Advanced Topics in Deep Reinforcement Learning

Hyperparameters

Theresa Eimer

What Is A Hyperparameter?

What Is A Hyperparameter?

Low-level design decision that contributes to training

What Is A Hyperparameter?

Low-level design decision that contributes to training

Examples:

- Learning rate
- Exploration epsilon
- Batch size
- Gradient clipping range
- ...

What Is A Hyperparameter?

Low-level design decision that contributes to training

Examples:

- Learning rate
- Exploration epsilon
- Batch size
- Gradient clipping range
- ...

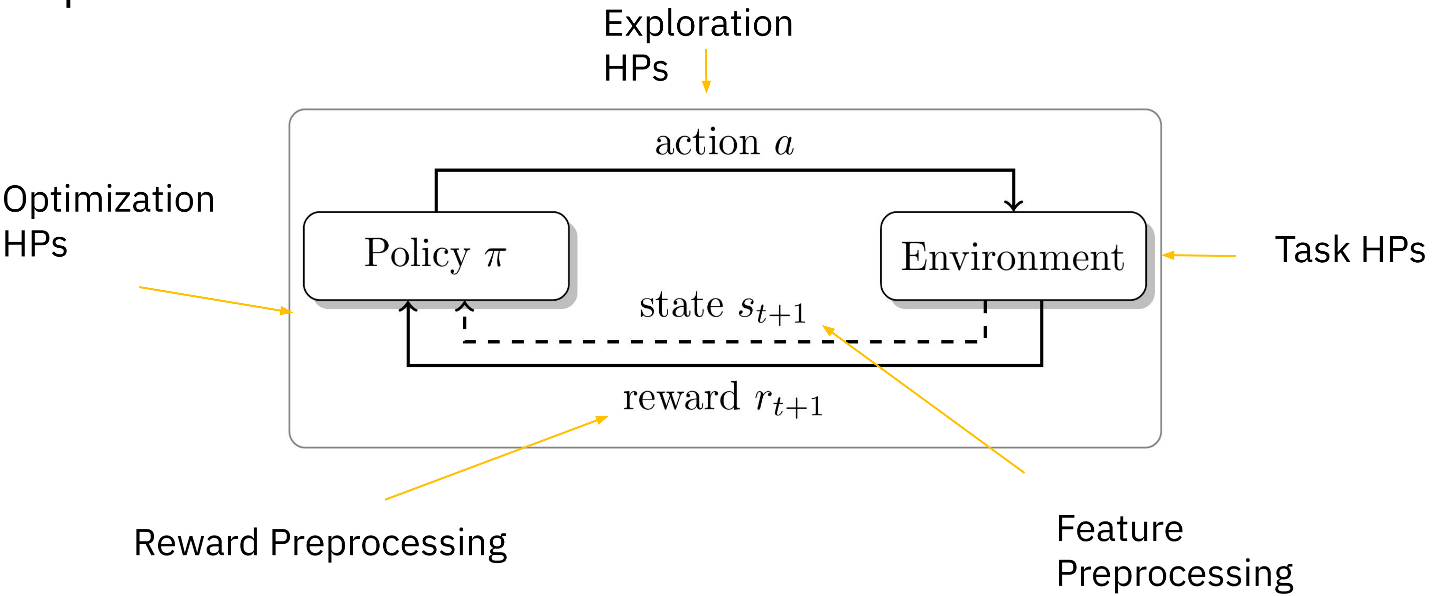
These are often integral to successful training

What Do Hyperparameters Do?

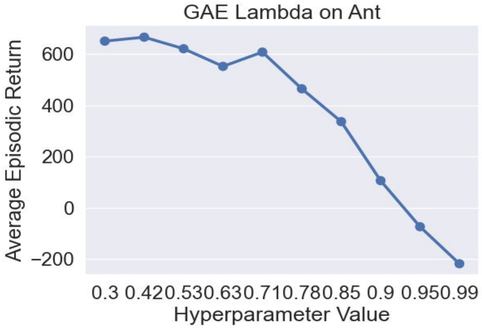
- Control exploration
- Shape environment interactions
- Manage Updates
- …

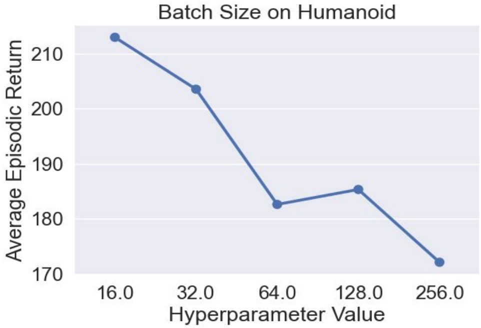
What Do Hyperparameters Do?

- Control exploration
- Shape environment interactions
- Manage Updates
- …



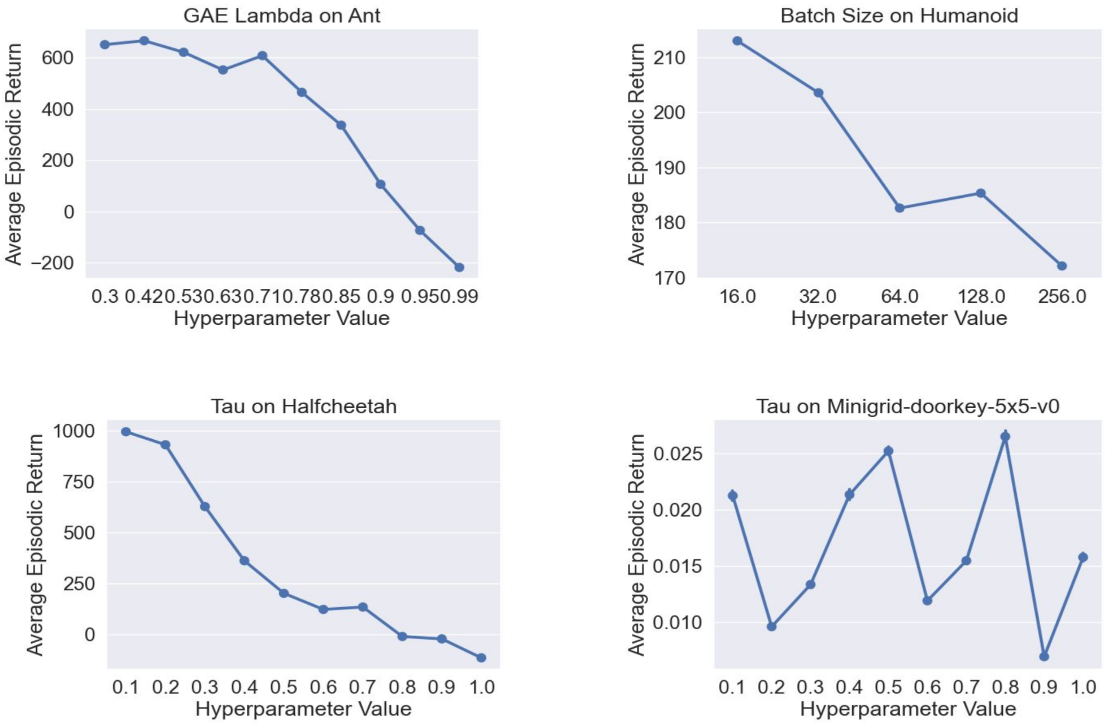
What Is A Hyperparameter?





Top: Sweeps on PPO Brax.

What Is A Hyperparameter?



Top: Sweeps on PPO Brax.
Bottom: Sweeps on DQN.
[Eimer et al. 2023]

Case Studies: Batch Size

Case Studies: Batch Size

- Batch size is usually not regard as an important RL hyperparameter

Case Studies: Batch Size

- Batch size is usually not regard as an important RL hyperparameter
- Compared to supervised learning: smaller batches between 32-256

Case Studies: Batch Size

- Batch size is usually not regard as an important RL hyperparameter
- Compared to supervised learning: smaller batches between 32-256
- But: very small batches can be helpful: [Obando-Ceron et al. 2023]

Case Studies: Batch Size

- Batch size is usually not regard as an important RL hyperparameter
- Compared to supervised learning: smaller batches between 32-256
- But: very small batches can be helpful: [Obando-Ceron et al. 2023]
  - Fewer plateaus
  - Supports exploration
  - Less wallclock time

Case Studies: Batch Size

- Batch size is usually not regard as an important RL hyperparameter
- Compared to supervised learning: smaller batches between 32-256
- But: very small batches can be helpful: [Obando-Ceron et al. 2023]
  - Fewer plateaus
  - Supports exploration
  - Less wallclock time
- This suggests batch size is much more complex in RL than supervised learning

Case Studies: Exploration Schedules

Case Studies: Exploration Schedules

- Likely we don’t want to explore the same amount at each point in training

Case Studies: Exploration Schedules

- Likely we don't want to explore the same amount at each point in training
- Can we schedule exploration even for simple exploration strategies?

Case Studies: Exploration Schedules

- Likely we don't want to explore the same amount at each point in training
- Can we schedule exploration even for simple exploration strategies?
- Common approach: linear epsilon schedules

Case Studies: Exploration Schedules

- Likely we don’t want to explore the same amount at each point in training
- Can we schedule exploration even for simple exploration strategies?
- Common approach: linear epsilon schedules
- Exploration decays from 1 to close to zero over a pre-determined amount of timesteps (often about 50% of training steps)

Case Studies: Exploration Schedules

- Likely we don't want to explore the same amount at each point in training
- Can we schedule exploration even for simple exploration strategies?
- Common approach: linear epsilon schedules
- Exploration decays from 1 to close to zero over a pre-determined amount of timesteps (often about 50% of training steps)
- Different decay curves are possible, e.g. exponential decay

Case Studies: Exploration Schedules

- Likely we don't want to explore the same amount at each point in training
- Can we schedule exploration even for simple exploration strategies?
- Common approach: linear epsilon schedules
- Exploration decays from 1 to close to zero over a pre-determined amount of timesteps (often about 50% of training steps)
- Different decay curves are possible, e.g. exponential decay
- This makes simple exploration a lot more capable

Meta-Algorithmics

Meta-Algorithmics

Examples:

- Hyperparameter Optimization
- Algorithm Selection
- Landscape analysis
- Meta-learning initializations, algorithms, etc.
- Task ordering
- ...

AutoRL [Parker-Holder et al. 2022]

If we want to optimize anything that goes into an RL pipeline, we’re doing AutoRL.

AutoRL [Parker-Holder et al. 2022]

If we want to optimize anything that goes into an RL pipeline, we're doing AutoRL.

Formally, if RL finds a policy θ s.t.:

$$\max_\theta J(\theta, \zeta) \text{ where } J(\theta, \zeta) = E\,[\, \Sigma_{t >= 0}\, \gamma^t\, r_t]\,]$$

AutoRL [Parker-Holder et al. 2022]

If we want to optimize anything that goes into an RL pipeline, we're doing AutoRL.

Formally, if RL finds a policy θ s.t.:

$$\max_\theta J(\theta, \zeta) \text{ where } J(\theta, \zeta) = E\left[\, \Sigma_{t >= 0}\, \gamma^t r_t \right]$$

Then AutoRL can be defined as a function f optimizing ζ:

$$\max_\zeta f(\zeta, \theta^*) \text{ s.t. } \theta^* \text{ in } \operatorname{argmax}_\theta J(\theta, \zeta)$$

AutoRL [Parker-Holder et al. 2022]

If we want to optimize anything that goes into an RL pipeline, we're doing AutoRL.

Formally, if RL finds a policy θ s.t.:

$$\max_\theta J(\theta, \zeta) \text{ where } J(\theta, \zeta) = E\left[\, \Sigma_{t>=0}\, \gamma^t r_t \right]$$

Then AutoRL can be defined as a function f optimizing ζ:

$$\max_\zeta f(\zeta, \theta^*) \text{ s.t. } \theta^* \text{ in } \operatorname{argmax}_\theta J(\theta, \zeta)$$

Best policy found by RL algorithm

AutoRL [Parker-Holder et al. 2022]

If we want to optimize anything that goes into an RL pipeline, we're doing AutoRL.

Formally, if RL finds a policy θ s.t.:

$$\max_\theta J(\theta, \zeta) \text{ where } J(\theta, \zeta) = E\left[\sum_{t>=0} \gamma^t r_t\right]$$

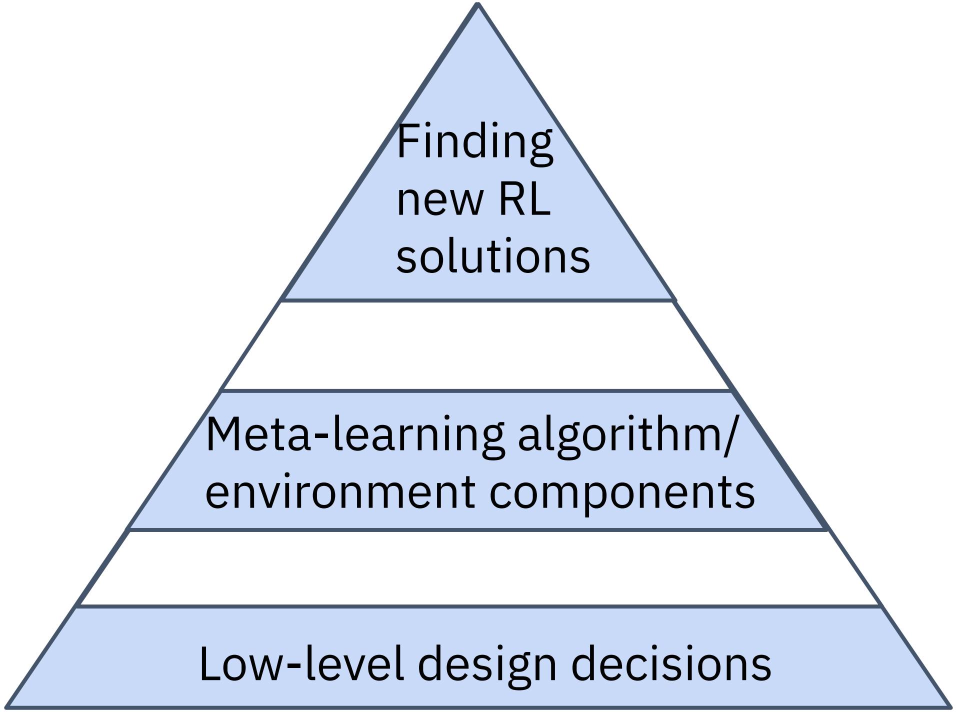
Then AutoRL can be defined as a function f optimizing ζ:

$$\max_\zeta f(\zeta, \theta^*) \text{ s.t. } \theta^* \text{ in } \operatorname{argmax}_\theta J(\theta, \zeta)$$

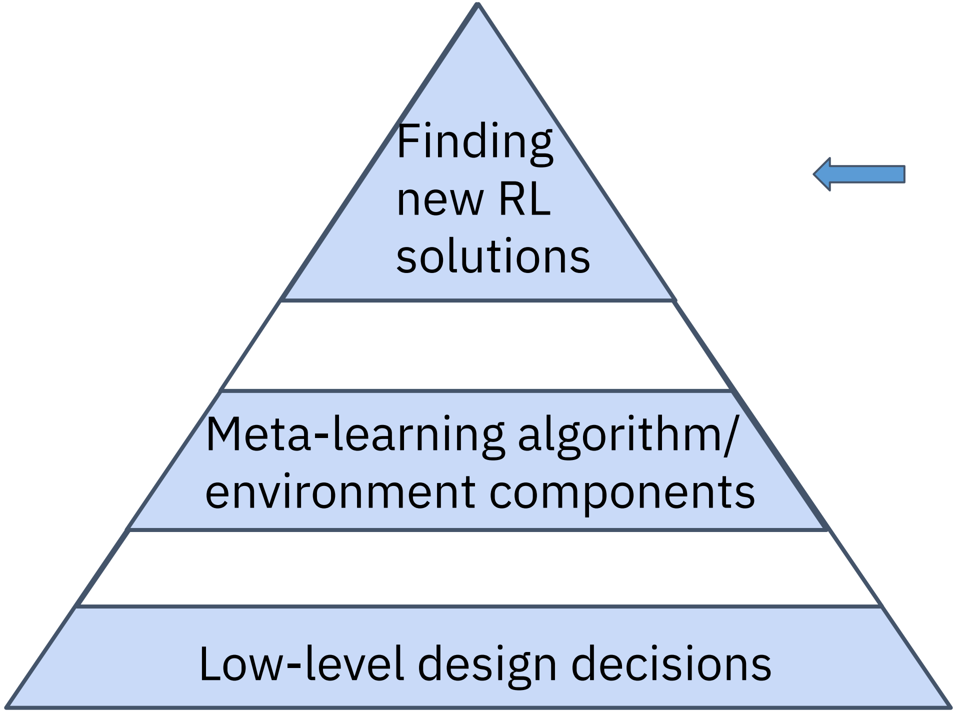
Hyperparameters, Algorithms, Tasks, etc.

Best policy found by RL algorithm

AutoRL

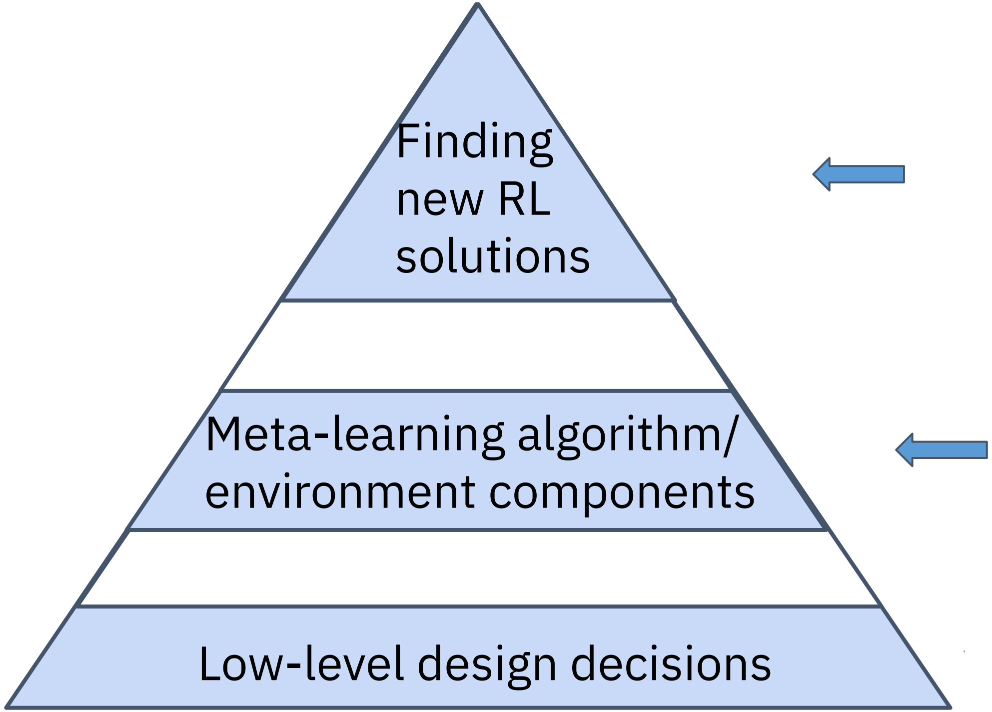

Finding new RL solutions
Meta-learning algorithm/ environment components
Low-level design decisions

AutoRL



Completely new algorithms

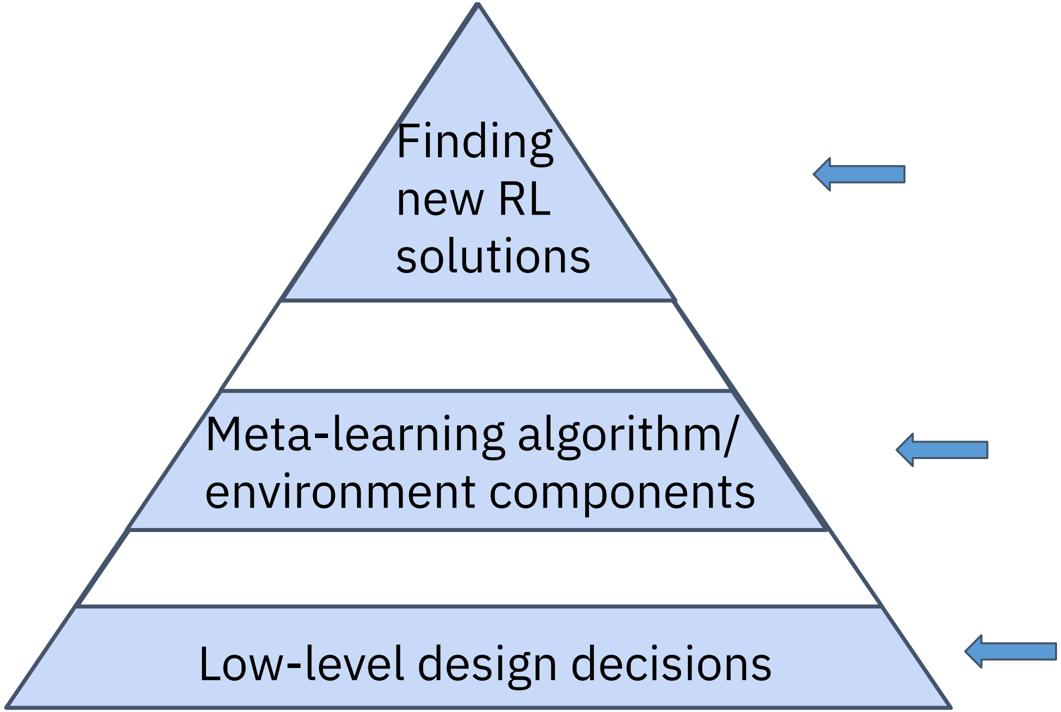
AutoRL



Completely new algorithms

Automatic reward shaping, learned exploration, ...

AutoRL



Completely new algorithms

Automatic reward shaping, learned exploration, ...

Hyperparameters, Task Ordering, ...

Examples of AutoRL Methods

- HPO via Population-based Training
- RL pipeline configuration with ARLO
- Curriculum learning via PLR
- Online exploration adjustment with Bootstrapped Meta-Gradients
- Meta-learned objective functions
- Evolved RL algorithms
- …

Hyperparameter Optimization: Basics

- We try to find a configuration that gives us the best outcome

Hyperparameter Optimization: Basics

- We try to find a configuration that gives us the best outcome
- Outcome is most commonly defined by final evaluation reward, alternatives:
  - final reward
  - environment steps until solution
  - time to solution
  - deviation between seeds
  - …

Hyperparameter Optimization: Basics

- We try to find a configuration that gives us the best outcome
- Outcome is most commonly defined by final evaluation reward, alternatives:
  - final reward
  - environment steps until solution
  - time to solution
  - deviation between seeds
  - ...
- Multiple objectives are possible

Hyperparameter Optimization: Basics

- We try to find a configuration that gives us the best outcome
- Outcome is most commonly defined by final evaluation reward, alternatives:
  - final reward
  - environment steps until solution
  - time to solution
  - deviation between seeds
  - ...
- Multiple objectives are possible
- Manual search means trying combinations until we find a working solution

Hyperparameter Optimization: Basics

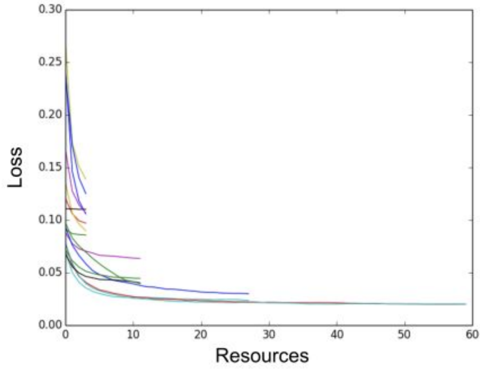
- We try to find a configuration that gives us the best outcome
- Outcome is most commonly defined by final evaluation reward, alternatives:
  - final reward
  - environment steps until solution
  - time to solution
  - deviation between seeds
  - ...
- Multiple objectives are possible
- Manual search means trying combinations until we find a working solution
- Optimizers can do this automatically and fairly well

Hyperparameter Optimization: Basics

- We try to find a configuration that gives us the best outcome
- Outcome is most commonly defined by final evaluation reward, alternatives:
  - final reward
  - environment steps until solution
  - time to solution
  - deviation between seeds
  - ...
- Multiple objectives are possible
- Manual search means trying combinations until we find a working solution
- Optimizers can do this automatically and fairly well
- Good approach: BO-based optimizer with multi-fidelity scheduling

Hyperparameter Optimization: Basics

- We try to find a configuration that gives us the best outcome
- Outcome is most commonly defined by final evaluation reward, alternatives:
  - final reward
  - environment steps until solution
  - time to solution
  - deviation between seeds
  - ...
- Multiple objectives are possible
- Manual search means trying combinations until we find a working solution
- Optimizers can do this automatically and fairly well
- Good approach: BO-based optimizer with multi-fidelity scheduling



Resource allocation in Hyperband [Li et al. 2018]

Considerations For RL

Before applying HPO to RL, we have to consider a few RL-specific issues in:

- Performance estimation
- Search spaces
- Objectives
- Dynamism
- Generalization

Considerations For RL: Performance estimates

Considerations For RL: Performance estimates

- Performance estimation is hard for most meta-algorithmic approaches

Considerations For RL: Performance estimates

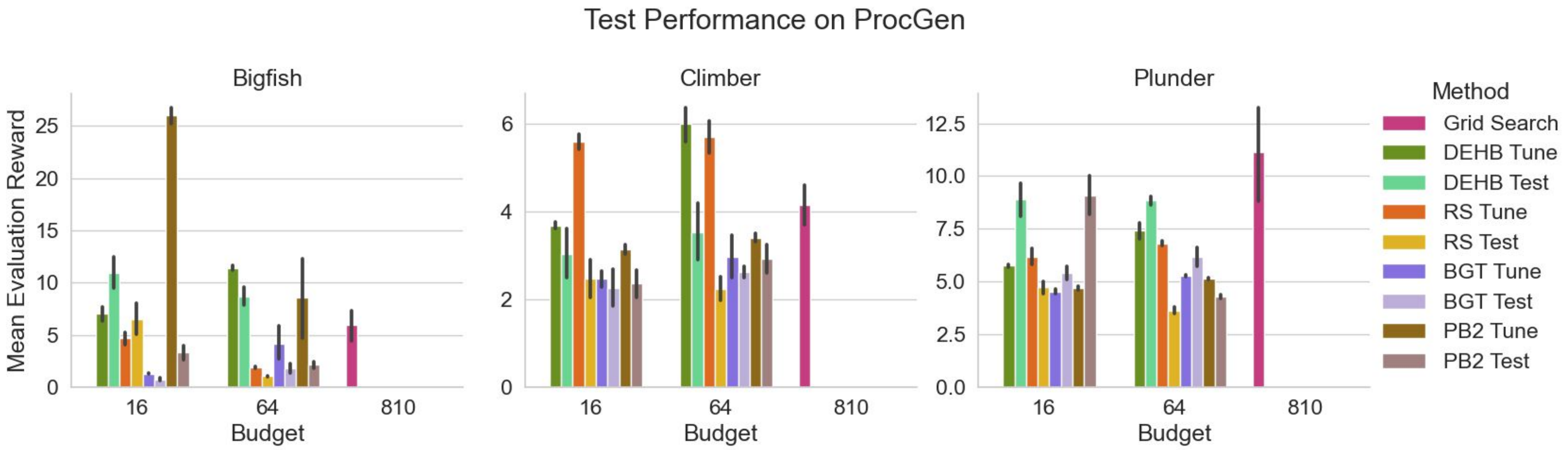
- Performance estimation is hard for most meta-algorithmic approaches
- Usually: try to estimate the performance of a configuration spending as little time and compute as possible

Considerations For RL: Performance estimates

- Performance estimation is hard for most meta-algorithmic approaches
- Usually: try to estimate the performance of a configuration spending as little time and compute as possible
- In RL, we want to estimate the average performance across seeds

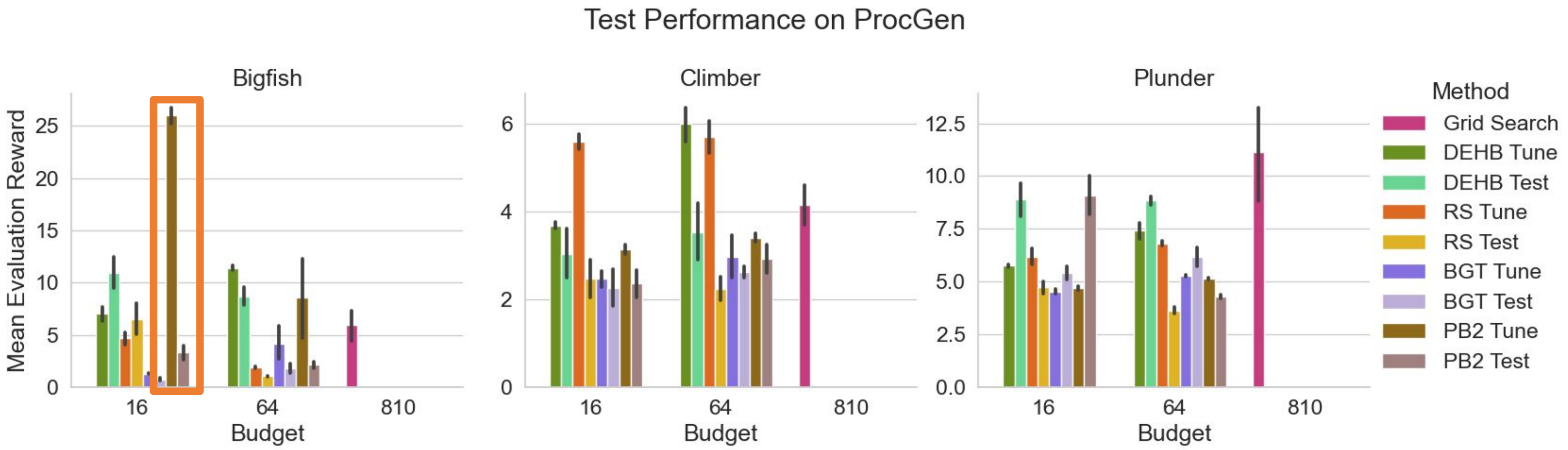
Considerations For RL: Performance estimates

- Performance estimation is hard for most meta-algorithmic approaches
- Usually: try to estimate the performance of a configuration spending as little time and compute as possible
- In RL, we want to estimate the average performance across seeds



Considerations For RL: Performance estimates

- Performance estimation is hard for most meta-algorithmic approaches
- Usually: try to estimate the performance of a configuration spending as little time and compute as possible
- In RL, we want to estimate the average performance across seeds



Considerations For RL: Search Spaces

Considerations For RL: Search Spaces

- Search spaces define which hyperparameters we adapt and which values they can take

Considerations For RL: Search Spaces

- Search spaces define which hyperparameters we adapt and which values they can take
- RL algorithms have many relevant hyperparameters [Eimer et al. 2023]

Considerations For RL: Search Spaces

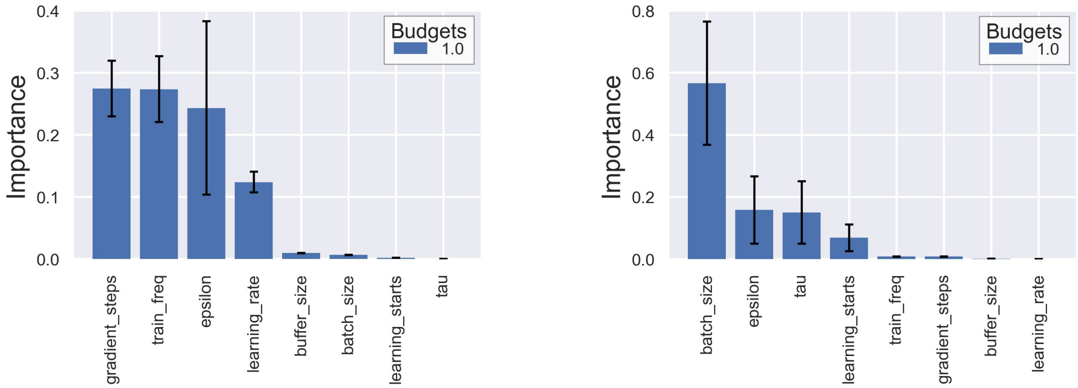
- Search spaces define which hyperparameters we adapt and which values they can take
- RL algorithms have many relevant hyperparameters [Eimer et al. 2023]
- The importance of individual hyperparameters depends on the environment

Considerations For RL: Search Spaces

- Search spaces define which hyperparameters we adapt and which values they can take
- RL algorithms have many relevant hyperparameters [Eimer et al. 2023]
- The importance of individual hyperparameters depends on the environment
- Using narrow search spaces requires prior studies or expert knowledge

Considerations For RL: Search Spaces

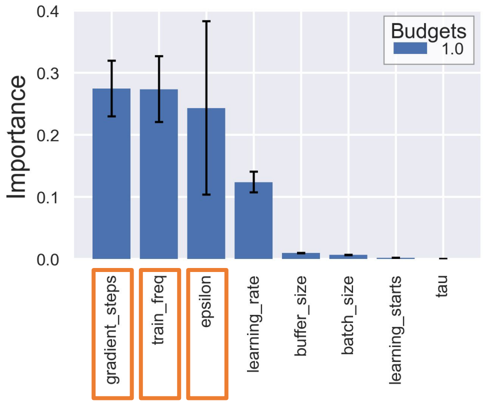
- Search spaces define which hyperparameters we adapt and which values they can take
- RL algorithms have many relevant hyperparameters [Eimer et al. 2023]
- The importance of individual hyperparameters depends on the environment
- Using narrow search spaces requires prior studies or expert knowledge

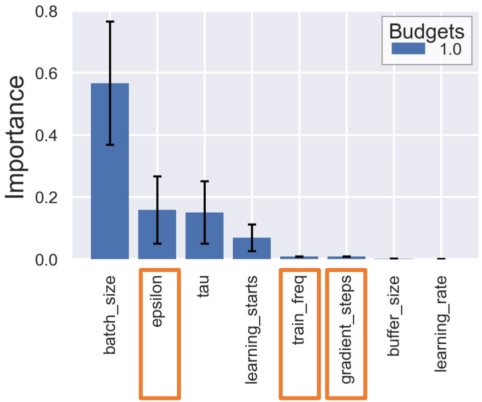


Left: DQN HP importance on Acrobot.
Right: DQN HP importance on MiniGrid 5x5.

Considerations For RL: Search Spaces

- Search spaces define which hyperparameters we adapt and which values they can take
- RL algorithms have many relevant hyperparameters [Eimer et al. 2023]
- The importance of individual hyperparameters depends on the environment
- Using narrow search spaces requires prior studies or expert knowledge

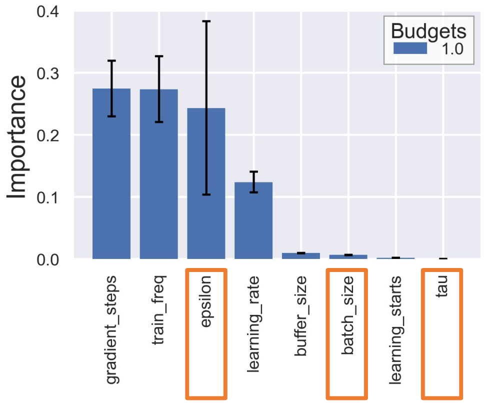


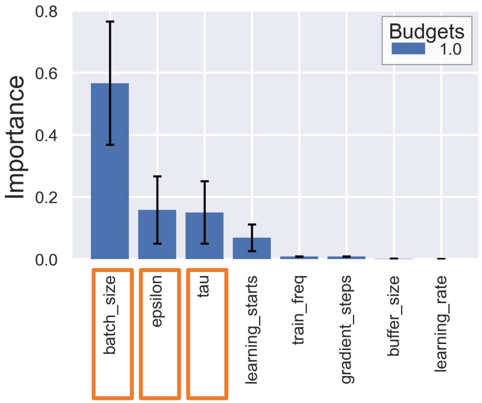


Left: DQN HP importance on Acrobot.
Right: DQN HP importance on MiniGrid 5x5.

Considerations For RL: Search Spaces

- Search spaces define which hyperparameters we adapt and which values they can take
- RL algorithms have many relevant hyperparameters [Eimer et al. 2023]
- The importance of individual hyperparameters depends on the environment
- Using narrow search spaces requires prior studies or expert knowledge





Left: DQN HP importance on Acrobot.
Right: DQN HP importance on MiniGrid 5x5.

Considerations For RL: Objectives

- What do we want to accomplish?

Considerations For RL: Objectives

- What do we want to accomplish?
  - Best solution?
  - Best result?
  - Best anytime performance?
  - Best stability in training?
  - Best wallclock time?
  - Best resource efficiency?

Considerations For RL: Objectives

- What do we want to accomplish?
  - Best solution?
  - Best result?
  - Best anytime performance?
  - Best stability in training?
  - Best wallclock time?
  - Best resource efficiency?
- We can’t always reliably measure these

Considerations For RL: Objectives

- What do we want to accomplish?
  - Best solution? => best anytime evaluation reward
  - Best result?
  - Best anytime performance?
  - Best stability in training?
  - Best wallclock time?
  - Best resource efficiency?
- We can’t always reliably measure these

Considerations For RL: Objectives

- What do we want to accomplish?
  - Best solution? => best anytime evaluation reward
  - Best result? => best final evaluation reward
  - Best anytime performance?
  - Best stability in training?
  - Best wallclock time?
  - Best resource efficiency?
- We can’t always reliably measure these

Considerations For RL: Objectives

- What do we want to accomplish?
  - Best solution? => best anytime evaluation reward
  - Best result? => best final evaluation reward
  - Best anytime performance? => training reward AUC
  - Best stability in training?
  - Best wallclock time?
  - Best resource efficiency?
- We can’t always reliably measure these

Considerations For RL: Objectives

- What do we want to accomplish?
  - Best solution? => best anytime evaluation reward
  - Best result? => best final evaluation reward
  - Best anytime performance? => training reward AUC
  - Best stability in training? => deviation between seeds
  - Best wallclock time?
  - Best resource efficiency?
- We can’t always reliably measure these

Considerations For RL: Objectives

- What do we want to accomplish?
  - Best solution? => best anytime evaluation reward
  - Best result? => best final evaluation reward
  - Best anytime performance? => training reward AUC
  - Best stability in training? => deviation between seeds
  - Best wallclock time? => time measurement
  - Best resource efficiency?
- We can't always reliably measure these

Considerations For RL: Objectives

- What do we want to accomplish?
  - Best solution? => best anytime evaluation reward
  - Best result? => best final evaluation reward
  - Best anytime performance? => training reward AUC
  - Best stability in training? => deviation between seeds
  - Best wallclock time? => time measurement
  - Best resource efficiency? => resource measurements
- We can’t always reliably measure these

Considerations For RL: Objectives

- What do we want to accomplish?
  - Best solution? => best anytime evaluation reward
  - Best result? => best final evaluation reward
  - Best anytime performance? => training reward AUC
  - Best stability in training? => deviation between seeds
  - Best wallclock time? => time measurement
  - Best resource efficiency? => resource measurements
- We can’t always reliably measure these
- In theory, we can use multiple objectives, but then have to choose a solution from the solution pareto front

Considerations For RL: Objectives

- What do we want to accomplish?
  - Best solution? => best anytime evaluation reward
  - Best result? => best final evaluation reward
  - Best anytime performance? => training reward AUC
  - Best stability in training? => deviation between seeds
  - Best wallclock time? => time measurement
  - Best resource efficiency? => resource measurements
- We can't always reliably measure these
- In theory, we can use multiple objectives, but then have to choose a solution from the solution pareto front
- Final evaluation reward is by far the most common objective in RL so far

Considerations For RL: Dynamism

Considerations For RL: Dynamism

- As we have seen, RL is very dynamic

Considerations For RL: Dynamism

- As we have seen, RL is very dynamic
- Very likely we want to adapt at least some hyperparameters over time

Considerations For RL: Dynamism

- As we have seen, RL is very dynamic
- Very likely we want to adapt at least some hyperparameters over time
- Example: linear schedules for epsilon exploration

Considerations For RL: Dynamism

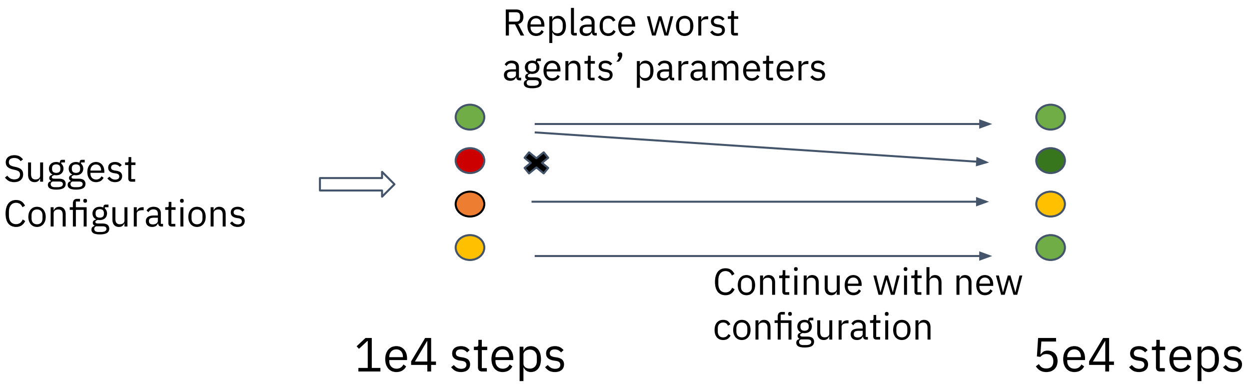
- As we have seen, RL is very dynamic
- Very likely we want to adapt at least some hyperparameters over time
- Example: linear schedules for epsilon exploration
- But: making schedules adaptive is very hard! [Adriaensen et al. 2022]

Considerations For RL: Dynamism

- As we have seen, RL is very dynamic
- Very likely we want to adapt at least some hyperparameters over time
- Example: linear schedules for epsilon exploration
- But: making schedules adaptive is very hard! [Adriaensen et al. 2022]
- Alternative: finding schedules via HPO (e.g. PBT)

Considerations For RL: Dynamism

- As we have seen, RL is very dynamic
- Very likely we want to adapt at least some hyperparameters over time
- Example: linear schedules for epsilon exploration
- But: making schedules adaptive is very hard! [Adriaensen et al. 2022]
- Alternative: finding schedules via HPO (e.g. PBT)



Considerations For RL: Generalization

Considerations For RL: Generalization

- Two level problem: finding general policies and general hyperparameters

Considerations For RL: Generalization

- Two level problem: finding general policies and general hyperparameters
- These intersect: finding hyperparameters for generalization can be harder

Considerations For RL: Generalization

- Two level problem: finding general policies and general hyperparameters
- These intersect: finding hyperparameters for generalization can be harder
- Since settings require different hyperparameter settings: one size fits all is unlikely to work in RL

Considerations For RL: Generalization

- Two level problem: finding general policies and general hyperparameters
- These intersect: finding hyperparameters for generalization can be harder
- Since settings require different hyperparameter settings: one size fits all is unlikely to work in RL
- Learned approaches can take the task setting into account

Considerations For RL: Generalization

- Two level problem: finding general policies and general hyperparameters
- These intersect: finding hyperparameters for generalization can be harder
- Since settings require different hyperparameter settings: one size fits all is unlikely to work in RL
- Learned approaches can take the task setting into account
- Best examples for RL: meta-gradient methods [Xu et al. 2018]

Meta-Gradients in RL

Hyperparameters η

Performance t

Meta-Gradients in RL

Hyperparameters η

Train

Performance t

Meta-Gradients in RL

Hyperparameters η

Train

Performance t

Performance t+1

Meta-Gradients in RL

Hyperparameters η

Update η via SGD using a differentiable meta-objective

Train

Performance t

Performance t+1

Meta-Gradients in RL

Hyperparameters η

Train

Performance t

Performance t+1

Update η via SGD using a differentiable meta-objective

=> Highly adaptive hyperparameter schedule learned during training

My Understanding of Hyperparameters in RL

- ❑ I understand what hyperparameters do
- ❑ I can name a few RL hyperparameters
- ❑ I understand the difficulties in performance estimation for RL

- ❑ I can describe AutoRL & give some examples
- ❑ I know at least 3 considerations for HPO in RL

- ❑ I can 1-2 HPO approaches for RL
- ❑ I can discuss which considerations we need for HPO in RL
- ❑ I can propose a simple HPO pipeline