Teamprojekt RL

Leif Döring, Bene Wille, Daniel Schmidt

Lehrstuhl Stochastik

11.2.26

Problem

All actor-critic methods are based on some form of the policy gradient theorem

$$\nabla J(\theta) = \sum_t \gamma^t \mathbb{E}^{\pi_\theta}[\nabla \log(\pi_\theta(A_t; S_t))\mathbb{A}(S_t, A_t)]$$

In practice, e.g. PPO, the discount factor is ignored, discounting is only used in advantage estimation.

Why? Because in robot problems, researchers don't care about discounting, discounting is merely used as algorithmic hyperparameter.

Problem: Most people are not aware, but other problems rely on discounting (e.g. everything with money).

What are we going to do?

Experiment with different RL environments and experiment with deep RL effects on the choice of γ.

Algorithm: We start with DQN and PPO.

Application: We start with simple supply chain problems.

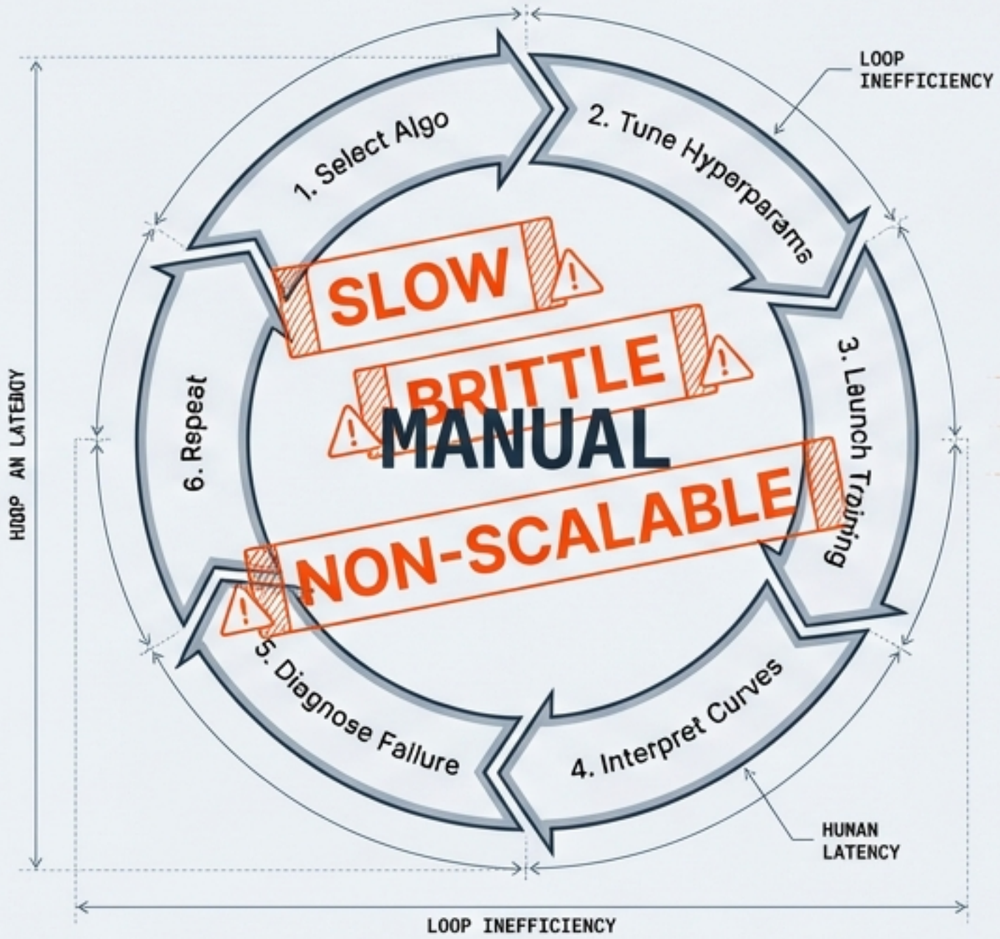
What should you know?

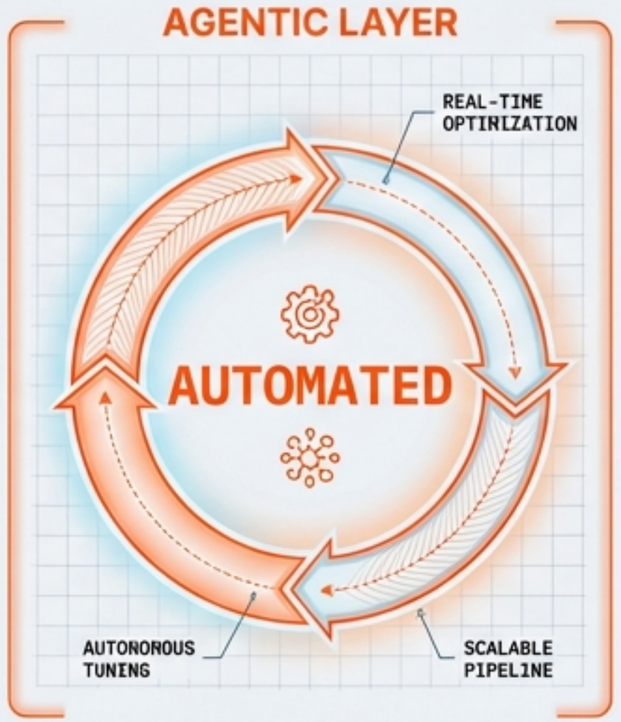
Understand RL.

Strong coding.

RL ALGORITHMS ARE SOLVED. AGENTIC CONFIGURATION IS NOT.

THE HUMAN BOTTLENECK





SYSTEM ARCHITECTURE NOTES

The Bottleneck:

Frameworks like Stable Baselines3 provide the algorithms, but the *experimentation loop* is the enemy.

Every new environment demands a manual, multi-turn process of tuning and diagnosis.

PROBLEM DEFINITION

The Gap:

AutoML solved this for supervised learning. For RL, it remains a craft.

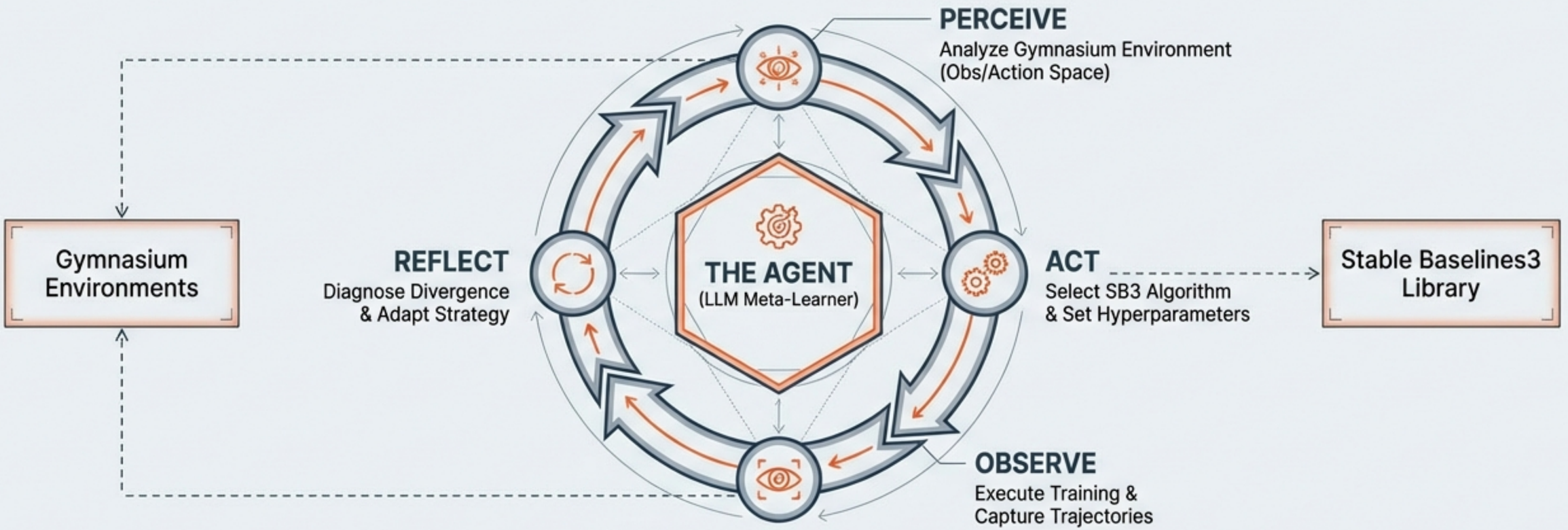
SYSTEM ARCHITECTURE NOTES

The Insight:

Recent work (ROME/ALE, Agent²) proves LLMs can operate in real-world environments over multiple turns.

We will apply this to the RL pipeline itself.

BUILDING AN AGENTIC LEARNING LOOP FOR AUTOMATED RL



The Mission:

Design an LLM-based agent that navigates the full RL experimentation cycle—not as an advisor, but as an actor.

The Benchmarks:

Test against SB3 defaults, Optuna-based HPO, and Zero-shot LLMs across classic environments (CartPole, LunarLander, Atari, MuJoCo).

The Deliverable:

An open-source tool for the SB3 ecosystem that the RL community can actually use.