Dynamic Upper-Body Reach with Trajectory Prediction

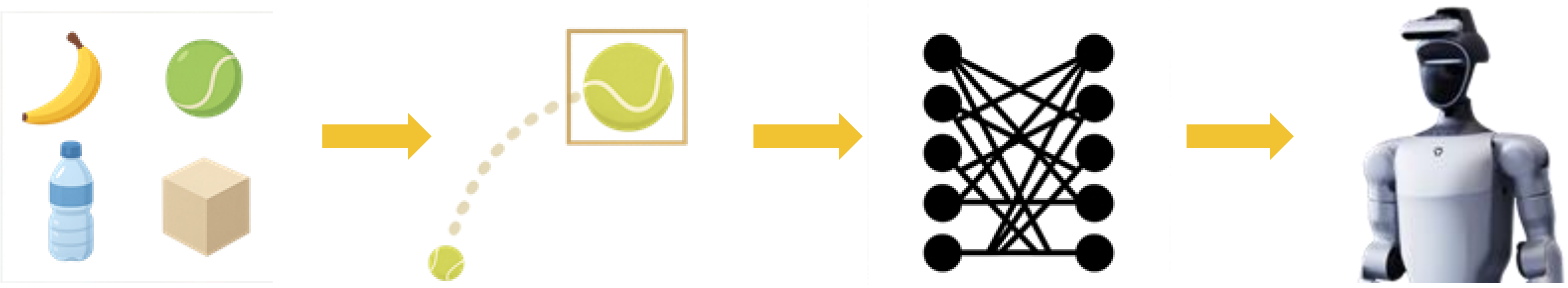
Presenter: Patrick (Huaze) Liu
Mentor: Kehlani Fay, Arth Shukla
2025.9.29

Overall Project Goal

- Build a unified platform for catching moving objects in simulation and real robots.
- Combine trajectory prediction, tactile sensing, and humanoid control into one system.

My Focus

- Designed and implemented **Trajectory Predictor**
- Developed **Upper-Body Reach Policy** (humanoid arm with locked legs)



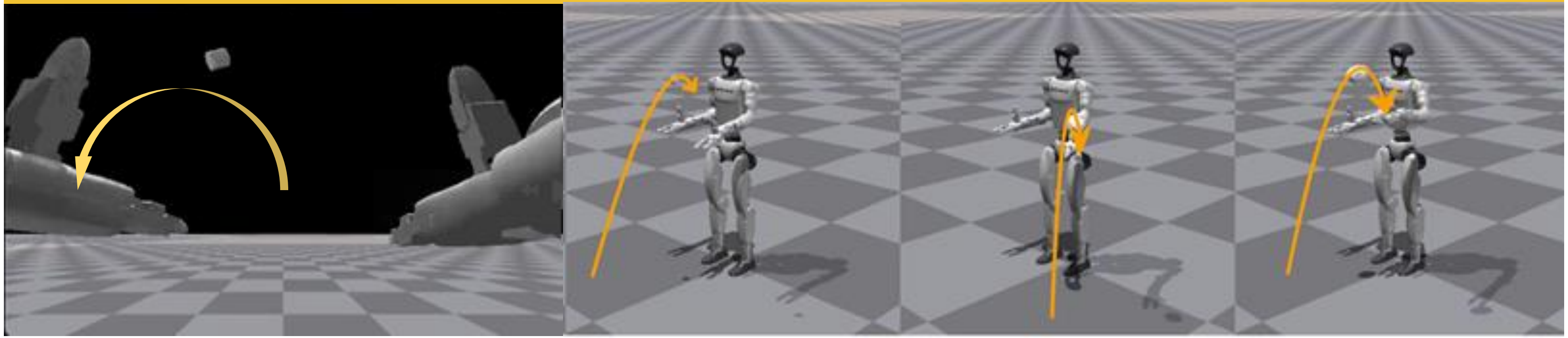
1. Motivation & Tasks

Constraints

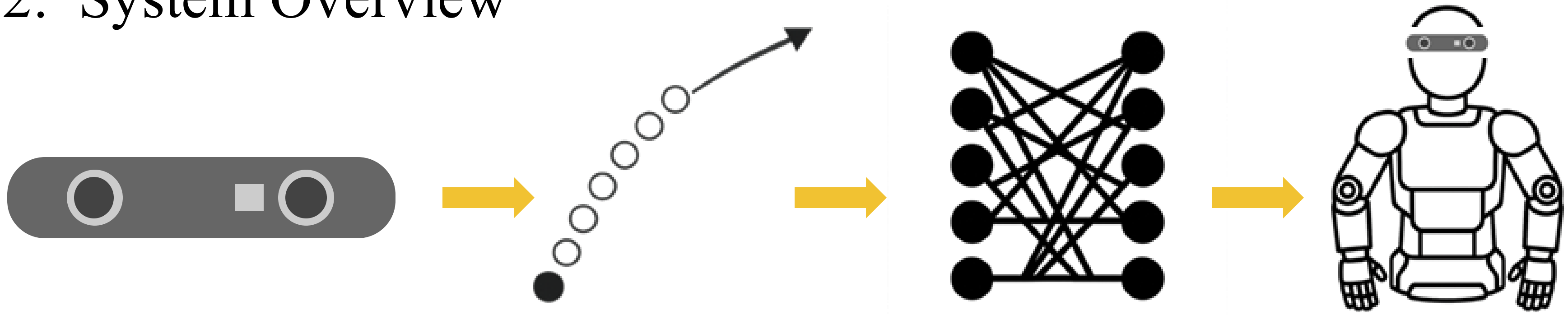
1. Upper body only (legs locked)
2. Noisy and delayed camera input
3. Ballistic object motion
4. Tight real-time control budget

Success Criteria

1. "Near" = within a few radius
2. "Still" = very low speed for a short time



2. System Overview



RGB-D Camera

Observations

Trajectory Predictor

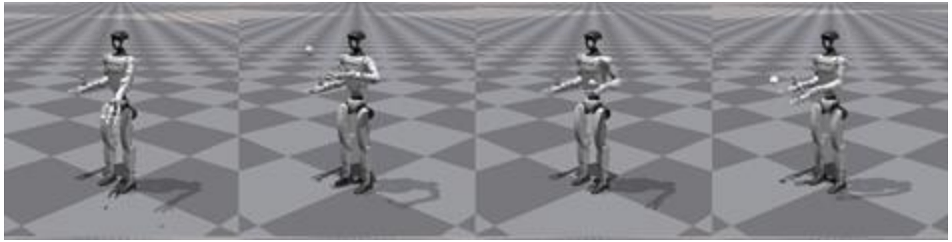
Future Positions

Policy (PPO-LSTM)

Actions

Upper Body Controller

Joint Commands



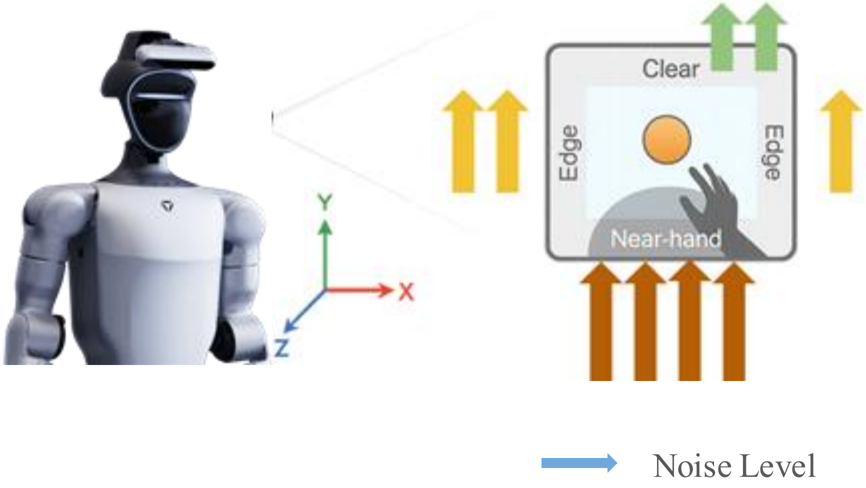
Simulator

Next State

Predict object's future
→ Act ahead of time

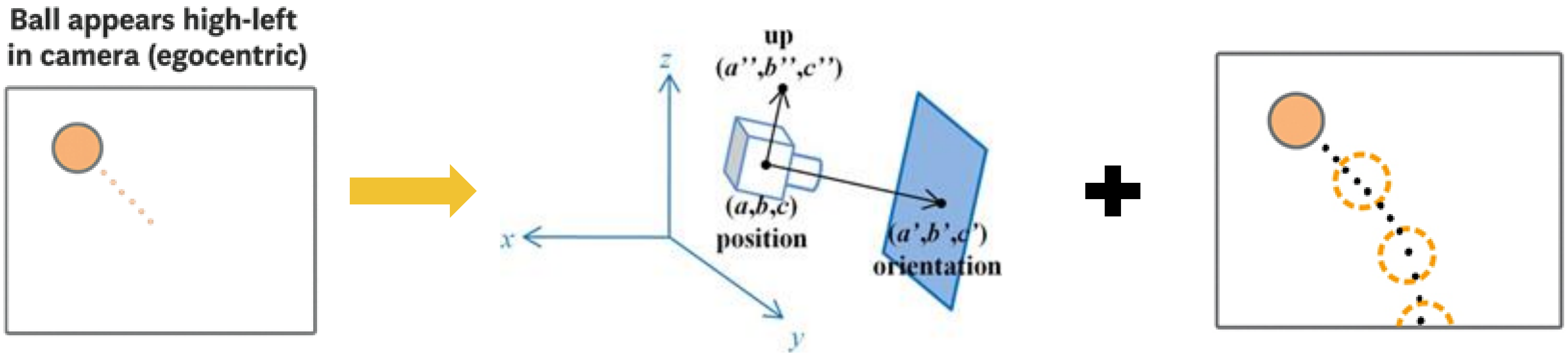
2.1 Sensing & Camera

- Setups: Head-mounted camera, world→camera transform
- Region-aware noise
- Dynamic scaling: farther/faster objects ⇒ noisier.
- Takeaway: All observations expressed in camera frame → **simplifies control**



2.1 Sensing & Camera

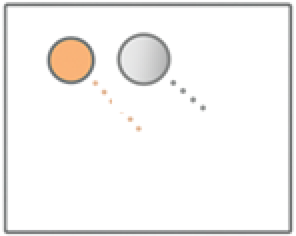
All observations are expressed in camera frame to simplified the control



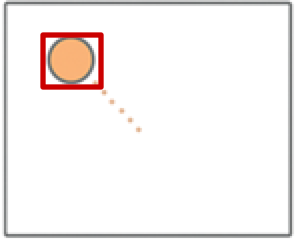
2.2 Trajectory Prediction

Perception

ZED 2i
RGB-D Camera



RGB Image
& Depth Map



Yolo-World
Object Detection

$[x, y, z]$

$[vx, vy, vz]$

Back-projected
3D/6D pose

Prediction

Model Fitting
X, Y → linear velocity
Z → quadratic (gravity)

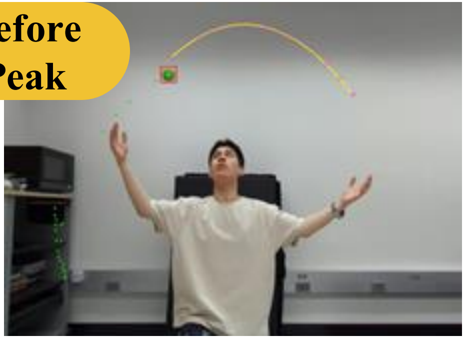
Algorithm
Z-axis
Concave Down

Output
Predicted States

ZED SDK 5.0 - Download | StereoLabs. (n.d.). https://www.stereolabs.com/developers/release

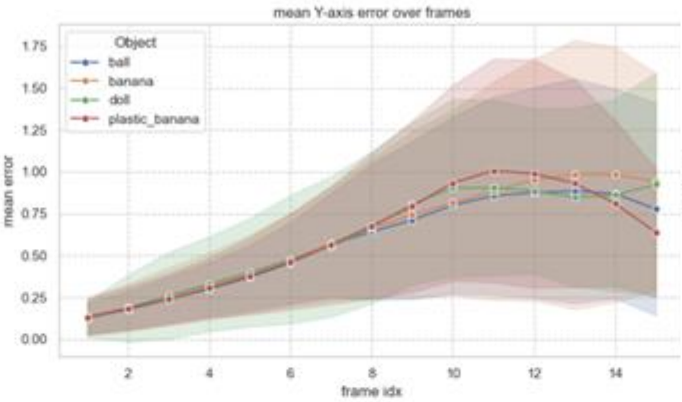
2.2 Trajectory Prediction

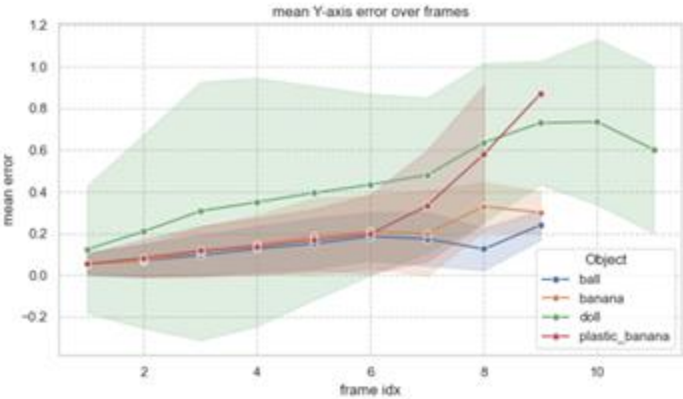
Before Peak



After Peak

Distance Error Analysis

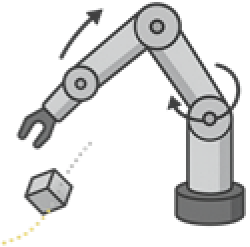




2.3 Policy Training – Observations & Actions

Observations

Arm States

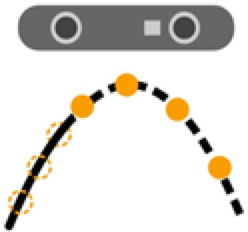


Arm angles

Arm speeds

Last command

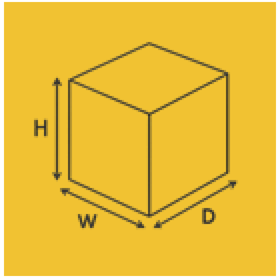
Object now + Future



Current Object pos (3D or 7D)

Future Object pos (N * 3D)

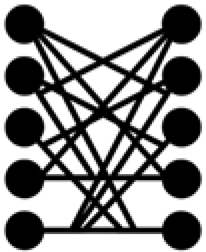
Size



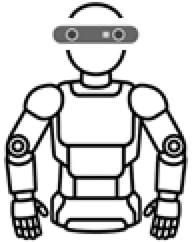
Actions

- Arm joint targets
- 🚫 Legs locked

obs

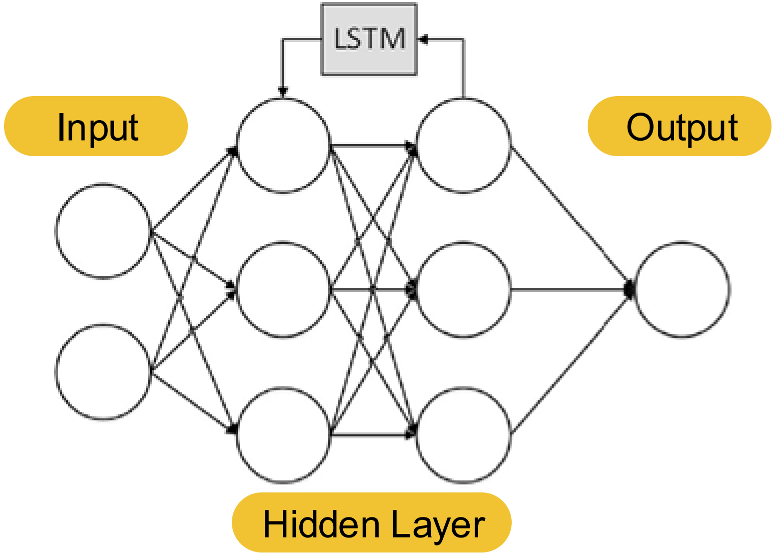


action



2.3 Policy Training – PPO-LSTM & Training Loop

- **PPO-LSTM** (partial observability & latency)
- **Rollout** – collect T steps from N parallel sims → build sequences for LSTM → reset hidden state at episode ends
- **Update** – PPO clipped objective + advantage estimates; early-stop by KL target
- **Repeat** – iterate collect



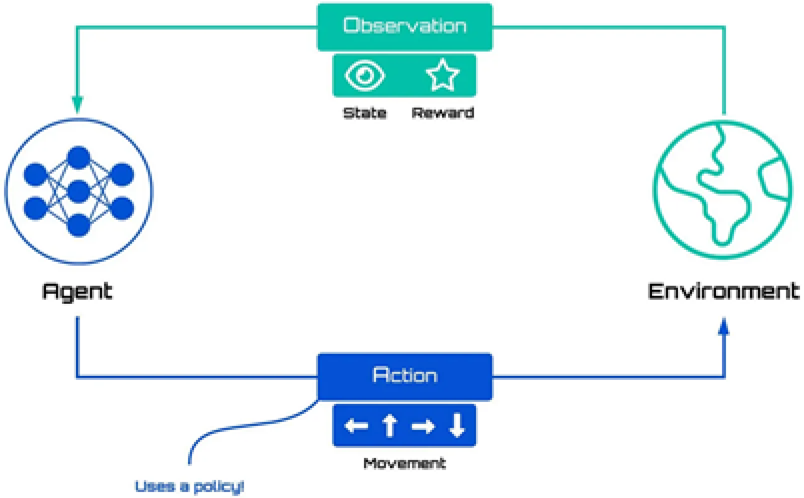
Why LSTM?

It remembers recent observations/actions.
Smoother and more robust control under laggy vision

Hodge, Victoria & Hawkins, Richard & Alexander, Rob. (2021). Deep reinforcement learning for drone navigation using sensor data.

2.3 Policy Training – Rewards

Rewards: What the robot is taught to care about.

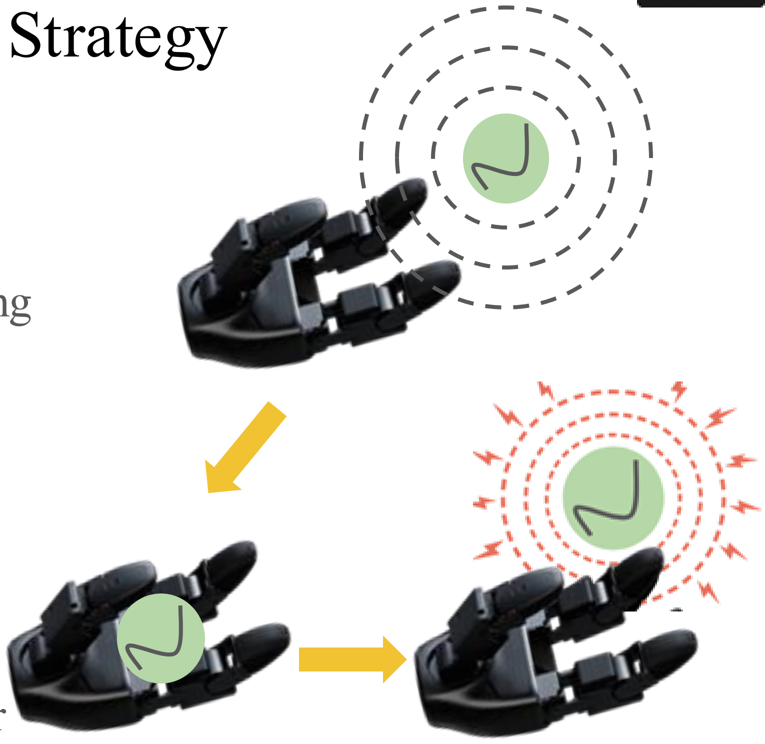


Reward Structure

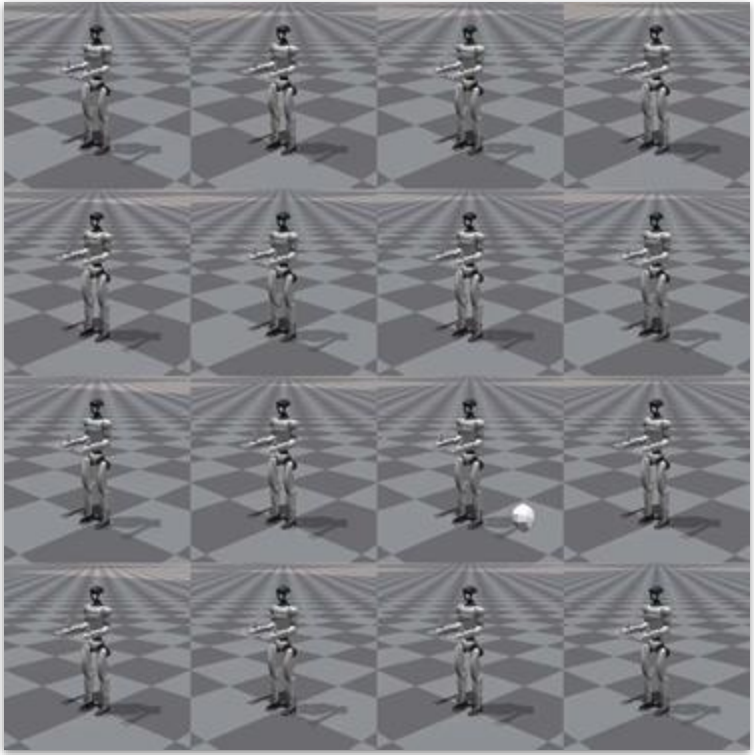
Reward type	What it measures
Proximity (✅ Get closer)	Hand–object distance, stable hold
Smoothness (✅ Stay stable)	Joint velocity / acceleration, tracking error
Safety (✅ Stay safe)	Torque, joint limits, contacts
Efficiency (✅ Don't waste effort)	Torque/energy Time shaping

2.3 Policy Training – Rewards Tuning Strategy

- Phase 1 – Reach first
  - High weight on proximity & hold rewards
  - Keep penalties light so policy learns basic reaching
- Phase 2 – Stabilize
  - Increase smoothness penalties (vel/acc)
  - Penalize contacts more → motions smoother
- Phase 3 – Safety hardened
  - Raise torque/limit penalties until spikes disappear
  - Ensure action clipping stays low
  - Keep success rate stable



3. Results – Baseline performance



Observation of Object	Components	
3D	[x, y, z]	Without angular info
7D	[x, y, z, qx, qy, qz, qw]	With angular info

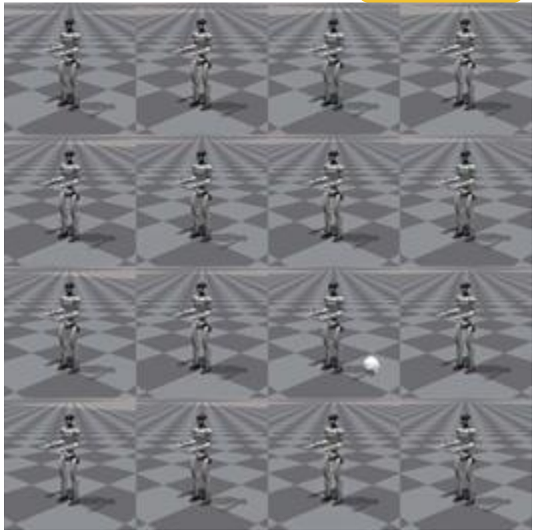
7D full pose: 83% success rate
3D pos: 73% success rate

➡️ **Orientation info is critical — without it, policy loses some stability.**

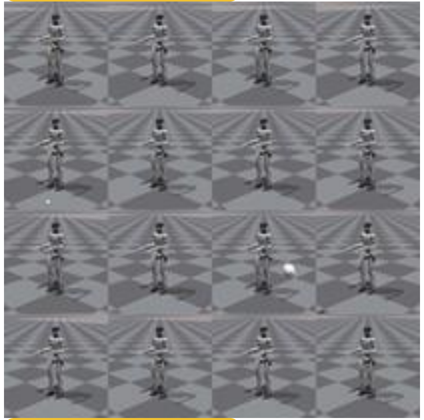
3. Results – Effect of noise

Noise Type	Success Rate	Comparison
No noise	83%	
DoF/Vel Noise	75%	-5%
Camera Reading Noise	70%	-10%, high oscillation

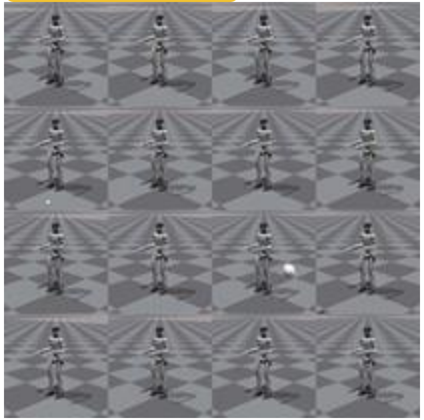
No Noise



DoF/Vel Noise



Camera Noise



➡ **Noise reduces stability (oscillation) more than raw accuracy.**

3. Results – Target position matters!

- What is the Target Position?

Observation Type	W/ Target Position	W/O Target Position	Comparison
7D	83%	76%	-7%
3D	73%	32%	-41%

➡️ **Target position is critical for 3D-only observations; without it, success collapses.**

UC San Diego

HARVEY MUDD COLLEGE

3. Results – Traj pred & critic observation

3D obs

Ablation	SR effect	Note
Traj pred (any # states)	~30% (w/o target position)	Ignored by policy
Critic 7D vs 3D	+ 5–8%	Marginal gain
Linear vs Quad Z	No major differences	Policy not using physics prior

➡️ **Trajectory prediction and critic variations had little impact → likely ignored by policy.**

UC San Diego

HARVEY MUDD COLLEGE

4. Summary

System & Framework

- Built an end-to-end trajectory prediction + control system in Isaac Gym
- Designed & tuned reward functions for real-robot constraints
- Implemented noise models to mimic real perception errors and latency

Experiments

- Ablation studies to explore each attribute's effect on the control policy
- Separate system blocks (perception vs. prediction vs. control)

Broader Takeaway

- Importance of sim2real realism
- Cross-disciplinary integration

UC San Diego

HARVEY MUDD COLLEGE