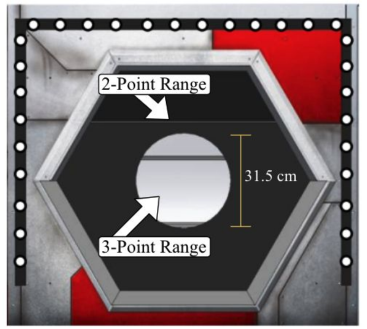
Simulated Data Generation Through Algorithmic Force Coefficient Estimation for AI-Based Robotic Projectile Launch Modeling

Sajiv Shah[1*], Ayaan Haque[1*], Fei Liu[2]

[1]Saratoga High School, [2]University of California, San Diego
[*]Authors contributed equally

Problem

- Objective
  - Determine launcher configurations necessary for scoring object into target
  - Generate predictions for a large range of shooting distances
- Trial and error testing requires lots of time
- Analyzing forces on the ball and simulating trajectories is highly inaccurate
  - There exist unknown dynamics
  - Aerodynamic forces have unknown coefficients
- Data driven approaches are not useful
  - RL is unsafe and requires expensive architecture
  - DL requires a large data pool, which renders the need for prediction obsolete



Key Contributions

Simulator

A trajectory generator that models launches based on multiple input parameters. Developed using known dynamics and kinematics of object flight, and used in conjunction

Neural Network

3-Layer neural network which inputs launcher configurations to make predictions on outcome of launch. Learns underlying patterns and factors that dictate object flight which are unaccounted for.

Experimentation

Implementation of model to a mobile robot equipped with a launcher. Comparison of accuracy in multiple dimensions using neural network and various other methods.

Data

Experimental Dataset

- 100 Configurations and Outcomes
- Collected with robot in field
- 20% for evaluation

Simulated Dataset

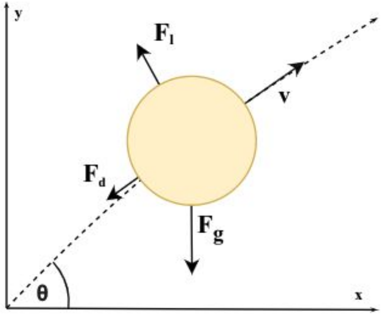
- Varying training labels
  - seg (10, 50, full)
  - class (100, 1000, full)
- Experiments repeated 5 times

Experimental Trials on Simulated Data

- Sampled 50 of Simulated Configurations and received their real world outcomes
- Used to evaluate real field accuracy

Simulator

Dynamics



$$F_g = mg$$

$$F_l = C_l \frac{4}{3}(4\pi^2 r^3 s \rho v)$$

$$F_d = \frac{C_d}{2} A v^2 \rho$$

Kinematics

$$\theta = \int_0^t \arctan(\frac{v_y}{v_x})dt + \theta_0$$

$$v_x = \int_0^t (\frac{dv_x}{dt})dt + v_0\cos(\theta_0)$$

$$v_y = \int_0^t (\frac{dv_y}{dt})dt + v_0\sin(\theta_0)$$

$$x = \int_0^t (v_x)dt$$

$$y = \int_0^t (v_y)dt$$

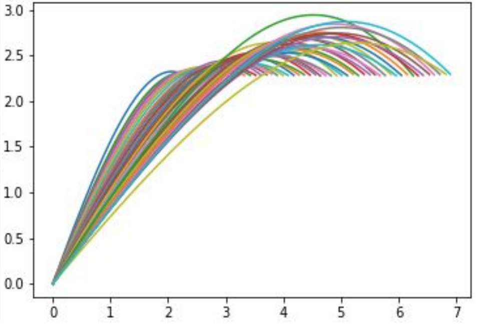
$$\frac{dv_x}{dt} = -(a_l\sin(\theta) + a_d\cos(\theta))$$

$$\frac{dv_y}{dt} = a_l\cos(\theta) - a_d\sin(\theta) - a_g$$

Input

- distance, launch angle, velocity

Output

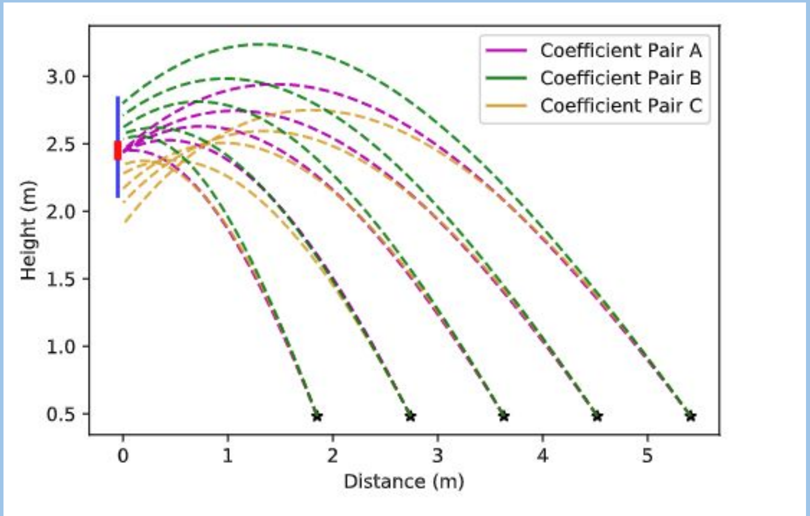


$$(d, h_{target} \pm 0.07) \rightarrow Y_r = (1,1)$$

$$(d, h_{target} \pm 0.35) \rightarrow Y_r = (1,0)$$

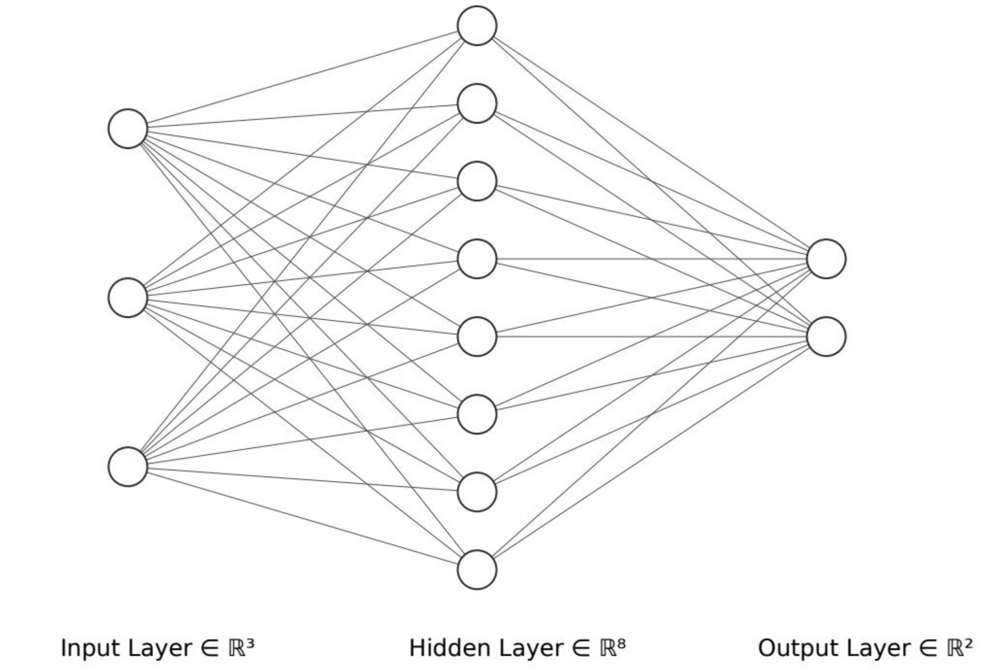
Force Coefficient Estimation

- Lift and Drag Coefficients are unknown
- Using collected data (X_r) we can estimate
- Test 1000^2 combinations
  - (0.00,5.000) with step side 0.005
- Evaluate best pair using multiple metrics
  - 3-point accuracy
  - 2-point accuracy
  - mean deviation



Neural Network

- 3 Input Nodes
  - One node per launcher configuration
- 3 Layers, output layer has 2 nodes
  - One node for the binary outcome of 2 and 3 point values
- Trained on a combined loss of experimental data and simulated data
  - Simulated loss is weighted lower

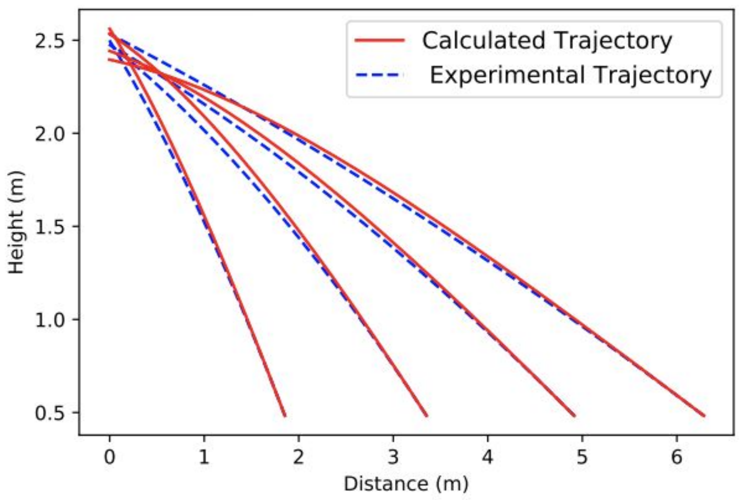


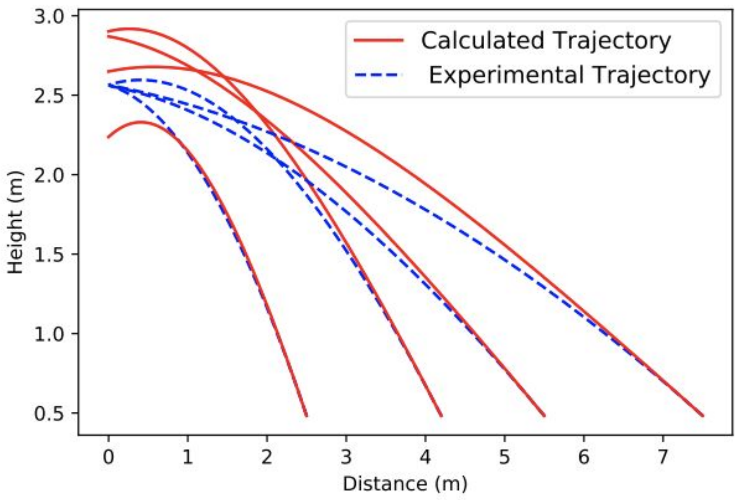
$$L(X_r, Y_r, X_s, Y_s) = L_R(\mathcal{F}_\theta(X_r), Y_r) + \lambda L_S(\mathcal{F}_\theta(X_s), X_s)$$

Implementation

- Ball launcher
  - Double-flywheel mechanism
  - Variable hood-angle using linear actuator (acme screw)
- Launch process
  - Generate predictions using FCE-NN and store data in csv
  - Use vision camera on mobile robot to determine distance to target
  - Set launch angle and motor speed

Results (Force Coefficient Estimation)





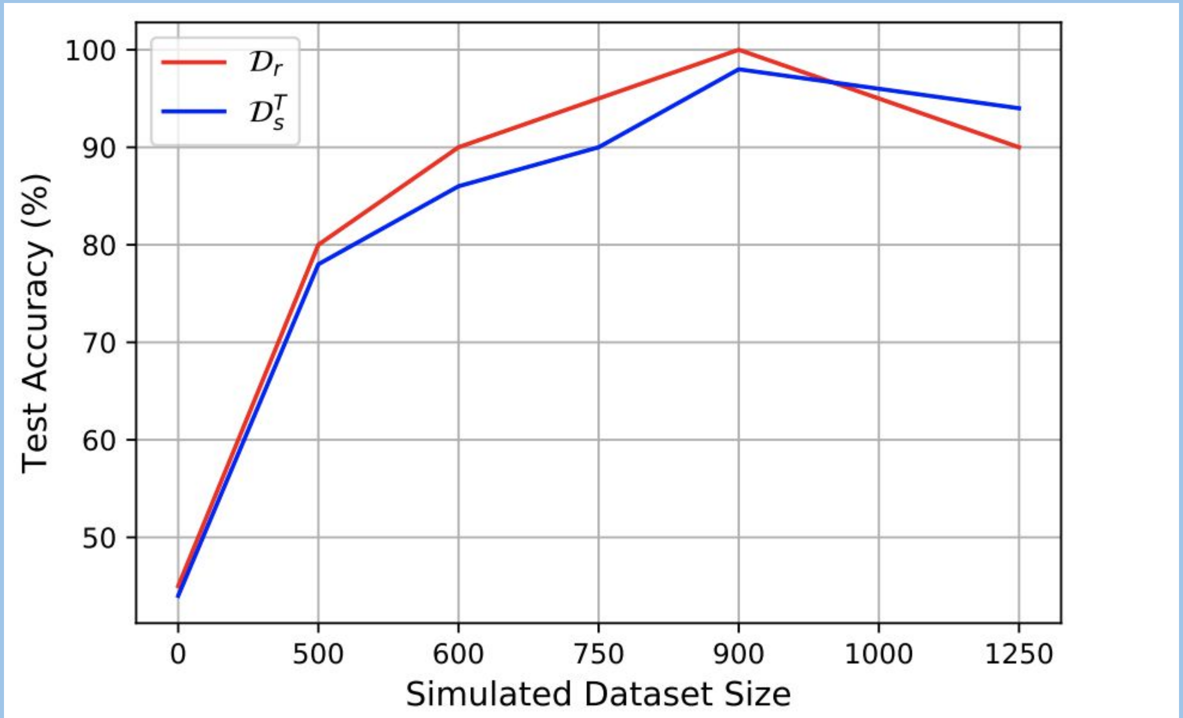
Results (Force Coefficient Estimation)

$\mathcal{D}_r$ Train/Test	Metrics			
	2pt Acc (%)	3pt Acc (%)	Mean Dev	Median Dev
Training	100	92.31	0.051	0.058
Testing	92.86	**85.71**	0.283	0.124

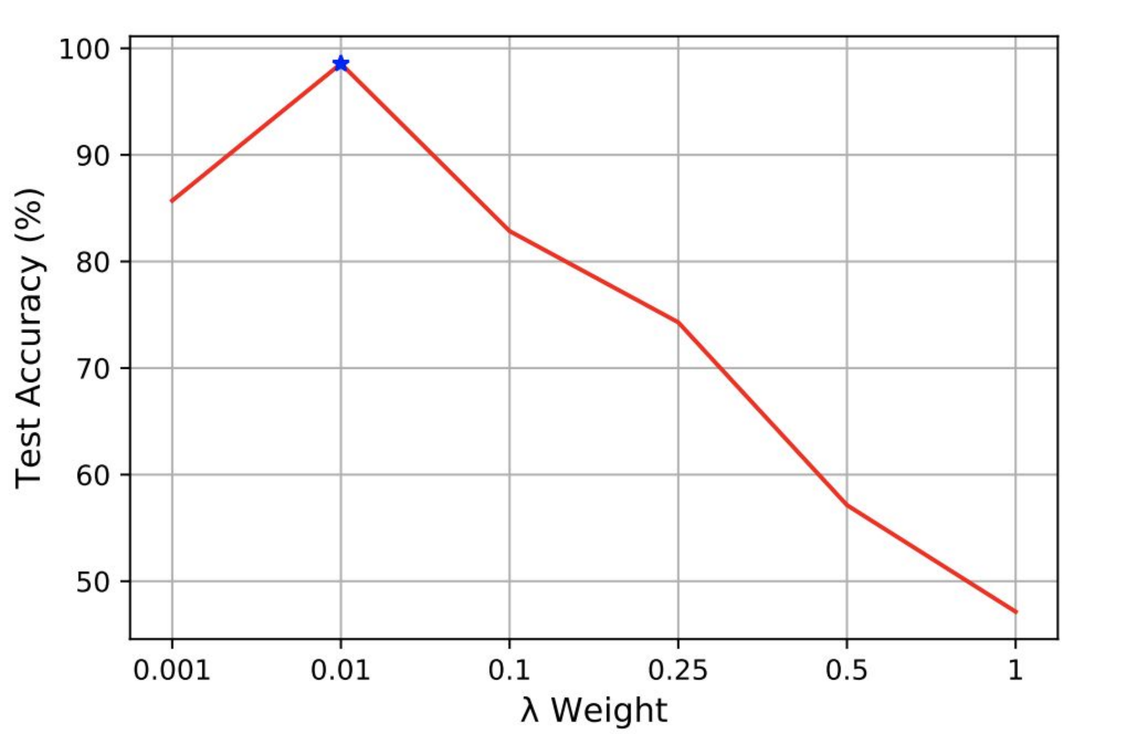
Results (Neural Network Performance)

$\mathcal{D}_s$ Size	Metrics (Combined evaluation of $\mathcal{D}_r$ and $\mathcal{D}_s^T$)		
	Overall Acc (%)	F1-3pt (%)	F1-2pt (%)
0	44.29	4.17	66.66
500	78.57	50.00	87.50
600	87.14	70.83	91.67
750	91.43	79.16	95.83
900	**98.56**	**95.83**	**100.00**
1000	95.71	91.67	95.83
1250	92.85	87.50	91.67

Results (Neural Network Performance)



Results (Neural Network Performance)



Conclusion and Application

- New method for modeling non-rigid object trajectory
- Algorithmic approach for estimating force coefficients based on experimental data
  - Force estimation is used to generate simulated data
- Simulated data is provided in supplemental to experimental data
  - Used to train a neural network to predict launcher outcomes
- Results confirm importance of simulated data
- System can be integrated with mobile robot to provide live-feedback predictions

Thank You For Listening!
Email or contact us with questions!

sajiv.shah@gmail.com, ayaanzhaque@gmail.com

•••

Sajiv Shah[1*], Ayaan Haque[1*], Fei Liu[2]

[1]Saratoga High School, [2]University of California, San Diego
[*]Authors contributed equally