Behavioral Cloning for Robotic Manipulation

Ekeno Lokwakai

1. Introduction

Robotic manipulation is crucial in autonomous systems, particularly for industrial automation and human-assistive robotics. This experiment explores behavioral cloning (BC), a type of imitation learning, to train robots based on expert demonstrations.

Three experiments were conducted:

- Lift Task – A simple object-lifting task.
- Can Task – A pick-and-place task with a cylindrical object.
- Custom Dataset Task – A custom dataset for manipulation.

Each experiment was evaluated based on return, success rate, horizon, loss metrics.

2. Problem Formulation

2.1 Configuration and Task Space

The robot used in all experiments is the Panda, a 7-DOF manipulator [1].

- Configuration space: 7 joint angles defining the robot's posture.
- Task space: 3D Cartesian space (x, y, z) with orientation and gripper control.
- Degrees of Freedom (DOF): 7, including position, orientation, and gripper movement.

2.2 State and Action Representation

Variable	Description
State	End-effector position, orientation, linear and angular velocities, joint positions, joint velocities, and object pose.
Action	7D continuous control vector: translation (x, y, z), rotation (roll, pitch, yaw), and gripper actuation.
Observation Modality	Low-dimensional proprioceptive inputs (joint states, velocities) and object state information.

The goal is to train a neural network to predict optimal actions based on observed states, mimicking expert demonstrations.

3. Methodology

3.1 Learning Approach Behavioral Cloning (BC), which maps states to actions through supervised learning is used. The model minimizes prediction error using:

- L2 loss (Mean Squared Error)
- L1 loss (Absolute Error)
- Cosine Similarity Loss (Angle differences between actions)

The network is RNN-based, with two LSTM layers and fully connected output layers.

3.2 Ensuring Physical Limits To ensure feasible motion, the following are applied:

- Action Clipping: Limits action values between [−1,1].
- Velocity & Acceleration Constraints from expert data.

- Regularization: Prevents overfitting and improves generalization.

4. Experiments and Results

4.1 Lift Experiment (Successful Learning)

- Dataset: 200 demonstrations, avg. trajectory length 48.3 steps.
- Training Duration: 2000 epochs.

Metric	Start Value	End Value	Change
Success Rate	0%	78.5%	Significant Improvement
Return Score	0.0	0.7852	Increased
Training Loss	High	Converged	Good Learning

Observations:
- Model successfully learned the lifting task.
- Loss steadily declined, improving execution.
- Validation loss fluctuated, indicating some overfitting.

4.2 Can Experiment (No Learning)

- Dataset: 200 demonstrations, avg. trajectory length 16 steps.
- Training Duration: 2000 epochs.

Metric	Start Value	End Value	Change
Success Rate	0%	0.0%	No Improvement
Return Score	0.0	0.0	Stagnant
Training Loss	Flat Line	Unstable	No Learning

Observations:
- Failure to learn, as indicated by constant horizon value.
- Possible reasons:
  - Incorrect action labels.
  - Task complexity exceeds BC's capabilities.
  - Insufficient sequence length for long-horizon dependencies.

4.3 Custom Dataset Experiment (No Learning)

- Dataset: Custom-collected manipulation dataset.
- Training Duration: 2000 epochs.

Metric	Start Value	End Value	Change
Success Rate	0%	8.0% (4/50)	Minimal Improvement
Return Score	0	0.08	Slight Increase
Training Loss	little drop	Unstable	Poor Learning

Observations:
- Little drop in horizon.
- Poor learning curve suggests dataset issues.

5. Discussion

5.1 Summary of Results

Experiment	Success?	Observations
Lift	✅ Yes	Loss converged, success rate increased.
Can	❌ No	No learning, dataset issues suspected.
Custom	❌ No	Poor dataset quality prevented learning.

5.2 Key Takeaways

- Behavioral Cloning works well for short-horizon tasks like Lift.
- Fails for long-horizon tasks like Can due to compounding errors.
- Dataset quality is crucial incorrect labels prevent learning.

6. Recommendations for Improvement

To address failures in Can and Custom datasets, suggestions will be tried:

1. Improve Dataset Quality:
   - Verify correct action labels.
   - Collect diverse demonstrations.
2. Enhance Model Architecture:
   - Use Transformers for better long-horizon dependencies.
   - Apply Diffusion Models for smoother actions.

7. Conclusion

The study examined Behavioral Cloning (BC) for robotic manipulation tasks.
- Lift task succeeded, showing BC works well for short-term tasks.
- Can and Custom tasks failed, highlighting BC's limitations in complex tasks.
- Future work would be generating and preprocessing quality datasets.

Bibliography

[1] A. Mandlekar, Y. Zhu, R. Mart\in-Mart\in, S. Savarese, and L. Fei-Fei, "Robosuite: A Modular Simulation Framework for Robot Learning," *arXiv preprint arXiv:2011.03548*, 2020, [Online]. Available: https://robosuite.ai/