CS-503 Guidelines

Visual Intelligence: Mind and Machines
Spring 2025
EPFL, Lausanne, Switzerland

1 Important Information

The course project counts for **60%** of the final grade and is made up of the following checkpoints:

- Project proposal (15%): Due **Sunday, April 13, 23:59**
- Project progress report (15%): Due **Friday, May 09, 23:59**
- Final project presentation (15%) and report (15%): The report is due by **Friday, May 30, 23:59**, and the project presentation video is due by **Friday, May 21, 23:59** (no late submissions allowed for this). The presentations will be organized in the last week (**May 22 and 27**).

Five homeworks (40%) during the semester (further instructions will follow) make up the rest of the grade. A total of **3 late days (72 hours)** are provided for any of the material (except final presentation videos) with no penalty. After that, late submissions count as 0 points.

Projects can be done individually or in groups. There is no minimum or maximum group size, but please consult us if you're less than 3 or more than 4 people.

Individual Contributions: For group projects, please include an author contributions section, that elaborates on what each member of the group did throughout the course of the project (e.g., you can refer to this exemplary author contribution statement to give you an idea). This should be included in both milestones and the final project report.

2 Example Projects & Inspiration

- Make a visual agent inspired by a simple biological organism (*e.g.* a jumping spider).
- Add a camera to any visually blind system out there: Go to a toy store, pick any cheap remote controlled agent, make it controlled with a visual sensor.
- Extract intelligent actionable information from stationary cameras (*e.g.* a camera in the corner of the living room, the passive surveillance cameras, the fridge camera, etc.)
- Take a concept shown in the class and analyze it computationally: *e.g.* a car with two light sensors from Lecture 2, page 9, or a soft robot navigating an environment through growth (Hawkes et al. 2017).
- Robustness: Analyze / demonstrate / enhance any existing vision method in terms of robustness.
- Active Vision: Design computational experiments using any simulation environment *that can render images* (e.g., ViZDoom, various OpenAIGym environments, DeepMindLab, Habitat, Gibson/iGibson, ThreeDWorld, various AI2THOR environments), train a *visually-guided* active agent (e.g., using an RL library, or any other method), and provide an empirical analysis. You can also further participate in any of the relevant CVPR/ICCV/ECCV/NeurIPS challenges, (e.g., Habitat Challenge, AWS DeepRacer, various challenges from the EmbodiedAI workshop).

- Analysis / debunking study: For examples see these papers on ImageNet generalization, metric learning reality check, or cheating on visual kinship challenges.
- Recent lab projects: *e.g.* mid-level visual representations, Gibson environment, cross-task consistency, cross-domain ensembles, semantically-aware sketching, 3D corruptions / augmentations, multi-modal / multi-task learning, long-horizon planning, task discovery, modality-invariant visual odometry.
- Think about the unsolved! Remember the picture of a busy street in Lecture 2, slide 22? There is a plethora of questions you can try to answer about this picture and current computer vision only scratches the surface.

	Train	Test
Classification	ImageNet	ObjectNet, Imagenet-C, ImageNet-3DCC, Imagenet-R, Imagenet-A
Regression	Cityscapes	Cityscapes+rain/blur/fog
Mix	WILDS, Taskonomy	WILDS, 3D Common Corruptions

Table 1: Examples of out of distribution datasets where labels are given. Although any data from the internet e.g. YouTube videos, query images can be used as OOD data.

Below are some samples that would be viable course projects. They are not meant to be your course projects as-is, but samples for you.

Example 1: Computational Experiments for Active Vision

Vision-science literature contains numerous qualitative hypotheses that aim to identify general principles behind our empirical observations of visually-intelligent behavior (e.g., the hypothesis of cognitive maps as an attempt to provide an explanation for how rats can learn the spatial layout of a maze). The classical way to investigate such hypotheses is through designing real-world experiments that can validate them (e.g., experiments of Edward C. Tolman related to cognitive maps). A computational alternative is to instead design and perform such experiments in simulation, by training policies to solve an experimental setup and analyzing the associated parameters that characterize the experiment.

For example, we can investigate the impact of visual landmarks on sample efficiency.

Step 1 Build various visual landmarks within a scene. As pointed out in some literature, convolutional neural networks tend to favor textures in classification tasks. Thus, a hypothesis can be that adding textures to certain paths can help the agent recognize and remember them. To this end, we can create two groups of training mazes that share the same topology, but contain different textures: i) a first group where all mazes consist of blank walls only, and ii) a second group where the right path in any of the mazes is clearly marked by some visual indicator that is the same for all mazes in the group (e.g., a collectible item or statue in ViZDoom). We can also create two groups of test mazes (i.e., again, one group with landmarks and one without) that weren't seen during RL training.

Step 2 Analyze the impact of visual landmarks on training efficiency. We can train some end-to-end reinforcement learning policies to solve the mazes in these two groups. Then, we log the sample efficiency in these two groups, i.e., the number of episodes or frames required for training to attain a certain level of performance. We could measure the performance by the time or steps used to escape from the maze. If the training has higher sample efficiency for the second training group, and also the resulting policy generalizes better to the corresponding group of test mazes, it can be argued that the inclusion of visual landmarks is useful for the solution of the problem.

Caution: Training RL agents will be computationally intensive, and it might therefore be better to use simple simulation environments (e.g. ViZDoom), render low-resolution images (e.g. 64x64), and train smaller models (e.g., ~1M parameters).

Example 2: Computational Evidence for the Eye Design Adaptation.
The eye design varies significantly between different species. Most of these design choices are hypothesized to be specific *adaptations* to the ecology of that species. For example, many insects have two acute zones (instead of a single one in humans and many animals), one of which points skyward. It is assumed to help with prey detection by placing it against the blue sky instead of a background of foliage where the detection would be hard (see ch.7, p.184 of *Animal Eyes*). There are many other adaptations one can find in nature, see *Animal Eyes* or *Visual Ecology* (both can be found in the lab, INJ211) for inspiration and more examples. In this project, you can choose an exemplar adaptation and test computationally whether it brings the deemed benefits.

Step 1 Choose one (or a few) examples of an adaptation of an eye design you want to study. Design an experimental setup by replicating the *core aspects* of the species ecology, s.t., the visual appearance and a task that the agent needs to solve, and the eye design, s.t., multiple acute zones. In the example above, it could be an object navigation task and the visual aspect is the background that makes it hard to detect the object (e.g., they have a similar texture). Develop the corresponding environment and agent in one of the simulators, and make sure you can train an agent with a "standard/default" camera.

Step 2 Equip and train the agent with different eye (i.e., camera) designs. Analyze whether one camera design is superior over the other, e.g., by comparing the convergence speed, sample complexity and the final reward. Ablate the ecology design choices to identify the which aspects are the most important for this eye adaptation to occur.

Example 3: Emerging navigation perception and behavior in different environments
The environment gives rich information for animals to perceive for action. For example, an open environment enables locomotion everywhere while a cluttered one only affords the possibility of moving in unoccupied space. Accordingly, animals need to distinguish obstacles from traversable areas in pursuit or escape. A more rigorous description and other examples could be found in *The Ecological Approach to Visual Perception*.

In this project, we can analyze the perception and navigation in various cluttered environments. An example pipeline is shown below.

Step 1 Generate various cluttered environments with different levels of complexity. Say, we have only boxes as obstacles. We can simply vary the complexity by the number of boxes. Another complexity metric used in Gibson is the largest ratio between the A* navigation distance and the straight line length.

Step 2 Analyze the pattern of navigation behavior w.r.t. the environment complexity. With the environments generated, we can train end-to-end navigation policies with reinforcement learning. The task can be ImageGoal Navigation. We can then analyze the emerging behavior w.r.t. the environment complexity. A commonly-used measurement is the Success weighted by Path Length, which measures both the success rate and the ratio between the current path length and the shorted one. Besides, the diversity of objects the agent sees in an episode could also be an interesting metric. It is intuitive that the agent may want to explore more places if it cannot see the goal object at the start position.

Step 3 Analyze the feature representation of the perception module. A typical structure of an end-to-end policy is a sequence of a feature extractor and a recurrent neural network. As pointed out much in literature, such as DINO, the feature representations (e.g., attention matrices in DINO), can have a strong correspondence to important objects in view. Thus, analyzing the feature representation of the feature extractor could help with understanding the navigation behavior.

Example 4: Robust multi-modal perception

Humans and animals have the ability to perceive the world and act within it using a variety of active and passive sensory modalities, e.g. sight, touch, sound, proprioception, vestibular sense, temperature, magnetoreception, echolocation, and more. Even when some of these sensors are noisy or unavailable, we can rely on others to still perform various tasks. Imagine walking into a room and suddenly the light turns off. You will still be able to navigate, but your behavior needs to adapt to deal with the missing visual modality. Perhaps you will use your hands to detect the walls and walk along them, or perhaps you have built a cognitive map of the room and can use it to more directly navigate to the goal. Indeed, it has been argued that multi-modality is a key driver behind biological intelligence (The Development of Embodied Cognition: Six Lessons from Babies, Smith and Gasser, 2005). More practically speaking, we would like our agents to be robust to failing or noisy input modalities, or to be able to switch off sensors that have a high energy usage when more efficient sensors are sufficient for solving a certain task.

In this project, we want to analyze different aspects of how multi-modality, and especially changing sensor suites affect agent behavior.

Step 1 Come up with a sensor suite you want your agent to have, and make sure to choose a simulation environment that can provide these (or allows you to add them yourselves). Come up with hypotheses of how an agent's behavior might change if certain sensory inputs are not available or are corrupted. Choose the environment and task the agent should solve by taking into account the sensor suite you chose.

Step 2 Train an agent to be robust w.r.t. changing sensor suites through multi-modal masking and/or corruption. Analyze the agent's behaviour w.r.t. dropping certain modalities at test time. What parts of the scene and of the sensory modalities does the agent use/attend to if some modalities go missing or get corrupted? Do agents build cognitive maps even when they don't have access to visual inputs? For example, see this paper on the Emergence of Maps in the Memories of Blind Navigation Agents, by Wijmans et al.

Example 5: Robustness of vision models

Analyzing and improving the robustness of vision models for application or task X:

Step 1 Define different axes of generalization, e.g. input perturbations, viewpoint changes, semantic shifts (urban → rural / country). Identify sources of data to evaluate models on e.g. challenges, YouTube videos. Also see Table 1. Evaluate the model's robustness on these axes. Perform ablation studies to understand sources of failure.

Step 2 Propose a method to improve model robustness. This could be an architectural change (e.g. using attention layers), a new form of pre-training or data augmentation, imposing real-world constraints (e.g. multi-view consistency), using multiple information sources (i.e. modalities), or any other inductive bias that you can incorporate. Demonstrate the improvement in robustness both quantitatively (e.g. by evaluating on robustness benchmarks) and qualitatively (e.g. run the method on YouTube queries to show it yields better predictions, temporally consistent, etc.)

Example 6: Test time adaptation (TTA)
You don't need to limit yourself to training-time robustness mechanisms only. You can also improve model performance by adapting to distribution shifts at test-time, like these papers are aiming to do: A, B, C, D, etc. Compare different test time adaptation (optimization based) methods. These methods adapt the model at test time via optimizating all or a subset of the parameters. Then propose a better way to perform TTA.

Step 1 Identify and categorize different kinds of TTA methods, e.g. according to the loss functions, adaptation parameters. Select several datasets with out of distribution (OOD) shifts. Evaluate these methods, and analyze the assumptions they need. Are these assumption reasonable? What are their limitations?

Step 2 Come up with a better way to perform TTA. This could be designing a better proxy task to use at test-time, architectural changes that yield flexibility or efficiency for adaptation (e.g. hypernetworks, feature-wise linear modulation layers, in-context learning, etc.) or a new training strategy.

3 Project Proposal Guidelines

The aim of the proposals is for you to get feedback from us before diving deep into the projects. The proposal document should be **at least one, and at most two pages** long. **All page limits are excluding references.** Please use the template that we will provide on Moodle in the corresponding week.

We would like you to think and write at least about the following three questions:

1. **What** is the problem you want to solve?
2. **Why** is it important that this problem be solved?
3. **How** do you solve this problem? Be as specific as possible.

The projects should have some connection to vision, perceptual agents, etc, per the general topic of the course. The more creative the project, the better. Creativity can appear in various forms, e.g. in terms of the problem selection, solution formulation, implementation, experimentation, or demonstration. We will cast a wide net and will be flexible with any meaningful efforts, so focus your energy on doing something interesting and less on worrying about grades. It is up to you if you want to implement a project purely in software (*e.g.* a passive vision problem or using a simulator), or whether you want to do something with hardware.

The submission will be handled through the course moodle. **One submission per group is sufficient. Make sure to include in the report the names and SCIPER numbers of every group member.**

Additional words of advice:

- Try to communicate and motivate your idea using visuals, diagrams, charts, or any other appropriate tools you see fit.
- Try to link your project to the recent advancements in the field and what has been proposed before to solve the problem that you're interested.
- If you choose to work on a challenge, e.g. Habitat, make it clear how your approach differs from the ones in the leader-board.
- Allocate your time well between the milestone and the final project delivery date.

4 Milestone Report Guidelines

The milestone report is intended to make sure that projects advance in a uniform and continuous pace. For the milestone, please provide an **at most two-page long** report that includes at minimum a clear description of the following:

- The steps taken so far in the implementation of the project, and deviations from the original proposal together with explanations.
- A discussion about the problems you encountered, the solutions you explored, and problems that you are likely to encounter as your project progresses.
- A tentative but concrete list of the action items you are planning to work on until the final report, and how they relate to the overall goal of the project.

Please use the template that is provided on Moodle alongside these guidelines when writing your report. One submission per group is sufficient, and submissions should be done by **the same person that submitted the project proposal**.

5 Final Report & Presentation Guidelines

5.1 Final Report Guidelines

Your final report should be **5 pages maximum**, excluding references. You can use appendices without any limits on the page number, but make sure that the main material is provided in the main report. Please use the template that is provided on Moodle alongside these guidelines when writing your report. As before, one submission per group is sufficient. We also ask you to submit your **code** with proper running instructions (e.g. a ReadMe file) in a **zip file**. The ReadMe file should be self-contained: specify the packages to install, their version numbers, commands on how to run your code to reproduce your results and the file hierarchy with a description of each file. Please make sure the code is understandable, e.g. through documentation. You will not be graded on your programming quality, but we need the code for verification and reproduction.

Please try to organize your report in the following suggested way for a better understanding.

- **Abstract:** Provide a brief description of your problem, approach, and key results.
- **Introduction:** Describe the problem you are solving, its significance (i.e. why are you solving it?), and how do you solve it. You can organize this section similar to the introduction of your proposal report while being *more concrete and specific.*
- **Related Work:** How is this problem currently solved (if solved)? Discuss the relevant works to your project and pose your approach against them. Indicate the *differences* and *similarities* between your project and these works as clearly as possible.
- **Method:** Explain your approach for solving the problem. Justify the design choices you made and mention other alternatives, if any. Make sure to include figures, diagrams, pseudo-code, etc. to strengthen your case. It is important that your method is explained in an understandable way for a fair evaluation.
- **Experiments:** Discuss your experimental setup in detail. Explain your baselines and justify why you picked them. Support your results with *quantitative and qualitative evaluations* comparing

your method to these baselines (e.g. include tables for performance metrics and qualitative figures.). If relevant, perform ablation studies to provide further insight into the inner workings of your method.

If your project *did not work as expected*; and you instead managed to systematically invalidate an apriori sensible hypothesis; that is also a perfectly meaningful contribution. If this is the case, provide a detailed and sensible analysis that identifies the main modes of failure of your original hypothesis, discusses their potential reasons, and distills what one can learn from them.

The projects will not be regarded as successful only if they "work"; any project that extracts interesting insights or contributes a signal towards evaluating a meaningful hypothesis will also be regarded as successful. The main evaluation criteria will be the degree of creativity, motivation, and scientific rigor with which you managed your project (e.g., asking interesting and sensible questions, forming and validating hypotheses in a systematic way using the appropriate baselines), rather then the end score you obtained on some pre-defined benchmark.

- **Conclusion and Limitations:** Provide a brief summary of and takeaways from your project. Mention the limitations of your method and how can they be solved. Also mention possible future extensions or other use cases.
- **Individual contributions:** If it is a group project, include an author contribution section explaining the role of each group member throughout the project. You can refer to this exemplary author contribution statement to give you an idea.

Additional suggestions:

- Make sure you proofread your report (or ask an external person to do it, if possible).
- Visuals (figures, tables, etc.) can be more effective at conveying information than writing. But do make sure that your visuals are helpful, e.g. include captions that describe and explain the take home message for your figures and tables, plots are understandable (e.g., with proper labels and readable font size for axes, etc).
- You can include videos as part of your results if you wish. You can also provide extended image results in your appendix.

Using alternative mediums for the final report: For your final report, you are generally encouraged to use other suitable mediums when your project can benefit. For example, you can prepare a webpage under the following conditions:

- If you use an online report (as opposed to sending the offline webpage package), please use a platform where meeting the deadline can be verified, e.g. use GitHub Pages by creating a repository where the last commit is no later than the deadline.
- Put a corresponding amount of material to what the page limit for the standard PDF report would allow, i.e. 5 pages excluding references, by using a comparable number of words.

See some examples here (from CS-503 Spring 2023), here, here, and here for inspiration.

Best project: After the presentations and submission of the reports and completion of grading, the best project will be announced. We will honor the best project members with (virtual) registration to their favorite AI/ML/CV/robotics conference of their choice (NeurIPS, CVPR, CoRL, etc.).

Evaluation criteria: Our grading criteria will be based on the novelty of your project and the scientific contribution it brings, thoroughness of your experiments and the rigor of your consequent analysis, and the quality of your delivery and writing. The weights of these aspects will be adjusted based on what makes sense for each project and to help you – your project does not necessarily have to hit all the marks to get a full grade, as long as it does what makes sense. For example, an analysis oriented project may not have a “novel" component, but the analysis will be valuable. Again, if your project *did not work as expected*, provide a clear and compelling discussion about why this is the case and what one can learn from it.

Furthermore, please make sure the audio quality is good. We encourage you to test the clarity of audio using the speakers in classroom. Another common issue with students' presentation is that, to fit in the time limit, they resort to inferior solutions such as talking fast, going through the slides too quickly, etc. Not being able to understand the presentation has a significant negative impact. The right way to address the time limit is to distill what should and shouldn't be said. Specifically, avoid talking too fast or showing the slide content for a duration less than it'd be needed to be understood by first time visitors.

5.2 Final Presentation Guidelines

Final presentations will be in person and in the form of pre-recorded videos. We require a 4 minute video for each project that will be streamed in the class room. This will be followed by a few minute of live Q&A session. Students of other groups are encouraged to also engage and ask questions. We will divide the presentations into two across the last 2 scheduled classes (see course calendar): half of them will happen in the exercise hour and the other half in the lecture hour. You will be randomly assigned to one of these two time slots. The students are expected to attend all sessions regardless of whether they are presenting or not.

We suggest creating a slide deck based on the provided structure in the sample deck (or another structure if it makes more sense for your case). Then record yourselves presenting these slides. You can use Zoom for recording, and please make sure the resolution is good enough. We ask you to include your face in the video too (i.e. not just a screen recording plus audio) so as to verify your identity. If the project has multiple students, you can pick one representative to speak or split the material across students. We encourage the latter option.

Following is a suggested structure for video presentation:

- **Problem Statement:** Briefly describe the problem and its importance.
- **Related Work:** Briefly describe the previous works tackling the problem you wanted to solve and their shortcomings.
- **Your Approach:** Describe your approach to solve the problem.
- **Results:** Describe evaluations you made and their significance.
- **Limitations and Future Work:** Describe the limitations of your work and possible future extensions.

Additional suggestions:

- Make sure you allocate enough space for visualizations, (i.e. optimize the canvas in a way that everything is easy to read and understand). Do not try to put too much text on the slides.

- Guide the audience through the slides, e.g., by revealing slide content like bullet points and images only when you talk about them.
- Try to include videos and animations if it helps explain your case.
- Make sure you speak in a clearly understandable way (e.g. do not speak very fast).
- Use a quiet place for recording and avoid using a poor microphone, e.g., those embedded in over-the-head headphones. Laptops and phones typically have good built-in microphones. Also, avoid speeding up your video to cram in more content leading to becoming unintelligible.
- You have a rehearsal option with your buddy TA (i.e., the TA who graded your project so far). Please reach out to your TA to book a rehearsal time to get feedback before the actual presentation.

Evaluation criteria: In addition to the evaluation criteria for the final reports, grading criteria will also include the clarity and conciseness of your presentation.

Good luck! ☘