Stanford University
Department of Chemical Engineering
CHEMENG 177/277, MATSCI 166/176
Data Science and Machine Learning Approaches in Chemical and Materials Engineering
Winter 2025

Final Project Instructions

Project Overview
You will develop an original machine learning application in chemical engineering, materials science, or a related field, reporting your work in the format of a machine learning conference workshop paper. The project can be completed in groups of 1-3 people.

Deliverables
1. Conference-style paper (max 5 pages excluding references and appendices)
2. GitHub repository with reproducible code and documentation
3. All data or clear instructions for data acquisition

The complete project is due on **Thursday, March 13, 2025**.

Paper Structure and Format

Length Requirements
- Main content: Maximum 5 pages
- References and appendices: Not included in page limit

Suggested Sections
A typical machine learning conference workshop paper will have the following sections. For concrete examples see "Accepted papers" in https://www.mlsb.io/ and "Papers" in https://ml4physicalsciences.github.io/2024/.

> **Title and Information.** Begin your paper with a descriptive, informative title that accurately represents your work. Include all authors' names and their institutional affiliations. Provide contact information for the corresponding author(s). This front matter helps establish the paper's identity and enables reader communication.

Abstract. Your abstract serves as a mini-version of your paper, condensed into 200-250 words. Open with a clear statement of the problem you're addressing. Follow this with a concise overview of your methodology, highlighting novel aspects. Present your key quantitative results, using specific numbers when possible. Close with your main conclusions and the broader implications of your work. A well-crafted abstract helps readers quickly determine the paper's relevance to their interests.

Introduction. The introduction, spanning roughly half to one page, sets the stage for your work. Start by establishing the broader context and motivation – why should readers care about this problem? Present your specific problem statement clearly and concisely. Include a brief overview of existing approaches, highlighting their limitations. Then articulate your novel contributions, explaining how your work advances the field. Be specific about what's new and important about your approach.

Related Work. Dedicate approximately half to one page to thoroughly reviewing relevant literature. Organize this section thematically rather than chronologically, grouping similar approaches together. Discuss existing methods' strengths and limitations objectively. Most importantly, identify specific gaps in current approaches that your work addresses. This section should naturally lead readers to understand why your approach is needed.

Methods. The methods section forms the technical core of your paper, typically running 1-1.5 pages. Begin with your data collection and preprocessing steps, providing enough detail for reproduction. Describe your model architecture and justify key design choices. Detail your training procedure, including hyperparameters and optimization decisions. Clearly specify your evaluation metrics and justify their selection. Include implementation details that would be necessary for replication, such as software frameworks used and computational resources required.

Results and Discussion. Use 1-1.5 pages to present and analyze your results. Start with your primary quantitative findings, using precise numbers and uncertainty measurements where appropriate. Include ablation studies to demonstrate the importance of different components of your approach. Compare your results against relevant baselines, explaining any significant differences. Analyze your model's behavior, including failure cases and limitations. Be honest about any shortcomings – this strengthens your paper's credibility.

Conclusion. Your half-page conclusion should do more than summarize. While briefly recapping your main contributions, focus on broader implications and future directions. Discuss potential applications of your work and its broader impact on the field. Suggest specific promising directions for future research, including both immediate next steps and longer-term possibilities.

References and Appendices. List all references using a consistent citation style throughout. The reference section doesn't count toward your page limit. Consider including an appendix for additional experimental details, extended results, or detailed hyperparameter settings that would disrupt the main paper's flow. Like references, appendices don't count toward the page limit.

Visual Elements. Include 1-3 carefully chosen figures or tables that enhance your paper's clarity. Each visual element must have a clear, descriptive caption that helps readers understand its significance without referring to the main text. Always reference and discuss figures/tables in your main text – never leave them "floating." Ensure all text in figures is legible at the final paper size, paying particular attention to axis labels and legends.

Remember that every element in your paper should serve a clear purpose in communicating your research. Be concise but complete, and maintain a professional, scientific tone throughout. Your goal is to make your work both accessible and reproducible by other researchers in the field.

Code Structure and Format

Your code repository should enable other researchers to easily reproduce your results. Create a simple, well-organized structure with three key components: a README file, a data directory, and your analysis code. This straightforward organization helps others understand and run your code without getting lost in complex directory structures.

Writing an Effective README. The README file serves as the entry point to your repository. Write it with the assumption that readers are encountering your project for the first time. Begin with a clear project overview that explains the problem you're solving and your approach. Then provide step-by-step instructions for data preparation – explain where to download the data, any preprocessing steps required, and where to place the files in your repository structure. Detail the Python packages required to run your code, ideally listing them in a separate requirements.txt file for easy installation.

Running Instructions. Most importantly, include clear instructions for running your analysis. If your code accepts any parameters or configuration options, document these thoroughly. Consider including example commands that demonstrate common usage scenarios.

Code Organization and Documentation. In main.py, organize your code into well-structured functions with clear purposes. Each function should have descriptive comments explaining what it does, what inputs it expects, and what outputs it produces. Group related functionality together logically – for example, keep data preprocessing functions separate from model training code. While your code should be well-documented, avoid excessive comments that merely restate the obvious. Instead, focus on explaining the "why" behind important decisions and any non-obvious implementations.

Ensuring Reproducibility. The goal is to make reproduction of your results as straightforward as possible. Another researcher should be able to clone your repository, follow your README instructions, and reproduce your paper's findings with minimal friction. Remember, reproducibility is a cornerstone of good scientific practice, and your code repository is as important as your paper in contributing to the scientific community.

Evaluation Criteria

Your project will be evaluated across four key dimensions, weighted to reflect their relative importance in scientific research and development.

Technical Merit (30%). The largest portion of your grade reflects the technical strength of your work. We will evaluate the soundness of your methodology – whether your approach is well-reasoned and appropriate for the problem at hand. The validity of your results is crucial – we'll examine whether your conclusions are well-supported by your data and analysis. Finally, we'll assess the depth of your analysis, looking for thoughtful investigation of your model's behavior, limitations, and implications.

Paper Quality (30%). Strong scientific work must be communicated effectively. Your paper should demonstrate clear, concise writing that conveys complex ideas accessibly with informative figures.

Code Quality (30%). The reproducibility of your work is essential to its scientific value. Your code should be well-documented, allowing others to understand and build upon your work. The organization should be logical and efficient, making it easy to navigate and modify. Most importantly, your code must reliably reproduce the results presented in your paper.

Innovation (10%). While a smaller component of the grade, innovation is meant to distinguish exceptional projects. We'll consider the novelty of your problem choice – whether you're addressing an important but underexplored area. Your methodological creativity matters – how you've approached the problem in new or interesting ways. Finally, we'll assess the potential impact of your work – its applications, implications, and potential to influence future research or practical applications.

Remember that these components work together: technical excellence supports clear writing, well-organized code enables result verification, and innovation is most valuable when properly executed and communicated. Focus on delivering a cohesive project that demonstrates strength across all these dimensions.

Project Timeline

A suggested timeline for the project looks like:

Week 1-2: Project proposal and data collection
Week 3-4: Model development and initial experiments
Week 5-6: Results analysis and paper writing
Week 7-8: Code cleanup and documentation

Some deadlines:

Thursday, January 30, 2025 (required): Submit a one-page project proposal to receive feedback from the course staff.
Tuesday, February 25, 2025 (optional): Meet with the course staff to discuss project progress, get feedback.
Thursday, March 13, 2025 (required): Submit the final project report and code.