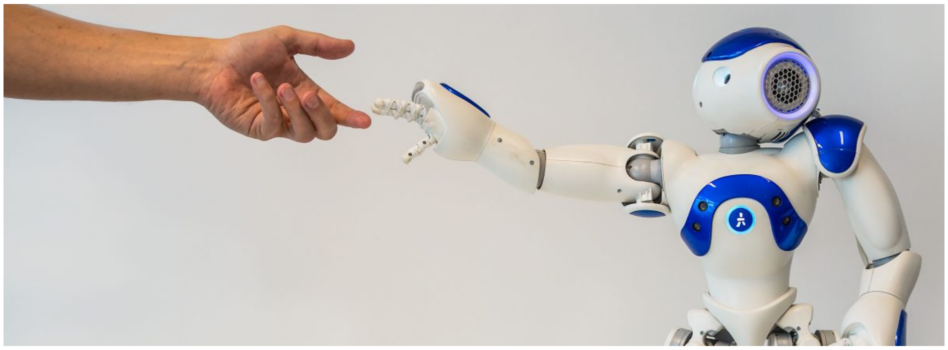
Final Project - Social Robotics



Context

Building up on your experience with the two lab sessions on imitation learning and evaluative feedback, you will have to propose an advanced framework of interactive reinforcement learning, scaling to more complex environments and better ways to integrate human signals. Starting with simple Behavioral Cloning and TAMER algorithm, you will have to focus on the following areas of improvement:

1. Scaling to high-dimensional and/or continuous state and action space in possibly complex and noisy environments,
2. Proposing novel and user-friendly ways to provide and include demonstrations and feedback.

While every aspect is important, your model needs to be interactive! Therefore, bear in mind that you need to design your system around the user, providing intuitive cues and interfaces for interactively training your agent.

Examples

We provide here examples of models for dealing with more complex human teaching signals. A list of alternative environments suited for interactive reinforcement learning is also given. You can also suggest models or environments based on your interests or challenges you want to address. *Beware that some environments and models might require the use of a GPU for training.*

Imitation Learning

For models, you can explore the following models with different integration of human demonstration: Generative Adversarial Imitation Learning (GAIL), Adversarial Inverse Reinforcement Learning (AIRL), DAgger, Density-Based Reward Modeling, Maximum Causal Entropy Inverse Reinforcement Learning (MCE IRL), Preference Comparisons, Soft Q Imitation Learning (SQIL). If you have substantial computational resources, consider exploring the Learning by Cheating method (CoRL 2019), developed for the 2020 CARLA Challenge.

Learning from Human Feedback

For those interested in learning from human feedback, explore and compare various TAMER-based approaches, building on the version tested in your practical session. Consider incorporating advanced feedback methods, such as counterfactual explanations and deep reinforcement learning. Key concepts include: Interactive Shaping: Learning from human-generated rewards, TAMER: Foundational (myopic) system for

Header image source: https://www.bartneck.de/2021/08/09/free-online-program-on-human-robot-interaction/

interactive shaping, Non-myopic Learning: Maximizing human-generated rewards over the long term, aligning the agent's performance with the trainer's intended task, TAMER+RL: Combining human-generated rewards with predefined rewards from a Markov Decision Process (MDP).

Environment

Here is a list of gymnasium environments you could use. These are suggestions, and you are free to choose others from GitHub or develop your own using Gymnasium wrappers and Pygame.

`Classical Gymnasium Environment`: `https://gymnasium.farama.org/`
→ From classic control to Mujoco ...

`Flappy Bird`: `https://github.com/robertoschiavone/flappy-bird-env`
→ Perfect for Imitation Learning-based projects

`Connect Four`: `https://github.com/lucasBertola/Connect-4-Gym-env-Reinforcement-learning`
→ Both suited for Imitation Learning and Learning from Human Feedback. The library provides a set of AI-opponent with various level!

`MIMO`: `https://github.com/trieschlab/MIMo`, `https://babybench.github.io/2025/`
→ This gymnasium environment allows researchers to investigate emerging behaviors during infants' development. You can work in various scenarios, *e.g.*, self-touching, reaching, and develop models with implicit rewards, *i.e.*, no reward/goal from the environment. This environment is particularly suited for feedback-based frameworks.

Guidelines

Project will be done in groups of 4 people maximum. Once you and your group have decided which problem you want to tackle, send an email to louis.simon@sorbonne-universite.com with 1) Group name, 2) Names of members, 3) a brief outline of your project, mostly the environment, the type of model, and demonstration/feedback you plan to include.

Project Timeline

You will have two in-person sessions to work, ask questions, and get feedback about your work. *It is highly recommended to work on your project before these sessions!*

- **Group composition and project outline submission**: November 5th
- **In-person supervised sessions**: November 17th and 24th
- **Oral presentation**: December 1st
- **Project files submission deadline**: November 28th

Required Project Files

Ensure the following files are included in your project folder:

- **Code**: A zip folder or Jupyter Notebook with the implementations.
- **Report**: A PDF document presenting your results and discussion.
- **Presentation Slides**: A PDF file containing slides for your oral presentation.

Code should be running on any machine, meaning that you should provide every element for reproducibility, *e.g.*, a `environment.yaml` or a `requirements.txt` file for conda/pyvenv. *We highly encourage you to use GitHub for versioning and sharing code with other members of your group. Please share your repository with Louis Simon and Silvia Tulli.*

Project Evaluation Criteria

Your project will be graded based on the following components (out of 20 points):

- **Background** (2/20) Theoretical understanding and intuition behind the topic.
- **Algorithm** (4/20) Parameters and algorithm choice, fidelity of the implementation (if you have implemented an existing algorithm and/or if you have followed specific standards for the implementation of an existing algorithm), proposed alternatives.
- **Algorithm Details** (3/20) Depth of understanding of the algorithm(s).
- **Results** (5/20) Depth of analysis and number of comparisons.
- **Discussion** (3/20) Understanding the limitations identified and insights from your work.
- **Code and Report Quality** (2/20) Clarity, conciseness.
- **Presentation** (1/20) Clarity of presentation and responses to the questions.

This structure serves as a guide for organizing your code, report, and presentation.

Ressources

`Interaction: https://imitation.readthedocs.io/`
→ Offers multiple implementations of imitation learning models.

`Minari: https://minari.farama.org/`
→ Hosts a collection of popular Offline Reinforcement Learning datasets.

`Stable Baseline: https://stable-baselines3.readthedocs.io/`
→ Gathers a set of reliable implementations of reinforcement learning algorithms in PyTorch

For any additional questions, please contact louis.simon@sorbonne-universite.com or silvia.tulli@sorbonne-universite.fr