In-Class Work Session (Trustworthy Deep Learning)

Initial grouping (by application area)

We'll form brainstorming groups by field, since this drives data, models, and evaluation:

- Computer Vision
- Natural Language Processing / Large Language Models
- Sciences (biology, chemistry, physics, climate, etc.)
- Networks / Graphs
- Signals / Audio

Want a different category? Email cpsc471_fall25_staff@googlegroups.com before the session with a brief justification.

What you should leave the session with

By the end of the session, your group should complete the one-page worksheet below. Keep answers brief (1–2 sentences per item).

Quick worksheet

- Team & area:
- Task & dataset(s):
- Baseline model:
- Trust goal (primary/secondary):
- Method (key idea):
- Metrics & stress tests:
- Success criteria (what result would convince you?):
- Feasibility notes (compute, timeline, risks):

Brainstorming prompts (work through in order)

Use these to guide your discussion in class. You don't need to answer them in the worksheet, but keep in mind that these questions are critical and should be addressed in your final project.

A) Map the space

- What data modalities are standard in your area? (images, text, time series, graphs, molecules, multimodal…)
- List 3–5 common tasks for that modality (e.g., classification, segmentation, retrieval, summarization, link prediction, ASR). They could be discriminative or generative tasks.
- Which tasks need trust the most in your application (and why)? Consider end-users and failure costs.

B) Baselines & resources

- Hugging Face: identify top models and datasets for your task.
- How are they evaluated? What assumptions do those evaluations make?
- In what realistic scenarios might a user not trust these models?
- Assume you will start from one public dataset and one baseline model from these hubs.

C) Choose a trust objective

- Pick one primary axis and (optionally) one secondary:
    - Explainability / Interpretability (saliency, concept attribution, mechanistic interpretability, sparsity/structure)
    - Adversarial / Corruption Robustness (ℓp attacks, patch, common corruptions, backdoors)
    - Privacy (membership inference resistance, DP training, PATE)
    - Fairness / Bias (group, individual; representation shift)
    - Efficiency / Compression (pruning/quantization that preserves trust signals)
    - Calibration / Uncertainty (ECE, NLL, selective prediction, abstention)

    - Reliability under Shift (OOD detection, conformal prediction, test-time adaptation)
- Questions to consider:
    - Who is the stakeholder (clinician, auditor, end-user)?
    - What do they need to trust? Why is it critical?
    - What's your threat model (attacker capability, shift type, sensitive attribute availability)?

D) Methods: what will you actually do?

- Use Papers With Code / arXiv or surveys to find 2–3 candidate methods. For each:
    - What's the core idea?
    - What motivated it? (threat model, stakeholder needs)
    - Where else could it be useful (modalities/tasks)?
- Your plan: Will you (i) reproduce it, (ii) adapt it to your data, or (iii) propose a novel twist?

E) Metrics: how will you prove trust?

- Pick at least one primary metric plus a sanity/robustness check.
- Do these metrics actually reflect the user risk you care about?
- What are the strengths/limitations? (e.g., saliency sanity checks, calibration under class imbalance, fairness under shift)
- Can you design a simple stress test (augments, synthetic confounders, OOD split) that makes failure modes visible?
- How will you communicate results (plots, dashboards, case studies, failure analyses)?

Scoping & feasibility (keep it doable)

- Data: accessible? size? preprocessing hurdles? licensing?
- Compute: fits within course resources? plan a smaller pilot first.
- Baselines: runnable in <1 day? identify one fallback.
- Risks: what could block you, and what's Plan B?

Notes & reminders

- Improving trust may or may not improve accuracy; that's okay—justify with your metric(s).
- You can intervene at training, inference, or data (or a combination).
- Data transformations (e.g., privacy-preserving pipelines, fairness-aware sampling, attribution-preserving augmentations) are valid methods—explain why they should help and how you'll test it.

Useful resources:

An excellent starting point would be the reading materials (surveys, papers) mentioned in slides. You can also explore the following sites for example implementations:

- Captum library
- Papers With Code

These tutorials may also be useful for inspiration or scoping:

- Explainability:
    - LIME Tutorial: https://captum.ai/tutorials/Image_and_T ext_Classification_LIME
- Adversarial robustness
    - Defense examples by Nicholas Carlini: https://github.com/google-research/selfstudy-adversarial-robustness
- Privacy
    - Flower tutorial on federated learning: https://flower.dev/docs/framework/example-pytorch-from-centralized-to-federated.html
- Fairness
    - Learning a fair loss function in pytorch: https://andrewpwheeler.com/2021/12/22/learning-a-fair-loss-function-in-pytorch/
- Efficiency

- Quantization tutorial in pytorch lightning: https://lightning.ai/docs/pytorch/stable/advanced/post_training_quantization.html

Please refer to the proposal rubrics for the project and proposal requirements

In-class Programming Session

Session Preparation

The VQA V2 validation data, questions, and annotations from this webpage are downloaded on Bouchet at this location "/nfs/roberts/project/cpsc4710/shared/ViLT/Datasets/VQAv2/", which you can access using this folder path. Make sure you see "v2_mscoco_val2014_annotations.json v2_OpenEnded_mscoco_val2014_questions.json val2014" when running "ls /nfs/roberts/project/cpsc4710/shared/ViLT/Datasets/VQAv2/".
Please ensure that you have access to the GPU cluster.
Check that the GPU access is working by importing torch and placing a tensor on the GPU.

```
import torch
z = torch.zeros(3, 3).cuda()
```

If you have time, it's best to get familiar with this before the class.

Part 1: Multimodal Interpretability

In this part, you will fine-tune a simple Visual Question Answering (VQA) model. Then, you will apply the Layer Integrated Gradients method for interpretability of the multimodal model. You will use the following notebook for this part, which contains most of the necessary code.

Steps:

1. Follow instructions to fine-tune the ViLT model using the GPU cluster. You may need to adapt the file paths accordingly for your particular environment.
2. Answer two questions on the existing code. Then complete the implementation by instantiating an appropriate LayerIntegratedGradients object.
3. Generate interpretability visualizations for the three example images.

Part 2: Dataset Exploration

In this part, you will download a dataset for your project and conduct some preliminary analysis.

Steps:

1. Visualize a few examples from the dataset. Do you observe any interesting patterns? Are there any issues that you notice with the dataset that may impact the results?
2. Plot the class distribution. Are there any categories with relatively few examples?
3. *(Extra credit)* Perform additional qualitative/quantitative analyses: feature distribution, clustering, word clouds, correlation analysis, error analysis of baseline models, etc.

Submission Guidelines

Submit your notebook along with your responses in PDF format on Canvas.