Proceedings of Machine Learning Research vol xxx:1–16, 2021

Are Learned Perception-Based Controllers Bound by the Limits of Robust Control?

Jingxi Xu JXU@CS.COLUMBIA.EDU
Columbia University, New York City, NY

Bruce Lee BRUCELE@SEAS.UPENN.EDU
Nikolai Matni NMATNI@SEAS.UPENN.EDU
Dinesh Jayaraman DINESHJ@SEAS.UPENN.EDU
University of Pennsylvania, Philadelphia, PA

Abstract

The difficulty of optimal control problems has classically been characterized in terms of system properties such as minimum eigenvalues of controllability/observability gramians. We revisit these characterizations in the context of the increasing popularity of data-driven techniques like reinforcement learning (RL) in control settings where input observations are high-dimensional images and transition dynamics are not known beforehand. Specifically, we ask: to what extent are quantifiable control and perceptual difficulty metrics of a control task predictive of the performance of various families of data-driven controllers? We modulate two different types of partial observability in a cartpole "stick-balancing" problem – the height of one visible fixation point on the cartpole, which can be used to tune fundamental limits of performance achievable by any controller, and by using depth or RGB image observations of the scene, we add different levels of perception noise without affecting system dynamics. In these settings, we empirically study two popular families of controllers: RL and system identification-based H_∞ control, using visually estimated system state. Our results show the fundamental limits of robust control have corresponding implications for the sample-efficiency and performance of learned perception-based controllers.

Keywords: Perception, robust control, reinforcement learning

1. Introduction

Data-driven techniques for robotic control such as deep reinforcement learning have recently become increasingly popular, especially for settings where input observations are high-dimensional, such as images, and state transition dynamics are not known in advance. These techniques have shown great promise for controlling a variety of robots ranging from manipulators (OpenAI et al., 2018) to legged robots (Lee et al., 2020), drones (Molchanov et al., 2019), and autonomous cars (Kendall et al., 2019). However, these techniques have largely been studied and developed within the confines of stylized, often simulated settings, where performance metrics are naturally divorced from important real-world concerns such as safety and robustness. In the light of recent catastrophic failures of learning-based control systems such as fatal autonomous car collisions (F. Siddiqui, 2018), we argue that it is imperative to study and characterize the *limitations* of these approaches in challenging settings that present realistic difficulties for observation and control. In particular, how do such difficulties affect the performance and sample complexity of learned controllers?

Classical robust control theory provides a rich set of tools characterizing fundamental limits on achievable performance in terms of system properties such as open-loop unstable poles and zeros. However, analogous theoretical results in the learning-based control literature are not nearly as well developed, especially in the context of controllers that involve high-capacity functional approximators such as deep reinforcement learning from pixels. Rather than seeking theoretical limits, we try a different tack, empirically studying various families of learned controllers in a setting where control and observation difficulty can be carefully tuned.

Our empirical study focuses on the visuomotor control task of stabilizing a cartpole in the upright position using only the visual observations from a camera with a head-on view. All visuomotor controllers must implicitly or explicitly solve two important and closely intertwined problems. The first is visual perception, i.e., how to map raw high-dimensional visual observations o to their task-relevant latent causes, denoted as the state representation x? The second is the task of synthesizing optimal action policies $\pi(u|x)$ conditioned on those state estimates.

In real world settings, perception is often an underconstrained problem. For example, an autonomous car with on-board cameras cannot see around the corner of a street on a left turn, or a pedestrian occluded behind a parked vehicle, or a pedestrian with dark clothes on a poorly lit street. In all such instances, the observations o are a non-invertible function of the relevant state x, and the estimated states $\hat{x}$ output from perception cannot match the state x perfectly, even with the most optimal perception system. This imperfect perception problem may be represented formally as a partially observable Markov decision process (POMDP). The successes of reinforcement learning in the last few years have been largely demonstrated on fully observed tasks, and general methods for tackling POMDPs remain elusive. Even when they are evaluated on real world robotic systems, robots and environments are typically overinstrumented to ensure near-full observability of all relevant state information, which is impractical for in-the-wild applications like autonomous driving.

Our visual cartpole balancing task permits the modulation of two realistic sources of partial observability, within the context of a well-studied classical control problem. First, a set fraction of the cartpole's length is constantly occluded from the camera. As we will explain, this type of information loss has been shown to induce fundamental limits on the performance of any controller for this system. Second, the sensing abilities of the camera itself may also limit perception. For example, to estimate the distance from the camera of an object in the scene such as the cartpole, a perception system with access to RGB camera observations would be harder to train and produce more noisy estimates than one receiving inputs from a stereo depth camera.

We study the impact of tuning these sources of perception noise on the performance of two families of learned controllers: system identification-based H_∞ control and reinforcement learning, both using visually estimated system state. Our careful empirical studies clearly show that increasing occlusion and deteriorating sensing quality affect both families of controllers in ways that align well with theoretically predicted limits for classical robust controllers. In particular, sample complexity increases and final task performance decreases, and the effect of sensing noise is exacerbated as more of the cartpole is occluded from view.

1.1. Related Work

Fundamental limits of learning-enabled control. Much of the research in the learning for control community has focused on characterizing *achievable upper bounds* on the performance of learning based control strategies. While such upper bounds are common in the literature, *lower bounds*

are few and far between. For model-free RL methods as applied to the Linear Quadratic Regulator, such lower-bounds can be found in (Tu and Recht, 2019), where the authors derive asymptotic lower bounds on the number of samples needed by both the least-squares-temporal-differencing estimator for policy evaluation, and policy gradient methods for policy improvement. In (Simchowitz et al., 2018; Simchowitz and Foster, 2020) information-theoretic lower bounds (Tsybakov, 2008; Duchi, 2016) are derived for arbitrary estimators, which in turn are used to show the optimality of certainty equivalent control for the LQ problem. We note that in both cases, such lower bounds are only available for *full-state* observation settings. Most similar in spirit to this paper in (Bernat et al., 2020), the authors empirically investigate the effects of loss of controllability and increased instability on the sample-complexity of policy gradient and certainty equivalent based methods. In (Venkataraman and Seiler, 2019), the authors show that traditional gradient descent methods converge to solutions with poor margins if applied to the counter-example system of (Doyle, 1978): however, they use analytic expressions in lieu of stochastic approximations of the gradients, and thus do not explore questions related to sample-complexity. To the best of our knowledge, no investigation of the effects of such fundamental limits on the sample-complexity and generalizability of perception-based learning-enabled control methods exist.

Reinforcement learning under partial observability. Reinforcement learning (RL)-based approaches have largely been studied in settings where the full Markov state information is available to the controller. When observations are noisy or incomplete, RL settings are typically framed as *partially observed* Markov decision processes (POMDP) (Jaakkola et al., 1995). In POMDPs, the system state x_t at time t is no longer available for training and running a standard RL control policy $\pi(u_t|x_t)$. In its place, all we have are observations o_t, which are non-invertible functions of the state x_t. To adapt standard RL algorithms to work in POMDPs, two broad families of approaches have been studied: memory-based RL and belief state RL. In memory-based approaches (McCallum, 1993; Hausknecht and Stone, 2015; Zhu et al., 2017), the input to the controller is no longer just the current observation, but instead the full history of observations and actions, so that the policy function is $\pi(u_t|o \leq t, u < t)$. Truly infinite history may become computationally intractable, so that a limited history window of size H may sometimes be used, containing only the last H observations and actions. In belief state RL (Kaelbling et al., 1998; Gregor et al., 2019; Gangwani et al., 2020; Weisz et al., 2018; Igl et al., 2018), a variable x_t is typically explicitly nominated by the control engineer as the Markov state based on knowledge about the task, and the conditional distribution $p(x_t|o \leq t, u < t)$ is maintained and updated with every new observation, as in standard recursive filtering approaches for state estimation. This conditional distribution, called the "belief state" b_t, is treated as the sufficient statistic of the full history for determining the optimal next action, so that the policy function is $\pi(u_t|b_t)$. This is equivalent to running RL on a newly constructed fully observed Markov decision process (MDP), called the "belief MDP", whose states are the beliefs b_t. To the best of our knowledge, RL approaches for POMDPs have not thus far been systematically evaluated under realistic sources of incompleteness or noise in high-dimensional visual observations. In recent works proposing RL algorithms for POMDPs, evaluation is performed exclusively with synthetic noise added to low-dimensional state observations, or with "flickering" visual observations of Atari games (Hausknecht and Stone, 2015; Zhu et al., 2017; Igl et al., 2018). These evaluations neither resemble real perceptual difficulties, nor involve controlled experiments where the degree and type of partial observability is tuned. We address these gaps in our work.

2. Preliminaries

Robust Control and Fundamental Limits. Consider a single-input, single-output (SISO) linear-time-invariant (LTI) system

$$x_{t+1} = Ax_t + Bu_t, \quad z_t = Cx_t + Du_t \tag{1}$$

with transfer function representation given by $z(\zeta) = C(\zeta I - A)^{-1}B + D =: P(\zeta)u(\zeta)$, obtained via the z-transform of (1), and ζ is a complex number frequency parameter.

Furthermore, consider the feedback control system illustrated in Fig. 1. In this setup, the reference input r is injected along with the control input u into the system, which produces a controlled output z, which the controller $C(\zeta)$, itself a LTI system, attempts to keep small by using knowledge of the system P and noisy measurements $y = z + n$. Through straightforward algebra, one can compute that the true output z of the system P under this feedback interconnection is given by

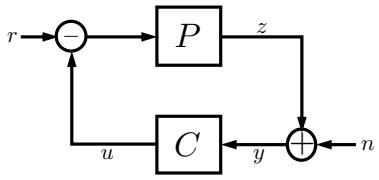

Fig. 1: Linear feedback control diagram.

$$y(\zeta) = P(\zeta)S(\zeta)r(\zeta) - T(\zeta)n(\zeta), \quad S(\zeta) := \frac{1}{1 + P(\zeta)C(\zeta)}, \quad T(\zeta) := \frac{P(\zeta)C(\zeta)}{1 + P(\zeta)C(\zeta)},$$

where $S(\zeta)$ and $T(\zeta)$ are the *sensitivity and complementary sensitivity functions* (Doyle et al., 2013) of the feedback interconnection of Fig. 1, respectively. These objects capture the closed-loop maps from reference input $r(\zeta)$ and sensor noise $n(\zeta)$ to the regulated output $w(\zeta)$, and thus through an appropriate quantification of their magnitudes, they can be used as measures of closed-loop performance. One commonly used measure of system magnitude is the H_∞-norm.

Definition 1 *The H_∞ norm of a transfer function $G(\zeta)$ is defined as $\|G\|_\infty = \sup_{\omega \in [-\pi, \pi]} |G(e^{j\omega})|$.*

We note that via Parseval's theorem, the H_∞ norm of a system also admits a time-domain interpretation as the worst-case $\ell_2 \to \ell_2$ gain of the filter G_t satisfying $\mathcal{Z}(G_t) = G(\zeta)$, where $\mathcal{Z}$ is the z-transform. As our study focuses on perception-based control, much of our analysis will be focused on characterizing the effects of sensing noise $n(\zeta)$, and hence, the object of concern will be the H_∞ norm of the complementary sensitivity function $T(\zeta)$. To that end, we conclude this section with a useful theorem that allows us to lower bound $\|T(\zeta)\|_\infty$ as a function of the open-loop unstable poles and zeros of the system $P(\zeta)$ for *any possible controller* $C(\zeta)$, thus establishing fundamental limits on achievable performance.

Theorem 2 (Ch.6, Doyle et al. (2013)) *Assume that the system $P(\zeta)$ has an unstable pole p and unstable zero q. We then have that $\|T(\zeta)\|_\infty \geq \left|\frac{pq-1}{q-p}\right|$.*

The proof of this Theorem is in Appendix A.

Understanding Limits of Perception-Based Control via Stick Balancing. We propose using the "stick-balancing" example from (Leong and Doyle, 2016; Doyle et al., 2013), shown in Fig. 2 (Left), as a case study. The dynamics of such a one-dimensional inverted pendulum on a moving cart are given by the following second order ordinary differential equations:

$$(M+m)\ddot{h} + m\ell(\ddot{\theta} - \dot{\theta}^2 \sin\theta) = u + r, \; m(\ddot{h}\cos\theta + \ell\ddot{\theta} - g\sin\theta) = 0, \; z = h + \ell_0 \sin\theta$$

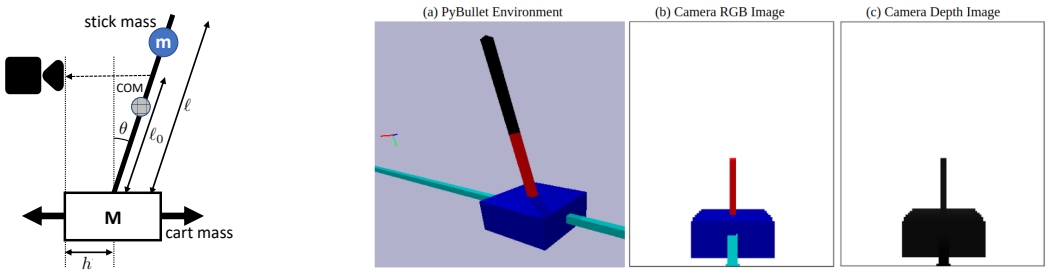

Fig. 2: (Left) Schematic of the simplified cartpole system. (Right) Our custom PyBullet environment: (a) a side-on view of the scene to visualize the camera pose with respect to the cartpole system, as well as the occluded portion of the cartpole (in black). (b) RGB image from the camera, (c) depth image from the camera.

where z is the distance to point of interest on the pole, which we call the *fixation point*, u is the control force applied to the cart, r is the reference point (set to 0 for all of our experiments), θ is the pendulum tilt angle from vertical, and h is the horizontal displacement of the cart from the origin. Additional parameters describing the system are the cart mass M, the pole mass m, the acceleration due to gravity g, the *fixation point* ℓ_0, and the length of the pole ℓ. We further introduce a sensor measurement reading $y = z + n$ for n the sensor noise, to capture the effects of imperfect perception in our model. Discretizing the system with euler integration using a step size of τ and linearizing the system about the upright position $x = \dot{x} = \theta = \dot{\theta} = 0$, one can check that the poles of the resulting linear open-loop system are found at 1 and $1 \pm \tau\sqrt{\frac{(M+m)g}{M\ell}}$. The system has no zeros if $\ell_0 = \ell$, and zeros at $1 \pm \tau\sqrt{\frac{g}{\ell-\ell_0}}$ when $\ell_0 < \ell$.

This system features two important properties: (i) it unstable, and when $M \gg m$, it has an open-loop unstable pole $p \approx 1 + \frac{\tau g^{1/2}}{\ell^{1/2}}$, and (ii) when $\ell_0 < \ell$, it is non-minimum phase, with open-loop unstable zero $q = 1 + \frac{\tau g^{1/2}}{(\ell-\ell_0)^{1/2}}$ where we recall that ℓ_0 denotes the *fixation point* on the stick where the camera is looking (see Fig. 2). It then follows from Theorem 2 that we can can make the control problem *quantitatively* more difficult by letting $\ell_0 \to 0$, i.e., by letting the open-loop unstable zero $q = 1 + \frac{\tau g^{1/2}}{(\ell-\ell_0)^{1/2}}$ approach the open-loop unstable pole $p \approx 1 + \frac{\tau g^{1/2}}{\ell^{1/2}}$, which in turns implies that $\|T(s)\|_\infty \gtrsim 1/\ell_0$. We emphasize that for small $|\theta|$ this bound holds *for all possible controllers* (Dahleh and Diaz-Bobillo, 1994), including the optimal H_∞ controller, and thus represents a fundamental lower bound on achievable performance.

3. Experiments

In this section, we investigate the performance of data-driven techniques for learning controllers on a customized cartpole balancing environment in PyBullet (Coumans and Bai, 2016), which allows for varying fixation lengths ℓ_0, observation quality, and injecting camera sensing noise. We study two techniques under different measurement models based on learned perception modules: a model-free RL algorithm and a system identification-based robust H_∞ control algorithm. We simulate and vary the fixation point variable discussed above, as well as measurement models and camera sensing noise, in order to tune them to observe their effects on controller performance. Our experiments aim to answer: **(1)** Is the performance of these learned perception-based controllers subject to the fundamental limits on achievable performance specified by Theorem 2? **(2)** What

effect does incomplete sensing (as measured by fixation length and the corresponding bounds of Theorem 2) and noisy sensing (as measured by the magnitude of simulated camera sensing noise) have on learning perception-based controllers?

Appendices, implementation details, and supplementary material for all experiments, are at the following URL: https://bit.ly/2Hj3364

3.1. Environment

We developed a custom "stick-balancing" environment in PyBullet, illustrated in Fig. 2. A cartpole system is actuated by moving the cart along a sliding track (in cyan) along x axis of world frame. The mass of the cart $M = 1\text{kg}$, the mass of the pole $m = 0.1\text{kg}$ and the length of the pole $l = 1\text{m}$.

Fixation	Depth	RGB
1.0	0.43	3.41
0.9	0.38	3.25
0.8	0.33	3.13
0.7	0.31	3.00

Tab. 1: ℓ_1 norm of perception error for estimating the fixation point depth z in mm.

The camera has a *head-on* view of the cartpole, as depicted in Fig 2(a). To simulate varying fixation points, we occlude a fixed fraction of the pole starting from its top, by setting it to be invisible in PyBullet. This is depicted in black in the simulator view in Fig 2(a). In our experiments, the controller's observations are either direct depth measurements of the fixation point (labeled z in Fig. 2), depth images (Fig 2(b)), or RGB images (Fig 2(c)). We simulate the parameters of an Intel RealSense D415 camera [1], and downsample and crop images to 120x100 pixels in all our experiments.

At the beginning of every episode of training and testing, the configuration variables $(x, \dot{x}, \theta, \dot{\theta})$ of the cartpole are all initialized randomly from a uniform distribution over $[-0.05, 0.05]$ (respectively m, m/s, rad and rad/s for the four variables). This environment is simulated at 50Hz (each time step equals 0.02s) for 500 steps (10s) in an OpenAI Gym framework. The episode terminates and resets after 500 steps, or when x goes outside $[-0.6\text{m}, 0.6\text{m}]$ or θ goes outside $[-15°, 15°]$.

3.2. Methods

Learning Perception Models. When dealing with images from the camera as input, the role of the perception system is to "invert" the observations into an estimate of the depth z of the fixation point, as depicted in Fig 3. Input images are either depth images or RGB images to allow us to tune partial observability from sensing limitations: we expect that depth images will enable more reliable estimates of z than RGB images.

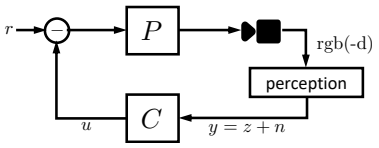

Fig. 3: Perception-based feedback control diagram.

For each value of the fixation point, and each of depth / RGB images, we train a convolutional neural network (see Appendix for architecture details) to minimize a mean squared error regression loss on target z labels, using stochastic gradient descent with the Adam optimizer (Kingma and Ba, 2014). Our training set contains 40K images with associated z labels, collected by uniformly sampling x and θ from the allowable range. Each model is trained with early stopping based on performance on held-out data.

Perception errors for different sensing modalities and fixation heights are shown in Table 1. As expected, the error in perceiving z from camera images is significantly higher with RGB cameras

1. https://www.intelrealsense.com/depth-camera-d415/

than with depth cameras. Errors are slightly lower at lower fixation heights, which may be due to smaller range of variation in z.

Robust Control with System Identification. Here we take a classical system identification and robust control approach, where the only difference between variants are the inputs to the system identification and robust controller.

System Identification: We first randomly generate system trajectories starting from initial state satisfying $\|x(0)\|_\infty \leq .05$, and record the resulting depth measurement outputs. We excite the system with control inputs drawn as $u(t) \overset{\text{iid}}{\sim} U[-10, 10]$ until the horizontal position of the cart deviates by more than 0.6m from its initial position, or the pole deviates more than 15° from the vertical. This data-collection step is repeated for differing numbers of trajectories, for fixation values of $\ell_0 = 1.0, 0.9, 0.8, 0.7$, and for outputs consisting of true depth measurements, depth estimates produced by a perception-map acting on depth images, and depth estimates produced from a perception-map acting on RGB images. We also record the full system state, as this will be used to identify a "baseline" system model for comparison. The result is a collection of input/output trajectories $\{z^{(i)}(0:T_i), u^{(i)}(0:T_i)\}_{i=1}^N$ to be used by a system identification algorithm.

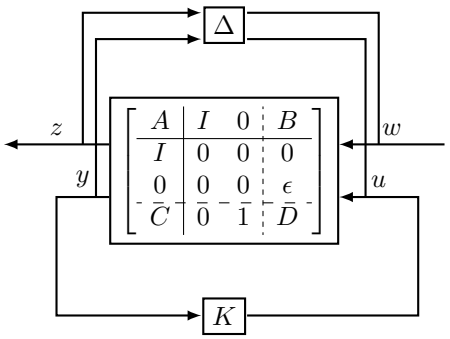

Fig. 4: The H_∞ controller K minimizes the worst case H_∞ norm of the closed loop system over all uncertainties $||\Delta||_\infty < 1$.

We fit a strictly causal linear time invariant model (1) with parameters $(\hat{A}, \hat{B}, \hat{C})$ from the collected trajectories $\{z^{(i)}(0:T_i), u^{(i)}(0:T_i)\}_{i=1}^N$. For the collected data consisting of true and estimated depth measurements, we use both N4SID (Van Overschee and De Moor, 1994) and a standard two step procedure which first fits an auto-regressive model $\hat{G}$ of order p by solving the least squares problem

$$\sum_{i=1}^{N}\sum_{t=p}^{T_i}\left\|z^{(i)}(t) - \begin{bmatrix} z^{(i)}(t-1) & u^{(i)}(t-1) & \dots & z^{(i)}(t-p) & u^{(i)}(t-p)\end{bmatrix} G\right\|_2^2,$$

and then applies the Ho-Kalman algorithm (Ho and Kalman, 1966) to obtain state-space parameters $(\hat{A}, \hat{B}, \hat{C})$. We refer to this latter approach as ARXHK. Exploiting our prior knowledge of the underlying physics of the system, we set the state-dimension (dimension of $\hat{A}$) to $n = 4$. We set the auto-regressive order $p = 10$ for ARXHK. For the collected data consisting of full state, we fit the state transition matrices $(\hat{A}, \hat{B})$ by solving a least-squares problem

$$\sum_{i=1}^{N}\sum_{t=0}^{T_i-1}\left\|x^{(i)}(t+1) - Ax^{(i)}(t) - Bu^{(i)}(t)\right\|_2^2,$$

and set $C = [1,\ 0,\ \ell_0,\ 0]$ such that $z = h + \ell_0\theta$.

Robust Control Synthesis: Once parameters $(\hat{A}, \hat{B}, \hat{C})$ are identified, we use tools from robust control to synthesize a controller that can mitigate the effects of uncertainty in the learned model introduced by noise in the measurements and by approximating nonlinear dynamics with a linear time invariant model (1). In particular, an H_∞ controller is synthesized by drawing on tools

Fixation	Noise-Free z		Depth Images		RGB Images	
	RL	H_∞	RL	H_∞	RL	H_∞
1.0	500.00, 1.00	380.44, 0.76	500.00, 1.00	396.42, 0.66	419.76, 0.58	360.80, 0.35
0.9	500.00, 1.00	295.56, 0.57	499.75, 0.99	240.74, 0.41	252.05, 0.01	228.63, 0.31
0.8	500.00, 1.00	127.31, 0.26	471.12, 0.69	90.17, 0.00	124.75, 0.00	23.89 ,0.00
0.7	88.79, 0.00	13.99,0.00	93.17, 0.00	3.80, 0.00	80.79, 0.00	5.9, 0.00

Tab. 2: Learned perception-based controller performance reported as "average reward, success rate" for both RL and H_∞ controllers, as a function of fixation height and observation quality.

from structured singular value, or μ, synthesis (Zhou et al., 1996) (see Appendix for more details). Specifically, we introduce unstructured uncertainty in feedback with the learned model, as shown in Fig. 4. To synthesize the controller we perform cross-validation on the parameter ε, which penalizes control effort, until a suitably high-performing but robust controller is found. Once a value of ε has been found to work at the fixation point of 1.00 for a perception map it is kept fixed throughout the remaining experiments with the perception map.

Reinforcement Learning from Estimated State. We use Soft Actor-Critic (Haarnoja et al., 2018) for training our RL agents. SAC is a widely used state-of-the-art model-free RL algorithm that has been demonstrated to work well in continuous control settings. The state of the RL agent is the sequence of estimates of the fixation depth value z from the past H steps. At each time step, a new z estimate output by the perception model is appended to the history buffer and the oldest one is abandoned. We set history size $H = 200$ based on validation performance. The reward is structured as a survival reward: the agent earns a unit reward for every timestep survived in the environment without episode termination. Since the maximum length of an episode is $T = 500$, the maximum achievable reward is 500. Each agent is trained up to 10K episodes with early stopping. We report results based on 100 trials.

Performance Metrics. At each fixation depth, for each type of sensor (noise-free observations of the true z, depth image observations, RGB image observations), and for each family of learned controller, we report the average reward earned per episode (same as the survival time), over 100 episodes, and also the success rate, which is the fraction of episodes in which the agent survived successfully up to $T = 500$.

3.3. Results

Tab. 2 shows the performance of RL and H_∞ controllers as the fixation height and observation quality are varied. Each RL controller is trained for a maximum of 10K episodes with early convergence, and each H_∞ controller is trained with up to 20k data points used for system identification.

Tab. 2 presents two main trends. First, performance uniformly deteriorates as the fixation height decreases and more of the pole is occluded from view. This is true for both RL and H_∞ controllers, and at all observation qualities. Next, not only does the performance uniformly worsen as the observation quality deteriorates (from noise-free to depth images to RGB images), but the impact of poorer observation quality is higher at low fixation heights, as can be seen by comparing rows 1 and 3 in the table. Both these experimental findings are closely aligned with the robustness limitations predicted by Theorem 2, which states that sensing noise (which will be larger when observation

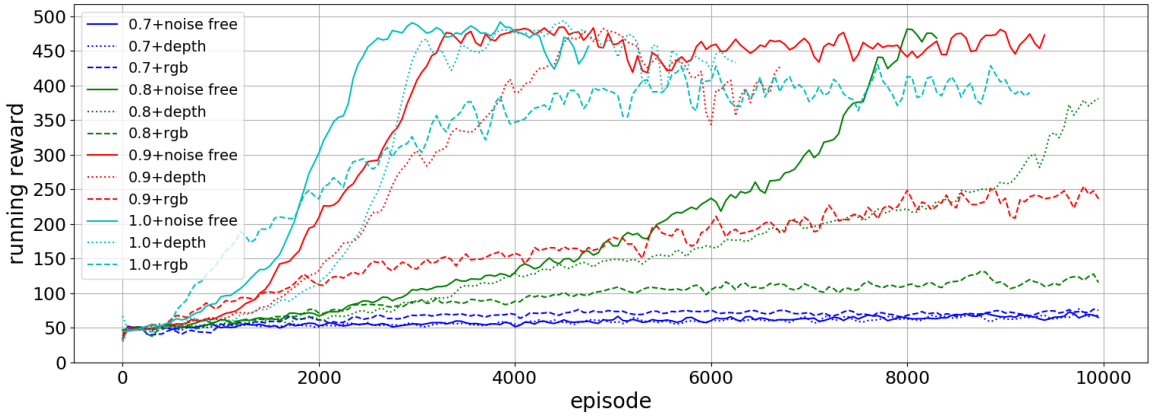

Fig. 5: As the fixation length ℓ_0 decreases from 1.0 through 0.7 and the quality of perception deteriorates from noise-free true fixation point depth to depth images to RGB images, the reinforcement learning agent finds it harder to stabilize the cartpole. It takes longer to train, and also achieves lower eventual reward at convergence. Furthermore, the effect of perception noise is higher at lower fixation lengths.

quality is poorer) will be amplified more (as measured by $\|T(s)\|_\infty$) at lower fixation points ℓ_0.This suggests that incomplete and noisy sensing act synergistically to further compound the difficulty of the control task when they co-occur.

Overall, Tab. 2 establishes that fixation height and observation quality are very effective at modulating the achievable performance levels for both families of learned controllers. Next, we ask: do these factors also predict the learning speeds for these controllers? Fig. 5 shows the training plots (running average of rewards vs. the number of training episodes) for the RL agents in each setting. Agents take longer to learn at lower fixation heights and lower observation quality, and in keeping with the results in Tab. 2, they also eventually converge to worse performance. We conjecture that the increased difficulty in the underlying control problem leads to systems for which only near-optimal policies provide meaningful reward signals, which manifests itself in the increased learning times observed in Fig. 5: we leave a formal investigation of this phenomenon to future work.

Model fit	**True state**			**Noise-free z**
Fixation	Noise-free z	Depth	RGB	Noise-free z
1.0	7.73	8.94	7.05	7.05
0.9	7.33	7.28	7.16	5.84
0.8	7.28	6.93	6.70	5.44
0.7	6.53	6.53	6.30	4.98

Tab. 3: Range of stabilized initial angles for a controller synthesized from a model fit using full states from 100 trajectories and tested on the three observation scenarios (left) a model fit using true depth measurements with N4SID, with data from 100 trajectories (right).

Finally, we investigate why the performance of the H_∞ controller almost uniformly lags behind that of the RL agent in Tab. 2. Note that H_∞ controllers tend to have higher success rates at lower average reward than RL agents: for example, observe the performance of the RL and H_∞ controllers at fixation 0.9, with RGB images. This happens because H_∞ controllers tend to perfectly stabilize the cartpole when it is initialized with small deviations from the vertical, but they fail almost immediately outside this basin of attraction. To illustrate, the 50th and 75th percentile of H_∞ controller rewards at fixation point 0.9 using depth images are 3, and 500 respectively.

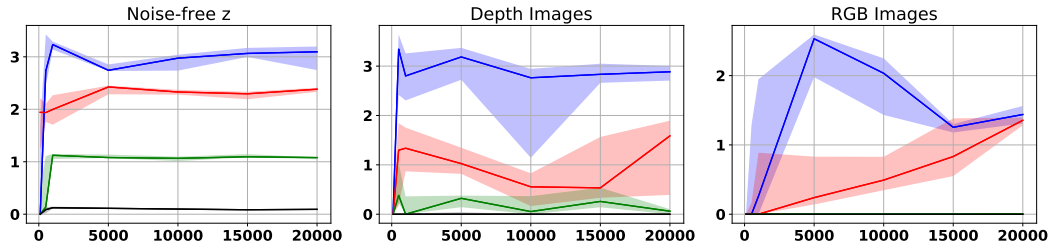

Fig. 6: The range angles stabilized by the H_∞ controller fit to a model using ARXHK are plotted with varying amounts of data. The x axes are the number of samples used by the identification algorithm. The blue, red, green and black curves are for fixations of $1.0, 0.9, 0.8$ and 0.7 respectively.

To better understand this phenomenon, we estimate the range of initial angles (in degrees) for which the H_∞ controllers stabilize the systems. First, Tab. 3 considers the angles for which a controller fit to the *full state* or the true z observations successfully stabilizes our system. As these models use full state observations, the estimated stabilizing range of angles serves as a rough upper bound for the stabilizing range of controllers synthesized from only noisy observations.

Now, we measure the stabilizing range of our H_∞ controllers synthesized using ARXHK and noisily perceived z from depth and RGB images. For each type of observation, we plot the stabilizing range vs. the amount of data used to fit the model in Fig 6. As the perception problem becomes more difficult with lower fixation points and noisier z measurements, the stabilizing region grows smaller. Also of note is the step-like response in the sample-complexity curves of Fig. 6: the ARXHK and H_∞ based method required only a few hundred data points to saturate the performance achievable by their model class. This further suggests that by fitting a slightly richer model (e.g., piecewise linear) and relying on a slightly more sophisticated robust control method (e.g., gain scheduling), the regions of attractions could be expanded to match those of the RL controllers while still requiring much less data.

4. Discussion

Through the use of a theoretical model of a simple one-dimensional cartpole system and corresponding customized experimental environment, we have empirically evaluated the consequences of well understood fundamental limits on the performance achievable by learned perception-based controllers. In particular, we examined the effects of limits imposed by unstable dynamics combined with realistic sources of partial observability through incomplete or noisy visual perception. Our results suggest that these fundamental limits propagate through to other aspects of the learning and control pipeline. For example, Fig. 5 suggests that training time required to achieve a given level of performance is negatively affected by both poor (low fixation point ℓ_0) and noisy sensing. We also observed similar trends in performance (see Tab. 2), as measured by reward and success rate for the RL controller, and success rate for the robust H_∞ controller. We believe that our results are not the consequences of phenomenological behavior unique to the simple system studied in this paper, but that they rather hint at a deeper, more fundamental interplay between how difficult it is to sense and control a system, and how difficult it is to learn to control it.

Acknowledgments

We thank Natalie Bernat and John C. Doyle for helpful feedback and comments.

References

Natalie Bernat, Jiexin Chen, Nikolai Matni, and John Doyle. The driver and the engineer: Reinforcement learning and robust control. In *2020 American Control Conference (ACC)*, pages 3932–3939. IEEE, 2020.

Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. 2016.

Munther A Dahleh and Ignacio J Diaz-Bobillo. *Control of uncertain systems: a linear programming approach*. Prentice-Hall, Inc., 1994.

John C Doyle. Guaranteed margins for lqg regulators. *IEEE Transactions on automatic Control*, 23 (4):756–757, 1978.

John C Doyle, Bruce A Francis, and Allen R Tannenbaum. *Feedback control theory*. Courier Corporation, 2013.

John Duchi. Lecture notes for statistics 311/electrical engineering 377. *URL: https://stanford. edu/class/stats311/Lectures/full_notes. pdf. Last visited on*, 2:23, 2016.

M. Laris F. Siddiqui. Self-driving uber vehicle strikes and kills pedestrian, 2018.

Tanmay Gangwani, Joel Lehman, Qiang Liu, and Jian Peng. Learning belief representations for imitation learning in pomdps. In *Uncertainty in Artificial Intelligence*, pages 1061–1071. PMLR, 2020.

Karol Gregor, Danilo Jimenez Rezende, Frederic Besse, Yan Wu, Hamza Merzic, and Aaron van den Oord. Shaping belief states with generative environment models for rl. *Advances in Neural Information Processing Systems*, 32:13475–13487, 2019.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *2015 AAAI Fall Symposium Series*. aaai.org, 2015.

B. L. Ho and Rudulof Kalman. Effective construction of linear state-variable models from input output functions. *at - Automatisierungstechnik*, 14(1-12):545 – 548, 1966.

Maximilian Igl, Luisa Zintgraf, Tuan Anh Le, Frank Wood, and Shimon Whiteson. Deep variational reinforcement learning for POMDPs. June 2018.

Tommi Jaakkola, Satinder P Singh, and Michael I Jordan. Reinforcement learning algorithm for partially observable markov decision problems. In *Advances in neural information processing systems*, pages 345–352, 1995.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artif. Intell.*, 101(1):99–134, May 1998.

Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8248–8254. IEEE, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Sci Robot*, 5(47), October 2020.

Yoke Peng Leong and John C Doyle. Understanding robust control theory via stick balancing. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 1508–1514. IEEE, 2016.

R Andrew McCallum. Overcoming incomplete perception with utile distinction memory. 1993.

Artem Molchanov, Tao Chen, Wolfgang Hönig, James A Preiss, Nora Ayanian, and Gaurav S Sukhatme. Sim-to-(Multi)-Real: Transfer of Low-Level robust control policies to multiple quadrotors. March 2019.

OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous In-Hand manipulation. August 2018.

Max Simchowitz and Dylan J Foster. Naive exploration is optimal for online lqr. *arXiv preprint arXiv:2001.09576*, 2020.

Max Simchowitz, Horia Mania, Stephen Tu, Michael I Jordan, and Benjamin Recht. Learning without mixing: Towards a sharp analysis of linear system identification. In *Conference On Learning Theory*, pages 439–473, 2018.

Alexandre B Tsybakov. *Introduction to nonparametric estimation*. Springer Science & Business Media, 2008.

Stephen Tu and Benjamin Recht. The gap between model-based and model-free methods on the linear quadratic regulator: An asymptotic viewpoint. In *Conference on Learning Theory*, pages 3036–3083, 2019.

Peter Van Overschee and Bart De Moor. N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75 – 93, 1994. Special issue on statistical signal processing and control.

Harish K Venkataraman and Peter J Seiler. Recovering robustness in model-free reinforcement learning. In *2019 American Control Conference (ACC)*, pages 4210–4216. IEEE, 2019.

Gellért Weisz, Paweł Budzianowski, Pei-Hao Su, and Milica Gašić. Sample efficient deep reinforcement learning for dialogue systems with large action spaces. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(11):2083–2097, 2018.

Kemin Zhou, John Comstock Doyle, Keith Glover, et al. *Robust and optimal control*, volume 40. Prentice Hall, 1996.

Pengfei Zhu, Xin Li, Pascal Poupart, and Guanghui Miao. On improving deep reinforcement learning for pomdps. *arXiv preprint arXiv:1704.07978*, 2017.

Appendix A. Proof of Theorem 2

The continuous-time analog of this result can be found in Chapter 6 of Doyle et al. (2013).

Proof Assume no unstable pole/zero cancellations between the plant $P(\zeta)$ and the controller $C(\zeta)$. Then note that for any unstable pole p of $P(\zeta)$,

$$S(p) = \frac{1}{1 + P(p)C(p)} = \frac{1}{\infty} = 0,\ T(p) = 1 - S(p) = 1$$

We say that a transfer function is *all-pass* if it is one on the unit disc. Any all pass function can be expressed, up to a sign, as a product of factors of the form

$$\frac{\zeta^{-1} - \bar{a}}{1 - \zeta^{-1}a},\ |a| < 1$$

A transfer function is said to be *minimum phase* if it has no zeros outside of the unit disc. We see that any transfer function G may be factored as $G = G_{mp}G_{ap}$ where G_{ap} is all-pass and G_{mp} is minimum phase. In particular, G_{ap} will receive the unstable zeros of G.

Using the fact that $T(p) = 1$,

$$T_{mp}(p) = T_{ap}^{-1}(p) = \frac{1 - p^{-1}q^{-1}}{p^{-1} - q^{-1}} \left(\prod_{k=1}^{m} \frac{p^{-1} - q_k^{-1}}{1 - p^{-1}q_k^{-1}} \right) = \frac{pq - 1}{q - p} \left(\prod_{k=1}^{m} \frac{p^{-1} - q_k^{-1}}{1 - p^{-1}q_k^{-1}} \right)$$

where q_k, $k = 1, \ldots, m$ are unstable zeros of $P(z)$ excluding q.

Then by the maximum modulus theorem

$$\|T\|_\infty = \sup_{\omega \in [-\pi, \pi]} |T(e^{jw})| = \sup_{|\zeta| > 1} |T(z)| = \sup_{|\zeta| > 1} |T_{mp}(p)| = \|T_{mp}(\zeta)\|_\infty \geq \left| \frac{pq - 1}{q - p} \right|$$

■

Appendix B. System Identification Methods

B.1. ARXHK

Given the collection of input output trajectories $\{z^{(i)}(0 : T_i), u^{(i)}(0 : T_i)\}_{i=1}^N$ and an autoregressive order p, and model order n , we will estimate a linear model $(\hat{A}, \hat{B}, \hat{C})$. As an intermediary step, we identify the observer $(\hat{A} - \hat{L}\hat{C}, \begin{bmatrix} \hat{B} & \hat{L} \end{bmatrix}, \hat{C})$ under the assumption that $\hat{A} - \hat{L}\hat{C}$ is stable. To simplify notation, define $\tilde{A} := \hat{A} - \hat{L}\hat{C}$. In our experiements, ARXHK was used with $p = 10$, $n = 4$.

First fit an autoregressive model $\hat{G}$ as

$$\hat{G} = \min_{G} \sum_{i=1}^{N} \sum_{t=p}^{T_i} \|z^{(i)}(t) - \begin{bmatrix} z^{(i)}(t-1) & u^{(i)}(t-1) & \dots & z^{(i)}(t-p) & u^{(i)}(t-p) \end{bmatrix} G\|_2^2,$$

Next define the Toeplitz matrix

$$\mathcal{H} = \begin{bmatrix} \hat{G}(2p-1) & \hat{G}(2p) & \hat{G}(2p-3) & \hat{G}(2p-2) & \dots & \hat{G}(1) & \hat{G}(2) \\ 0 & 0 & \hat{G}(2p-1) & \hat{G}(2p) & \dots & \hat{G}(3) & \hat{G}(4) \\ \vdots & & & \ddots & & & \\ 0 & & \dots & & 0 & \hat{G}(2p-1) & \hat{G}(2p) \end{bmatrix}$$

Identify the elements of $\hat{G}$ with the Markov parameters of our desired observer and use the fact that $\tilde{A}$ is assumed to be stable so that $\hat{C}\tilde{A}^p\hat{B} \approx 0$ for p sufficiently large. Then

$$\begin{aligned} \mathcal{H} =: & \begin{bmatrix} \hat{C}\tilde{A}^{p-1}\hat{B} & \hat{C}\tilde{A}^{p-1}\hat{L} & \hat{C}\tilde{A}^{p-2}\hat{B} & \hat{C}\tilde{A}^{p-2}\hat{L} & \dots & \hat{C}\hat{B} & \hat{C}\hat{L} \\ 0 & 0 & \hat{C}\tilde{A}^{p-1}\hat{B} & \hat{C}\tilde{A}^{p-1}\hat{L} & \dots & \hat{C}\tilde{A}\hat{B} & \hat{C}\tilde{A}\hat{L} \\ \vdots & & \ddots & & & & \\ 0 & & \dots & & 0 & \hat{C}\tilde{A}^{p-1}\hat{B} & \hat{C}\tilde{A}^{p-1}\hat{L} \end{bmatrix} \\ \approx & \begin{bmatrix} \hat{C}\tilde{A}^{p-1}\hat{B} & \hat{C}\tilde{A}^{p-1}\hat{L} & \hat{C}\tilde{A}^{p-2}\hat{B} & \hat{C}\tilde{A}^{p-2}\hat{L} & \dots & \dots & \hat{C}\hat{B} & \hat{C}\hat{L} \\ \hat{C}\tilde{A}^{p}\hat{B} & \hat{C}\tilde{A}^{p}\hat{L} & \hat{C}\tilde{A}^{p-1}\hat{B} & \hat{C}\tilde{A}^{p-1}\hat{L} & \dots & \dots & \hat{C}\tilde{A}\hat{B} & \hat{C}\tilde{A}\hat{L} \\ \vdots & & \ddots & & & & & \\ \hat{C}\tilde{A}^{2p-2}\hat{B} & \hat{C}\tilde{A}^{2p-2}\hat{L} & \dots & & \hat{C}\tilde{A}^{p}\hat{B} & \hat{C}\tilde{A}^{p}\hat{L} & \hat{C}\tilde{A}^{p-1}\hat{B} & \hat{C}\tilde{A}^{p-1}\hat{L} \end{bmatrix} \end{aligned}$$

If the observability matrix and reversed controllability matrix of our desired observer are defined as

$$\hat{\mathcal{O}} := \begin{bmatrix} \hat{C} \\ \hat{C}\tilde{A} \\ \vdots \\ \hat{C}\tilde{A}^{p-1} \end{bmatrix}, \hat{\mathcal{C}} = \begin{bmatrix} \tilde{A}^{p-1}\hat{B} & \tilde{A}^{p-1}\hat{L} & \tilde{A}^{p-2}\hat{B} & \tilde{A}^{p-2}\hat{L} & \dots & \hat{B} & \hat{L} \end{bmatrix}$$

respectively, then by the assignment above, $\hat{\mathcal{O}}\hat{\mathcal{C}} \approx \mathcal{H}$. Then to recover a realization up to some similarity transformation, we may first compute the singular value decomposition of $\mathcal{H}$ and set

$$\mathcal{H} = (U\Sigma^{1/2})(:, 1:n)(\Sigma^{1/2}V^\top)(1:n, :) =: \tilde{\mathcal{O}}\tilde{\mathcal{C}}$$

where MATLAB indexing notation is used in the above expression. We may immediately recover $\hat{B}$, $\hat{C}$ and $\hat{L}$ as $\hat{C} := \tilde{\mathcal{O}}(1:1, :)$, $\hat{B} := \tilde{\mathcal{C}}(:, 2p-1:2p-1)$, $\hat{L} := \tilde{\mathcal{C}}(:, 2p:2p)$, and solve for $\tilde{A}$ as $\tilde{A} = \min_W \tilde{\mathcal{O}}(:-2)W = \tilde{\mathcal{O}}(2:)$. From $\tilde{A}$ we determine $\hat{A}$ as $\hat{A} = \tilde{A} + \hat{L}\hat{C}$.

B.2. N4SID

The tests using N4SID leveraged the n4sid function from MATLAB. It was used with N4Weight set to CVA and N4Horizon set to [1 10 10]. Further details about the algorithm can be found in (Van Overschee and De Moor, 1994).

Appendix C. Control Synthesis

To account for the uncertainties of the identified model $(\hat{A}, \hat{B}, \hat{C})$ in controller synthesis. we consider a disturbance w_x upon the state and w_y upon the output of the system (stacked as $w = \begin{bmatrix} w_x^\top & w_y^\top \end{bmatrix}^\top$) and define an auxiliary signal z which contains the values we desire to be becomes small, in particular the state variables and the actuation effort. The penalty upon the actuation effort is scaled by ε, a parameter which can be tuned to achieve a controller with good performance. Accounting for these disturbances in our model, and the auxiliary output signal in our model results in the following system

$$\begin{aligned} x_{k+1} &= \hat{A}x_k + \hat{B}u_k + w_{xk} \\ y_k &= \hat{C}x_k + w_{yk} \\ z_k &= \begin{bmatrix} I \\ 0 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ \epsilon \end{bmatrix} u_k \end{aligned}$$

To further enhance the controller's robustness to model uncertainties, we place the resulting system in feedback with a unstructured linear time invariant uncertain system. This may be performed in MATLAB by calling the feedback function along with ultisys. The controller is then synthesized using the MATLAB function musyn with nmeas and ncont both set to one.

Appendix D. Controller Synthesis Experimental Procedure

A version of Fig 6 which includes the success rate and the average reward is available as Fig 7. To obtain the control results in Tab 2 and Figs 6, 7 we collected 3000 runs with the initial states and actuation signals described in Subsection 3.2. For each of the three perception variations we first determined a value of ε to use in the control synthesis step. To do so, we fit a model using ARKHK with 20000 data points from a the trials with a fixation point of 1.0 for each perception type, and varied ε until the control performance was acceptable. These values were fixed for perception map for the remaining experiments. The values of ε use $5 \times 10^{-3}, 1 \times 10^{-6}$ and 1×10^{-6} for the no noise setting, depth perception map, and the rgb perception map respectively.

Now, for each perception map and each fixation point, we fit a linear model using ARXHK autoregressive horizon using $p = 10$, and state dimension $n = 4$ using $100, 1000, 5000, 10000, 15000$ and 20000 data points to fit the model. We then synthesize a controller according to Appendix C. The resulting controller was then tested by running 100 trials of the PyBullet simulator with max length 500. The number of times the controller stabilized the system for 500 steps was recorded, and divided by 100 to determine the success rate. The average reward was also recorded as the average number of steps for which the controller stabilized each trial. An estimate for the maximum initial angular displacement from which the controller was capable of stabilizing the system was also determined by setting all initial states to zero and bisecting on the initial angle. These experiments were repeated seven times. The medians are plotted and the quartiles are shaded. Only the medians for each quantity are reported in Tab 2.

Appendix E. Soft Actor-Critic Training Details

Hyperparamters. For training the SAC agent with depth image observations, we use temperature parameter $\alpha = 0.2$, target smoothing coefficient $\tau = 0.005$, reward discount factor $\gamma = 0.99$ and

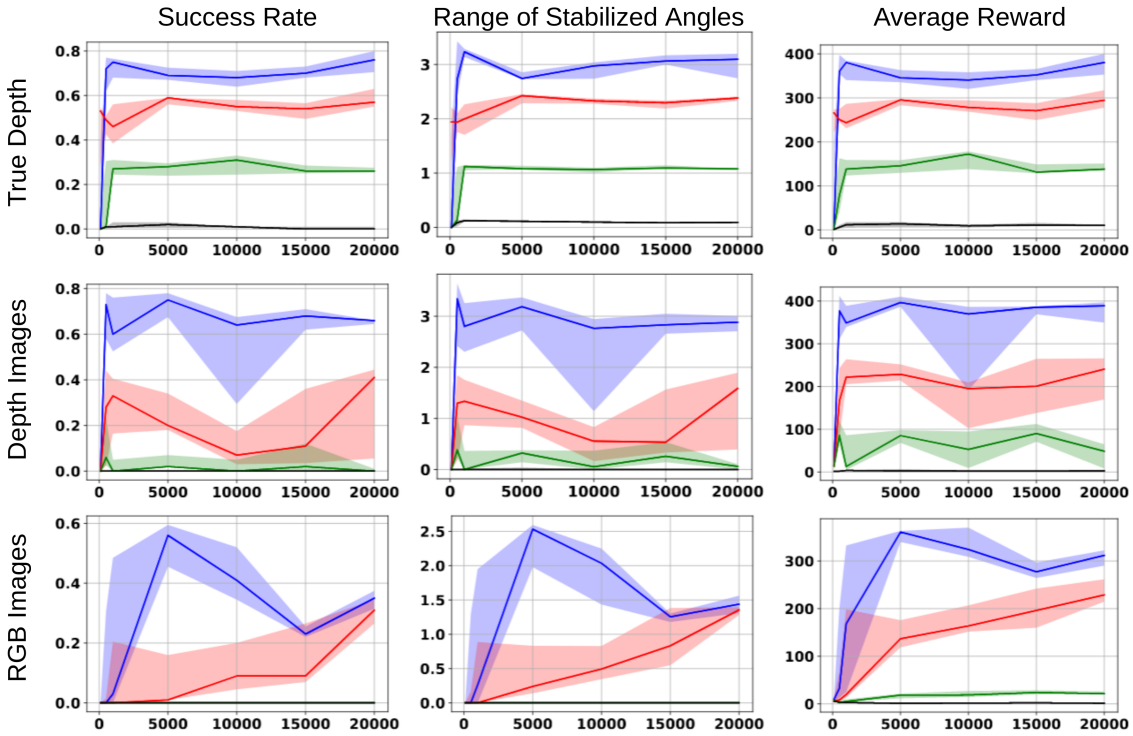

Fig. 7: The blue, red, green and black curves are for fixations of $1.0, 0.9, 0.8$ and 0.7 respectively. The shaded regions contain the first and fourth quartiles over seven trials.

learning rate $lr = 0.0003$. We decrease the temperature parameter α to 0.01 for RGB observations to decrease the importance of entropy term against reward. We use Adam (Kingma and Ba, 2014) for stochastic gradient descent.

Network architectures. The input to both the Q-network and policy network is the sequence of estimates of the fixation depth value z from the past 200 steps. Both networks are deep neural networks with 2 fully-connected layers, each with 256 hidden units and followed by the ReLU activation. For the Q-network, a third fully-connected layer outputs the Q value. For the policy network, a third fully connected layer outputs the action mean and standard deviation. The final action is sampled from the Gaussian distribution defined by this action mean and standard deviation.

Appendix F. Perception Model Architecture

The perception models are deep convolutional neural networks with the following architecture: $32 \times 3 \times 3$ conv (stride 2) $\rightarrow$ ReLU $\rightarrow$ $64 \times 3 \times 3$ conv (stride 2) $\rightarrow$ ReLU $\rightarrow$ $128 \times 3 \times 3$ conv (stride 2) $\rightarrow$ ReLU $\rightarrow$ $256 \times 3 \times 3$ conv (stride 2) $\rightarrow$ ReLU $\rightarrow$ flatten $\rightarrow$ 1024×64 fully-connected $\rightarrow$ ReLU $\rightarrow$ 64×1 fully-connected. For depth observations, the first layer takes 1-channel input, and for RGB, it takes 3-channel inputs.