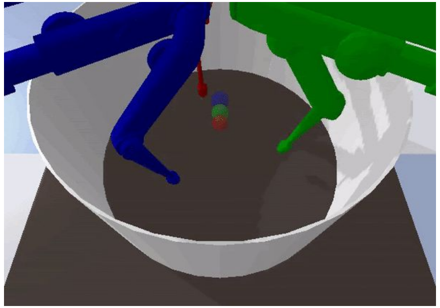
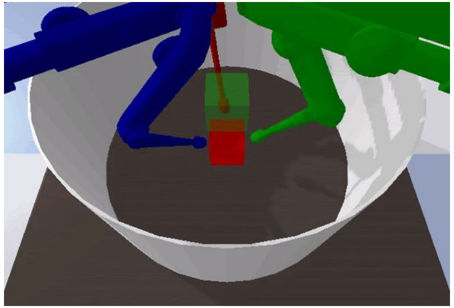
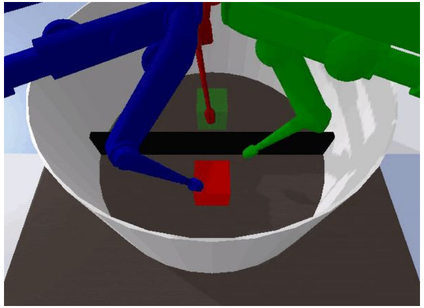
Adapting PAD to Trifinger Robotic Manipulation in CausalWorld

Adaptation and Adaptive Computation

Fall 2025







CausalWorld

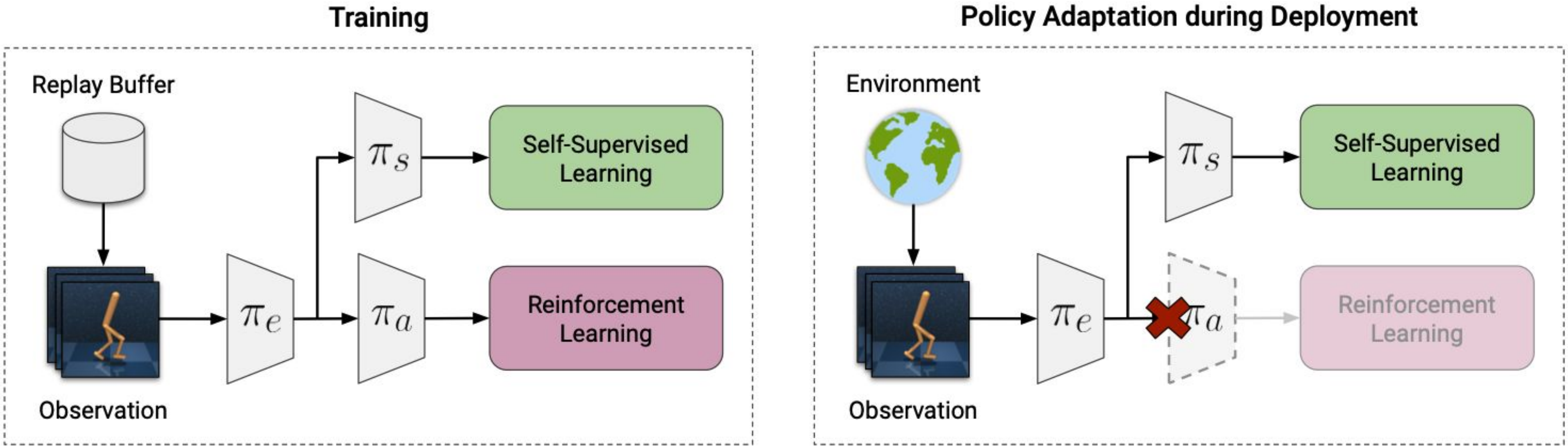
The Environment: CausalWorld

- Simulation of the TriFinger robot.
- Supports interventions on physical variables (friction, link mass, visual appearance).
- Perfect for testing **Generalization** and **Robustness** to Out-of-Distribution shifts.

Our Experimental Setup

- **Task:** Multi-Arm Reaching.
- **Easiest task,** serves as a proof of concept for this method.

PAD: Self-Supervised Policy Adaptation during Deployment



Original PAD

- Input: Pixel-based (Images)
- Task: Simple Control (2D locomotion or Single-Arm Reaching)
- Domain Shift: Visual (Background colors)

This Project

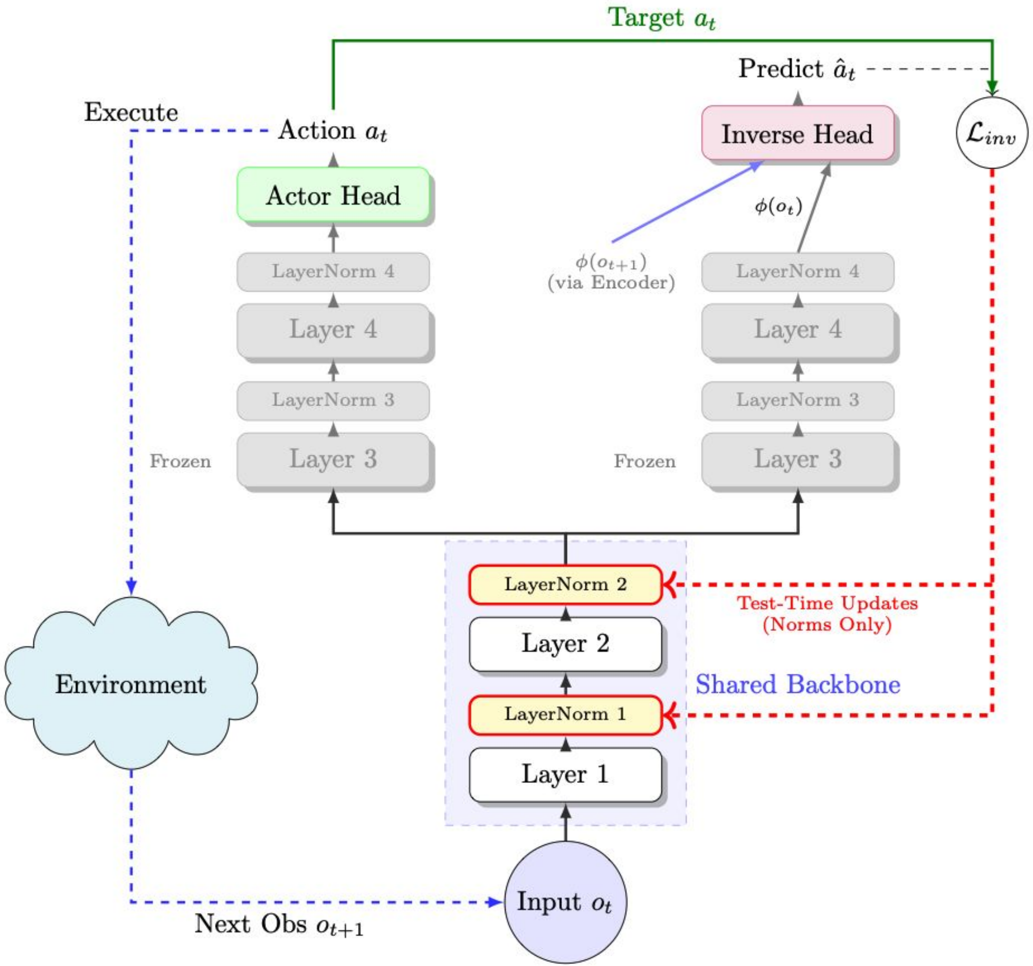
- Input: State vectors
- Task: Complex Manipulation (TriFinger / Three-Arm Reaching)
- Domain Shift: Physical Dynamics (Link masses, Goal positions)

Phase 1: Training

- **Objective: Joint Optimization**. The encoder learns features useful for *both* maximizing reward and predicting physics.
- Update **all** network parameters.

Phase 2: Testing

- **Objective: Self-Supervised Adaptation**. No reward signal available; adaptation relies purely on physics prediction.
- **Freeze all weights.** Update only Normalization statistics in the shared backbone.



Experimental Setup

Methods Compared:

1. **No Auxiliary Task:** Standard RL baseline.
2. **No Adaptation:** Trained with auxiliary task, but frozen at test time.
3. **Episodic Adaptation:** Updates during rollout, resets to baseline after each episode.
4. **Lifelong Adaptation:** Updates during rollout, maintains learned parameters across episodes.

Training Details:

- All models trained for ~50 epochs.
- Total training time: ~8 minutes on a laptop.
- Default environment (fixed reaching goal/mass), no randomization.

Test Environment:

- Models are tested on increasingly difficult unseen goals (unseen joint link mass is too easy).
- Metrics: Average Episode Reward and Success Rate.

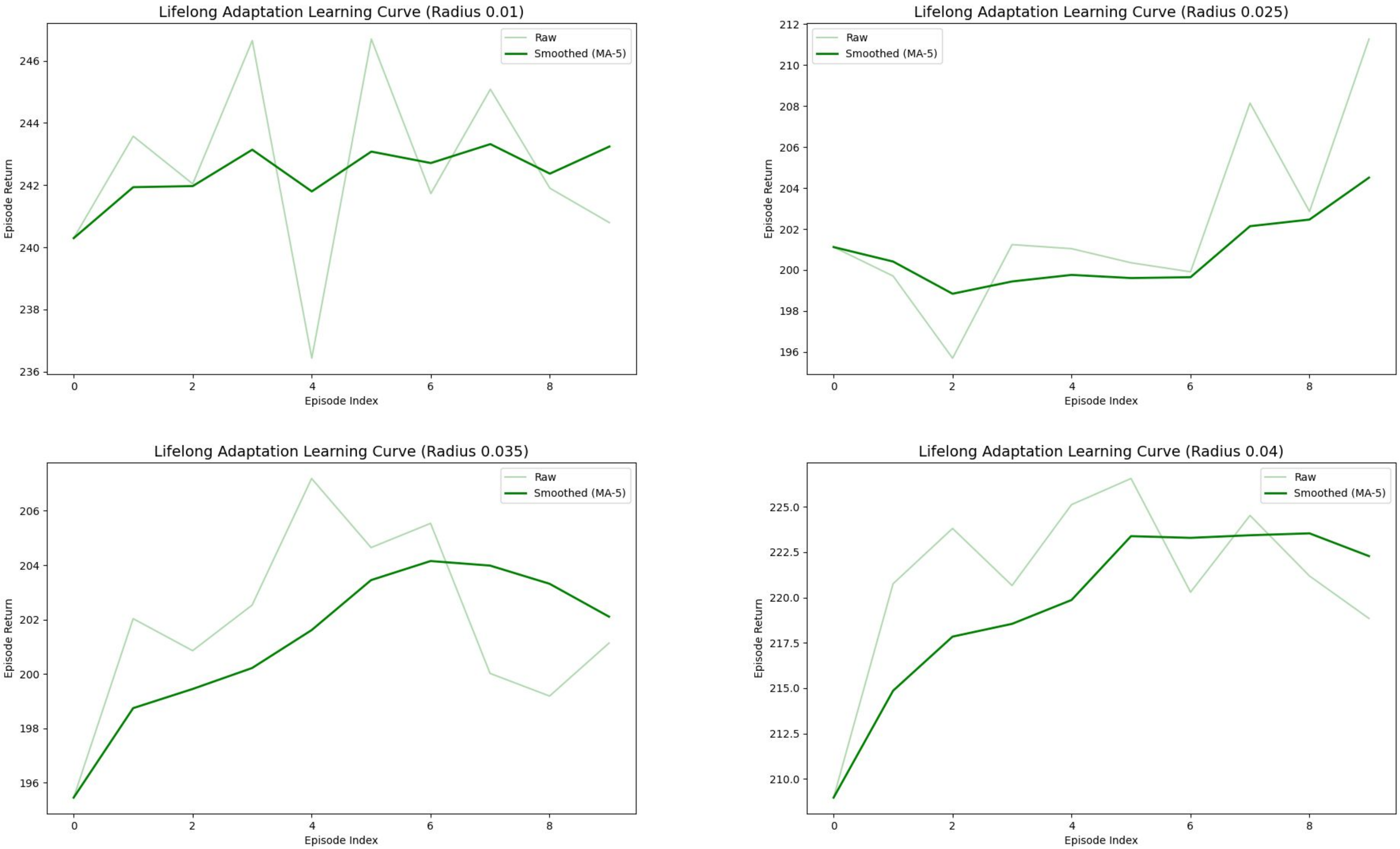
Evaluated 10 episodes per radius with a fixed goal displacement relative to the training set.

Table 1: Comparison of average episode rewards across increasing goal displacement.

		Average Episode Reward (↑)							
Method ↓	**Radius →**	**0.0**	**0.01**	**0.02**	**0.025**	**0.03**	**0.035**	**0.04**	**0.045**
No Aux Task (Baseline)		89.5	151.1	151.4	106.0	188.3	95.1	94.8	61.5
No Adaptation		248.4	244.0	**219.2**	199.4	228.2	198.6	220.1	**167.6**
Episodic Adaptation		246.6	240.3	217.9	201.1	**229.6**	200.5	209.0	166.5
Lifelong Adaptation		**248.6**	**244.5**	218.9	**202.1**	227.1	**201.9**	**221.1**	164.2

Table 2: Comparison of success rate over 10 episodes across increasing goal displacement.

		Success Rate (↑)							
Method ↓	**Radius →**	**0.0**	**0.01**	**0.02**	**0.025**	**0.03**	**0.035**	**0.04**	**0.045**
No Aux Task (Baseline)		0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
No Adaptation		100.0%	100.0%	100.0%	100.0%	10.0%	0.0%	0.0%	0.0%
Episodic Adaptation		100.0%	100.0%	100.0%	100.0%	40.0%	0.0%	0.0%	0.0%
Lifelong Adaptation		100.0%	100.0%	100.0%	100.0%	**60.0%**	**20.0%**	0.0%	0.0%


Lifelong Adaptation Learning Curve (Radius 0.01)
Raw
Smoothed (MA-5)
Episode Return
246
244
242
240
238
236
Episode Index
0
2
4
6
8
Lifelong Adaptation Learning Curve (Radius 0.025)
Raw
Smoothed (MA-5)
Episode Return
212
210
208
206
204
202
200
198
196
Episode Index
0
2
4
6
8
Lifelong Adaptation Learning Curve (Radius 0.035)
Raw
Smoothed (MA-5)
Episode Return
206
204
202
200
198
196
Episode Index
0
2
4
6
8
Lifelong Adaptation Learning Curve (Radius 0.04)
Raw
Smoothed (MA-5)
Episode Return
225.0
222.5
220.0
217.5
215.0
212.5
210.0
Episode Index
0
2
4
6
8

Conclusion

Robustness on joint mass:

- The jointly trained backbone (even without TTA) was sufficiently robust to handle mass variance, proving the value of the auxiliary task itself.

Unexpected Success on Goals:

- Adaptation improved performance on Goal Shifts, despite no change in underlying physics.
- Successful test-time adaptation: corrected OOD observation statistics (e.g., unfamiliar distance_to_goal values).

Future Directions:

- The "Reaching" task is likely too simple.
- Method should be validated on contact-rich tasks (Pushing/Picking) with friction and object mass shifts.