



Dr. Phil Winder

W Winder
ML | RL | MLOps | DATA SCIENCE

REINFORCEMENT LEARNING:

ReFrame Pt. 2: How We Overcame Key RL Challenges

in **LIVE** -ish



Last Event Recap

- Multi-step, long-term rewards, agent affects environment & outcome
- Simulations are useful
- Lots of engineering still to be done
- Rewards are hard



Challenge 1: Framing the Problem

Issue: The MDP abstraction obfuscates “solvability”

- **Sequential:** Is this problem *really* sequential?
- **Representative:** Does the observation *really* represent the state?
- **Imagine:** yourself trying to solve it? How would you learn? What’s missing?
- **Simplify:** the task as much as possible, then keep iterating.
- **Hierarchy/Separation of concerns:** Could you split it up?
- **History:** Do you need to remember what you did? If yes, can you think of actions that removes the need for having history?



Challenge 2: The Environment

Issue: Real life is expensive, dangerous, and hard

- **Useful:** during development, during training, during testing - but not real life
- **Don't overcomplicate** until you're happy
- **Slowly** introduce actions and observations
- Develop a suite of environments. Ideally testing:
 - Simple
 - Regression tests
 - Different models
 - Different rewards
 - Challenges in your domain - e.g. noise, sequence length, observability, exploration
 - Different difficulties
- **Experiment** through environments - think of environments like ML experiments



Challenge 3: Rewards

Issue: No clear reward

- Rewards define optimality **AND** how to get there
- What can rewards be? **Anything**. A number, a linear model, a function, an ML model.
- **Scale**: super important, especially when you have competing concerns
- **Clipping**: avoid throwing away info
- **Start simple**: treat changes to the reward as an experiment
- **Intermediate rewards**: guide the agent
- **Incorporate domain expertise**: e.g. robot must stand, user must be shown beginner videos, A/C must be powered on before you can alter temperature



Challenge 4: Training & Development

Issue: Tempting to skip to the end


- **Simplicity**
- **Baseline agents**
 - Random
 - Fixed pick a single action (e.g. most popular)
 - Simpler RL algorithms like MCMC or Cross Entropy Method
- **Regression tests**
 - Develop regression tests for situations where “it must get it right”
- Beware of long training durations
- Keep track of your experiments
- Randomness - averaging, seeds, stability, etc.
- Sensitivity to hyper parameters



Challenge 5: Evaluation

Issue: Hard to prove performance

- **Visualise:** Be careful, compare like for like, understand the why
- **What's important?** Algorithmic performance improvements aren't everything
- Very **stochastic:** Sample vigorously
- **Outliers:** Performance metrics don't tell the whole story
- **Truth:** Performance is only realised in production

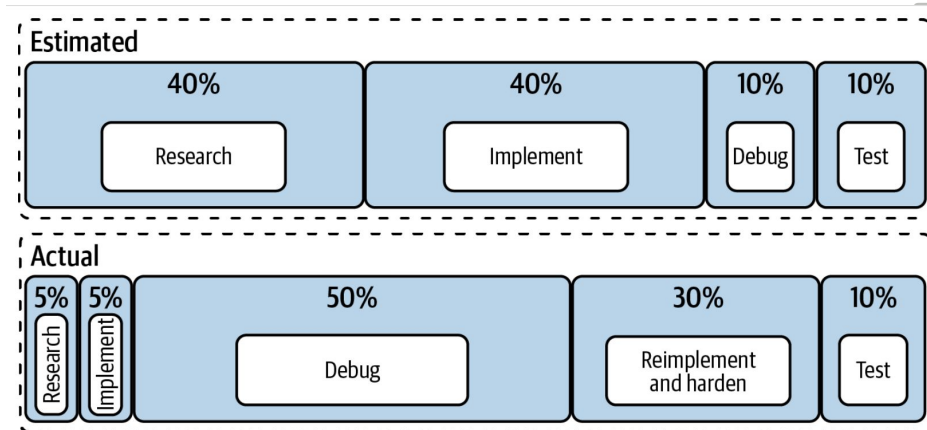


Challenge 6: Deployment

- **Expect a lot of engineering:** deployment tooling is immature
- **You must continue exploring:** unless you know the population state is static and your model is perfect
- **You must continue learning:** unless you know that your model is perfect
- **A/B testing is a must** for new algorithms/observations/actions: MLBs!

Challenge 7: Debugging

- Debugging is hard -
 - one study showed that code-level optimizations improved performance more than the choice of algo
 - Another showed how a single line bug (zeroing an array) caused oscillation in the value estimates
- Standard software engineering debugging techniques are useful
- Monitoring training metrics, evaluate, provide the ability to experiment
- If in doubt, start with something simpler
- Most modern “state-of-the-art” algos are hardware optimisations
- Apps “fail” because the problem isn’t suited to RL





Summary

- Many challenges!
- **KISS - Iterate, don't jump**
- Experiment and simulate
- Evaluate carefully: true performance isn't known until production



[https://Winder.AI/events/
phil@winder.ai](https://Winder.AI/events/phil@winder.ai)
DrPhilWinder

<https://winder.ai>