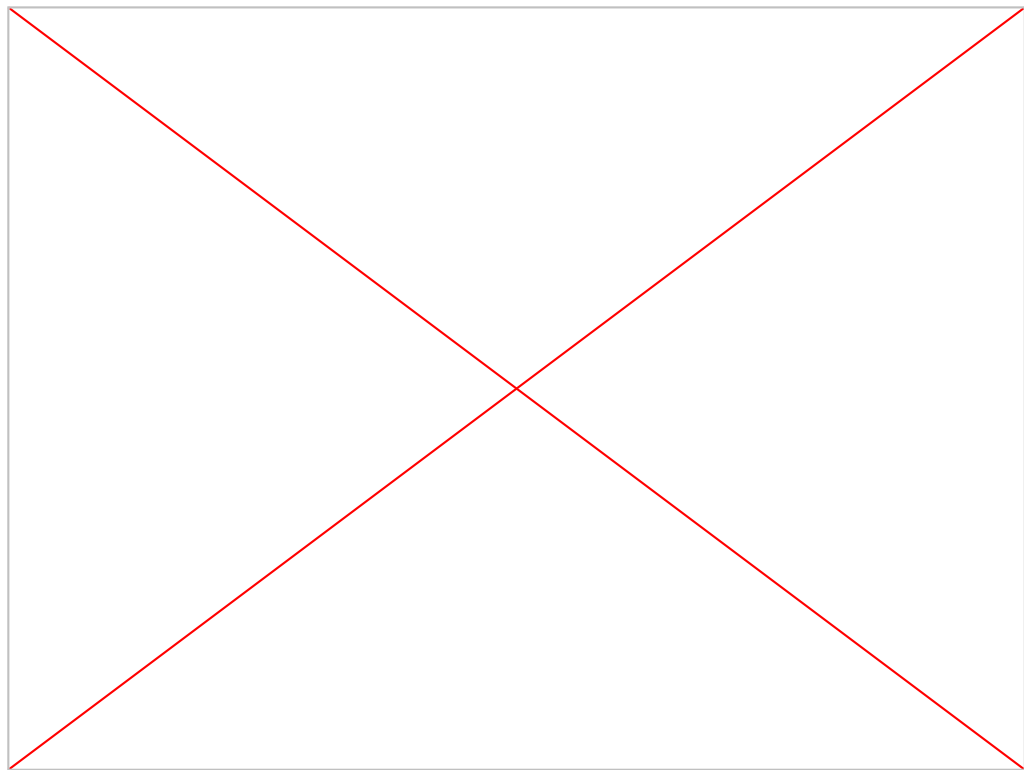


Avg. Reward & Loss v. Time

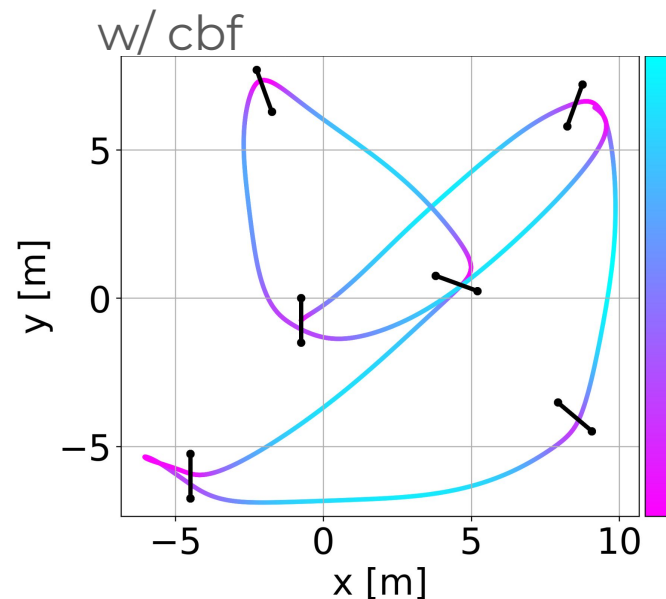
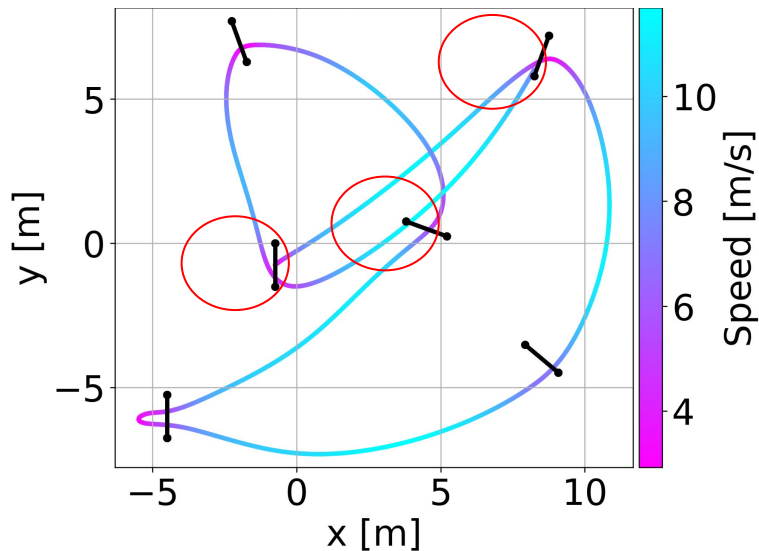
Training Results



Evaluation



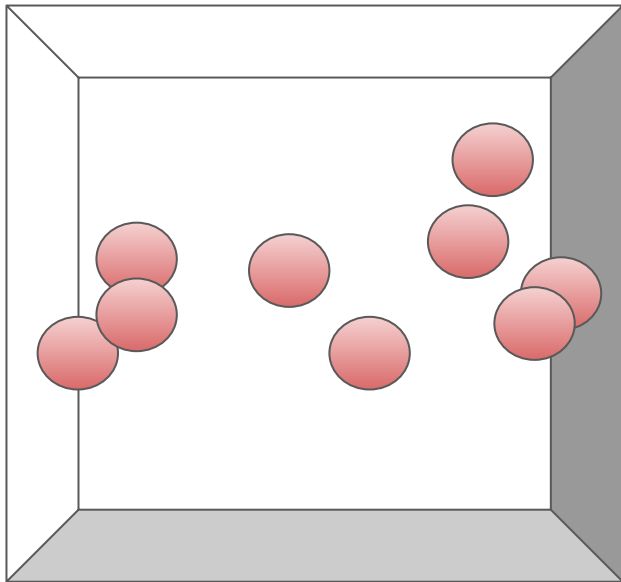
Comparison



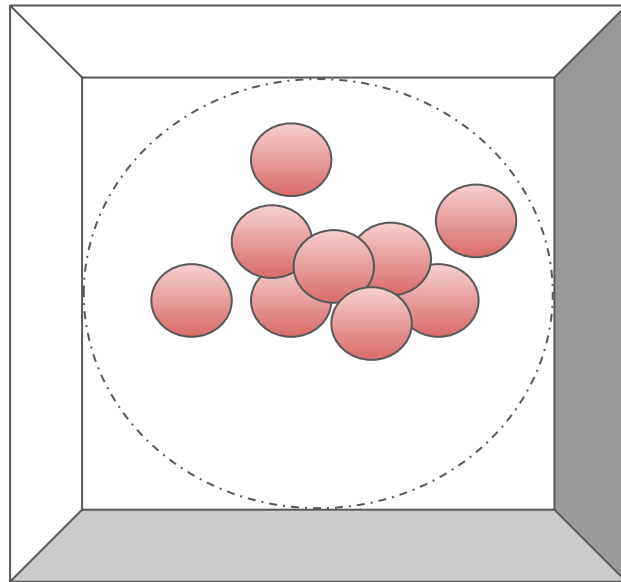
Lap Time (s)	11.56	7.2
Avg. Speed (m/s)	9.76	13.87

Comparison

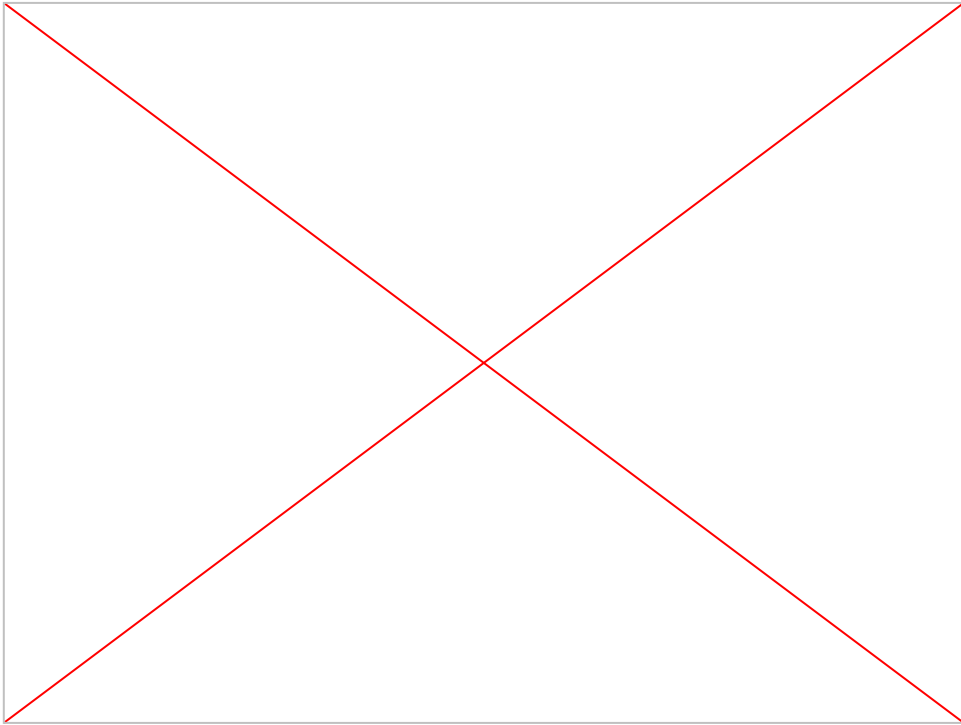
Wang et al.



w/ cbf



- Successfully integrated CBFs into a time-optimal racing RL framework to act as effective guidance funnels for high-speed navigation.
- Agent strategized better and ensure safe gate traversal while improving on speed.
- Optimize for FPV navigation by locking the heading to the direction of motion, preventing the camera from losing sight of future gates



Lap Time (s)	9.82
Avg. Speed (m/s)	14.37

- Successfully integrated CBFs into a time-optimal racing RL framework to act as effective guidance funnels for high-speed navigation.
- Agent strategized better and ensure safe gate traversal while improving on speed.
- Optimize for FPV navigation by locking the heading to the direction of motion, preventing the camera from losing sight of future gates
- Deploy on real hardware

THANK YOU