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 Adversarial Imitation Learning with Transformer-Based Policy Net
Category	Applications of Generative Models
Supervisor	Mohammadreza Safavi

Objective Implement a modernized version of *generative adversarial imitation learning (GAIL)* [1], where the generator, i.e., the policy network, is modeled using a Transformer. The goal is to train the generator to imitate expert behavior in a simple reinforcement learning (RL) environment through adversarial training.

Motivation GAIL frames the problem of *imitation learning*, i.e., learning to imitate an expert agent [2], as a generative adversarial process. Though the original idea is well-established, the framework can be modernized using more advanced generator architectures. This project aims to realize the GAIL using a *Transformer-based generator*. The project also introduces the opportunity to *style* policies or analyze their interpretability.

Requirements The final submission should address the following requirements while allowing freedom in design decisions.

1. Select a simple environment from OpenAI Gym[1], e.g., CartPole, FrozenLake, LunarLander, with *known* expert behavior.
2. Implement the GAIL framework. In this respect, consider the following items.
   - Use a *Transformer-based generator* to model the policy. Basic MLP or CNN architectures are not accepted for *full-mark* evaluation.
   - The discriminator should classify trajectories as expert versus generated.
3. Train the agent using imitation learning with expert trajectories that are collected either from a pretrained PPO agent or from a publicly available datasets.
4. Evaluate the trained policy by comparing it to
   - an agent trained by classic deep RL method, e.g., Q-learning or policy gradient methods, **Hint:** No need to implement the deep RL approach. Available implementations can be used directly while cited properly.

[1] `https://gymnasium.farama.org/`

- **[Optional]** a traditional GAIL model with a basic generator, e.g., an MLP generator.

5. **[Optional]** Add stylistic control to the policy, e.g., conservative versus aggressive behavior, or visualize attention in the Transformer-based generator.

MILESTONES

1. *Literature review and environment setup.* Study GAIL to understand the framework. Select a transformer-based policy model. Choose a Gym environment and collect or generate expert trajectories.
2. *Implement GAIL with Transformer-based generator.* Build the Transformer-based policy model, discriminator, and adversarial training loop.
3. *Training and evaluation.* Train the imitation agent and evaluate its performance on the chosen task. Compare it with benchmark.
4. **[Optional]** *Extensions.* Introduce stylistic variations, modular policies, or attention-based interpretability.
5. *Final report.* Summarize the findings, including architectural choices, training dynamics, comparison results, and lessons learned.

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

[1] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Proc. Advances in Neural Information Processing Systems (NuerIPS)*, 29, 2016.

[2] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.