Tutorial Project 01

Project Plan: Fine-tuning VLA in MuJoCo (2-Month)

📌 Goal

Train and test a Vision-Language-Action (VLA) model to follow simple natural-language navigation commands in a MuJoCo simulation.

🎯 Objectives

1. **Environment Setup**: Create a small MuJoCo arena with colored pedestals.
2. **Data Collection**: Generate simulation data with vision, language, and action labels using an oracle controller.
3. **Model Fine-tuning**: Apply LoRA/QLoRA to fine-tune an OpenVLA model.
4. **Evaluation**: Test the fine-tuned model in unseen layouts and instructions.
5. **Reporting**: Summarize results and discuss improvements.

🛠 Tools & Resources

- **Simulator**: MuJoCo
- **Model**: OpenVLA
- **Fine-tuning**: LoRA/QLoRA (via HuggingFace PEFT)
- **Hardware**: Google Colab (T4 or A100)
- **Languages**: Python, PyTorch

📁 Dataset Description

The dataset will consist of:

- **Images**: RGB frames (84×84) from the robot's front camera.
- **Language Instructions**: Short natural-language commands (e.g., "go to the red pedestal").
- **Action Labels**: Discrete control commands chosen by the expert/oracle policy.
  - Example actions: `{0: forward, 1: turn left, 2: turn right, 3: stop}`
  - These represent the *correct action* to take given the current image and instruction.

Dataset size target: ~5,000 episodes, each 30–80 timesteps long, with random object positions and lighting for generalization.

🗓 Timeline (2 Months)

Week 1–2: Environment & Tools

- Install MuJoCo, set up Python environment.
- Build arena with 3 colored pedestals (red, green, blue).
- Implement camera rendering (84×84 RGB).
- Test simulation loop.

Week 3–4: Data Collection

- Create templated commands (“go to the red pedestal”).
- Implement oracle controller to navigate to target.
- Log `(image, text, action)` per timestep.
- Collect ~5k episodes with domain randomization.

Week 5–6: Fine-tuning

- Load pretrained OpenVLA.
- Freeze vision/language backbones; attach small action head.
- Fine-tune with LoRA/QLoRA on Colab.
- Track validation accuracy and loss.

Week 7: Evaluation

- Test in unseen layouts and with synonym commands.
- Measure success rate, path efficiency.
- Run ablations (no-language baseline).

Week 8: Reporting

- Write final report with:
  - Task description & setup
  - Dataset summary
  - Training details
  - Evaluation results
  - Discussion of limitations & improvements
- Prepare short demo video.

✅ Success Criteria

- Model reaches **>80% success** on seen layouts.
- Shows **>50% success** on unseen layouts/instructions.
- All notebooks run end-to-end on Colab without errors.