Project Guidelines (2024/2025)

Intro

Your final project is your opportunity to dive deep into a problem at the frontier of Deep Learning and Applied AI. You are encouraged to use powerful tools like large language models (GPT o3, Gemini 2.5 Pro, Claude, etc.) to refine your ideas, explore open research directions, draft proposals, and accelerate your technical thinking. Start by identifying a compelling question, study the foundational and recent papers in that area, and don't be afraid to be bold -- many impactful ideas start as creative hypotheses!

These projects are more than exercises; several from previous editions of this course have evolved into publications at top-tier conferences such as NeurIPS, ICML, ICLR, and CVPR. Think of this as a real research endeavor, not just a class assignment.

Examples of past student projects that became full publications:

- MERGE3: Efficient Evolutionary Merging on Consumer-grade GPUs, **ICML 2025**
- Task Singular Vectors: Reducing Task Interference in Model Merging, **CVPR 2025**
- COCOLA: Coherence-Oriented Contrastive Learning of Musical Audio Representations, **ICASSP 2025**
- MASS: MoErging through Adaptive Subspace Selection, **ICCV 2025** (*under review*)
- STAGE: Stemmed Accompaniment Generation through Prefix-Based Conditioning, **ISMIR 2025** (*under review*)
- LoopGen: Training-Free Loopable Music Generation, **ISMIR 2025** (*under review*)
- Activation Patching for Interpretable Steering in Music Generation, **ISMIR 2025** (*under review*)
- ATM: Improving Model Merging by Alternating Tuning and Merging, **ECAI 2025** (*under review*)

Evaluation Criteria (Total: 30 points)

- **Problem clarity & motivation (5 pts):** Is the problem well defined and relevant? Is the motivation grounded in literature or application need?
- **Literature review (5 pts):** Have key papers been cited and understood? Is the novelty of the approach contextualized clearly?
- **Technical depth (10 pts):** Is there a thoughtful method or architecture proposed and implemented? Are experiments well designed?
- **Creativity & ambition (5 pts):** Does the project show originality and an exploratory mindset?
- **Clarity & communication (5 pts):** Is the final report or presentation clear, well structured, and easy to follow?

Before you start

- Fill up this google sheet with your **matr. number** and the **project id** of your choice. If you do a team project, use a coherent background color for your matr. number, the same for all team members. Ignore the **title** column. Each team should use its own color.
- **Team projects** must be approved beforehand, with clear motivations and separation of tasks.
- If you want to propose your own project, see **project ID 0** below. If approved, add the project to the google sheet with project id 0, and write a short title in the **title** column.
- Projects that are not registered in the google sheet **will not be graded**.

What to deliver
(exact submission requirements are subject to change until May 20th, check back in a few days)

Each project submission must consist of:

- A **report** describing the project and its outcomes, adhering to the **provided template**. **Check the page limits** as described in the template.
- A GitHub (or similar) **repository** with code, data, notebooks, checkpoints, results, and any other supplementary material.

Send links and pdf to rodola@di.uniroma1.it and solombrino@di.uniroma1.it with the subject line starting with **[DLAI Project]**.

Additional Submission Details:

REPORT. Your report is the most important evaluation tool for your project. It needs to be clear and concise, but still present in detail what you did and how. It should be structured as follows (section sizes are approximate guidelines):

- Introduction (~1/2 col): To avoid wasting space, please assume that we know the topic of the project and the preliminary knowledge required to understand your project. The introduction should be brief and highlight the MAIN SCIENTIFIC CONTRIBUTION presented in your project (i.e: the idea you implemented that you are proudest of).

- Related Work (~1/2 col): Here, you should briefly summarise the current state of the art in your field. If someone else already solved the problem, it should be presented here. If not, you should find the scientific work that is closest to your field of interest and briefly present it.

- Method (>= 1 col) : This should be roughly divided in two sections.

- BASELINES: here you present either (a) your implementation of previous work done by other researchers on the same topic, or (b) a very naive and simple solution that you can take as a starting point for your research. For example, if your project is on image super-resolution, the baselines could be (a) a naive UNet implementation, (b) the current state of the art model for the super-resolution task.

- CONTRIBUTION: your own scientific contribution to the solution of the problem. Ideally, this should contain your implementation of your own innovative idea that could (and should) improve the results achieved by the baselines.

NOTE on method: This *baseline/contribution* is not a rigorous subdivision. Depending on the problem, baselines might be difficult to achieve, and could be skipped, or the entire project could consist of an implementation of a strong baseline, if particularly challenging. In any way, this METHOD section should include the explanation of the main thing you did for this project.

- Experiments/Result: (>= 1 col) this is just as important as the method. Your goal is to convince yourself first, and the reader secondly, that your method works, by testing it on carefully designed experiments. You should try to evaluate your method with a KNOWN metric whenever possible. Look at the related works: how did they evaluate their methods? Can you apply the same evaluation procedure? If so, you need to. Always assume that your results might be biased, and fight to prove the validity of your method. Remember: the numbers in your results don't matter as much as you think. What matters is how fairly and thoroughly you evaluate your method and what knowledge you extracted from the research process.

- Conclusions (optional) (~ 1/2 col): if space allows, here you can discuss your results, presenting the considerations that arose from them. Please avoid filler content ("We can see that our method outperforms the baseline … and more research could be done with larger datasets / larger models …). Only include this if you have meaningful considerations that might impact future research on the topic. Otherwise, keep further considerations for the appendix.

APPENDIX. The report has to be very concise. The appendix is where you should include everything that you did but did not end up in the report. This is a fundamental tool for us to evaluate your project. What brought you to the final method you proposed? Did you try other methods that didn't work? What didn't work is just as valuable as what did, as long as you asked yourself why and how. We suggest including a brief summary of every single experiment you ran, including hyperparameter searches, different evaluation procedures, failed ideas. There isn't a page limit on the appendix, but you should still try to get straight to the point. You can include as many tables, charts and plots as you'd like. Remember your main goal: taking us through the thought process that guided your research, helping us understand how you came to your ideas and conclusions.

CODE. We will read, run and evaluate your code. Please make sure that (a) it is clear which python files / bash scripts should be executed to reproduce your results (b) your environment is perfectly reproducible (either use UV, or include a well-made list of requirements, specifying versions of the core libraries). It is good practice to include a README.md file that explains the structure of your project folder. You can, and should include links to your checkpoints (if you have any worth sharing), and to download the data you used. While code quality is not the main evaluation factor for your project, it is still very much taken into account. We suggest not to structure your whole project into a giant unreadable Jupyter notebook. If you ran multiple experiments, it is good practice to keep track, inside your code, of all the experiments you ran (you could keep a python / bash script for each experiment you ran, or configure the experiments with YAML configuration files using hydra, or keep your hyperparameters in python dictionaries/dataclasses…). Before submitting, run your code in a new environment and try to look at it with an external point of view: does it clearly convey what you did and how?

Good luck!

When to deliver

There is no fixed date, you can deliver the project anytime **until May 31st, 2026**. After that date, you must pick a new project from the new project pool.

Remember to **register on Infostud** for the exam session when you want your project to be graded.

ID 4: VQ-VAE a posteriori with Geodesic Quantization

This project revisits the classic VQ-VAE pipeline with a twist: instead of jointly learning a discrete latent space, you will first **train a standard continuous VAE**, then perform **a posteriori vector quantization** using **geodesic distances** in the latent space, rather than Euclidean ones.

Here's a possible workflow:

1. Train a standard VAE (e.g., on MNIST, audio spectrograms, etc.).
2. Apply **K-means clustering in the latent space**, but replace the usual Euclidean distance with a **geodesic metric** that better reflects the curvature or topology of the latent manifold (e.g., compute distances via shortest paths over a k-NN graph or use Riemannian approximations).
3. Assign each latent vector to its nearest centroid and build a discrete codebook.
4. Train an **autoregressive model** (e.g., PixelCNN or Transformer) over the resulting discrete codes.
5. Compare results (reconstruction quality, sample fidelity, perplexity) to a standard VQ-VAE trained end-to-end.

This project challenges you to think critically about the geometry of learned latent spaces, explore non-Euclidean clustering and its impact on generative modeling, and understand the tradeoffs between post-hoc discretization vs. learned quantization.

You may work on any modality; MNIST is a good sandbox, but extending to audio, images, or other structured data is encouraged.