L361: Part II Project Guidelines

2024-2025

For Part II, L361 comprises four practicals and one final project. The final practical slot is used to discuss the early results of the final project. The Part MPhil/Part III students receive separate guidelines. This document describes the project expectations and is supposed to guide you toward achieving style standards, experimental standards, and good engineering practices.

The final project will be expected to be a well-presented, full-scale technical engineering report on a meaningful piece of work worth a large portion of the overall grade. This hands-on project will be assessed based on the *report source code*, *related documentation*, and a brief 8-minute *pre-recorded talk* summarising key project elements (any slides used are also submitted as part of the project). For Part II students, this presentation is recommended but not examinable.

Unlike the practicals, the final project is more open-ended, depending on the exact nature of the implemented system/evaluated method. The contents and format of a technical report generally include the *motivations* of the system; the *goal* of the proposed work (e.g., improving the speed of the Flower simulator, the cohesion of an API or providing experimental validation of an existing sytems/algorithm); a *background* contextualizing previous production/research systems justifying the need for the proposed solution or the experiments proposed, the *design* applied to construct the solution or the *methods* that are explored, the *experimental setup* used to test the effectiveness of the design/methods with clear standards and criteria for success (very similar to the scientific work done by the MPhil/Part III students, please refer to their research project guidelines for this part if in doubt); the *results* and their *interpretation* (including sources of error in the evaluation); *limitations* of the engineering solution/experimental analysis; any *conclusions* drawn from building it.

As the project report is one of the primary forms of assessment for the module, it is important that you invest a significant amount of time in writing and refining the report, paying attention to detail and presentation.

All project reports **must** use the NeurIPS template.

1 Contents

For the purposes of L361, we require that project reports follow this structure concerning both section and overall report length. Marks will be deducted for failing to observe the final project's overall page limits, excluding references and appendices. The recommended lengths for each section provide a lower and upper bound; choosing the upper bound for every section will likely result in you going over the page limit. Thus, care should be chosen into how the focus is split depending on the nature of your work.

The final project has a page limit of **9 pages**, including figures. If collaborating, try to split the commits to the GitHub repo between your accounts or provide a statement regarding your contributions if doing a practice like pair programming.

1.1 Title; author name/names; date

1.2 Abstract

The abstract should contain a high-level summary of the context of the work, the need for the work, the proposed solution, and the solution's effectiveness based on the experimental results. It should be no longer than 140 words.

1.3 Introduction

Frames the report by providing (succinctly) context, motivations, approach, and results. (1-2 paragraphs)

1.4 Background

A presentation of work done around the topic of the final project with particular care towards identifying strengths and weaknesses of previous work. This may include references to scientific works, technical reports, and code implementations with relevant algorithms, formulas, and/or design diagrams presented. If your work has been done before, the background **must** identify why a new solution/implementation/experimental evaluation would be needed. If you are implementing novel work, the background **must** identify the gap it is trying to fill and why it is relevant. (1-1.5 pages)

1.5 Design/Methods

For works closer to software engineering, this section should cover the design of the system containing algorithms, API examples, or relevant diagrams describing the system (e.g., data flow, state machines, UML, etc...). Every component should be justified in conformity to standard engineering practices (e.g., performance, usability, extensibility, reliability, etc.). Most importantly, the Design section **must** have a clear set of goals that the design is trying to achieve, which **must** be testable in the experimental section.

For technical/experimental/lab reports (e.g., testing the numerical stability of an algorithm), this section should provide an overview of the method being proposed/experimented with, together with relevant formulas and potential pseudocode. (1-1.5 pages)

1.6 Experimental setup and methodology

An exploration of the goals/hypotheses, experimental setup, the procedure used in the experiment, and details of any steps taken to mitigate potential errors or problems. All of these should relate directly to the design goals of the project or the methods.

Clear success guidelines must be chosen for the tested system/algorithm, and choices relating to evaluating them must be justified. If your work is more scientific/experimental in nature, refer to the Research Project Guidelines document for Part III students.

Reproducibility should be explicitly considered via proper library versioning and seeding, although details for reproducing experiments should be delegated to a GitHub repository or the appendix. A separate paragraph should be included in the appendix to mention at a high level what steps have been taken to ensure reproducibility, similarly to the ICLR guidelines. (1-2 pages)

1.7 Results and discussion

The results obtained, graphs illustrating those results, and exploration and interpretation of the results, including important artifacts, the validity of the results, and conclusions they lead us to. Tables and figures required to explain the results should be present in all reports; in most cases, graphs will be expected. This is the body of the report.

Explicit mention of which results support/contradict which of the provided design goals/experimental hypotheses is expected. For both successes and failures, the quality of argumentation for the likely cause of the outcome will be used to determine marking. (2 pages-3 pages, including figures)

1.8 Conclusion

A summation of the results and thoughts on potential future directions. (1-2 paragraphs)

1.9 References

References to the most relevant implementations/code bases and/or scientific literature.

1.10 Appendices

Additional material that supplements prior sections - e.g., it might be desirable to reference content such as scripts to perform experiments in explaining material in the body, additional data tables, or more detailed illustrations of an experimental setup. As mentioned above, details on reproducibility and data distributions used should be here. (There is no length limit, but moderation is encouraged when including additional results)

1.11 Recap

Abstract (1 paragraph, 140 words), Intro (1-2 paragraphs), Background (1-1.5 pages), Design/Methods (1-1.5 pages), Experimental setup and methodology (1-2 pages), Results and discussion (2-3 pages), Conclusion (1-2 paragraphs). Choose appropriate section lengths to stay within the 9-page limit.

1.12 Caveats

It is important that both the abstract, introduction, and conclusion describe the hypotheses/goals, experimental approach, and results - which will feel redundant but is the nature of the format.

Appendices should be included only where they improve understanding of the body of the report. Only pages within the page limit of the final project will be assessed; if the appendices are too long in terms of text or, in exceptional circumstances, in terms of figures, they will not be read.

2 Style and presentation

Reports must be clearly written, spell-checked, and formatted to make them easy for the reader to follow, including concerning figure labels. Given length limitations, they will, of necessity, be high-level presentations of your experiments and cannot explore every detail — but instead, they should focus on important and interesting results and how they relate to effects observed in the labs or topics discussed in lectures/wider FL literature.

Particular attention should be paid to graphs and tables that will present the results: axes must be labeled, scales should be selected with care to avoid misunderstanding, and if, for example, there are clear artifacts of interest, then an additional graph may be appropriate to explore those in greater detail. All graphs must be described in the text's body and have a suitable (but brief) descriptive caption. It will be important to include error bars or other error information and explain when confidence intervals have been used. Thought should be put into stacking graphs vertically (for comparing results at the same x), horizontally (for comparing y-values), or if two graphs should be merged. Two y-axes may also be appropriate depending on context (e.g., inner-product and cosine similarity).

For the sake of experimental reproducibility, at least three experiments are required to present a median, min, and max result with other methods becoming available (e.g., interquartile range, mean and standard deviation, mean and confidence interval) as the number of experiments increases. If you are unable to run enough experiments to asses error, prefer changing the approach (e.g., move to only single-round analysis and simulate multiple client selection situations) rather than reporting single numbers without any context.

All graphs must be vector-based rather than raster images and be prepared to be clear even if printed in black and white (so use different line styles/fillings). It may be appropriate to use diagramming packages such as tikz, code rendering via the listings package, and additional tools such as matplotlib, **R**, and graphviz to analyse and present results.

Students are cautioned that many of these tools are complex and subtle and, when used incautiously, tend to consume all available time. If you run into difficulties, seek help from the course instructor or one of the teaching assistants - and when in doubt, avoid exciting-sounding features in LaTeX!

3 Assessment

Improper: extremely poor (or incomplete) design/experimental procedure or writeup that might include an incoherent description of the work, improper experimental design incapable of testing the design/experimental hypotheses/proposed method, poor experimental practice that leads to incorrect results, failure to discuss potential sources of error, lack of comprehension of the explored/proposed FL system/method, and/or poor data analysis that draws incorrect conclusions despite clear evidence to the contrary. This marking category will also be used if insufficient original thought is put into the argumentation.

Proper: adequately performed design/experimental procedure and writeup, but with a few (but not many) of the following problems: (1) the design will be coherent but insufficiently detailed or justified with clear oversights (2) the experimental approach will have been roughly right, but failed to avoid potential sources of error, used inadequate procedures to manage variance, failed to pursue an important behavior or effect; (3) the writeup will have drawn reasonable conclusions, but failed to make proper use of statistics to prove the reproducibility of effects or failed to investigate a surprising effect or result; or (4) graphs will present useful results but are unclear or disagree with the experimental analysis.

Good: most or all of the following hold: a superior writing style and clarity; well-justified and thought-out design without clear oversights; strong experimental procedure and error analysis, in which surprising results or artifacts are adequately illustrated via graphs and explained in the text; strong or even new insights into system design/the evaluated method are gained.

4 Collaboration

Students must write up and submit project reports individually or with one teammate. If part of a team of two, try to split the commits to the GitHub repo between your accounts or provide a statement regarding your contributions if doing a practice like pair programming. The excessive similarity of graphs, text, and analysis across groups may be penalized as plagiarism.

5 Credit

Thanks to Professor Robert N. M. Watson for his lab report guide used in the Advanced Operating Systems course.