

Final Project

Dates:

- Paper approval: Sunday, February 1st, 2026.
- Due date: Wednesday, April 2nd, 2026.

General Guidelines

- The project is due in **teams of four**.
 - In special cases, different group sizes need to be approved by mail.
 - Each team should send an email to talrosenwein@gmail.com with team members name, IDs, and a link to the paper for implementation.
- The final report should contain the following:
 - **project.pdf** file - **no more than 5 pages long** (not including references).
 - Specify the names and ID numbers of the group members clearly on top of the first page.
 - Should be written in OverLeaf (Latex editor).
 - Attach a **project_code.zip** file containing the code of your implementation:
 - The code should have **requirements.txt** file with all dependencies required to pip-install.
 - Add a **readme.txt** file that specifies how to run the following scripts:
 - Train script
 - Evaluation script
 - Once installed the requirements.txt file, the code should run as is.
 - Coding language: Python3.10
 - Please attach audio samples from the training set and validation set.
- An **up to 5 points bonus** will be granted to groups that will publish a medium post on their work / publish a research paper at a conference.

Instructions:

- You can select any research paper related to any topic discussed in the course, as long as it uses DNNs. You should **approve your papers** by Sunday, February 1st, 2026.
 - Online materials:
 - You can use online tools / Github repos / etc., but you should train the model by yourselves.
 - Specify and cite the tools you used. Make sure it's clear what you wrote from scratch, and what tools you used during the project.
 - You must understand what's happening under the hood!
 - I may ask you to explain some functions / specific lines of code to test your understanding.
 - Select papers with relatively small datasets:
 - Good idea: LibriSpeech, LJSpeech, etc.

- Bad idea: Whisper's dataset.
- Select papers with relatively small compute required
- You can implement papers that do only the fine-tuning stage, and download the pre-trained models from Hugging-Face.
- Implement the research paper
 - Try to reconstruct results. If achieved- great!, if not achieved, try to explain why.
- Write a report that summarizes your work:
 - The report should look have the following structure:
 - **Abstract**- explains shortly in simple words what you've done and your results.
 - **Introduction**: Explain what problem the research paper addresses, and how it solves it.
 - **Related work**- specifies (and cites) relevant research papers / prior work done in the field.
 - **Method** - The main part of the report.
 - Architecture
 - Explain the evaluation metrics - add equation(s) if needed.
 - Experimental setup (compute, hyper parameters used, datasets, etc.)
 - Specify the challenges you faced when using the Git repository to reproduce the results from the paper, including any modifications or additional steps you had to take to make it work.
 - **Results + Discussion**:
 - Show interesting results and explain them
 - Add the convergence graphs (loss, acc. Of the model during training) you can take screen-shots from Tensorboard.
 - **Future work**:
 - Suggest an idea for improving the proposed method. Do NOT implement it.
 - Explain why you think it will work, and what are the pros and cons of your idea.
 - **Limitations & Broader impact**:
 - Specify the broader impact and risks of the model you've trained.
 - **References** (can exceed the 5 pages):
 - List of relevant references.

Important Note:

If you see something that is really off the charts, please let me know **in advance** so we can think together on alternatives.