6.8300/6.8301 Final Project Grading Guideline

General Instructions:

1. Every student needs to be responsible for hands-on coding / experimentation work. It is not acceptable to split the work into “writing” and “running experiments / coding” - every student on a project team is expected to interact with the codebase / model in a meaningful way.
2. For every additional student, your report needs to include:
    a. An additional experiment (which means: an ablation study, a benchmark, an additional baseline (that you actually ran yourself)).
    b. For that additional experiment, we then expect an additional output: either a quantitative comparison, for instance a table with numbers, or a qualitative comparison, for instance a figure that compares model outputs. Examples:
        i. For an ablation study, compare the model with and without the changes you are proposing.
        ii. For a baseline, compare quantitative performance with the baseline and compare model outputs.
3. Every report is required to have a final paragraph that explains who did what. **Specifically, each student needs to identify the experiment(s) that they ran.**

Report Grading Rubric:

- Abstract (10%)
    - The abstract should: 1) Briefly motivate problem; 2) Describe project and identify key techniques/methods; 3) Describe results and identify contributions for CV/solving the problem
    - The abstract should not use hyperbole or overstatement; show, don’t tell; avoid vague language like “we investigate a number of interesting phenomena”.
- Introduction, (10%)
    - Is the problem described clearly? Why is the problem important (if you solved it, who would care)?
    - What is your goal: What are you going to check / test / validate / analyze / improve?
- Related work (5%):
    - We expect a *separate* related work section.
    - Prior work: How have people tackled this problem before?
    - How is your method different / what ideas from prior work does it combine / build upon?
- Overview/Method Figure (5%):
    - A figure that illustrates the problem and / or your method.
    - Has a caption that describes the problem and / or the method.
- Methodology and technical correctness (25%)
    - The hypothesis is stated clearly: what are you trying to show / investigate?
    - The method is described clearly and actually addresses the problem laid out in the introduction
    - Performance metrics are meaningful and clearly defined
    - For students of the graduate version of the course, we expect a slightly more ambitious methodology than for undergraduates.
    - Grading guideline:
        - If the method is described very well and the grader understands what the method is, that is full score.
        - If significant parts of the method are unclear or there are significant technical mistakes, that’s 10% off
        - If there is no methods section / no discussion at all, this is 15% off
    - New ideas & contributions (10 out of 25%)
        - Does the project go beyond re-implementing an existing method and add something that has not been done before? Could be a new analysis of the model, a change to the model, or a new application.

        - If you re-implement something (e.g. the suggested NeRF project), we expect you to discuss where you follow the paper and where you diverge and to discuss the design decisions with reference to follow-on work. Code copying is unacceptable.
    - The larger the team, the higher the expectation. For teams of 1, adding an ablation study / running a new analysis is sufficient. For teams of 2 and more, we expect more, either analyzing the method along several different axes, or making & ablating a proposed change to the model. **(See top of page.)**
    - For 6.8301, similarly, we expect slightly less novelty. An ablation study or small analysis of the model along a previously unseen axis is sufficient. For 6.8300, we expect that the analysis would demonstrate some understanding of what insight would be gained from it, and why it would be important. Similarly, for model changes, we expect that changes to the model are clearly motivated with a hypothesis of what they will accomplish and an analysis that indicates whether the improvement materialized.
    - We require a final paragraph specifying individual contributions; otherwise there may be heavy penalties.
    - Grading guideline:
        - If students come up with their own thing entirely, that's the full score
        - If there are very small changes to an existing method / no discussion on what considerations were made about which parts to re-implement / change, we take of 5%
        - If there is no novelty at all, we take off 10%
- Experimental Results & Discussion (and technical correctness) (20%)
    - Readers understand which experiments you ran: which dataset, hyperparameters of your model, how long you trained it for, what the baselines were, etc.
    - The experiments align with what was described in the methods section.
    - Example experiments are ablations, baseline comparisons on different datasets, experiments that display different model capabilities, etc.
    - The more members in your team, the higher the expectation for experimental result, i.e., baselines, ablations, etc**. (See top of page).**
- Results Figures and Tables (10%)
    - For every student in your team, there needs to be one quantitative **or** qualitative result (i.e., a figure or a table) that displays the results of an independent experiment they ran.
    - For example, this could be an ablation study, a baseline comparison, runs on different datasets, etc.
- References and Conclusion (5%)
    - We expect that you cite *at least* 10 papers
- Clarity and reproducibility (10%)
    - Easy to understand? Be wary of using LLMs: the generated language may seem clear but often distorts your meaning.
    - Can the work be reproduced from the information given in the report?

Reference papers:

- pixelSplat, https://arxiv.org/pdf/2312.12337.pdf
    - Good reference abstract
    - Good related work section
    - Has nice methods figures (Figs. 1, 2, 3)
    - Lots of results figures and tables (Fig. 5, 6, 7; Tabs 1, 2)
- ALDI, https://arxiv.org/pdf/2403.12029
    - Figure 2 is nice explanation of the method
    - Fig 1 emphasizes the motivation
    - Fig 3-6, Tables demonstrate results, ablations, and analysis

- **Good paper abstracts:**

    - **Learning Internal Representations by Error Propagation, Rumelhart et al.**
    This paper presents a generalization of the perception learning procedure for learning the correct sets of connections for arbitrary networks. The rule, falled the generalized delta rule, is a simple scheme for implementing a gradient descent method for finding weights that minimize the sum squared error of the system's performance. The major theoretical contribution of the work is the procedure called error propagation, whereby the gradient can be determined by individual units of the network based only on locally available information. The major empirical contribution of the work is to show that the problem of local minima not serious in this application of gradient descent.

    - **Deep Unsupervised Learning using Nonequilibrium Thermodynamics, Sohl-Dickstein et al. 2015**
    A central problem in machine learning involves modeling complex data-sets using highly flexible families of probability distributions in which learning, sampling, inference, and evaluation are still analytically or computationally tractable. Here, we develop an approach that simultaneously achieves both flexibility and tractability. The essential idea, inspired by non-equilibrium statistical physics, is to systematically and slowly destroy structure in a data distribution through an iterative forward diffusion process. We then learn a reverse diffusion process that restores structure in data, yielding a highly flexible and tractable generative model of the data. This approach allows us to rapidly learn, sample from, and evaluate probabilities in deep generative models with thousands of layers or time steps, as well as to compute conditional and posterior probabilities under the learned model. We additionally release an open source reference implementation of the algorithm.