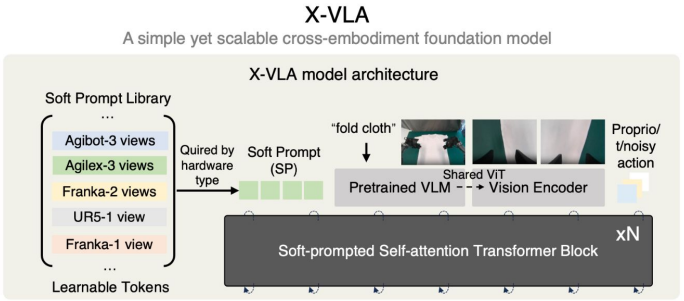
Romadic Winter school Exercise 2: VLAs

By Boris Kuster, JSI

Overview - What are VLAs?



- VLA: Vision-Language-Action model

“Path of evolution”:

- Large Language Models (LLMs):
  text input -> tokenization -> model -> output tokens -> text output

- Vision-Language Models (VLMs) - e.g. GPT-5o:
  text + image input -> text output

- VLAs:
  robot state + text instructions + image input -> robot action output

Key elements:

- Image encoder: Converts image(s) into feature vectors
- Text encoder: Converts text instructions (tasks) into feature vectors
- (Soft-prompt encoder - X-VLA specific: Converts robotic embodiment INDEX (e.g. Franka, UR10, …) to feature vector)
- “Main” transformer model: Converts input features into output feature embeddings
- Action head: Converts output feature embeddings into robotic actions

VLA: Observation space, Action space, Action head

Key design decisions about VLAs:

- Observation space - robot state (which robot data is necessary in order to be able to perform a task)
- Observation space - cameras (# of cameras, positioning, **image resizing dim.** before passing to image encoder)

- Action space: How is the output converted to robot commands?
  - Absolute position/pose (Joint vs. Cartesian)
  - Relative (delta) commands (Joint vs. Cartesian)
  - Optional stiffness “value”
  - Gripper command
  - “Done” flag (VLA indicates it believes a task is finished)
  -

- Action head: The module responsible for converting output feature vectors from “VLM” into robot actions
- Different optimization methods:
  - MSE (mean-squared-error), OR
  - Diffusion/flow-matching based action heads

VLA: Dataset collection

Key limitation of VLAs:
- Limited dataset scales, varying robots/embodiments, different action & observation spaces, …

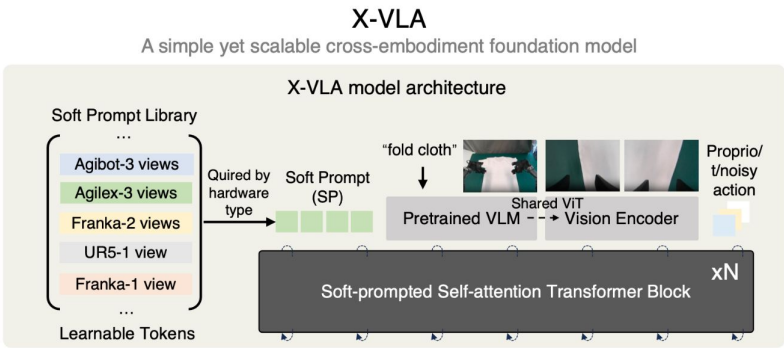
Real-world or simulation-based approaches

Oracle policy:
- Pre-program demonstrations, record them, fine-tune the VLA on gathered data

Teleoperation - human-acquired demonstrations:
- Devices such as Spacemouse, keyboard…

Q&A:
- Determine 5 tasks in which teleoperation is more suitable than oracle-based training.
- What is the downside of training in simulation?
- Example tasks which can NOT be trained in simulation?

- What are key requirements when performing teleoperation, In order to get a “useable” dataset for VLAs?
- Specify 4 downsides of teleoperation-acquired data.



Exercises:

These exercises will lead you through common steps for training and evaluating VLAs:

- Collecting training data manually (teleoperation) and using an Oracle
- Evaluating & filtering collected data
- Fixing code bugs
- Optimizing observation & action spaces
- Fine-tuning a VLA and monitoring progress
- Evaluating a fine-tuned VLA

1. Record a demonstration trajectory
2. Find and fix the bug in Spacemouse configuration file.
3. Run the pre-trained VLA
4. Modify the observation space (robot config file) - add F/T measurements in gripper frame
5. Record new trajectory
6. Run VLA training on updated dataset
7. Check progress (loss stats) on WandB
8. Add a new task to the oracle

A. Code review

- Sketch out the structure of a LeRobot custom module (robot, teleop device, …)

- Determine what the **observation** and **action** spaces are
- How would you **improve** and what would you **add** or **remove** from these spaces? HINT: A crucial element is **missing** in action space

- Determine which **tasks** the Oracle supports
- Think of an example **new task** to add

Relevant repos:

- https://repo.ijs.si/bkuster/lerobot_robot_robotblockset
- https://repo.ijs.si/bkuster/lerobot_camera_ros2
- https://repo.ijs.si/bkuster/lerobot_teleoperator_ros2_spacemouse
- https://repo.ijs.si/bkuster/lerobot_policy_oracle

A. Bugfix and teleoperation

Run the example teleoperation script (does **not record** to disk).

https://github.com/huggingface/lerobot/blob/main/src/lerobot/scripts/lerobot_teleoperate.py

```
lerobot-teleoperate \
    --robot.type=robotblockset_robot \
    --teleop.type=spacemouse \
    --display_data=false \
    --fps=15 \
    --robot.fps=15 \
    --robot.cameras='{
        "wrist": {
            "type": "ros2",
            "width": 640,
            "height": 480,
            "fps": 30
        }
    }'
```

- Unfortunately for you, the robot is misbehaving. It has something to do with the Spacemouse configuration. Find and fix the bug.
- Manually (using your hand) evaluate the robot stiffness. Where is it set and what is its value? Would you change the parameters?

Relevant repos:

- https://repo.ijs.si/bkuster/lerobot_robot_robotblockset
- https://repo.ijs.si/bkuster/lerobot_teleoperator_ros2_spacemouse
- https://repo.ijs.si/leon/robotblockset_python/-/tree/master/robotblockset?ref_type=heads

A. Episode recording - Teleoperation

Record an episode of a new task - Pick up the white socket

```
lerobot-record \
    --robot.type=robotblockset_robot \
    --teleop.type=spacemouse \
    --display_data=false \
    --dataset.fps=15 \
    --robot.fps=15 \
    --robot.cameras='{
        "image": {
            "type": "ros2",
            "width": 640,
            "height": 480,
            "fps": 30
        },
        "image2": {
            "type": "ros2",
            "width": 640,
            "height": 480,
            "fps": 30
        }
    }' \
    --dataset.repo_id=${HF_USER}/jsi_franka_peg_in_hole_v8 \
    --dataset.num_episodes=1 \
    --dataset.single_task="Pick up the white socket" \
    --dataset.episode_time_s=15 \
    --play_sounds=false
    --resume=true
```

Oh no, you've recorded a bad episode! It will ruin your day and your model. How will you delete it?

- https://huggingface.co/docs/lerobot/using_dataset_tools

A. Episode recording - Oracle

Record an episode of an existing Oracle-supported task - Gripper rotation in base frame.

```
lerobot-record \
    --robot.type=robotblockset_robot \
    --teleop.type=spacemouse \
    --display_data=false \
    --dataset.fps=15 \
    --robot.fps=15 \
    --robot.cameras='{
        "image": {
            "type": "ros2",
            "width": 640,
            "height": 480,
            "fps": 30
        },
        "image2": {
            "type": "ros2",
            "width": 640,
            "height": 480,
            "fps": 30
        }
    }' \
    --dataset.repo_id=${HF_USER}/jsi_franka_peg_in_hole_v8 \
    --dataset.num_episodes=1 \
    --dataset.single_task="ENTER TASK HERE" \
    --dataset.episode_time_s=ENTER TASK DURATION HERE \
    --play_sounds=false
    --resume=true
```

The rotation is too slow, we don’t have all day to wait for the VLM to finish a task. Change the relevant parameter.

A. VLA training

Fine-tune the VLA for 10 000 steps on the dataset which contains the new episodes. Monitor the progress using Weights and Biases website.

https://wandb.ai/home

Note: The VLA training will not finish within the duration of this exercise.

A. VLA evaluation

Evaluate the fine-tuned VLA on one of the available tasks.

scp -r bkuster@plankton:/home/bkuster/docker_lerobot_vla_training/workspace_volume/trained_vla .

```
lerobot-record  \
  --robot.type=robotblockset_robot \
    --robot.cameras='{
            "image": {
                "type": "ros2",
                "width": 640,
                "height": 480,
                "fps": 30
            },
            "image2": {
                "type": "ros2",
                "width": 640,
                "height": 480,
                "fps": 30
            }
        }' \
  --dataset.fps=15 \
  --robot.fps=15 \
  --display_data=false \
  --dataset.repo_id=${HF_USER}/jsi_franka_peg_in_hole_v3 \
  --dataset.single_task="Move in gripper positive x" \
  --dataset.episode_time_s=2 \
  --dataset.push_to_hub=false \
  --policy.path=/scripts_ws/trained_vla/checkpoints/last/pretrained_model/ \
  --policy.empty_cameras=0 \
  --resume=true
```

A. Oracle improvement

Evaluate the fine-tuned VLA on one of the available tasks.
Add a new task to the oracle and test it.
Suggestion: Pick up the white socket

A. Theoretical exercises & questions

1. **In which cases** should we prefer to use a VLA as opposed to a manually-programmed solution? **Specify 5 example tasks**.

2. Describe the X-VLA **action head** and its **loss function**.
3. What is the **motivation** for such a choice, and what are the **advantages** and **disadvantages**?

4. Suggest some **other** applicable VLA models.

5. What **other action head types** and **training** are commonly used for VLAs? Find examples of models which use it.

6. For the “peg-in-hole” task, suggest **optimal observation & action spaces**.

7. Based on training results **(“train_loss” graph)**, which **model checkpoint** would you pick, and **why**?

8. What are the **advantages** & **disadvantages** of VLAs compared to **Reinforcement-learning** approaches? Can RL be combined with VLAs?

A. Extra tasks

- **Can you modify the oracle to pick up the white socket?**
- **Record 5-10 hrs of teleoperation episodes for picking up the blue peanut bag.**