



The Final Project

The Exam

- ▶ The exam will take place on 18.08.
- ▶ Block the full day since this will be a poster session with everyone (no individual slots!)
- ▶ Each group will make a poster about their project and present it to us (5-10 minutes)
 - ▶ Maximum team size: 3
 - ▶ Larger teams (e.g., three-student team vs single-student team) should also deliver at least 50% larger scopes per student
- ▶ We will then ask questions about the project and related topics (e.g. if you used PPO, expect to be asked about PPO ask well)
- ▶ Total duration per group: roughly 20 minutes
- ▶ You will have the opportunity to see other groups' posters

Project Goals

The project is intended to test your skills in empirical RL:

- ▶ Understanding at least one lecture topic deeply enough to build upon it
- ▶ Designing reproducible experiments
- ▶ Working with RL algorithms and environments in code
- ▶ Analysing results

Nice to have, but not things we want to test:

- ▶ Your ability to write a conference paper
- ▶ Your ability to solve an especially hard environment
- ▶ How faithfully you can implement basic algorithms
- ▶ If you can apply an existing algorithm to an unspecific new environment

What Makes a Good Project?

- ▶ You ask an interesting research question related to any topic from the lecture
- ▶ You design and run experiments to investigate and answer this research question
- ▶ Your experimental results are clearly and comprehensively reported, and the experiments are reproducible
- ▶ You conduct additional ablations to solidify the reliability of your insights
- ▶ You make effective use of all available resources—existing methods, open-source code, your own work from the semester, etc.— and appropriately cite all sources, including papers and external codebases.

Reliable quality insights are the main scoring criterion. Quality over quantity!

- ▶ You try to do too much (exploring multiple research questions, many algorithms, or pushing too many different ideas or extensions)
- ▶ You neglect or overlook reproducibility (even if it is in the name of solving a harder problem)
- ▶ Your experiments are too computationally expensive to thoroughly validate or perform hyperparameter optimization
- ▶ You try to do everything by yourself, or work in isolation (e.g. by not researching methods aiming for similar goals)
- ▶ You try to run your experiments last minute (you will be almost guaranteed to have a destructive bug!)

Example Topics

Good project topic from a previous year:

- ▶ Research question: Does a curriculum help DQN solve hard MiniGrid environments?
- ▶ Used manual curriculum to test this
- ▶ Sequence of three increasingly difficult MiniGrid environments with evaluation performance as metric

Bad project topics we've seen:

- ▶ Solving Doom, MineCraft or another expensive environment¹
- ▶ Methods that frequently (e.g. every step) call large foundation models
- ▶ "Can PPO solve X?" without any further questions or insights
- ▶ Implementing and reproducing all of Rainbow DQN (a large study on a complex code base)

¹In general, no shooter or other violent tasks are allowed.

Proposing a Project



- ▶ You will submit a project proposal before our last session on 14.07.
- ▶ We provide a template at: <https://www.overleaf.com/read/fbsydrfzxhqh#f12042>
- ▶ This proposal contains an outline of your idea, related concepts, experiment plan and timeline
- ▶ We discuss all projects on 14.07. and you're allowed to make modifications afterward
- ▶ You can always ask for feedback beforehand

Working on the Project

- ▶ The project should equal around two weeks of full time effort even though you can start whenever you want
- ▶ Be sure to record all results you produce (incl. model weights) in case you need them again!
- ▶ Maintain a clean repository. We will mark down code submissions we cannot read.
- ▶ Document what your contribution is and what you took from others. Failure to do so counts as plagiarism!
- ▶ You have time until 11.08. end of day (CEST!) to submit your project report (incl. link to code) and link to poster in StudIP
- ▶ We will print all posters and bring them to the exam

- ▶ Show us what is important to support your pitch
- ▶ Do not put too much text on it
- ▶ Carefully check all font sizes for readability
- ▶ At most 10-minutes pitch! We will stop you if you go over time.
- ▶ Poster template: <https://docs.google.com/presentation/d/1sY3tm1Ned2nc09UwmZhqf8Le10C7Nwczraa9tLdIEBI/edit?usp=sharing>

- ▶ The report template contains guidelines on what to write:
<https://www.overleaf.com/read/fbsydrfzxhqh#f12042>
- ▶ There is no minimum length for the report as long as all important information is included
- ▶ Try to be brief. If your report far exceeds 4 pages for reasons other than many figures, you are likely writing more than we need from you
- ▶ An optional reproducibility checklist is included as a reference for yourself and us

Score Distribution

- ▶ Idea, Topic & Results (based on poster and report): at most 25
(sufficiently challenging, interesting beyond lecture, gained insights, results, ...)
- ▶ Implementation (based on code): at most 25
(correctness, reproducibility, code documentation, code quality, ...)
- ▶ Presentation (based on report and poster): at most 25
(concise slides, clear message, structure, citations, plots, ...)
- ▶ Answering of questions (based on Q&A in poster session): at most 25
(correct, concise, knowledge, own ideas, ...)

Grade distribution:

- ▶ 100-90: 1.0; 89 - 85: 1.3; 84 - 80: 1.7
- ▶ 79 - 75: 2.0; 74 - 70: 2.3; 69 - 65: 2.7
- ▶ 64 - 60: 3.0; 59 - 55: 3.3; 54 - 50: 3.7
- ▶ 49 - 45: 4.0; < 45 : failed