Advanced Topics in Deep Reinforcement Learning

Scientific Standards

What Even Is Science?

- Systematic
- Testable
- Unbiased
- ...

What Even Is Science?

- Systematic
- Testable
- Unbiased
- ...

How realistic are these assumptions?

Science in Theory vs Practice

Science in Theory vs Practice

- Experiments are designed by humans

Science in Theory vs Practice

- Experiments are designed by humans
- Resources are limited

Science in Theory vs Practice

- Experiments are designed by humans
- Resources are limited
- Not everything is perfectly observable

Science in Theory vs Practice

- Experiments are designed by humans
- Resources are limited
- Not everything is perfectly observable
- Human error can significantly alter results

Science in Theory vs Practice

- Experiments are designed by humans
- Resources are limited
- Not everything is perfectly observable
- Human error can significantly alter results
- Research directions are not chosen/funded equally

Science in Theory vs Practice

- Experiments are designed by humans
- Resources are limited
- Not everything is perfectly observable
- Human error can significantly alter results
- Research directions are not chosen/funded equally

=> Even with best intentions, we can often not live up to the scientific ideal

Examples from ML/RL

Examples from ML/RL

- Losses are only an approximation of the solution - can result in wrong predictions for low loss, e.g. detecting watermarks instead of image content

Examples from ML/RL

- Losses are only an approximation of the solution - can result in wrong predictions for low loss, e.g. detecting watermarks instead of image content
- Many design decisions means we can’t test every combination, potentially missing relevant factors [Engstrom et al. 2020]

Examples from ML/RL

- Losses are only an approximation of the solution - can result in wrong predictions for low loss, e.g. detecting watermarks instead of image content
- Many design decisions means we can’t test every combination, potentially missing relevant factors [Engstrom et al. 2020]
- Seemingly unrelated factors like reward scaling can sometimes skew results significantly [Henderson et al. 2018]

Examples from ML/RL

- Losses are only an approximation of the solution - can result in wrong predictions for low loss, e.g. detecting watermarks instead of image content
- Many design decisions means we can’t test every combination, potentially missing relevant factors [Engstrom et al. 2020]
- Seemingly unrelated factors like reward scaling can sometimes skew results significantly [Henderson et al. 2018]
- Software package updates can make big performance differences

Confounding Factors in RL Experiments

Confounding Factors in RL Experiments

- Seeding

Confounding Factors in RL Experiments

- Seeding
- Versions of environments

Confounding Factors in RL Experiments

- Seeding
- Versions of environments
- Implementation details

Confounding Factors in RL Experiments

- Seeding
- Versions of environments
- Implementation details
- Hyperparameters

Confounding Factors in RL Experiments

- Seeding
- Versions of environments
- Implementation details
- Hyperparameters
- Observation augmentation

Confounding Factors in RL Experiments

- Seeding
- Versions of environments
- Implementation details
- Hyperparameters
- Observation augmentation
- Reward processing

Confounding Factors in RL Experiments

- Seeding
- Versions of environments
- Implementation details
- Hyperparameters
- Observation augmentation
- Reward processing
- ...

Comparing RL Methods

Comparing RL Methods

Fundamental question: **How to show that method A is better than method B?**

Comparing RL Methods

Fundamental question: **How to show that method A is better than method B?**

- “Better” is not a metric - better at what?

Comparing RL Methods

Fundamental question: **How to show that method A is better than method B?**

- "Better" is not a metric - better at what?
- Best practice: hypothesis-driven research

Comparing RL Methods

Fundamental question: **How to show that method A is better than method B?**

- "Better" is not a metric - better at what?
- Best practice: hypothesis-driven research
    a. Formulate a specific and testable hypothesis

Comparing RL Methods

Fundamental question: **How to show that method A is better than method B?**

- "Better" is not a metric - better at what?
- Best practice: hypothesis-driven research
  a. Formulate a specific and testable hypothesis
  b. Design an experiment that can precisely test the hypothesis

Comparing RL Methods

Fundamental question: **How to show that method A is better than method B?**

- "Better" is not a metric - better at what?
- Best practice: hypothesis-driven research
  a. Formulate a specific and testable hypothesis
  b. Design an experiment that can precisely test the hypothesis
  c. Run the experiment with minimal confounding factors

Comparing RL Methods

Fundamental question: **How to show that method A is better than method B?**

- "Better" is not a metric - better at what?
- Best practice: hypothesis-driven research
  a. Formulate a specific and testable hypothesis
  b. Design an experiment that can precisely test the hypothesis
  c. Run the experiment with minimal confounding factors
- Design includes how outcomes are measured

Comparing RL Methods

Fundamental question: **How to show that method A is better than method B?**

- "Better" is not a metric - better at what?
- Best practice: hypothesis-driven research
  a. Formulate a specific and testable hypothesis
  b. Design an experiment that can precisely test the hypothesis
  c. Run the experiment with minimal confounding factors
- Design includes how outcomes are measured
- Before running anything, it should be clear how the results need to look to (dis-)prove the hypothesis

Comparing RL Methods

Fundamental question: **How to show that method A is better than method B?**

- "Better" is not a metric - better at what?
- Best practice: hypothesis-driven research
  a. Formulate a specific and testable hypothesis
  b. Design an experiment that can precisely test the hypothesis
  c. Run the experiment with minimal confounding factors
- Design includes how outcomes are measured
- Before running anything, it should be clear how the results need to look to (dis-)prove the hypothesis
- The outcome should not be explained by anything but the hypothesis

Comparing RL Methods

Fundamental question: **How to show that method A is better than method B?**

- "Better" is not a metric - better at what?
- Best practice: hypothesis-driven research
  a. Formulate a specific and testable hypothesis
  b. Design an experiment that can precisely test the hypothesis
  c. Run the experiment with minimal confounding factors
- Design includes how outcomes are measured
- Before running anything, it should be clear how the results need to look to (dis-)prove the hypothesis
- The outcome should not be explained by anything but the hypothesis

RL Code & Reproducibility

RL Code & Reproducibility

- Perfect reproduction is often not realistic

RL Code & Reproducibility

- Perfect reproduction is often not realistic
- Compute architecture can produce variations in results even with the same code

RL Code & Reproducibility

- Perfect reproduction is often not realistic
- Compute architecture can produce variations in results even with the same code
- Papers seldomly (can) contain all relevant implementation details, so reimplementation usually will not lead to same results

RL Code & Reproducibility

- Perfect reproduction is often not realistic
- Compute architecture can produce variations in results even with the same code
- Papers seldomly (can) contain all relevant implementation details, so reimplementation usually will not lead to same results
- Versioning is very important in RL!

RL Code & Reproducibility

- Perfect reproduction is often not realistic
- Compute architecture can produce variations in results even with the same code
- Papers seldomly (can) contain all relevant implementation details, so reimplementation usually will not lead to same results
- Versioning is very important in RL!
  - Standardized versions of environments
  - Standardized versions of algorithm libraries
  - Standardized versions of optimizers (e.g. Torch)

RL Code & Reproducibility

- Perfect reproduction is often not realistic
- Compute architecture can produce variations in results even with the same code
- Papers seldomly (can) contain all relevant implementation details, so reimplementation usually will not lead to same results
- Versioning is very important in RL!
  - Standardized versions of environments
  - Standardized versions of algorithm libraries
  - Standardized versions of optimizers (e.g. Torch)
- Code quality can make reproducibility easier and be automated

Statistical Testing

Statistical Testing

- In most empirical STEM disciplines: statistical testing is the most common comparison method

Statistical Testing

- In most empirical STEM disciplines: statistical testing is the most common comparison method
- ML violates pre-requisites of many statistical tests

Statistical Testing

- In most empirical STEM disciplines: statistical testing is the most common comparison method
- ML violates pre-requisites of many statistical tests
  - independence of variables

Statistical Testing

- In most empirical STEM disciplines: statistical testing is the most common comparison method
- ML violates pre-requisites of many statistical tests
  - independence of variables
  - importance of learning curves instead of single points

Statistical Testing

- In most empirical STEM disciplines: statistical testing is the most common comparison method
- ML violates pre-requisites of many statistical tests
  - independence of variables
  - importance of learning curves instead of single points
  - True/false might not be the most interesting insight into a method

Statistical Testing

- In most empirical STEM disciplines: statistical testing is the most common comparison method
- ML violates pre-requisites of many statistical tests
  - independence of variables
  - importance of learning curves instead of single points
  - True/false might not be the most interesting insight into a method
- Tests that are useful to know for RL:

Statistical Testing

- In most empirical STEM disciplines: statistical testing is the most common comparison method
- ML violates pre-requisites of many statistical tests
  - independence of variables
  - importance of learning curves instead of single points
  - True/false might not be the most interesting insight into a method
- Tests that are useful to know for RL:
  - Mann-Whitney U-Test (probability of improvement of A over B)

Statistical Testing

- In most empirical STEM disciplines: statistical testing is the most common comparison method
- ML violates pre-requisites of many statistical tests
  - independence of variables
  - importance of learning curves instead of single points
  - True/false might not be the most interesting insight into a method
- Tests that are useful to know for RL:
  - Mann-Whitney U-Test (probability of improvement of A over B)
  - ANOVA (parametric, mean of A is different than mean of B)

Statistical Testing

- In most empirical STEM disciplines: statistical testing is the most common comparison method
- ML violates pre-requisites of many statistical tests
  - independence of variables
  - importance of learning curves instead of single points
  - True/false might not be the most interesting insight into a method
- Tests that are useful to know for RL:
  - Mann-Whitney U-Test (probability of improvement of A over B)
  - ANOVA (parametric, mean of A is different than mean of B)
  - Spearman rank correlation (rank of A is different than rank of B)

Statistical Testing

- In most empirical STEM disciplines: statistical testing is the most common comparison method
- ML violates pre-requisites of many statistical tests
  - independence of variables
  - importance of learning curves instead of single points
  - True/false might not be the most interesting insight into a method
- Tests that are useful to know for RL:
  - Mann-Whitney U-Test (probability of improvement of A over B)
  - ANOVA (parametric, mean of A is different than mean of B)
  - Spearman rank correlation (rank of A is different than rank of B)
- For many insights, we still have to lean on learning curve comparisons

Seeding & Randomness

Seeding & Randomness

- Like environments or tasks, randomness introduces variability to your results

Seeding & Randomness

- Like environments or tasks, randomness introduces variability to your results
- This is equivalent to another subject in e.g. a survey study

Seeding & Randomness

- Like environments or tasks, randomness introduces variability to your results
- This is equivalent to another subject in e.g. a survey study
- Like in any other domain: number of runs is very important to your results

Seeding & Randomness

- Like environments or tasks, randomness introduces variability to your results
- This is equivalent to another subject in e.g. a survey study
- Like in any other domain: number of runs is very important to your results
- It’s possible to hack statistical tests by choosing the “correct” number

Seeding & Randomness

- Like environments or tasks, randomness introduces variability to your results
- This is equivalent to another subject in e.g. a survey study
- Like in any other domain: number of runs is very important to your results
- It’s possible to hack statistical tests by choosing the “correct” number
- Cherry picking seeds (whether accidental or not) can skew results

Seeding & Randomness

- Like environments or tasks, randomness introduces variability to your results
- This is equivalent to another subject in e.g. a survey study
- Like in any other domain: number of runs is very important to your results
- It's possible to hack statistical tests by choosing the "correct" number
- Cherry picking seeds (whether accidental or not) can skew results
- This is likely the single most important factor in comparing RL algorithms

Seeding & Randomness

- Like environments or tasks, randomness introduces variability to your results
- This is equivalent to another subject in e.g. a survey study
- Like in any other domain: number of runs is very important to your results
- It’s possible to hack statistical tests by choosing the “correct” number
- Cherry picking seeds (whether accidental or not) can skew results
- This is likely the single most important factor in comparing RL algorithms

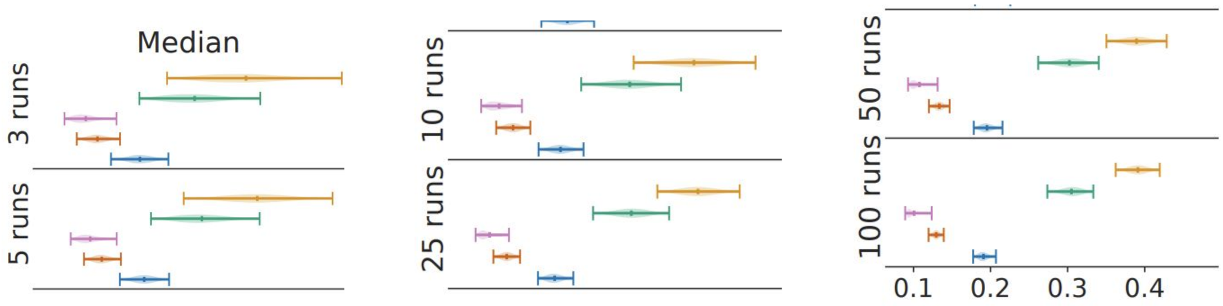

Figure: 95% CI of final performance of 5 RL algorithms on Atari [Agarwal et al. 2022]

Seeding & Randomness: What Is A Seed?

Seeding & Randomness: What Is A Seed?

- Seeding a random number generator means the pseudo-random sequence stays the same for every re-run

Seeding & Randomness: What Is A Seed?

- Seeding a random number generator means the pseudo-random sequence stays the same for every re-run
- Where can we find random number generators in RL?

Seeding & Randomness: What Is A Seed?

- Seeding a random number generator means the pseudo-random sequence stays the same for every re-run
- Where can we find random number generators in RL?
  - Action sampling in the policy
  - Transitions
  - SGD updates
  - (Mini-)batch sampling
  - ...

Seeding & Randomness: What Is A Seed?

- Seeding a random number generator means the pseudo-random sequence stays the same for every re-run
- Where can we find random number generators in RL?
  - Action sampling in the policy
  - Transitions
  - SGD updates
  - (Mini-)batch sampling
  - ...
- Common practice: seed everything the same

Seeding & Randomness: What Is A Seed?

- Seeding a random number generator means the pseudo-random sequence stays the same for every re-run
- Where can we find random number generators in RL?
  - Action sampling in the policy
  - Transitions
  - SGD updates
  - (Mini-)batch sampling
  - ...
- Common practice: seed everything the same
- Technically correct practice: use different seeds and more runs

Seeding & Randomness: What Is A Seed?

- Seeding a random number generator means the pseudo-random sequence stays the same for every re-run
- Where can we find random number generators in RL?
  - Action sampling in the policy
  - Transitions
  - SGD updates
  - (Mini-)batch sampling
  - ...
- Common practice: seed everything the same
- Technically correct practice: use different seeds and more runs
- Alternative: don’t seed at all (but results will differ on each re-run)

Seeding & Randomness: How Many Seeds?

1. Community standards: what are other people doing?

Seeding & Randomness: How Many Seeds?

1. Community standards: what are other people doing?

2. Separation: add runs until there is no overlap in uncertainty

Seeding & Randomness: How Many Seeds?

1. Community standards: what are other people doing?

2. Separation: add runs until there is no overlap in uncertainty

3. Power Analysis: run pre-study and determine how many runs you need

Seeding & Randomness: How Many Seeds?

1. Community standards: what are other people doing?
   - Pros: easy to do, direct comparison to prior work
   - Cons: might not produce reliable results
2. Separation: add runs until there is no overlap in uncertainty

3. Power Analysis: run pre-study and determine how many runs you need

Seeding & Randomness: How Many Seeds?

1. Community standards: what are other people doing?
   - Pros: easy to do, direct comparison to prior work
   - Cons: might not produce reliable results
2. Separation: add runs until there is no overlap in uncertainty
   - Pros: simple procedure, clear outcome
   - Cons: adding runs can be tedious, minimum number of runs required
3. Power Analysis: run pre-study and determine how many runs you need

Seeding & Randomness: How Many Seeds?

1. Community standards: what are other people doing?
   - Pros: easy to do, direct comparison to prior work
   - Cons: might not produce reliable results
2. Separation: add runs until there is no overlap in uncertainty
   - Pros: simple procedure, clear outcome
   - Cons: adding runs can be tedious, minimum number of runs required
3. Power Analysis: run pre-study and determine how many runs you need
   - Pros: statistically founded, principled approach
   - Cons: complex process, assumptions might be violated

ML/RL Checklists

- Many conferences have standards for papers

ML/RL Checklists

- Many conferences have standards for papers
  - Very broad, but good: NeurIPS

ML/RL Checklists

- Many conferences have standards for papers
  - Very broad, but good: NeurIPS
  - Stricter standards for reproducibility: AutoML Conf

ML/RL Checklists

- Many conferences have standards for papers
  - Very broad, but good: NeurIPS
  - Stricter standards for reproducibility: AutoML Conf
  - Proposal for the same in RL

ML/RL Checklists

- Many conferences have standards for papers
  - Very broad, but good: NeurIPS
  - Stricter standards for reproducibility: AutoML Conf
  - Proposal for the same in RL
  - Our extension for RL with HPO

ML/RL Checklists

- Many conferences have standards for papers
  - Very broad, but good: NeurIPS
  - Stricter standards for reproducibility: AutoML Conf
  - Proposal for the same in RL
  - Our extension for RL with HPO
- Good practices for first projects

ML/RL Checklists

- Many conferences have standards for papers
  - Very broad, but good: NeurIPS
  - Stricter standards for reproducibility: AutoML Conf
  - Proposal for the same in RL
  - Our extension for RL with HPO
- Good practices for first projects
- In case there is a good reason, it's okay to deviate from these

Practices I Recommend

- Never run less than 10 seeds

Practices I Recommend

- Never run less than 10 seeds
- Know why you treat seeds the way you do

Practices I Recommend

- Never run less than 10 seeds
- Know why you treat seeds the way you do
- Never just look at one metric, plot mean, IQM and median with CI over time

Practices I Recommend

- Never run less than 10 seeds
- Know why you treat seeds the way you do
- Never just look at one metric, plot mean, IQM and median with CI over time
- Quantify the drawbacks of your method (compute overhead? More time?)

Practices I Recommend

- Never run less than 10 seeds
- Know why you treat seeds the way you do
- Never just look at one metric, plot mean, IQM and median with CI over time
- Quantify the drawbacks of your method (compute overhead? More time?)
- Be sure that your hyperparameters and environment settings don’t become confounding factors (by e.g. making sure you know how the seeding works)

Practices I Recommend

- Never run less than 10 seeds
- Know why you treat seeds the way you do
- Never just look at one metric, plot mean, IQM and median with CI over time
- Quantify the drawbacks of your method (compute overhead? More time?)
- Be sure that your hyperparameters and environment settings don’t become confounding factors (by e.g. making sure you know how the seeding works)
- Don’t overtune only your method and use different seeds for reporting

Practices I Recommend

- Never run less than 10 seeds
- Know why you treat seeds the way you do
- Never just look at one metric, plot mean, IQM and median with CI over time
- Quantify the drawbacks of your method (compute overhead? More time?)
- Be sure that your hyperparameters and environment settings don’t become confounding factors (by e.g. making sure you know how the seeding works)
- Don’t overtune only your method and use different seeds for reporting
- Use tools to help you write good code: tests, linters, formaters, docstrings

Practices I Recommend

- Never run less than 10 seeds
- Know why you treat seeds the way you do
- Never just look at one metric, plot mean, IQM and median with CI over time
- Quantify the drawbacks of your method (compute overhead? More time?)
- Be sure that your hyperparameters and environment settings don’t become confounding factors (by e.g. making sure you know how the seeding works)
- Don’t overtune only your method and use different seeds for reporting
- Use tools to help you write good code: tests, linters, formaters, docstrings
- Open-source everything and collaborate on code

Practices I Recommend

- Never run less than 10 seeds
- Know why you treat seeds the way you do
- Never just look at one metric, plot mean, IQM and median with CI over time
- Quantify the drawbacks of your method (compute overhead? More time?)
- Be sure that your hyperparameters and environment settings don’t become confounding factors (by e.g. making sure you know how the seeding works)
- Don’t overtune only your method and use different seeds for reporting
- Use tools to help you write good code: tests, linters, formaters, docstrings
- Open-source everything and collaborate on code
- Whenever you can, use existing well-tested code

Practices I Recommend

- Never run less than 10 seeds
- Know why you treat seeds the way you do
- Never just look at one metric, plot mean, IQM and median with CI over time
- Quantify the drawbacks of your method (compute overhead? More time?)
- Be sure that your hyperparameters and environment settings don’t become confounding factors (by e.g. making sure you know how the seeding works)
- Don’t overtune only your method and use different seeds for reporting
- Use tools to help you write good code: tests, linters, formaters, docstrings
- Open-source everything and collaborate on code
- Whenever you can, use existing well-tested code
- Write down everything you do as well as you can

My Understanding of RL Standards

- ❑ I understand which factors can confound RL experiments
- ❑ I can describe a rough set of minimum standards experiments should follow
- ❑ I can give 1-2 examples of wrong result interpretations
- ❑ I understand how coding factors into RL reproducibility
- ❑ I know what factors to look out for when applying statistic test to RL results
- ❑ I have an overview of the different aspects of seeding in RL