Project Plan: Scam Detection with LLMs

1. Motivation & Background

- **Problem**: Online scams are highly diverse and evolve rapidly, making binary classification (scam vs. non-scam) insufficient for real-world defense.
- **Status quo**: LLMs have been explored for scam detection, but mostly in binary settings without fine-grained scam type identification or transparent reasoning.
- **Challenges**:
    - Scam tactics change over time → models must generalize to *emerging scams*.
    - Regulators and financial institutions require **explanations**, not just labels.
- **Opportunity**: LLM-generated **chain-of-thought (CoT) reasoning** and **structured outputs** can be repurposed as knowledge for retrieval and as weak supervision signals for fine-tuning, improving accuracy, explainability, and adaptability.

Is it possible to make it multimodal way? -> logo? Banner? What's in the email?

---

2. Research Questions

Main: how to use prompt engineer to effectively detect job scams.

1. How can LLM-generated **reasoning traces and structured outputs** be transformed into a dynamic retrieval knowledge base for scam detection?
2. Can these outputs serve as **weakly supervised training data** to improve fine-tuning performance under resource constraints?
3. Which strategies most effectively improve detection of **emerging scam types** and **fine-grained scam distinctions**?

Currently, we are working with imposter. Can we generalize to other types of scams?

3. Methodology

(A) Data

- Source: **CFPB consumer complaint dataset** (scam-related categories).
- Preprocessing: taxonomy normalization, text cleaning, temporal split (recent months reserved as *emerging scam evaluation*).

In hardware case: 1 rtx 5070 16G

Do we have another available source?

(B) Pipeline

1. **Baseline**: Zero-/few-shot prompting and embedding-based classifiers(KNN).
2. **Output + CoT generation**: Prompt LLM to produce `{scam type, structured fields, concise reasoning}`. Compress CoT into **evidence–clue–rule triples**.
3. **Knowledge base (RAG)**: Store definitions, prototypes, contrasting examples, and reasoning triples. Retrieval combines embeddings and structured filters.
4. **Weak supervision for fine-tuning**:
   - Keep only self-consistent, evidence-aligned outputs.
   - Augment with re-labeled difficult cases.
   - Fine-tune lightweight adapters (LoRA/SFT) on `{text, retrieved evidence, structured rationale}`.
5. **Error-driven refinement**: Iteratively update knowledge base and training samples with misclassified or novel scam cases.

   Other possible add up:

4. Evaluation

- **Metrics**: Macro-F1, pairwise confusion for closely related scam types, Precision/Recall
- **Comparisons**:
    - Prompt-only vs. Prompt+CoT
    - RAG without vs. with reasoning triples
    - SFT (one method of fine tuning)？
- **Generalization**: no idea……

---

5. Timeline（If idea ok）

- **Week 1–2**: Data cleaning, taxonomy mapping, baseline setup.
- **Week 3–4**: Output+CoT prompt design, initial RAG knowledge base.
- **Week 5–6**: Weak supervision pipeline, lightweight fine-tuning.
- **Week 7**: Full ablation and temporal generalization experiments.
- **Week 8**: Error analysis, draft report/paper.