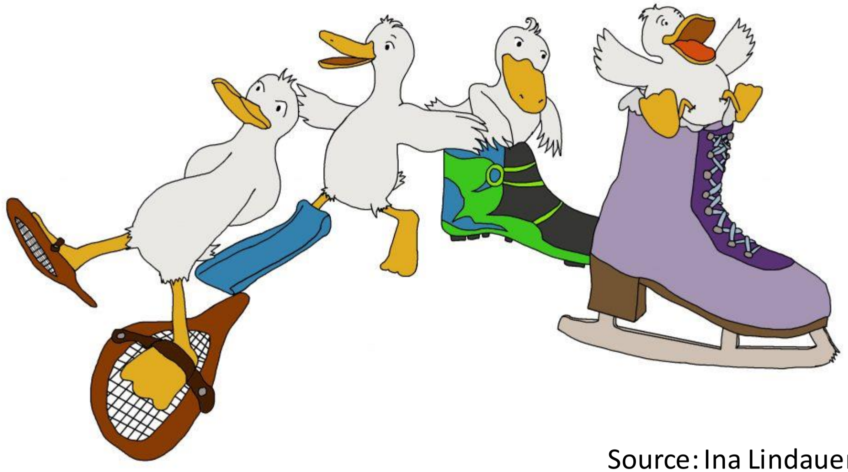
Institute of Artificial Intelligence

Leibniz Universität Hannover

DAC - Dynamic Algorithm Configuration



Source: Ina Lindauer

What is DAC?

- DAC is an algorithm configuration paradigm
- Task: find the best hyperparameter value for each instance **at each timestep**
- Goal: increase algorithm performance and efficiency
- Generalization of Algorithm Configuration (AC) and Per-Instance Algorithm Configuration (PIAC)
- Can be modelled as a sequential decision making problem

Why Configure Dynamically?

AC tools are already good at finding suitable hyperparameter configurations, But:

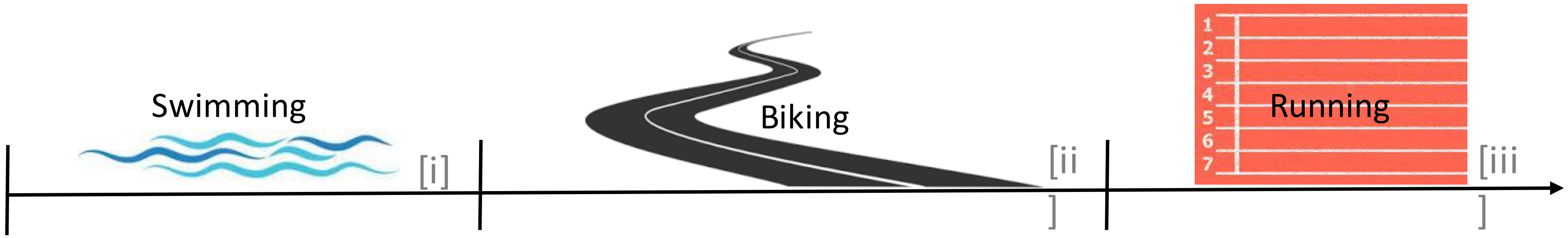
- In many algorithms, the role of hyperparameters shift during training
  Example: exploration hyperparameters in RL
- Optimal hyperparameter values can correspond to the algorithm's progress
  Example: learning rate in ML algorithms
- Both of these can vary between instances
  Example: CNN on MNIST or on ImageNet

Dynamic Configuration: Triathlons

- Three disciplines, each with its own equipment
- Goal: fastest time possible
- Disciplines: Swimming, Running, Biking
- Equipment: 👟 🚲 🥽

Dynamic Configuration: Triathlons

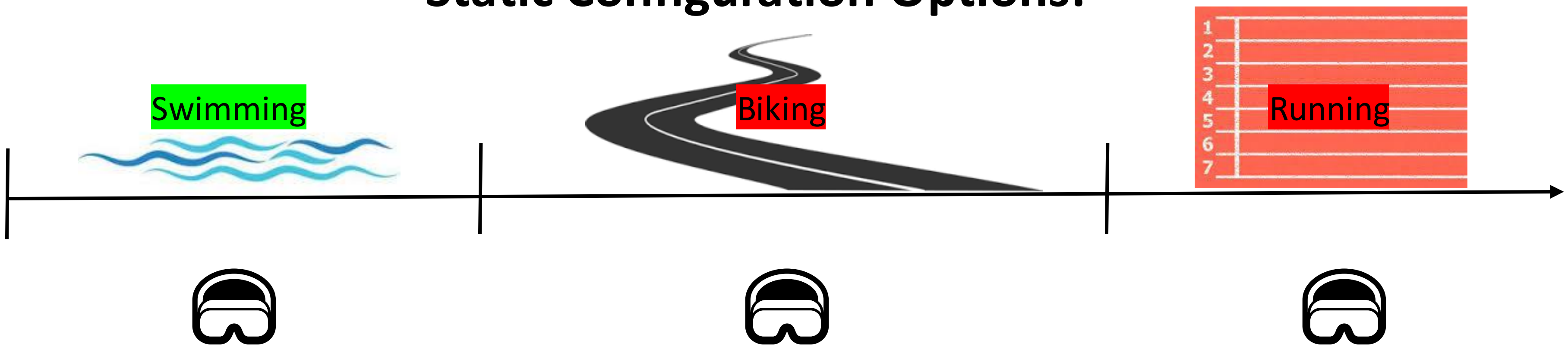
- Three disciplines, each with its own equipment
- Goal: fastest time possible
- Disciplines: Swimming, Running, Biking
- Equipment: 👟 🚲 🥽



Dynamic Configuration: Triathlons

- Three disciplines, each with its own equipment
- Goal: fastest time possible
- Disciplines: Swimming, Running, Biking
- Equipment: 👟 🚴 🥽

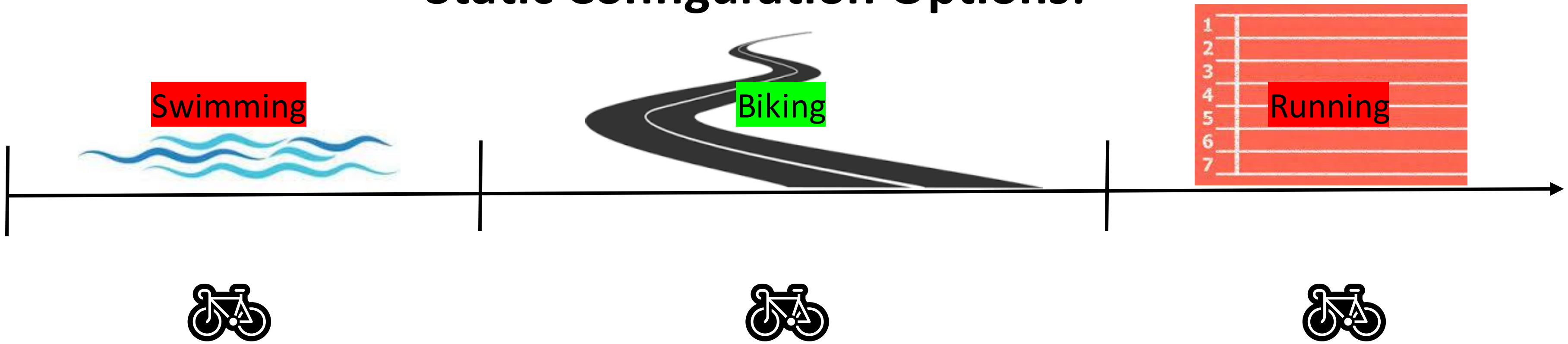
Static Configuration Options:



Dynamic Configuration: Triathlons

- Three disciplines, each with its own equipment
- Goal: fastest time possible
- Disciplines: Swimming, Running, Biking
- Equipment: 

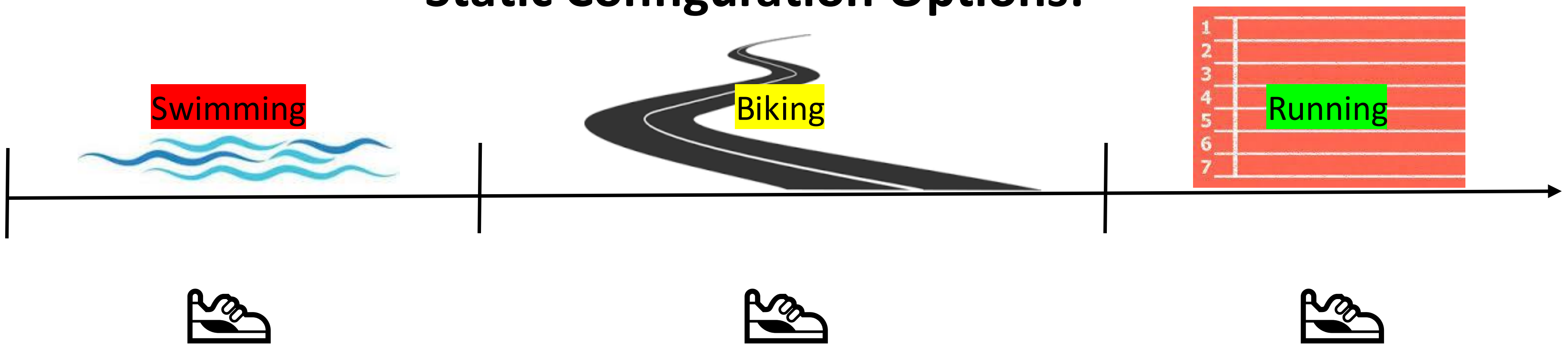
Static Configuration Options:



Dynamic Configuration: Triathlons

- Three disciplines, each with its own equipment
- Goal: fastest time possible
- Disciplines: Swimming, Running, Biking
- Equipment: 👟 🚴 🥽

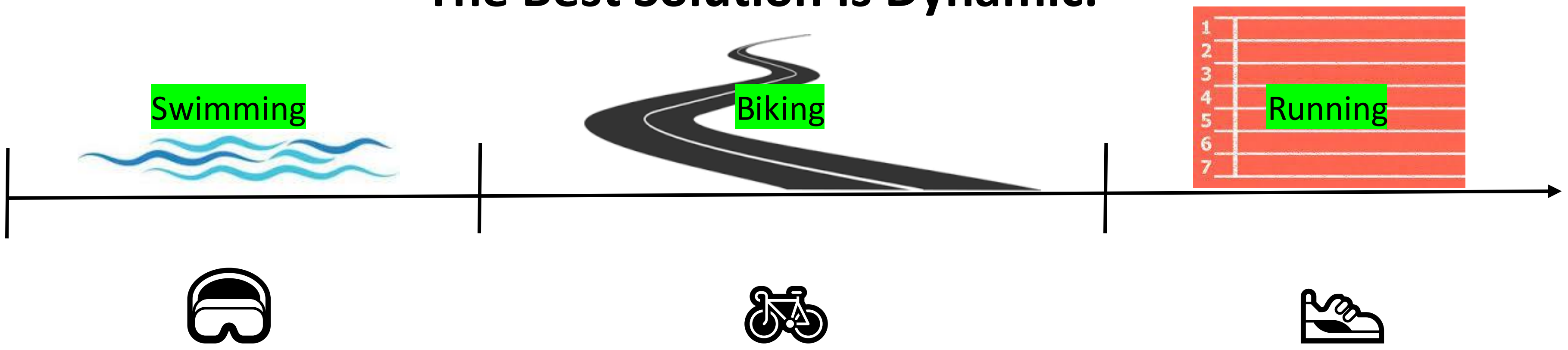
Static Configuration Options:



Dynamic Configuration: Triathlons

- Three disciplines, each with its own equipment
- Goal: fastest time possible
- Disciplines: Swimming, Running, Biking
- Equipment: 👟 🚴 🥽

The Best Solution is Dynamic:



Dynamic Configuration: Triathlons

- While their order is fixed, the length of each discipline can vary
- Thus: there's no one-size-fits all solutions
- Different triathlons can only be solved by accounting for their length
- Each triathlon length in this example is an **instance** of a triathlon
- We can also imagine future instances with different orders of the disciplines or even repeating disciplines

Dynamic Configuration in Machine Learning

- Initially: high learning rate to efficiently traverse loss landscape
- Once minimum is found: decrease learning rate to continue descending
- Possibly helpful: learning rate spikes during training to find global minimum

TODO: make plots for this

Overview

1. What is DAC?
2. Example: AC solvers for DAC
3. Example: DAC by Reinforcement Learning
4. The State of DAC & Open Questions

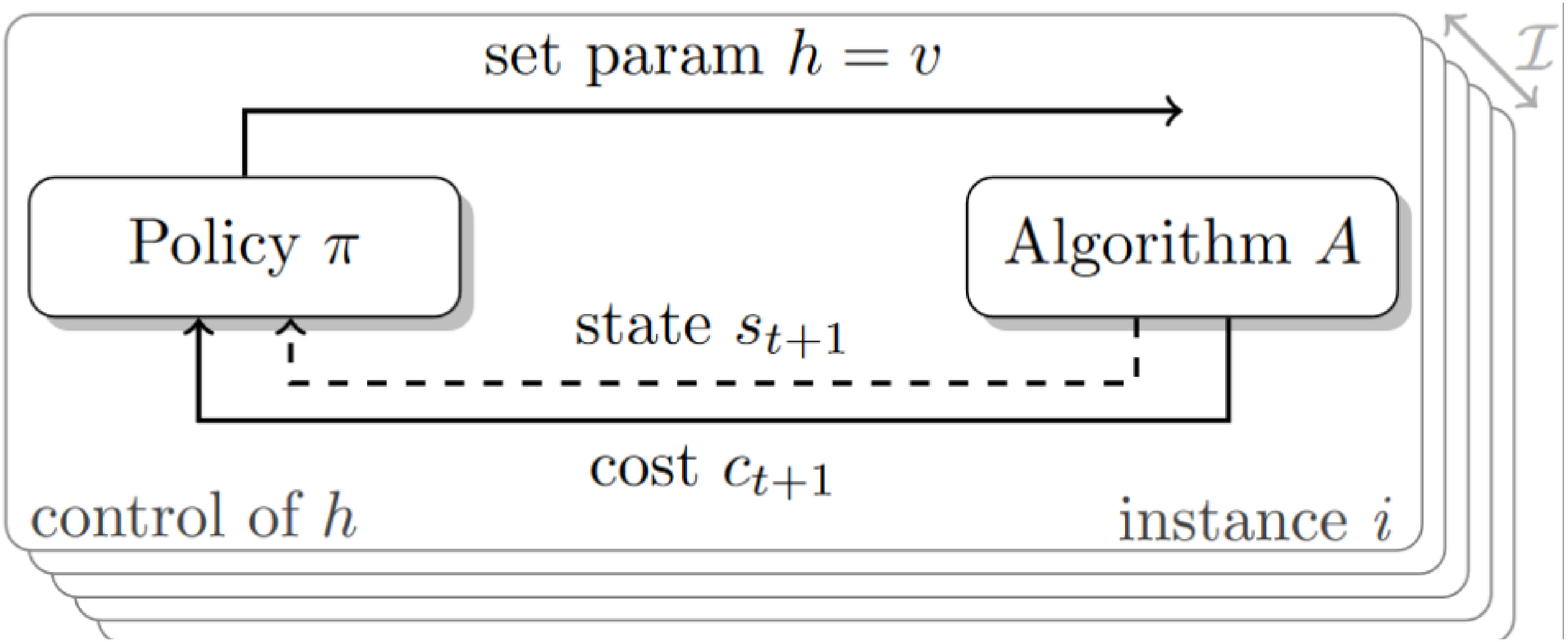
DAC Defined

Given:

- An algorithm $\mathcal{A}$ with configuration space Θ
- A distribution $\mathcal{D}$ over a target problem instances with domain $\mathcal{I}$
- A space of dynamic configuration policies Π with $\pi: \mathcal{S} \times \mathcal{I} \rightarrow \Theta$
- A cost metric $c: \Pi \times \mathcal{I} \rightarrow R$

Find $\boldsymbol{\pi}^* \in \mathrm{argmin}_{\pi \in \Pi} \, E_{i \sim \mathcal{D}} \, c(\pi, i)$ [Adriaensen et al. 2022]

DAC In Practice



AC Solvers for DAC

- DAC can be solved using classical AC solvers
- Idea: search for a hyperparameter per pre-set time interval
- Downside: tradeoff between number of intervals and search space size
- Upside: there are fairly sophisticated AC solvers that perform very well on many different tasks

Example: SMAC for Dynamic Learning Rates

- Setting: a set of different CNNs on MNIST and CIFAR10
- Task: control the learning rate to minimize prediction error
- SMAC [Lindauer et al. 2022] with Bayesian Optimization and Hyperband is used
- DAC outperforms the best static learning rate in almost all cases [Adriaensen et al. 2022]

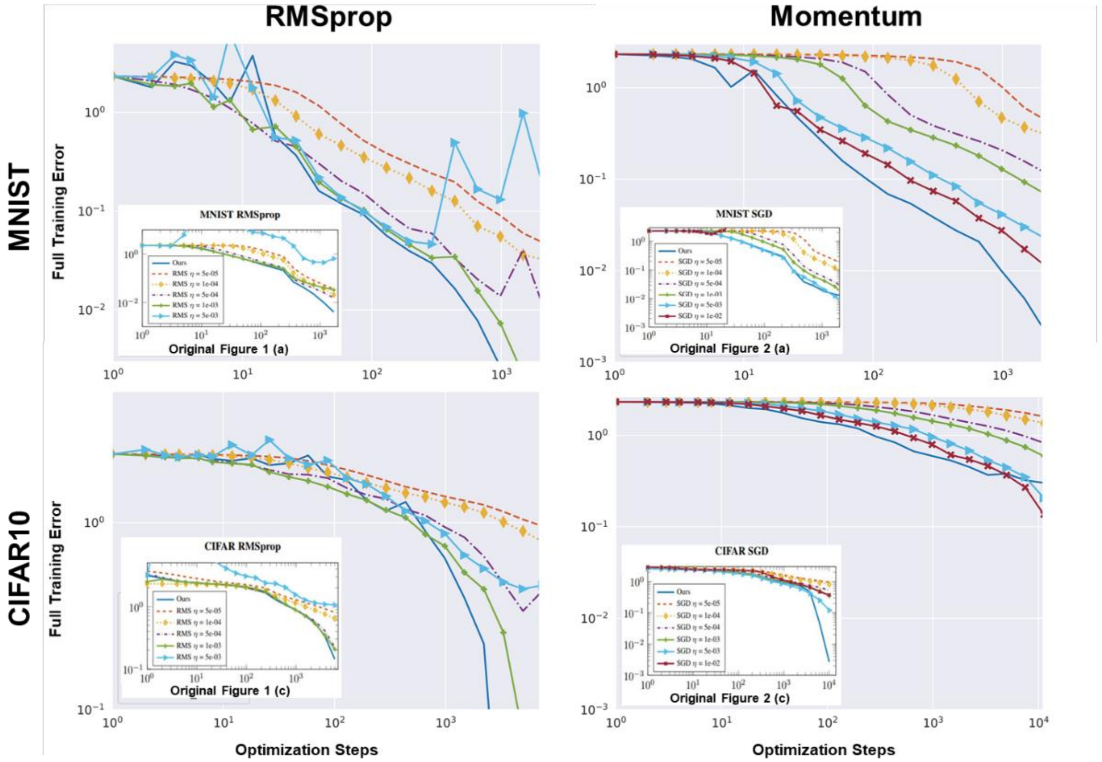


Figure: SMAC (dark blue solid) and static LR values on different optimizers

DAC by Reinforcement Learning

- As DAC can be modelled as a sequential decision problem, we can solve it using Reinforcement Learning
- Each action the agent takes changes the hyperparameter
- Downside: RL is often unreliable, scaling to multiple hyperparameters is hard
- Upside: search space is independent of the number of hyperparameter changes, generalization can be easier than with classical AC solvers

Example: GPS for CMA-ES

- Setting: CMA-ES of different functions
- Task: control the step size to minimize the current function
- RL agent learns from dynamic standard heuristic using GPS [Levine & Koltun 2013]
- While DAC needs training time, it beats even the static oracle [Adriaensen et al. 2022]

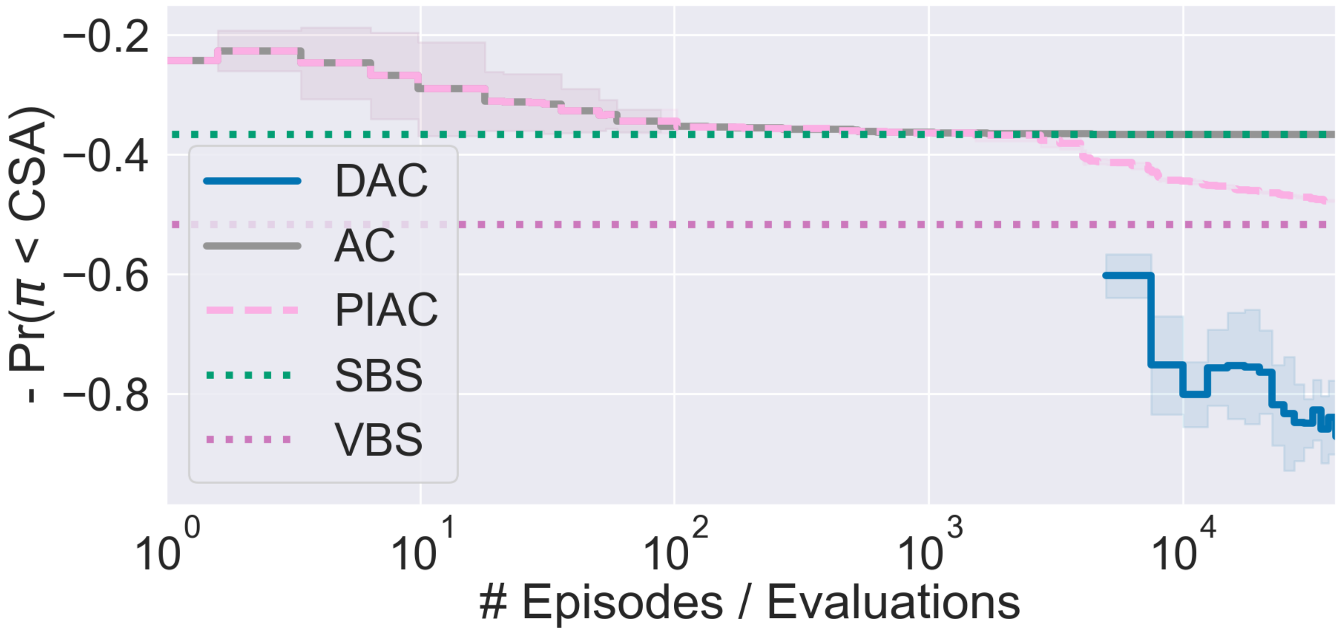


Figure: DAC and static configuration variants on CMA-ES

State of the DAC

- Several successful applications of DAC
- Dedicated benchmark library from different domains [Eimer et al. 2020]
- On-going research into better solutions methods
- Collaborative efforts to expand both available problems and solutions
- **What's next?**

The DAC4AutoML Competition Setting

- Competition for the AutoML-Conf 2022
- Two tracks: DAC for ML and DAC for RL
- Motivation: create a problem setting that reflects interesting ML and RL configuration problems
- Focus on generalization across several different instance options
- Task: beat both static baselines and well-known dynamic heuristics

The DAC4AutoML Competition Setting - DAC4SGD

- Image Classification on different image datasets, hyperparameter configurations and architectures
- Task: dynamic learning rate control across variations
- Test entropy loss is used for scoring
- Baselines: Static learning rate, cosine annealing [Loshchilov & Hutter 2017], reduce learning rate on plateau [Pytorch; Paszke et al. 2019]
- Results: two participants beat all baselines, all beat the static one

The DAC4AutoML Competition Setting - DAC4RL

- Training an RL agent each for 5 different environments with variations
- Task: controlling the algorithm and all its hyperparameters
- Rank across environments is used for scoring
- Baselines: SB3 Zoo optimized hyperparameters [Raffin et al. 2021], PB2 on the competition setting [Parker-Holder et al. 2020]
- Results: one participant could beat all baselines, setting overall is hard

Current Open Questions

- How can we scale to more hyperparameters?
- Is there a best DAC method? Can we combine existing ones?
- What are ways to bootstrap from existing solutions?
- How far can we push generalization in DAC for ML?

If You Want To Know More

- Find all our research on DAC on automl.org
- Read our recent overview paper on DAC
- Get started working on DAC with DACBench
- Get in touch!

Bibliography

[0] Steven Adriaensen, André Biedenkapp, Gresa Shala, Noor H. Awad, Theresa Eimer, Marius Lindauer, Frank Hutter: Automated Dynamic Algorithm Configuration. CoRR abs/2205.13881 (2022)

[1] Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Difan Deng, Carolin Benjamins, Tim Ruhkopf, René Sass, Frank Hutter: SMAC3: A Versatile Bayesian Optimization Package for Hyperparameter Optimization. J. Mach. Learn. Res. 23: 54:1-54:9 (2022)

[2] Sergey Levine, Vladlen Koltun: Guided Policy Search. ICML 2013

[3] Theresa Eimer, André Biedenkapp, Maximilian Reimer, Steven Adriaensen, Frank Hutter, Marius Lindauer: DACBench: A Benchmark Library for Dynamic Algorithm Configuration. IJCAI 2021

[4] Ilya Loshchilov, Frank Hutter: SGDR: Stochastic Gradient Descent with Warm Restarts. ICLR 2017

[5] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, Soumith Chintala: PyTorch: An Imperative Style, High-Performance Deep Learning Library. NeurIPS 2019

[6] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, Noah Dormann: Stable-Baselines3: Reliable Reinforcement Learning Implementations. J. Mach. Learn. Res. 22 (2021)

[7] Jack Parker-Holder, Vu Nguyen, Stephen J. Roberts: Provably Efficient Online Hyperparameter Optimization with Population-Based Bandits. NeurIPS 2020

Image Sources

[i] Triathlon waves: https://www.pinterest.at/pin/334533078571125502/

[ii] Triathlon road: https://publicdomainvectors.org/en/free-clipart/Country-road-vector-image/27110.html

[iii] Triathlon track: https://www.dreamstime.com/running-track-vector-stadium-pattern-illustration-background-image151507060