Measurable Deployment

Deploying an agent that is
1. incapable of optimality (due to resource/representational constraints),
2.but still receives an evaluative feedback signal (reward).

We posit such a deployment is a Continual Reinforcement Learning problem.

Why is it a CRL problem?

- Action-induced non-stationarity
- Changes in environment dynamics
- Emergent Novelty
- Evolving Goals

Is your problem continual?

Can the optimal policy π* be reached within the policy set, given available resources?

→ If **yes**: The problem is non-continual. Find and settle on π*.
→ If **no**: The problem is continual. Best agent never stops learning.

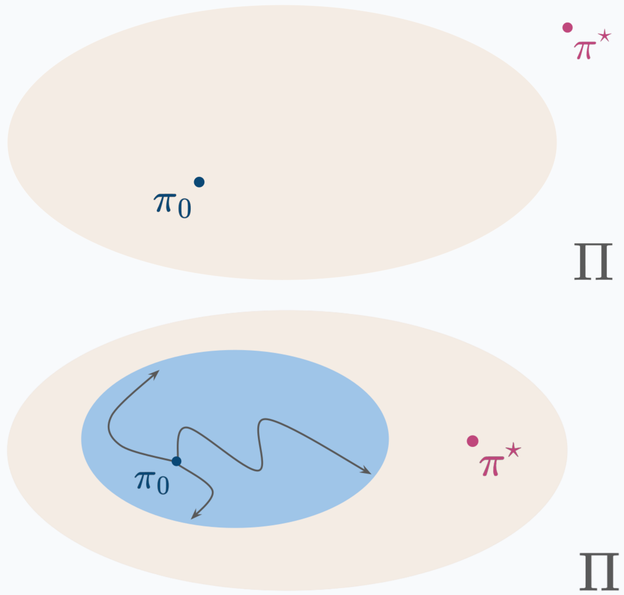
Alternative Views.

1. *Current approachesare already adaptive.* Retraining usually gets triggered by humans or MLOps pipelines, not discovered by the agent itself.
2. *Offline-trained policies generalize well enough.* Generalization gives a head start, not a substitute, it's bounded by what the training phase saw.
3. *Not all deployments need continual learning.* True, but it's a requirement specifically for measurable deployment.
4. *Reward signals may be sparse, delayed, or unobservable after deployment.* Delayed reward still teaches from value differences; total absence of reward is the real edge case Monitored-MDPs address.
5. *Safety and alignment concerns favor fixed policies over continual learners.* Fixed policies degrade unsafely too, safe RL treats adaptation as necessary, not opposed to it.

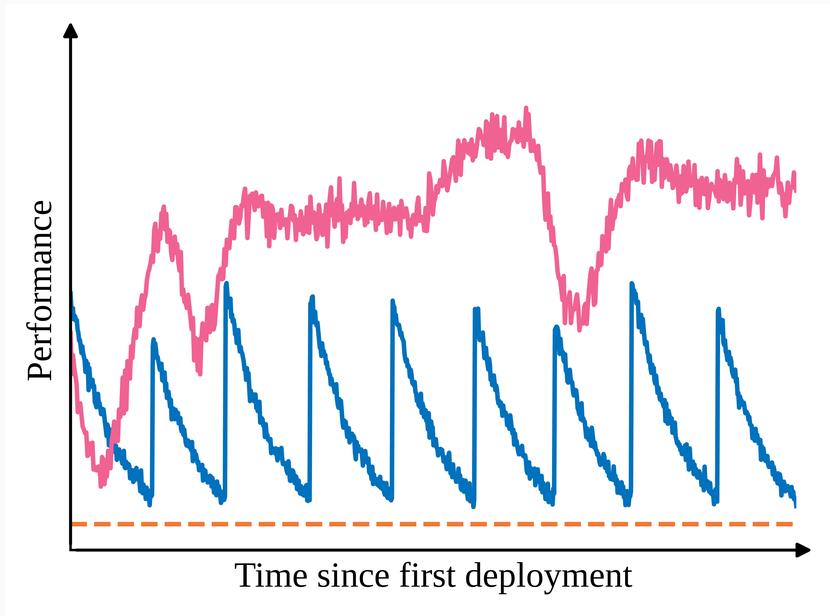
Continual learning isn't optional, it's *usually* a necessity.

The best deployed reinforcement learning agents are the ones that **never** stop learning.

Measurable Deployment.



We have been approximating continual learning problems as a series of non-continual problems.



Evidence from deployed systems.

</> Cursor Tab (Coding Agent)

Code completion improves from every accept and reject signal after deployment, not just pretraining.

Lyft Ride-sharing Matching Algorithm

Online RL that keeps adapting after launch added millions of rides and over $30M in annual revenue.

Sim-to-real gap in Robotics

Policies trained purely in simulation still drift once deployed, so on-robot adaptation stays necessary.

Recommendations for Researchers

- Study the performance of conventional methods implemented in continual settings.
- Rederive algorithm properties for history processes.
- Consider different evaluation metrics.
- Take resource constraints into account.
- Design benchmarks specifically for continual RL methods.

Recommendations for Practitioners

- Recognize your deployment regime.
- Build feedback loops, not just pipelines.
- Validate continually, not just before deployment.
- Introduce non-stationarity deliberately.

Position: Deployed Reinforcement Learning should be Continual
Parnian Behdin, Kevin Roice, Golnaz Mesbahi
parnian.behdin@amii.ca, kevin.roice@amii.ca