CS 175:
Project in Artificial Intelligence
Winter 2020

Lecture 1: Introduction

Roy Fox
Department of Computer Science
Bren School of Information and Computer Sciences
University of California, Irvine

Today's lecture

- Course overview and general information

- Project scope and resources

- Brief introduction to reinforcement learning

Course description for CS 175

- This is a project course in AI/ML
- Students will work in teams of 3 to develop and deploy reinforcement learning agents
- The agent will interact with one of 3 platforms:
  - The Malmo platform built on top of Minecraft
  - The Duckietown simulator and real robots
  - The ColosseumRL game arena
- Students can either participate in one of these 3 competitions, or propose a different task to perform

General information

- Office hours: Fridays 9–11am, DBH 4064
- Teaching assistant: Julian Collado
- No discussion section
  - Friday mornings will be used for meetings with teams
- No midterm or final exam
  - But there will be a progress report and a final report

Communication

- Course announcements: piazza.com/uci/winter2020/cs175/home
  - A link to the course website will be posted there
  - Links to resources and reading will be posted there
- Use Piazza for all questions
  - Course staff will monitor and answer
  - Feel free to also answer questions
  - Use "private mode" to limit visibility to course staff
- Use email as last resort if other options don't work

Academic integrity

- Please read the guidelines on academic integrity below. Academic integrity is taken seriously in this class. Failure to adhere to the policies below can result in a student receiving a failing grade in the class.

- For assignments: you are allowed to discuss the assignments verbally with other class members, but you are not allowed to look at or to copy anyone else's written solutions or code. All problem solutions and code submitted must be material you have personally written during this quarter, except for any standard library or utility functions.

- For class projects: all reports submitted must be written by you or members of your project team. Code generated for class projects can be a combination of code written by team members and publicly-available code. You should clearly indicate in your reports and in your code documentation which parts of your code was written by you or your team and which parts of your code was written by others.

- It is the responsibility of each student to be familiar with UCI's Academic Integrity Policies and UCI's definitions and examples of academic misconduct.

Course overview

- Early weeks:
  - Lectures, 2 assignments, and reading
  - Introducing general principles of RL and the platforms
  - In a nutshell: deep RL and advanced topics
- Later weeks: team project in RL
  - Proposal
  - Reading > implementation > experimentation > evaluation
  - Presentation + reports

Course schedule (subject to minor updates)

Week	Tuesday	Thursday
(1) Jan 6	Lecture: Introduction	Lecture: Platforms
(2) Jan 13	Lecture: Reinforcement Learning **Assignment 1 due**	Lecture: Deep RL
(3) Jan 20	Lecture: experimental RL **Assignment 2 due**	Office hours (no lecture)
(4) Jan 27	Office hours (no lecture) **Project proposal due**	Office hours (no lecture)
(5) Feb 3	Office hours (no lecture)	Office hours (no lecture)
(6) Feb 10	Office hours (no lecture)	Short lecture: progress reports
(7) Feb 17	Office hours (no lecture)	**Progress report due**
(8) Feb 24	Office hours (no lecture)	Office hours (no lecture)
(9) Mar 2	[Super Tuesday]	Short lecture: final report
(10) Mar 9	Project presentations (in class)	Project presentations (in class)
	Mar 15: **Final project report due**	

Project scope

- 3-person teams
  - Project grading will be partly team-based and partly based on individual contributions
  - Note that Assignments 1 and 2 are ***not*** team-based, these will be worked on and submitted individually
- Each team will propose its own project
  - Select one of 3 platforms: Malmo, Duckietown, or ColosseumRL
  - Choose whether to compete on that platform's standard tasks, or suggest your own
  - Your code should extensively use external libraries, and focus on a novel part
- Projects will be graded based on:
  - (1) initial proposal; (2) progress and final reports; (3) in-class presentation
  - We will discuss all of this in more detail in future lectures

Software tools

- Python 3 will be the primary language we will use in this class
  - All students are assumed to have a good working knowledge of Python 3
- Platforms:
  - Malmo: https://www.microsoft.com/en-us/research/project/project-malmo/
  - Duckietown: https://challenges.duckietown.org/
  - ColosseumRL: https://rl-competition.igb.uci.edu/

Software tools — RL libraries

- RLlib: https://ray.readthedocs.io/en/latest/rllib.html
  - Pros: scalable, designed for research
  - Cons: mostly tied to OpenAI Gym
- Baselines: https://github.com/openai/baselines
  - Pros: standalone
  - Cons: limited extendability
- Spinning up: https://spinningup.openai.com/
  - Pros: educational, readable
  - Cons: not optimized for performance

Assignment 1

- Will be published on the course webpage before the next class
- **Due next Monday, Jan 13, 11pm**
- Outline:
  - Complete the Malmo tutorial
  - Complete the AI-DO "Getting Started"
  - Complete the ColosseumRL "Getting Started"

Compute resources

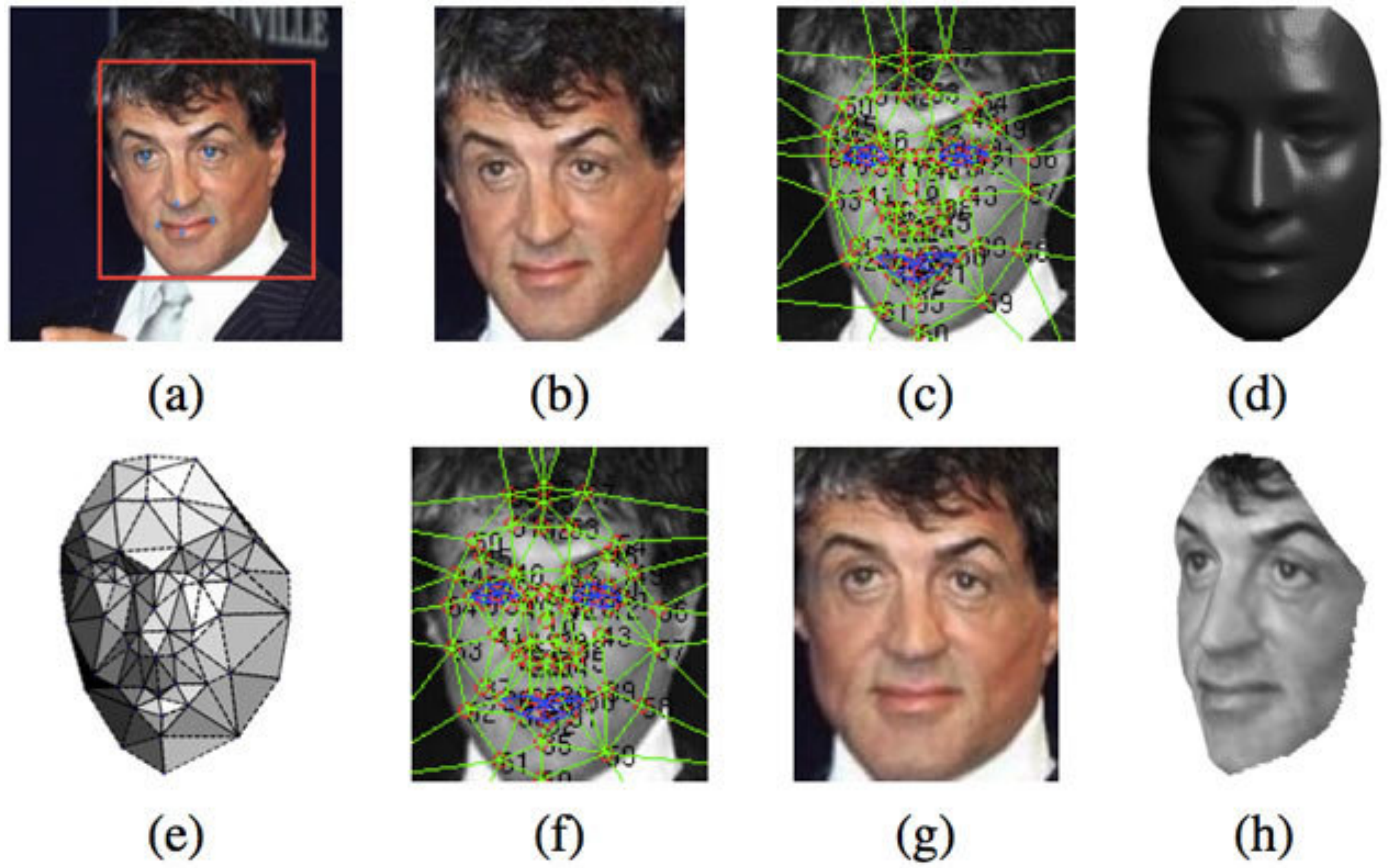
- Much of your development work can be handled by your laptop or desktop
  - Always test your code on a smaller challenge that "should" work
- When more compute resources are required:
  - Campus-wide cluster: https://hpc.oit.uci.edu/
  - Google Colab: https://colab.research.google.com/
  - We may be able to help with AWS / Google Cloud credits

What is Machine Learning

- Artificial Intelligence:
  - Can we build a machine with a property we would call "intelligence"?
- Machine Learning:
  - Can we build AI without explicitly figuring out all the details of its working?
    - Solution = problem-agnostic algorithm + problem-specific data
  - Learning = Statistics + Algorithms
  - ML = Learning + Implementation + Data

ML examples

Face recognition



Speech synthesis

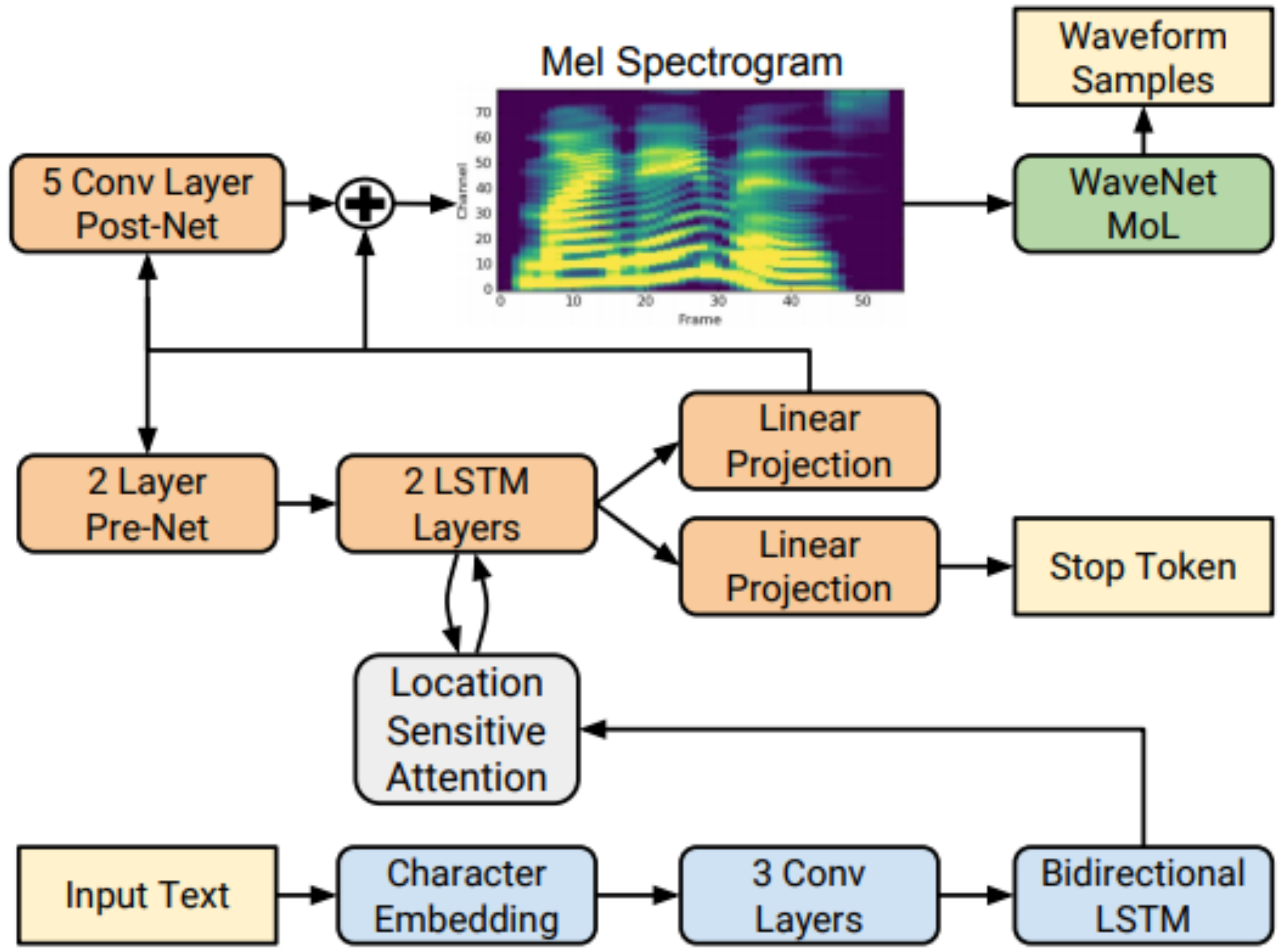
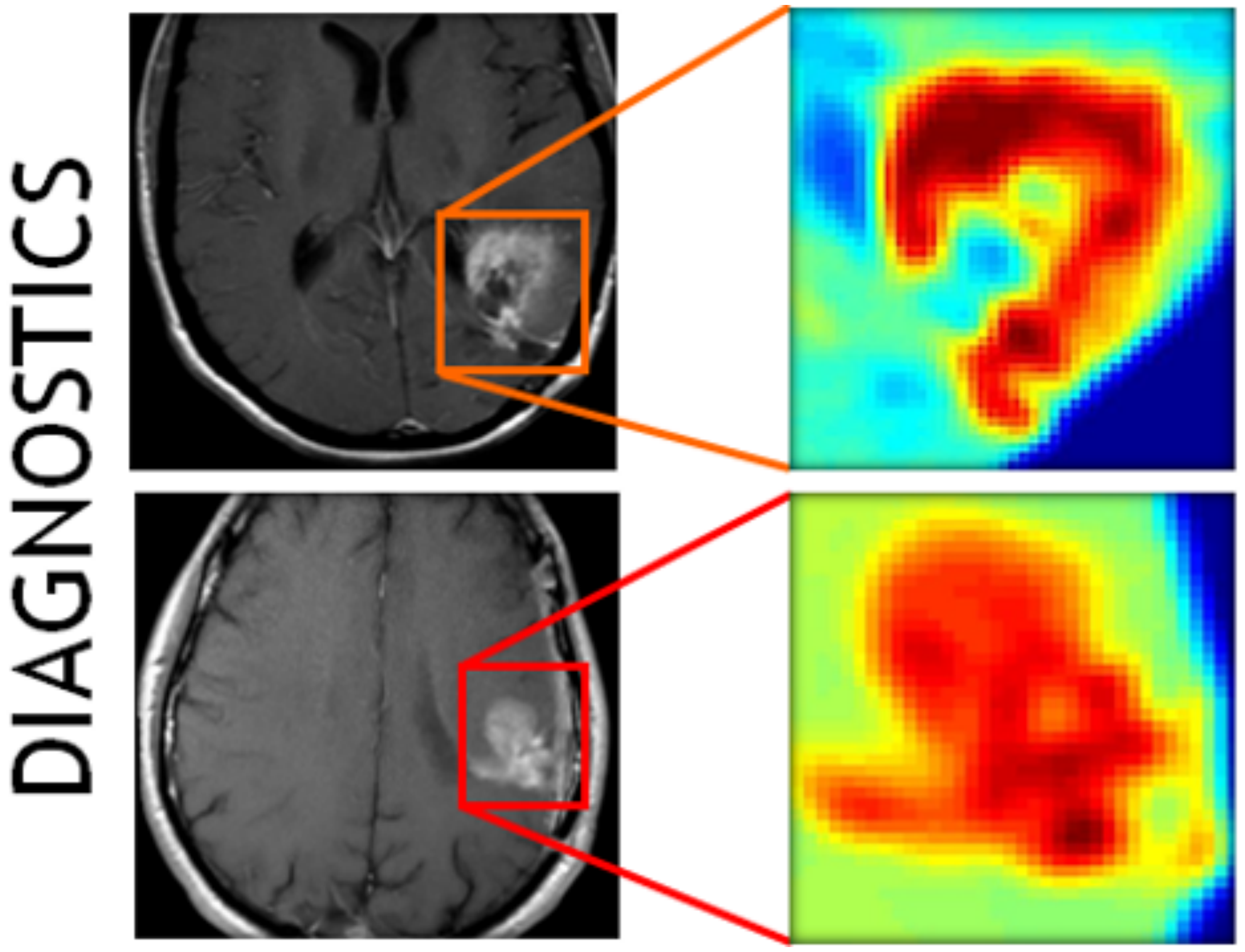


Fig. 1. Block diagram of the Tacotron 2 system architecture.

Medical diagnosis



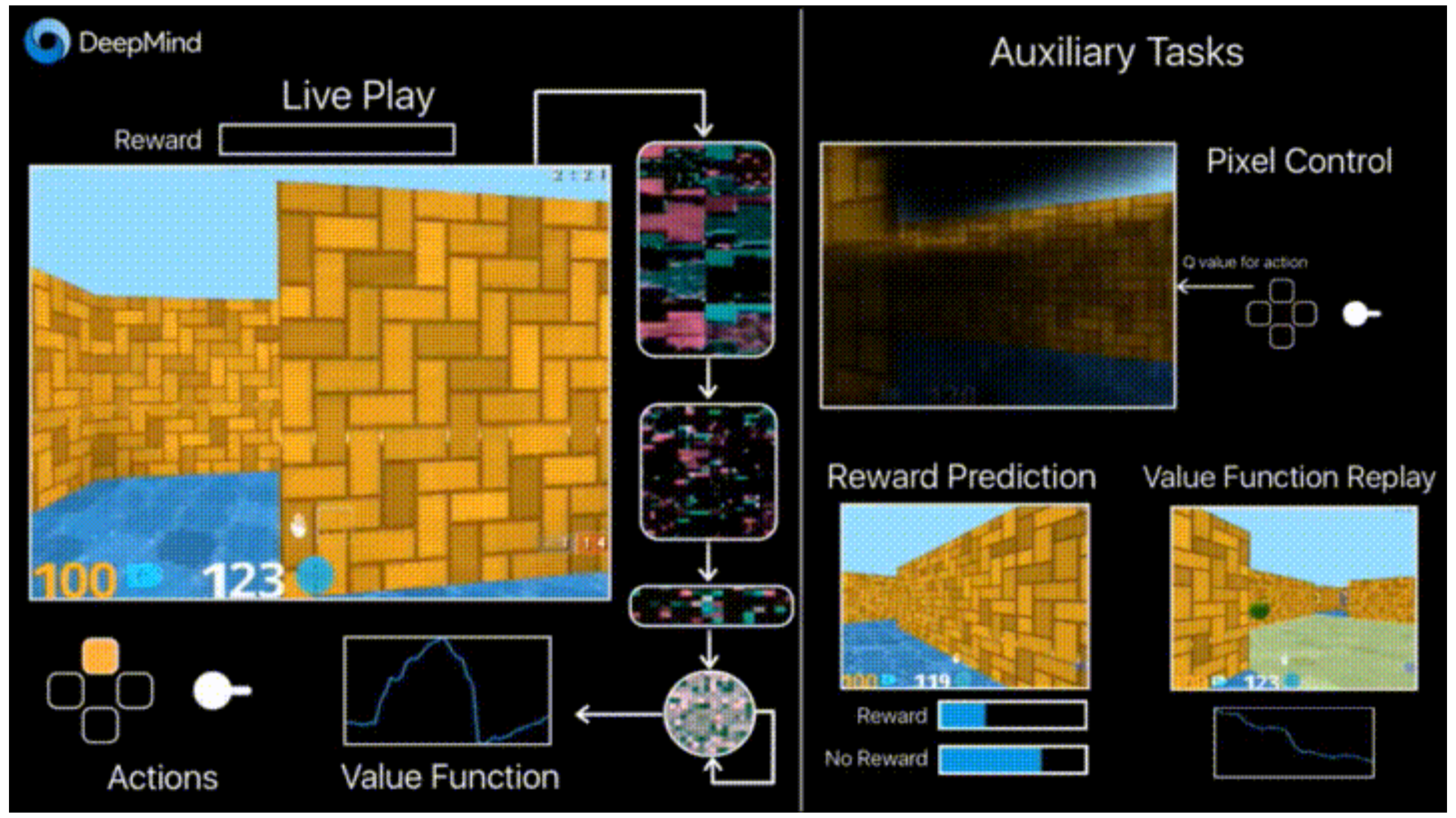
What is Reinforcement Learning

- Intelligence appears in interaction with a complex system, not in isolation
- An **agent** interact with an **environment**
- Performs **sequential** decision making:
  - Sense environment state ***s***
  - Take action ***a***
  - Repeat
- Success measured by the accumulation of reward ***r(s, a)***
  - As opposed to the "correct" action (that would be Imitation Learning)

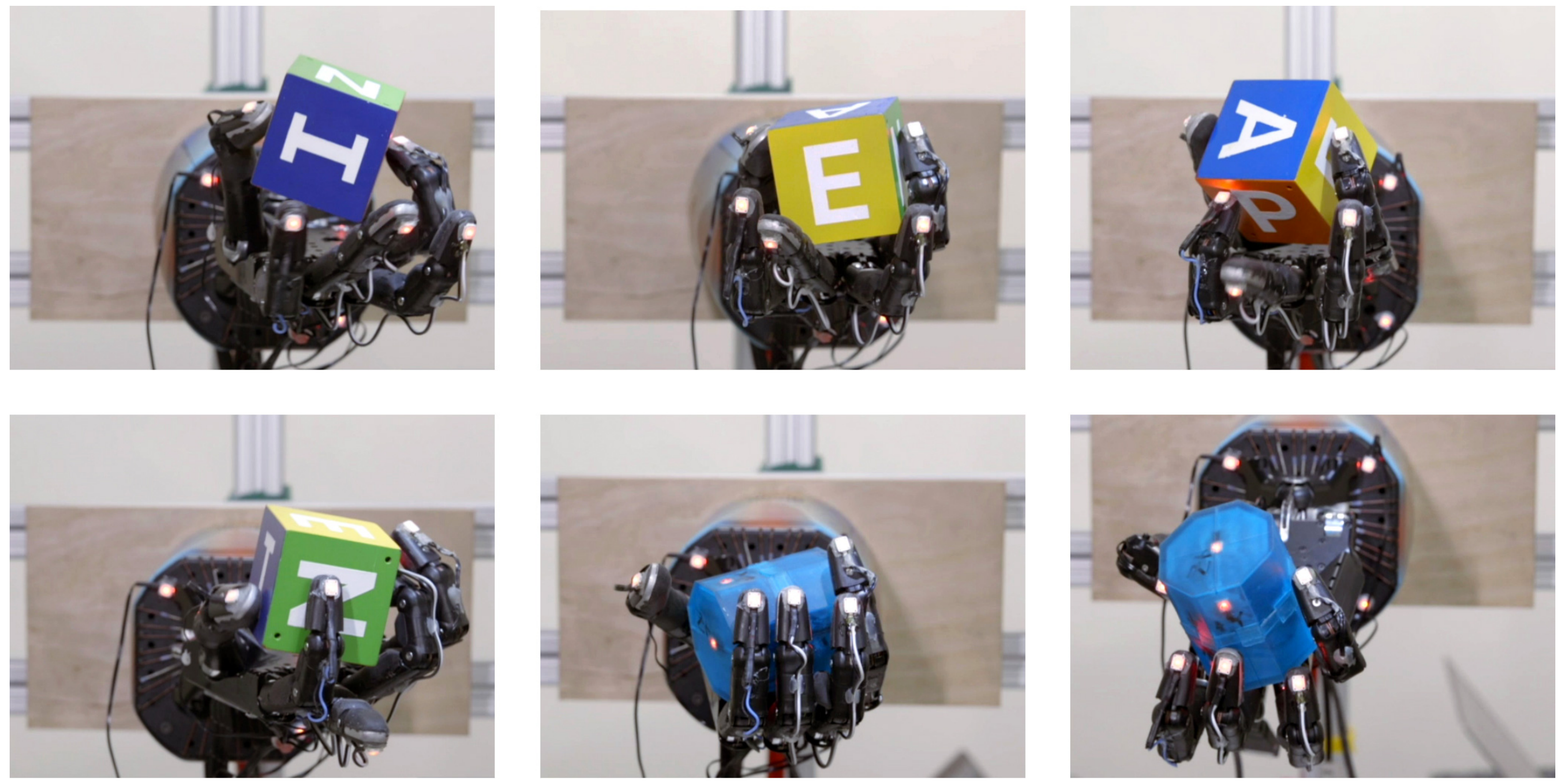
RL examples

Gameplay

Spacial navigation



Dextrous manipulation

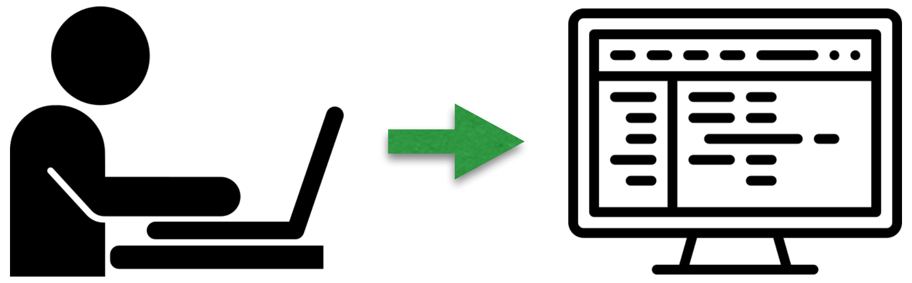
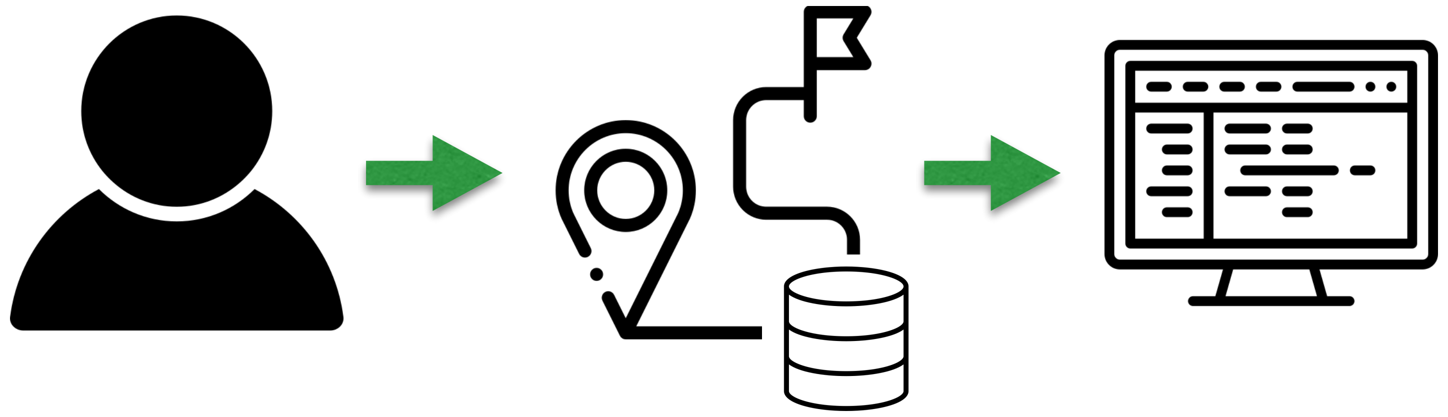
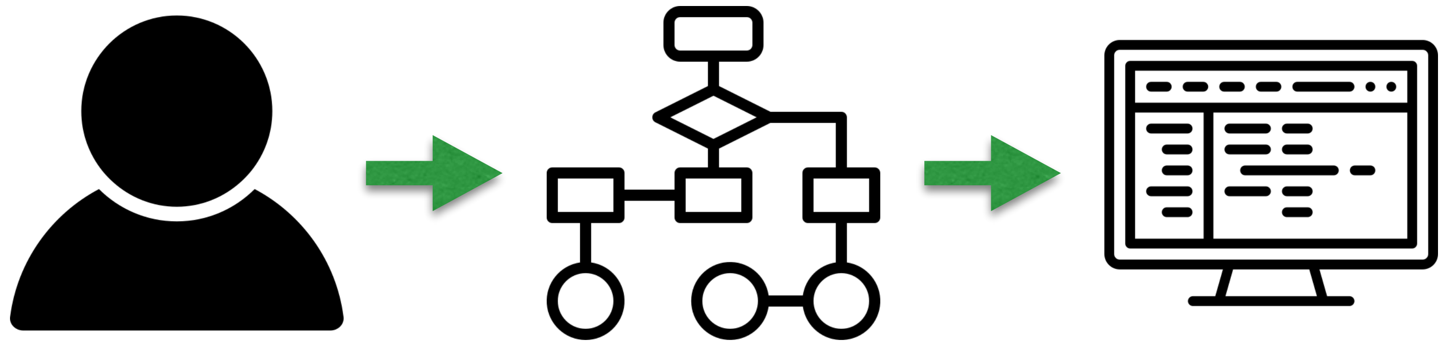
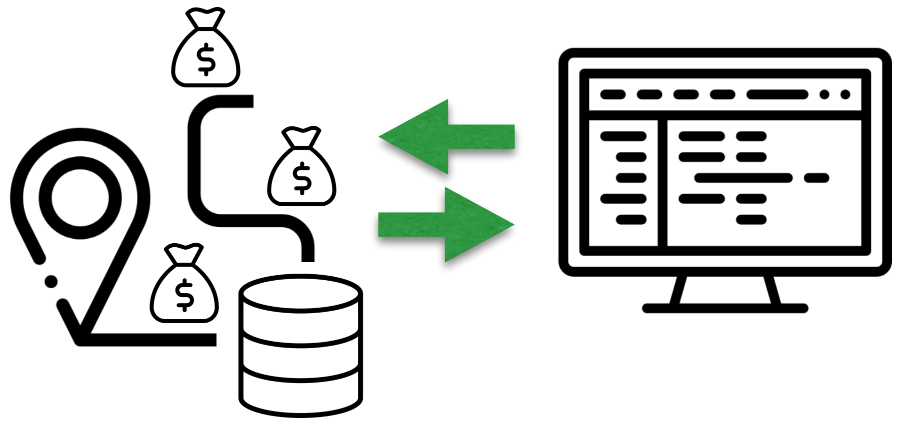


Basic RL concepts

- Dynamics $p(s_{t+1}|s_t, a_t)$
- Policy $\pi(a_t|s_t)$
- Trajectory $p(s_0, a_0, s_1, a_1, \ldots) = p(s_0) \prod_t \pi(s_t|a_t) p(s_{t+1}|s_t, a_t)$
- Return $R = \sum_t \gamma^t r(s_t, a_t) \qquad 0 \leqslant \gamma < 1$
- Value $V(s) = \mathbb{E}[R|s_0 = s]$

  $Q(s, a) = \mathbb{E}[R|s_0 = s, a_0 = a]$

Learning policies

	Explicit	Implicit
"how"	Programming 	Imitation Learning 
"what"	Specification 	Reinforcement Learning 

RL is ML... but special

- Test distribution of trajectories depends on the policy!
  - Cannot avoid train–test mismatch
  - To reduce it, learner interacts with the environment to collect data = exploration
  - Balanced exploration is challenging
- Policy space is strewn with local optima
  - Actions in a sequence need to be coordinated
- A good policy may require memory
  - Learning to remember is hard!

RL — the frontier

- How to perform better exploration?
- How to model / structure the agent's policy? in particular, its memory
  - Hierarchical RL
- How to jointly learn multiple tasks?
- How to learn from more kinds of data?
  - RL + imitation learning / NLP / vision / program synthesis
- How to interface with a human teacher?

What makes a good project

- Science: what have we learned?
  - Compare multiple methods
  - Demonstrate a failure mode of a method
  - Explain why the results are what they are
- Technology: how is this useful?
  - Propose a new method or component
  - Contribute an elegant design or implementation
- Art: what is the aesthetic value?
  - Make something cool!
  - Make something inspiring

Recap

Week	Tuesday	Thursday
(1) Jan 6	Lecture: Introduction	Lecture: Platforms
(2) Jan 13	Lecture: Reinforcement Learning **Assignment 1 due**	Lecture: Deep RL
(3) Jan 20	Lecture: experimental RL **Assignment 2 due**	Office hours (no lecture)
(4) Jan 27	Office hours (no lecture) **Project proposal due**	Office hours (no lecture)
(5) Feb 3	Office hours (no lecture)	Office hours (no lecture)
(6) Feb 10	Office hours (no lecture)	Short lecture: progress reports
(7) Feb 17	Office hours (no lecture)	**Progress report due**
(8) Feb 24	Office hours (no lecture)	Office hours (no lecture)
(9) Mar 2	[Super Tuesday]	Short lecture: final report
(10) Mar 9	Project presentations (in class)	Project presentations (in class)
	Mar 15: **Final project report due**	

Assignment 1

- Will be published on the course webpage before the next class
- **Due next Monday, Jan 13, 11pm**
- Outline:
  - Complete the Malmo tutorial
  - Complete the AI-DO "Getting Started"
  - Complete the ColosseumRL "Getting Started"