University of Toronto
Department of Electrical and Computer Engineering
ECE1508: **Deep Generative Models** A. Bereyhi - Summer 2025

Course Project

Code of Honor. All external resources used in the project, including research papers, open-source repositories, datasets, and any content or code generated using AI tools, e.g., ChatGPT, GitHub Copilot, Claude, Gemini, must be *clearly cited* in the final submission. The final report must also include *a clear breakdown of individual group member contributions.* Any lack of transparency in the use of external resources or in reporting group contributions will be considered academic dishonesty and will significantly impact the final evaluation.

Topic	Generative Travel Plan Advisor Using Language Models with Interactive Refinement
Category	Open-ended
Supervisor	Amirhosein Rostami

Objective The objective of this project is to create a generative AI-based travel planning agent that assists users in designing personalized, actionable, and adaptive travel itineraries. The system will receive structured user input, including destination, travel dates, price range, trip type, and interests, to produce end-to-end travel plans enriched with real-time web tool information. Apart from the first generation, the agent will support iterative refinement through interactive dialogue, enabling the dynamic updating of constraints and user preferences. By integrating language models with external APIs, monitored structured context, and a self-improvement cycle, the project will offer an intelligent and user-oriented travel planning experience.

Motivation As the pace of life increases in today's world, the majority of young people are eager to travel to new destinations but lack the time and expertise to create detailed travel plans. While AI-driven travel software already provides basic itinerary suggestions, it falls behind in two significant dimensions: editability and customization. Current systems are either overly structured or overly black-boxed, where users are not able to perform fine-grained edits without recreating the whole plan.
We saw a gap in the market for a smart, dynamic assistant that can both learn about and understand the user's personality. It can also enable fine-grain editing of individual trip components without disrupting the rest of the itinerary. This system will allow users to co-create flexible, customized, and endlessly revisable travel plans with minimal effort, making high-quality trip planning accessible to time-scarce travelers.

Requirements The final submission should address the following requirements while the details can be freely decided by the group members.

1. **Prompt Schema Design**
   Define a structured schema for user input that captures essential travel constraints such as:
   - Destination and origin
   - Dates and duration
   - Budget constraints
   - Trip style (e.g., relaxed, adventure, cultural)

  - Number of travelers and group composition
  - Personal interests and goals

2. **Plan Generation via LLM**
   Implement a pipeline that uses an LLM (e.g., GPT-4, Claude, Mistral) to generate:
   - Daily travel itineraries with recommended activities
   - Suggested transportation and accommodation options
   - Estimated costs per day and category
   - Reservation instructions for tickets, events, or activities

3. **Tool Integration with MCP Protocol**
   Develop a plugin-based agent that calls external web tools (e.g., flight/hotel search, Google Maps) using the MCP protocol. These tools retrieve real-time data to ground the plan in actual availability and prices.

4. **Context Management & User Interaction**
   Maintain a dynamic, structured state of the user's evolving preferences and constraints. Allow interactive updates (e.g., change destination, adjust budget), and reflect these changes in the regenerated plan.

5. **Self-Refinement Loop**
   Detect plan inconsistencies (e.g., unavailable hotel, activity outside time window, over-budget) and implement a self-refinement mechanism that:
   - Queries tools
   - Adjusts the plan via LLM with feedback
   - Reports are updated to the user transparently

6. **Evaluation and Benchmarking**
   Evaluate the travel plan output using metrics such as:
   - Feasibility
   - Personalization: alignment with user profile
   - Responsiveness: how well it reacts to feedback
   - Compare with human-designed plans or plans from online platforms (e.g., Google Travel, TripAdvisor)

7. **[Optional] Itinerary Visualization UI**
   Build a lightweight frontend to visualize and update travel plans in real-time, supporting interaction and a feedback loop.

Milestones The following milestones are to be accomplished throughout the semester.

1. **Prompt and State Schema Definition**
   Design the structured prompt format and the schema for representing user state and constraints. Define example inputs and edge cases.

2. **Core Plan Generation Pipeline**
   Implement the initial pipeline that sends the user state to the LLM and receives a full itinerary. Ensure correct parsing of model outputs.

3. **MCP Tool Integration**
   Connect the MCP client to tools such as `web_browse`, `flight_search`, and `hotel_lookup`. Enable real-time retrieval of content and structured data.

4. **Interactive Context Tracker**
   Implement a memory module to store user constraints and update them. Add interfaces for user feedback and plan modification.

5. **Self-Refinement and Error Handling**
   Build logic to detect infeasibilities and apply automatic fixes via LLM or rule-based patching.

6. **Evaluation and Comparative Study**
   Define evaluation criteria. Conduct a qualitative and quantitative analysis of the generated plans vs. baseline methods and human-curated examples.

7. **[Optional] Frontend and Final Integration**
   Develop a GUI for interactive planning and visualization. Package the agent for demo or deployment.

8. **Final Report and Presentation**
   Compile implementation details, experiment results, failure cases, ablations, and lessons learned into a comprehensive report.

Submission Guidelines The main body of work is submitted through Git. In addition, each group submits a final paper and gives a presentation. In this respect, please follow these steps.

- Each group must maintain a Git repository, e.g., GitHub or GitLab, for the project. By the time of final submission, the repository should have
  - Well-documented codebase
  - Clear `README.md` with setup and usage instructions
  - A `requirements.txt` file listing all required packages or an `environment.yaml` file with a reproducible environment setup
  - Demo script or notebook showing sample input-output
  - *If applicable,* a `/doc` folder with extended documentation
- A final report (maximum *5 pages*) must be submitted in a PDF format. The report should be written in the provided formal style, including an abstract, introduction, method, experiments, results, and conclusion.
  Important: Submissions that do not use template are considered *incomplete.*
- A 5-minute presentation (maximum *5 slides including the title slide*) is given on the internal seminar on Week 14, i.e., *Aug 4 to Aug 8,* by the group. For presentation, any template can be used.

Final Notes While planning for the milestones please consider the following points.

1. You are encouraged to explore innovative approaches to conditioning or generation as long as the core objectives are met.
2. While computational resources are limited, carefully chosen datasets and training setups can make even diffusion models feasible. Trade-offs, e.g., resolution, training steps, are expected and should be justified.
3. Teams are expected to manage their computing needs and are advised to perform early tests to estimate runtime and training feasibility. As graduate students, team members can use facilities provided by the university, e.g., ECE Facility. Teams are expected to inform themselves about the limitations of the available computing resources and design the model accordingly.

References