AdaStop: sequential testing for efficient and reliable comparisons of Deep RL Agents

Timothée Mathieu
timothee.mathieu@inria.fr

Riccardo Della Vecchia
riccardo.della-vecchia@inria.fr

Alena Shilova
alena.shilova@inria.fr

Hector Kohler
hector.kohler@inria.fr

Matheus Medeiros Centa
matheus.medeiros-centa@inria.fr

Odalric-Ambrym Maillard
odalric.maillard@inria.fr

Philippe Preux
philippe.preux@inria.fr
Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 – CRIStAL, F-59000 Lille, France.

Abstract

The reproducibility of many experimental results in Deep Reinforcement Learning (RL) is under question. To solve this reproducibility crisis, we propose a theoretically sound methodology to compare multiple Deep RL algorithms. The performance of one execution of a Deep RL algorithm is random so that independent executions are needed to assess it precisely. When comparing several RL algorithms, a major question is how many executions must be made and how can we assure that the results of such a comparison is theoretically sound. Researchers in Deep RL often use less than 5 independent executions to compare algorithms: we claim that this is not enough in general. Moreover, when comparing several algorithms at once, the error of each comparison accumulates and must be taken into account with a multiple tests procedure to preserve low error guarantees. To address this problem in a statistically sound way, we introduce AdaStop, a new statistical test based on multiple group sequential tests. When comparing algorithms, AdaStop adapts the number of executions to stop as early as possible while ensuring that we have enough information to distinguish algorithms that perform better than the others in a statistical significant way. We prove both theoretically and empirically that AdaStop has a low probability of making an error (Family-Wise Error). Finally, we illustrate the effectiveness of AdaStop in multiple use-cases, including toy examples and difficult cases such as Mujoco environments.

1 Introduction

We consider the problem of comparing a set of Reinforcement Learning (RL) agents, based on experimental runs only, in a theoretically sound way while using as few runs as possible.

A methodology crisis. In this paper, the end-goal is *result reproducibility* which consists in being able to get the "same" result when re-running an experiment even if the seed of randomness changes [1, 7, 13]. In RL, in particular, papers that contain an experimental study usually compare the performance of 2 or more agents facing a certain task. The common practice is to run each agent N times, typically using different pseudo-random number seeds. From these runs, the performance of the agent is

16th European Workshop on Reinforcement Learning (EWRL 2023).

measured (e.g. using an estimate of the value of the agent policy). This N is usually set *a priori*, hoping that it is enough to decide how agents rank, or at least the best one. When the training does not take too long, it is possible to take N large and be confident about the conclusion. As it is customary today, one may also test the agents on computationally heavy tasks, where an experimental run of one agent may take several days to complete. In such cases, practitioners may execute only $N = 3$ runs, and still proceed with conclusions. We scanned all RL papers published in the proceedings of ICML 2022: there are 82 and only one paper uses a statistical test to compare agents [19]. Most papers use less than $N = 5$ runs (see Appendix H.1). For a statistician, it is clear that the N being used is unlikely to be enough, except maybe the paper using $N = 80$. In any case, for all these 82 papers, we simply do not know if the conclusion is statistically significant. Indeed, from a probabilistic point of view, the performance of an agent is a random variable, and one run generates one sample from this random variable. The randomness may come from the agent itself (e.g. random initialization of some variables, or random decisions), from stochastic transitions or rewards of the environment, or both. For these reasons, comparing the performance of two agents in a reliable way usually requires more than a single run of each agent, to account for the statistical variability of the random variables and hence draw the correct conclusion. This creates a tension between collecting enough runs to ensure the correct conclusion and keeping the running time (also computational power, energy consumption) required by all experimental runs as low as possible. Moreover, the distribution of the performance of an agent rarely belongs to a classical parametric family and should rather be thought of as being non-parametric (see Appendix H.2), which creates an additional difficulty.

Motivation. Motivated by these critical methodological issues, we want to introduce a sound statistical test as a tool for practitioners to assess whether agent A_1 is statistically better than learning agents $A_2, ...A_L$. Furthermore, due to the stringent cost of performing one run of an agent, we want to achieve a statistically significant conclusion with as few runs as possible to rank the agents. In short, we want a sequential non-parametric multiple-hypothesis test with sound guarantees (for drawing the right conclusion), able to stop experiments as soon as enough evidence is acquired to rule out the wrong hypotheses. While designing such a test may be beneficial far beyond the field of reinforcement learning (e.g. computational optimization), in this paper we focus on RL due to its identified methodology issues and easily available benchmarks.

Literature overview of evaluation methods. The literature relevant to solving the evaluation methodology challenge is three-fold:

Non-adaptive approaches. In the literature, two approaches for RL agent comparison have been studied. In [6, 7], the authors show how to use hypothesis testing to test the equality between agents. Compared to our work, their approach is non-adaptive and only compares two agents. Another line of works can be found in [1] in which the authors compare many agents using confidence intervals. Their approach is non-adaptive and lacks theoretical guarantees, as they do not use multiple tests.

Sequential tests. A closely related method for adaptive hypothesis testing consists in sequential tests. Two particular classes of sequential tests that are commonly used are the Sequential Probability Ratio test [30] and the Generalized Likelihood Ratio test [18]. In the sequential testing, one deals with the performance of each single run, one after the other. This is not adapted to our situation because in RL practice, one often trains several agents in parallel, hence obtaining a batch of performances simultaneously. This motivates the use of group sequential tests [17]. In particular, the test we use is close to the one presented in [22] in which they use rank test with group-sequential testing. Contrary to our work, [22] does not provide theoretical guarantees.

Bandits (Best arm identification or ranking). Our objective is close to the one of bandit algorithms [20]. Similar to bandits, we *minimize the stopping time* (as in fixed-confidence setting) of the test and have a *fixed maximum budget* (as in a fixed-budget setting). In our test, we allow a type I error with probability $\alpha \in (0, 1)$, which is similar to the fixed confidence setting while still having a fixed budget. Compared to the fixed budget setting, we allow a larger error rate, which results in a test that is more sample efficient than bandit algorithms.

Contribution. The main contribution of this paper is to define AdaStop, a statistical test able to decide whether the number of runs already made is enough to rank a set of agents with some confidence level α. AdaStop is a new sequential test; we provide the theoretical analysis that proves that AdaStop is actually performing this test. Aside from the theoretical contribution, we report on an experimental study that demonstrates its use. AdaStop also comes as an off-the-shelf open source program that is easy to use.

By providing a statistical test to compare the performance of different agents, AdaStop improves on experimental reproducibility (we define this term below) by making researchers run their experiments just enough times, but not more. In this regard, AdaStop may also help to optimize the energy consumption by avoiding unnecessary runs, while ensuring statistically valid conclusions.

Overview of the paper: In Section 2, we state the problem and make a survey of existing evaluation workflows in RL. In Section 3, we provide a background on the statistical tools used to design AdaStop, especially group-sequential, permutation and multiple hypotheses tests. Our main algorithm, AdaStop, is introduced in Section 4, Algorithm 1, and its theoretical guarantees are stated in Section 4.1 Theorem 1. Finally, in Section 5, we give a detailed experimental study of potential benefits of AdaStop to draw statistically significant comparisons of agents.

To reproduce the experiments of this paper, the python code is freely available on github. In addition, we provide a library and command-line tool that can be used independently[1].

2 Problem setting

First, we fix some vocabulary that we will use throughout the article.

- Vocabulary in RL

  Agent: a program implementing an RL algorithm with its set of parameters and hyperparameters values, except the seed of the pseudo-random number generator.

  Seed & run: a run is one training of an agent and the subsequent evaluations of this trained agent. Because one run is initialized with a seed of the random number generator, it is common to refer to the number of runs as the number of seeds.

  Evaluation: cumulative reward obtained after running the policy of a trained agent for one episode.

  Performance: denoted $e_N(j)$ or $e_{N,k}(j)$: quantification of the performance of a trained agent on one run. Typically, we run the agent's policy on the environment for 100 epochs and recover the evaluation of each epoch. The performance returned is usually the mean of these 100 evaluations.

- Vocabulary on tests

  Interim: interim k is the kth iteration in a group sequential testing, when we take a decision whether to reject the hypotheses and stop early or continue with the group sequential test.

  Boundary: denoted $B_N^{(j)}$ or $B_{N,k}^{(j)}$, the boundary is the set of all the thresholds used in the group sequential test.

2.1 Goal and requirements for AdaStop

Given two agents A_1 and A_2, the goal of this article is to propose a sound way to evaluate how large N must be to be confident that either $\mathbb{E}[e_1(A_1)] = \mathbb{E}[e_2(A_2)]$ or $\mathbb{E}[e_1(A_1)] \neq \mathbb{E}[e_2(A_2)]$, i.e. if two agents perform similarly or if one is better than the other. This leads to manage a trade-off between the computational time and the correct assessment of the performances of A_1 and A_2. The main properties of AdaStop are as follows:

Non-Parametric The distribution of agents' performances is typically non-Gaussian, usually multimodal, skewed, etc.

Fixed Budget AdaStop should use a fixed maximum number of runs so that the computational time stays reasonable.

Sample efficient AdaStop should stop as soon as possible in practice, that is, as soon as a statistically significant conclusion can be drawn.

[1]Anonymous repository can be found here: AdaStop Library repo `https://anonymous.4open.science/r/adastop-1CF3`, paper reproducibility repo `https://anonymous.4open.science/r/Adaptive_stopping_MC_RL-5450/`.

Handling batches of data AdaStop should be able to manage batches of data to be run in parallel. Training one instance of an agent may take a long time, but training multiple instances of an agent is easy to parallelize, and thus should be done to speed-up computations.

Hence, AdaStop should be a distribution-free test.

A candidate statistical test that may verify all these properties can be found in group sequential permutation test (see the textbook [17] on general group sequential tests).

2.2 Survey of current evaluation workflows in RL

In the RL community, different approaches currently exist to compare agents and most of them are not based on any theoretically sound workflow. In what follows, we summarize some of the problems we see with the current approaches used to compare two or more RL agents in research articles.

Theoretically sound study of Atari environments? Atari environments are famous benchmarks in Deep RL [2]. Due to time constraints, when using these environments, it is customary to use very few seeds for one given game (typically 3 seeds) and compare the agents on many different games. The comparisons are then aggregated in: agent A_1 is better than agent A_2 on more than 20 games over the 26 games considered. In terms of rigorous statistics, this kind of aggregation is complicated to analyze properly because the reward distributions are not the same in all games. A_2 may be better than A_1 only on some easy games: does this mean that A_1 is a better than A_2? Up to our knowledge, there is not any proper statistical guarantee for this kind of comparison. Aggregating the comparisons on several games in Atari are still an *open problem*, and it is *beyond the scope of this article*. In this article, we suppose that we compare the agents only on one given task and leave the comparisons on a set of different tasks for future work.

Theoretically sound comparison of multiple agents? Statistical theory tells that to compare more than 2 agents (this is called multiple testing in statistics), we need more samples from each agent than if we compare only two agents. The basic idea is that there are a lot more occasions to make an error than when we compare only two agents, hence we need more data to have a lower probability of error at each comparison. This informal argument is made precise when using multiple testing, but the theory of multiple testing has never been used to compare RL agents. In this paper, we remedy this with AdaStop giving a theoretically sound workflow to compare many RL agents.

How many random seeds for Mujoco environments?
The number of runs used in practice in RL is quite arbitrary and often quite small (see Appendix H.1). An arbitrary choice of seeds do not allow us to make a statistically significant comparison of the agents.

3 Background material on hypothesis testing

In this section we describe the basic building blocks used to construct AdaStop: group sequential testing, permutation tests, and step-down method for multiple tests. We explain these items separately, and then we combine them to create AdaStop in Section 4. We also provide a small recap on hypotheses testing in the Appendix (Section B) for readers that are not used to hypothesis testing, and we provide an index of notations (Section A) defining the notations used in this article.

Group sequential testing. To choose the number of runs N adaptively, we propose to use group sequential testing (GST, see [17, 14, 25, 23]). GST often makes strong assumptions on the data, in particular it is often assumed that the data is i.i.d. and drawn from a Gaussian distribution (see [17]). With AdaStop, we propose a non-parametric approach to GST similar to [22]. In GST, the data are obtained sequentially, but the tests are done only at interim time points, with a new block of n data being obtained from one monitoring point to the next. At each interim, the boundary deciding which test to reject is derived from the permutation distribution of the statistics observed across all previously obtained data.

Permutation tests. Permutation tests (originally from [24, 11] and more recently [5, 27, 21]) are non-parametric tests that are exact for testing the equality of distributions. This means that the type I error of the test (e.g. the probability to make a mistake and reject the equality of two agents when their performances are statistically the same) is controlled by the parameter of the test α.

Let us recall the basic formulation of a two-sample permutation test. Let $X_1, \dots, X_N$ be i.i.d. sampled from a law P and $Y_1, \dots, Y_N$ i.i.d sampled from a law Q, we want to test $P = Q$ against $P \neq Q$. Let $Z_i = X_i$ if $i \leq N$ and $Z_i = Y_i$ if $i > N$, $Z_1, \dots, Z_{2N}$ is the concatenation of $X_1, \dots, X_N$ and $Y_1, \dots, Y_N$. Then, the test proceeds as follows: we reject $P = Q$ if $T(\mathrm{id}) = \left| \frac{1}{N} \sum_{i=1}^{N} (Z_i - Z_{N+i}) \right|$ is larger than a proportion $(1 - \alpha)$ of the values $T(\sigma) = \left| \frac{1}{N} \sum_{i=1}^{N} (Z_{\sigma(i)} - Z_{\sigma(N+i)}) \right|$ where σ enumerates all possible permutations of $\{1, \dots, N\}$. The idea is that if $P \neq Q$, then $T(\mathrm{id})$ should be large, and due to compensations, most $T(\sigma)$ should be smaller than $T(\mathrm{id})$. Conversely, if $P = Q$, the difference of mean $T(\sigma)$ will be closer to zero.

Multiple hypothesis testing. Multiple comparisons arise when a statistical analysis involves multiple simultaneous statistical tests [21, Chapter 9]. One possible error measurement in such a test is the family-wise error. The idea is that the confidence level for rejection probability of a true hypothesis (type I error) generally applies only to each test considered individually, but often it is desirable to have a confidence level for the whole family of simultaneous tests. Instead of the type I error considered in two-sample testing, we consider the classical family-wise error rate [29] which is defined as the probability of making at least one type I error.

Definition 1 (Family-Wise Error [29])**.** *Given a set of hypothesis H_j for $j \in \{1, \dots, J\}$, its alternative H'_j, and $\boldsymbol{I} \subset \{1, \dots, J\}$ the set of the true hypotheses among them, then*[2]

$$\mathrm{FWE} = \mathbb{P}_{H_j, j \in \boldsymbol{I}} \left(\exists j \in \boldsymbol{I} : \quad \textit{reject } H_j \right).$$

We say that an algorithm has a weak FWE control at a joint level $\alpha \in (0, 1)$ if the FWE is smaller than α when all the hypotheses are true, that is $\boldsymbol{I} = \{1, \dots, J\}$ but not necessarily otherwise. We say it has strong FWE control if FWE is smaller than α for any non-empty set of true hypotheses $\boldsymbol{I} \neq \emptyset$ (while $\boldsymbol{I}^c$ refer to false hypotheses).

There are several procedures that can be used to control the FWE. The most famous one is Bonferroni's procedure [4] recalled in the Appendix (Section B). As it can be very conservative in general, we prefer a *step-down* method [27] that performs better in practice because it implicitly estimates the dependence structure of the test statistic. The step-down method that we use is described in details in the case of two agents in the Appendix in Section F.2. The basic idea is to use the quantiles of the permutation law of the maximum over all the comparisons of the test statistics of the tests for two agents. This step corresponds to line 10 to 15 in AdaStop algorithm (Algorithm 1).

4 Adaptive stopping for non-parametric group-sequential multiple hypothesis testing

We now go further and propose a new statistical test, AdaStop (see Algorithm 1), to compare the performance of multiple agents in an *adaptive* rather than fixed way. We consider $L \geq 2$ agents $A_1, \dots, A_L$. The kth step of the algorithm is called the kth *interim*, where $k \in \{1, \dots, K\}$. We let $\mathbf{C}_0 = \{c_1, \dots, c_J\} \subseteq \{1, \dots, L\}^2$ be all the comparisons we want to make between agents. Therefore, $c_j = (c_{j,1}, c_{j,2}) \in \mathbf{C}_0$ denotes a comparison between a couple of agent's indices $c_{j,1}, c_{j,2} \in \{1, \dots, L\}$. To simplify the notation, we indicate a comparison between two agents directly with the index $j \in \{1, \dots, J\}$ instead of writing c_j, and we reserve the use of index j just for this purpose. We adopt the same shorthand notation also to re-define equivalently the set of all comparisons $\mathbf{C}_0 = \{1, \dots, J\}$ using just the indices of the comparisons. $\mathbf{I}$ denotes the set of indices of the true hypotheses among $\{1, \dots, J\}$.

We denote $e_{1,i}(j), \dots, e_{2N,i}(j)$ the concatenation at interim i of the $2N$ performance evaluations obtained for comparison j of two agents. We also consider permutations of these evaluations to define our test statistics $T_{N,k}^{(j)}$ below. Let $\mathfrak{S}_{2N}$ be the set of all the permutations of $\{1, \dots, 2N\}$. For a comparison j, and a concatenation of the evaluations of the two agents in the comparison, we consider a permutation $\sigma_i \in \mathfrak{S}_{2N}$ at interim i that reshuffles the order of the evaluations sending $n \in \{1, \dots, 2N\}$ to $\sigma_i(n) \in \{1, \dots, 2N\}$. Note that, if $n \in \{1, \dots, N\}$ and $\sigma_i(n) \in \{N+1, \dots, 2N\}$, we are permuting an evaluation of the first agent with an evaluation of the second agent in the

[2] Regarding the precise meaning of the notation $\mathbb{P}_{H_j, j \in \mathbf{I}}$, we refer to Appendix B.

comparison and vice versa. It can also happen that we instead permute evaluations of the same agents. The difference between the two cases is important for the definition of the following permutation statistic:

$$T_{N,k}^{(j)}(\sigma_{1:k}) = \left| \sum_{i=1}^{k} \left(\sum_{n=1}^{N} e_{\sigma_i(n),i}(j) - \sum_{n=N+1}^{2N} e_{\sigma_i(n),i}(j) \right) \right| . \tag{1}$$

In other words, $T_{N,k}^{(j)}(\sigma_{1:k})$ is the absolute value of the sum of differences of all evaluations until interim k after consecutive permutations of the concatenation of the two agents' evaluations by $\sigma_1, \ldots, \sigma_k \in \mathfrak{S}_{2N}$. Let $\mathbf{C} \subseteq \mathbf{C}_0$ be a subset of the set of considered hypothesis and denote

$$\overline{T}_{N,k}^{(\mathbf{C})}(\sigma_{1:k}) = \max_{j \in \mathbf{C}} T_{N,k}^{(j)}(\sigma_{1:k}) .$$

AdaStop: adaptive stopping algorithm using step-down method and group sequential permutation tests. Algorithm 1 specifies the AdaStop test. It depends on the values of the boundary thresholds. Some implementation details are discussed below.

Choice of permutations. Instead of using all the permutations as it was done previously when comparing two agents, one may use a random subset of all permutations $\mathcal{S}_k \subset \{\sigma_{1:k}, \quad \forall i \leq k, \sigma_i \in \mathfrak{S}_{2N}\}$ to speed-up computations. The theoretical guarantees persist as long as the choice of the permutations is made independent on the data. Using a small number of permutations will decrease the total power of the test, but with a sufficiently large number of random permutations (typically for the values of N and K considered, 10^4 permutations are sufficient) the loss in power is acceptable.

$T_{N,k}$ is invariant by permutation of the first half and the second half of a group. In essence, choosing a permutation is equivalent to choosing the signs in $\sum_{n=1}^{N} e_{\sigma_i(n),i}(j) - \sum_{n=N+1}^{2N} e_{\sigma_i(n),i}(j)$. And because we take the absolute value, we obtain that there are $\frac{1}{2}\binom{2N}{N}$ possible permutations in the first interim that give unique values to $T_{N,1}$. Then, by choosing permutation for the other interim, there are $\frac{1}{2}\binom{2N}{N}^k$ possible permutations giving unique values to $T_{N,k}$.

Then, we use a parameter $m \in \mathbb{N}$ and the number of permutations used at interim k will be $|\mathcal{S}_k| = m_k = \min\left(m, \frac{1}{2}\binom{2N}{N}^k\right)$, i.e. whenever possible, we use all the permutations and if this is too much, we use random permutations.

Definition of the boundaries. With these permutations, we define $B_{N,k}^{(\mathbf{C})}$ such that

$$B_{N,k}^{(\mathbf{C})} = \inf \left\{ b > 0 : \frac{1}{m_k} \sum_{\sigma \in \widehat{S}_k} \mathbb{1}\{\overline{T}_{N,k}^{(\mathbf{C})}(\sigma_{1:k}) \geq b\} \leq q_k \right\} . \tag{2}$$

where $\sum_{j=1}^{k} q_j \leq \frac{k\alpha}{K}$ and where $\widehat{S}_k$ is the subset of $\mathcal{S}_k$ such that the statistic associated to the permutation would not have rejected before. Formally, $\widehat{S}_k$ is the following set of permutations $\widehat{S}_k = \left\{ \sigma_{1:k} : \forall m < k, \quad \overline{T}_{N,m}^{(\mathbf{C})}(\sigma_{1:m}) \leq B_{N,m}^{(\mathbf{C})} \right\}$. Note that q_1 is not equal to α/K. Due to discreetness (we use an empirical quantile over a finite number of values), q_1 is chosen equal to $\lfloor \frac{\alpha}{2K}\binom{2N}{N} \rfloor / (\frac{1}{2}\binom{2N}{N})$, and similarly q_2 is chosen to be as large as possible while having $q_1 + q_2$ smaller than $2\alpha/K$, and so on for q_i for $3 \leq i \leq k$.

4.1 Theoretical guarantees

One of the basic properties of two-sample permutation tests is that when the null hypothesis is true, then all permutations are as likely to give a certain value and hence the law given the data is the uniform distribution over all permutations. Then, as a consequence of our choice of $\overline{B}_{N,k}$ as a quantile of the law given the data, the algorithm has a probability to wrongly reject the hypothesis bounded by α. This informal statement is made precise in the following theorem.

Theorem 1 (Controlled family-wise error)**.** *Suppose that $\alpha \in (0,1)$, and consider the multiple testing problem $H_j : P_j = P_k$ against $H'_j : P_j \neq P_k$ for all the couples $(j,k) \in \{c_1, \ldots, c_J\}$. Then, the*

Algorithm 1: AdaStop (main algorithm)

Parameters: Agents $A_1, A_2, \ldots, A_L$, environment $\mathcal{E}$, comparison pairs $(c_i)_{i \leq L}$ where c_i is a couple of two agents that we want to compare. Integers $K, N \in \mathbb{N}^*$, test parameter α.

Define LNK different seeds $(s_{l,n,k})_{l \leq L, n \leq N, k \leq K}$.
Set $\mathbf{C} = \{1, \ldots, J\}$ the set of indices for the comparisons we want to do.
for $k = 1, \ldots, K$ **do**
  for $l = 1 \ldots L$ **do**
    Train agent A_l on environment $\mathcal{E}$ with the seeds $s_{l,1,k}, \ldots, s_{l,N,k}$.
    Collect the N evaluations of agent A_l.
  end
  while *True* **do**
    Compute the boundaries $B_{N,k}^{(\mathbf{C})}$ from Equation (2).
    if $T_{N,k}^{(\boldsymbol{C})}(\mathrm{id}) > B_{N,k}^{(\boldsymbol{C})}$ **then**
      Reject $H_{j_{\max}}$ where $j_{\max} = \arg\max \left(\overline{T}_{N,k}^{(j)}(\mathrm{id}), \quad j \in \mathbf{C} \right)$.
      Update $\mathbf{C} = \mathbf{C} \setminus \{j_{\max}\}$
    else
      Break the while loop.
    end
  end
  if $\boldsymbol{C} = \emptyset$ **then** Break the loop and returns the answers. ;
  if $k = K$ **then** Then accept all hypotheses remaining in $\mathbf{C}$. ;
end

test resulting from Algorithm 1 has a strong control on the Family-wise error for the multiple test, i.e. if we suppose that all the hypotheses $H_i, i \in \mathbf{I}$ are true and the others are false, then

$$\mathbb{P}\left(\exists j \in \mathbf{I} : \quad \textit{reject } \mathbf{H_j}\right) \leq \alpha.$$

The proof of Theorem 1 is postponed to the Appendix (Section C). Remark that the theoretical results are not entirely satisfying: instead of comparing the means we compare the probabilities, and we don't have information on the power of the test. It can be shown (Section E) that for N large the test of comparing the means $\mu_j = \mu_k$ versus $\mu_j \neq \mu_k$ has the right guarantees (FWE smaller than α) and the power goes to 1 but we have no finite-time guarantees on this.

4.2 Heuristic for early accept for even faster decisions

AdaStop only rejects hypotheses early, it is in addition also possible to accept some hypotheses early. When comparing four agents, we could have Agent A_1 performs similarly to A_2 and A_3 performs similarly to A_4. In such a case, AdaStop would use up all the maximum number of comparisons and would stop only when $k = K$ because it will never be able to reject the comparisons A_1 vs. A_2, and A_3 vs. A_4. To solve this problem, we *early accept* the equality of agents that are statistically very similar. We proceed by analogy with the early reject methodology and construct a boundary under which the minimum of the test statistics $T_{N,k}^{(j)}$ must be to accept. The details of the implementation and its effect on the Walker environment are given in the Appendix (see Section G).

5 Experimental study

In this section, we first illustrate the statistical properties of AdaStop on toy examples for which the performances of agents are simulated. Then, we compare empirically AdaStop to non-adaptive approach. Finally, we exemplify the use of AdaStop on a real case to compare several deep-RL agents, each from a different library. We believe this is a key section demonstrating the strength of our approach from a practitioner perspective.

5.1 Toy examples

To start with, let us demonstrate the execution of our algorithm on toy examples. In what follows, we use $\mathcal{N}(\mu, \sigma^2)$ for the normal distribution with the mean μ and the standard deviation σ, $t(\mu, \nu)$ for the t-Student distribution with the mean μ and the degree of freedom ν, $\mathcal{M}^{\mathcal{N}}_{\frac{1}{2}}(\mu_1, \sigma_1^2; \mu_2, \sigma_2^2)$ for the mixture of Gaussians $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$ and $\mathcal{M}^{t}_{\frac{1}{2}}(\mu_1, \nu_1; \mu_2, \nu_2)$ for the mixture of t-Student distributions $t(\mu_1, \nu_1)$ and $t(\mu_2, \nu_2)$. In the two examples, we suppose that we compare two agents A_1 and A_2 for which we know the distributions of their performances. Those two examples are summarized in Fig. 1 (top), where Δ denotes the distance between two modes of the mixtures ($\Delta = |\mu_1 - \mu_2|$). For both cases, we execute AdaStop with $K = 5$, $N = 5$ and $\alpha = 0.05$. We also limit the maximum number of permutations to $B = 10\,000$. These two cases are executed without early accept. In Fig. 1 (bottom), we plot the rate of rejection of the null hypothesis, stating that compared distributions are the same. By varying Δ from 0 to 1, we observe the evolution of the power of tests. The bottom line (Case 1) shows that the power of the test stays around 0.05 level for all Δ. Indeed, even though the distributions in the comparison are different, their means remain the same. If the null hypothesis states that the means are the same, then AdaStop will return the correct answer with type I error not larger than 0.11 for $\alpha = 0.05$. This is an illustration of the fact that in addition to performing a test on the distributions, AdaStop approximates the test on the means as shown theoretically in the asymptotic result in Appendix Section E and as discussed at the end of Section 4.1. In contrast, the top line (Case 2) demonstrates the increasing trend, reaching the level close to 1 after $\Delta = 0.6$, which corresponds to the case when two modes are separated by 3 standard deviations from both sides. To obtain error bars, we have executed each comparison $M = 5\,000$ number of times, and we plot confidence intervals corresponding to $3\sigma/\sqrt{M}$ (more than 99% of confidence) where σ is a standard deviation of the test decision. In addition to Cases 1 and 2, we also provide a third experiment with a comparison of 10 agents in Appendix Section H.3.

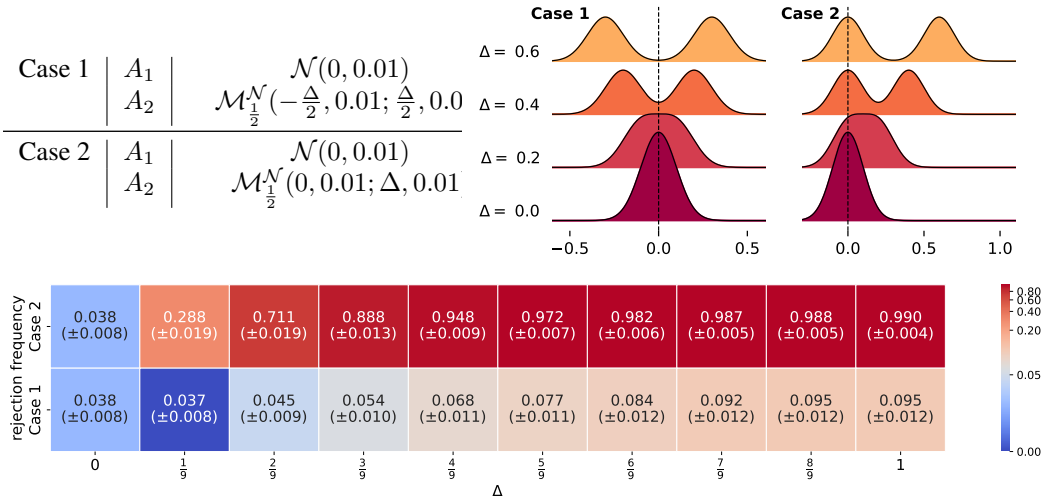

Figure 1: Toy examples 1 and 2 with an illustration of Gaussian mixtures (top) and rejection frequency of null hypothesis according to Δ (bottom).

5.2 Comparison with non-adaptive approach

The article [7] shares the same objective as ours. However, it uses non-adaptive tests unlike AdaStop. We follow their experimental protocol and compare AdaStop and non-adaptive approaches empirically in terms of statistical power as a function of the sample size (number of seeds). In particular, we use the data they provide for a SAC agent and for a TD3 agent evaluated on HalfCheetah (See Fig. 8 in Appendix). Similarly to [7, Table 15], we compute the empirical statistical power of AdaStop as a function of the number of seeds of the RL algorithms (Table 1). To compute the empirical statistical power for a given number of seeds, we make the hypothesis that the distributions of SAC and TD3 agents evaluations are different, and we count how many times AdaStop decides that one

Table 1: Empirical statistical power and effective number of seeds used by AdaStop as a function of the total number of seeds ($N \times K$) when comparing SAC and TD3 on HalfCheetah. The number of permutations is 10 000 and α is 0.05. AdaStop is run 1000 times for every (N, K) pair. The shades of blue are proportional to a value in $[0, 1]$ (we use the same color scheme as in [6])

N\K	2	3	4	5	6
1	0.0 (2.0)	0.0 (3.0)	0.277 (4.0)	0.465 (5.0)	0.56 (6.0)
2	0.005 (4.0)	0.33 (6.0)	0.531 (6.96)	0.602 (8.345)	0.704 (9.198)
3	0.213 (5.984)	0.506 (8.085)	0.627 (10.212)	0.689 (11.02)	0.785 (11.52)
4	0.371 (7.616)	0.611 (9.648)	0.744 (11.7)	0.82 (12.08)	0.845 (13.89)
5	0.465 (9.044)	0.691 (11.031)	0.78 (13.28)	0.853 (14.27)	0.884 (14.532)
6	0.534 (10.4)	0.73 (12.306)	0.837 (14.124)	0.89 (14.94)	0.911 (15.978)
7	0.599 (11.358)	0.779 (13.404)	0.879 (14.916)	0.92 (15.495)	0.939 (16.404)
8	0.635 (12.322)	0.818 (13.95)	0.885 (15.824)	0.942 (16.03)	0.961 (17.268)

agent is better than the other (number of true positives). As the test is adaptive, we also report the effective number of seeds necessary to make a decision with 0.95 confidence level. For each number of seeds, we have launched AdaStop 1000 times. For example, when comparing SAC and TD3 agents performances on HalfCheetah using AdaStop with $N = 4$ and $K = 5$, the maximum number of seeds that could be used is $N \times K = 20$ without early stopping. However, we observe in Table 1 that when $N = 4$ and $K = 5$, AdaStop can make a decision with a power of 0.82 using only 12 seeds. In [7, Table 15], the minimum number of seeds required to obtain a statistical power of 0.8 when comparing SAC and TD3 agents is 15 when using a t-test, a Welch test, or a bootstrapping test.

5.3 AdaStop for Deep Reinforcement Learning

In this section, we use AdaStop to compare four commonly-used Deep RL algorithms on the MuJoCo[3] [28] benchmark for high-dimensional continuous control, as implemented in Gymnasium[4]. More specifically, we train agents on the Ant-v3, HalfCheetah-v3, Hopper-v3, Humanoid-v3, and Walker-v3 environments using PPO from rlberry [9], SAC from Stable-Baselines3 [26], DDPG from CleanRL [16], and TRPO from MushroomRL [8]. PPO, SAC, DDPG, and TRPO are all deep reinforcement learning algorithms used for high-dimensional continuous control tasks. We chose these algorithms because they are commonly used and represent a diverse set of approaches from different RL libraries. We use different RL libraries in order to demonstrate the flexibility of AdaStop, as well as to provide examples on how to integrate these popular libraries with AdaStop. For each algorithm, we fix the hyperparameters to those used by the library authors in their benchmarks for one of the MuJoCo environments. In Appendix H.5, we provide the values that were used and further discuss the experimental setup.

We compare the four agents in each environment using AdaStop with $N = 5$ and $K = 6$. Fig. 2 shows the AdaStop decision tables in each environment, as well as the number of evaluations per agent and environment. As expected, SAC ranks first in every environment. In contrast, DDPG is ranked last in four out of five environments. Such performance may be a product of the restriction to deterministic policies, which hurts exploration in high-dimensional continuous control environments such as the MuJoCo benchmarks. Furthermore, we observe that the expected ordering between PPO and TRPO is generally respected, with TRPO outperforming PPO in only one environment. Finally, we note that PPO performs particularly well in some environments with its performance comparable to SAC, while also being the worst-performing algorithm for HalfCheetah-v3. Overall, the AdaStop rankings in these experiments are not unexpected.

Moreover, our experiments demonstrate that AdaStop can make decisions with fewer evaluations, thus reducing the computational cost of comparing Deep RL agents. For instance, as expected, SAC outperformed other agents on the environment HalfCheetah-v3, and AdaStop required only five evaluations to make all decisions involving SAC. Additionally, we observed that the decisions requiring the entire budget of $NK = 30$ evaluations were the ones in which AdaStop determined that the agents were equivalent in terms of their performance. With the early-accept heuristic proposed

[3]We use MuJoCo version 2.1, as required by https://github.com/openai/mujoco-py

[4]https://github.com/Farama-Foundation/Gymnasium

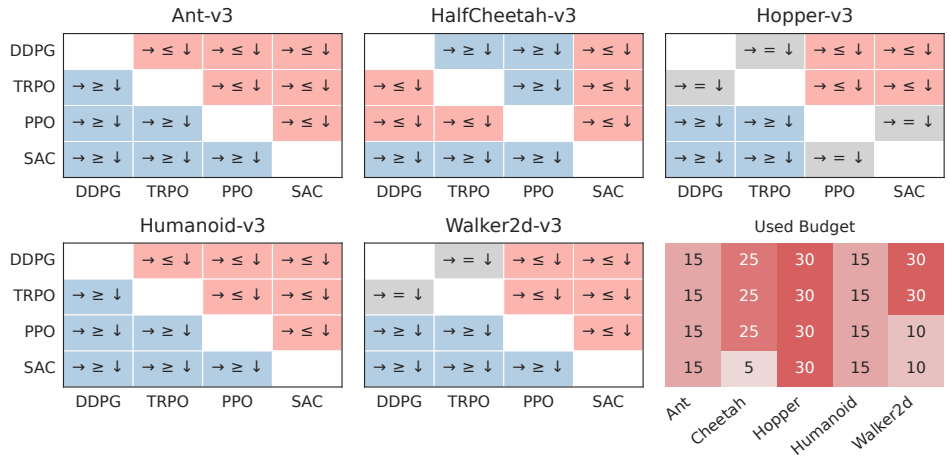

Figure 2: AdaStop decision tables for each MuJoCo environment, and the budget used to make these decisions (bottom right). See Appendix H.5 for further details.

in 4.2 and detailed in the Appendix (Section G), this process can be sped-up and for instance in the Walker2d-v3 environment, early accept allows us to take all the decisions after only 10 seeds.

6 Conclusion and future works

In this paper, we introduce AdaStop which is a sequential group test aiming at ranking the performance of agents. Our goal is to provide statistical grounding to define the number of times a set of agents should be run to be able to confidently rank them, up to some level α. This is the first such test, and we think this is a major contribution to computational studies in reinforcement learning and other domains. Using AdaStop is simple, and we provide open source software to use it. From a statistical point of view, we have been able to demonstrate the soundness of AdaStop as a statistical test. Experiments demonstrate how AdaStop may be used in practice, even in a retrospective manner using logged data.

Acknowledgments

O-A. Maillard and Ph. Preux acknowledge the support of the Métropole Européenne de Lille (MEL), ANR, Inria, Université de Lille, through the AI chair Apprenf number R-PILOTE-19-004-APPRENF. All the authors acknowledge Inria, Scool for the working environment.

References

[1] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34:29304–29320, 2021.

[2] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

[3] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300, 1995.

[4] Carlo Bonferroni. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, 8:3–62, 1936.

[5] EunYi Chung and Joseph P. Romano. Exact and asymptotically robust permutation tests. *The Annals of Statistics*, 41(2):484 – 507, 2013.

[6] Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. How many random seeds? statistical power analysis in deep reinforcement learning experiments. *arXiv preprint arXiv:1806.08295*, 2018.

[7] Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. A hitchhiker's guide to statistical comparisons of reinforcement learning algorithms. *arXiv preprint arXiv:1904.06979*, 2019.

[8] Carlo D'Eramo, Davide Tateo, Andrea Bonarini, Marcello Restelli, and Jan Peters. Mushroomrl: Simplifying reinforcement learning research. *Journal of Machine Learning Research*, 22(131):1–5, 2021.

[9] Omar Darwiche Domingues, Yannis Flet-Berliac, Edouard Leurent, Pierre Ménard, Xuedong Shang, and Michal Valko. rlberry - A Reinforcement Learning Library for Research and Education, 10 2021.

[10] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep policy gradients: A case study on ppo and trpo. *arXiv preprint arXiv:2005.12729*, 2020.

[11] Ronald Aylmer Fisher. Design of experiments. *British Medical Journal*, 1(3923):554, 1936.

[12] Yannis Flet-Berliac, Reda Ouhamma, Odalric-Ambrym Maillard, and Philippe Preux. Learning value functions in deep policy gradients using residual variance. In *ICLR 2021-International Conference on Learning Representations*, 2021.

[13] Steven N Goodman, Daniele Fanelli, and John PA Ioannidis. What does research reproducibility mean? *Science translational medicine*, 8(341):341ps12–341ps12, 2016.

[14] KK Gordon Lan and David L DeMets. Discrete sequential boundaries for clinical trials. *Biometrika*, 70(3):659–663, 1983.

[15] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[16] Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G.M. Araújo. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022.

[17] Christopher Jennison and Bruce W Turnbull. *Group sequential methods with applications to clinical trials*. CRC Press, 1999.

[18] Emilie Kaufmann and Wouter M Koolen. Mixture martingales revisited with applications to sequential tests and confidence intervals. *The Journal of Machine Learning Research*, 22(1):11140–11183, 2021.

[19] Pascal Klink, Haoyi Yang, Carlo D'Eramo, Jan Peters, and Joni Pajarinen. Curriculum reinforcement learning via constrained optimal transport. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 11341–11358. PMLR, 17–23 Jul 2022.

[20] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[21] Erich Leo Lehmann, Joseph P Romano, and George Casella. *Testing statistical hypotheses*, volume 3. Springer, 2005.

[22] Cyrus R Mehta, Nitin Patel, Pralay Senchaudhuri, and Anastasios Tsiatis. Exact permutational tests for group sequential clinical trials. *Biometrics*, pages 1042–1053, 1994.

[23] Sandro Pampallona and Anastasios A Tsiatis. Group sequential designs for one-sided and two-sided hypothesis testing with provision for early stopping in favor of the null hypothesis. *Journal of Statistical Planning and Inference*, 42(1-2):19–35, 1994.

[24] Edwin JG Pitman. Significance tests which may be applied to samples from any populations. *Supplement to the Journal of the Royal Statistical Society*, 4(1):119–130, 1937.

[25] Stuart J Pocock. Group sequential methods in the design and analysis of clinical trials. *Biometrika*, 64(2):191–199, 1977.

[26] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[27] Joseph P. Romano and Michael Wolf. Exact and approximate stepdown methods for multiple hypothesis testing. *SSRN Journal Electronic Journal*, 2003.

[28] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[29] John Wilder Tukey. The problem of multiple comparisons. *Multiple comparisons*, 1953.

[30] A. Wald. Sequential Tests of Statistical Hypotheses. *The Annals of Mathematical Statistics*, 16(2):117 – 186, 1945.

A Index of notations

- $\mathbb{E}[X]$: expectation of the random variable X.
- $\mathfrak{S}_n$: set of all the permutations of $\{1, \dots, n\}$.
- H_j: denotes hypothesis j in a multiple test, H'_j denotes the alternative of hypothesis H_j.
- σ: generic notation for a permutation. Element of $\mathfrak{S}_n$ for some $n \in \mathbb{N}^*$.
- $e_i(j)$ or $e_{i,k}(j)$: performance measure that corresponds to run number i when doing the test for comparison j. See the beginning of Section 2.
- $\sigma_{1:k}$: shorthand for the permutation that applies each σ_i to the elements $(e_{n,i})_{n \leq 2N}$, for interims from $i = 1$ to $i = k$.
- $T_N(\sigma)$ and $T^{(j)}_{N,k}(\sigma)$: test statistics. See Equation (9) and Equation (1).
- c_j: denotes a comparison. This is a couple in $\{1, \dots, L\}^2$.
- j: shorthand for denoting comparison c_j.
- $\mathbf{C}_0$: set of all the comparisons done in AdaStop.
- $\mathbf{C}$: current set of undecided comparisons in AdaStop, a subset of $\mathbf{C}_0$.
- $\mathbf{C}_k$: state of $\mathbf{C}$ at interim k in AdaStop.
- $\mathbf{I}$: set of true hypotheses.
- FWE: family-wise error, see Definition 1.
- $\mathcal{N}(\mu, \sigma^2)$: law of a Gaussian with mean μ and variance σ^2.
- $t(\mu, \nu)$: law of a translated Student distribution with center of symmetry μ and ν degrees of freedom.
- $\mathcal{M}^{\mathcal{N}}_{\frac{1}{2}}(\mu_1, \sigma_1^2; \mu_2, \sigma_2^2)$: mixture of two normal distributions.
- $\mathcal{M}^{t}_{\frac{1}{2}}(\mu_1, \nu_1; \mu_2, \nu_2)$: mixture of two Student distributions.
- $\mathbb{P}_{H_j, j \in \mathbf{I}}$: probability distribution when $H_j, j \in \mathbf{I}$ are true and $H_j, j \notin \mathbf{I}$ are false.

B Recap on hypothesis testing

To be fully understood, this paper requires the knowledge of some notions of statistics. In the hope of widening the audience of this paper, we provide a short recap of essential notions of statistics related to hypothesis testing.

B.1 Type I and type II error

In its most simple form, a statistical test is aimed at deciding, whether a given collection of data $X_1, \dots, X_N$ adheres to some hypothesis H_0 (called the null hypothesis), or if it is a better fit for an alternative hypothesis H_1. Typically, $H_0 : \mu = \mu_0$ and $H_1 : \mu \neq \mu_0$ where μ is the mean of the distribution of $X_1, \dots, X_N$. Because μ is unknown, it has to be estimated using the data, and often that is done using the empirical mean $\widehat{\mu} = \frac{1}{N} \sum_{i=1}^{N} X_i$. $\widehat{\mu}$ is random and some deviation from μ is to be expected, the theory of hypothesis tests is concerned in finding a threshold c such that if $|\widehat{\mu} - \mu_0| > c$ then we say that H_0 is false because the deviation is more than what was expected by the theory.

A slightly more complex problem is to consider two samples $X_1, \dots, X_N$ and $Y_1, \dots, Y_N$ and do a two-sample test deciding whether the mean of the distribution of the X_i's is equal to the mean of the distribution of the Y_i's.

In both cases, the result of a test is either accept H_0 or reject H_0. This answer is not a ground truth: there is some probability that we make an error. However, this probability of error is often controlled and can be decomposed in type I error and type II errors (often denoted α and β respectively, see Table 2). Please note that the problem is not symmetric: failing to reject the null hypothesis does not mean that the null hypothesis is true. It can be that there is not enough data to reject H_0.

	H_0 is true	H_0 is false
We accept H_0	No error	type II error β
We reject H_0	type I error α	No error

Table 2: Type I and type II error.

B.2 Multiple tests and FWE

When doing simultaneously several statistical tests, one must be careful that the error of each test accumulate and if one is not cautious, the overall error may become non-negligible. As a consequence, multiple strategies have been developed to deal with multiple testing problem.

To deal with the multiple testing problem, the first step is to define what is an error. There are several definitions of error in multiple testing, among which is the False discovery rate, which measures the expected proportion of false rejections. Another possible measure of error is the Family-wise error (this is the error we use in this article) and which is defined as the probability to make at least one false rejection:

$$\text{FWE} = \mathbb{P}_{H_j, j\in\mathbf{I}}\left(\exists j \in \mathbf{I}: \quad \text{reject } H_j\right),$$

where $\mathbb{P}_{H_j, j\in\mathbf{I}}$ is used to denote the probability when $\mathbf{I}$ is the set of indices of the hypotheses that are actually true (and $\mathbf{I}^c$ the set of hypotheses that are actually false). To construct a procedure with FWE smaller than α, the simplest method is perhaps Bonferroni correction [4] in which one would use one statistical test for each of the J couple of hypotheses to be tested. And then, one would tune each hypothesis test to have a type I error α/J where J is the number of tests that have to be done. The union bound then implies that the FWE is bounded by α:

$$\text{FWE } = \mathbb{P}_{H_j, j\in\mathbf{I}}\Big(\bigcup_{j\in\mathbf{I}}\{\text{reject } H_j\}\Big) \leq \sum_{i\in\mathbf{I}} \mathbb{P}_{H_j, j\in\mathbf{I}}\left(\text{reject } H_j\right) \leq |\mathbf{I}|\frac{\alpha}{J} \leq \alpha.$$

which is the probability of rejecting the hypothesis given that it is actually true. Bonferroni correction has the advantage of being very simple to implement, but it is often very conservative and the final FWE would be most often a lot smaller than α. An alternative method that performs well in practice is the step-down method that we use in this article and which is presented in Section F.2.

C Proof of Theorem 1

The proof of Theorem 1 is based on an extension of the proof of the control of FWE in the non-sequential case and the proof of the step-down method (see [27]). The interested reader may refer to Lemma 1 in the Appendix where we reproduce the proof of the bound on FWE for simple permutation tests as it is a good introduction to permutation tests. The proof proceeds as follows: first, we prove weak control on the FWE by decomposing the error as the sum of the errors on each interim and using the properties of permutation tests to show that the error done at each interim is controlled by α/K. Then, using the step-down method construction, we show that the strong control of the FWE is a consequence of the weak control because of monotony properties on the boundary values of a permutation test.

C.1 Simplified proof for $L = 2$ agents, and $K = 1$

The proof of the theorem for this result is a bit technical. We begin by showing the result in a very simplified case with $L = 2$ agents, and $K = 1$.

Lemma 1. *Let $X_1, \ldots, X_N$ be i.i.d from a distribution P and $Y_1, \ldots, Y_N$ be i.i.d. from a distribution Q. Denote $Z_1^{2N} = X_1, \ldots, X_N, Y_1, \ldots, Y_N$ be the concatenation of X_1^N and Y_1^N. Let $\alpha \in (0, 1)$ and define B_N such that*

$$B_N = \inf\left\{b > 0: \quad \frac{1}{(2N)!}\sum_{\sigma\in\mathfrak{S}_{2N}} \mathbb{1}\left\{\frac{1}{N}\sum_{i=1}^{N}(Z_{\sigma(i)} - Z_{\sigma(N+i)}) > b\right\} \leq \alpha\right\}.$$

Then, if $P = Q$, we have

$$\mathbb{P}\left(\frac{1}{N}\sum_{i=1}^{N}(X_i - Y_i) > B_N\right) \leq \alpha$$

Proof. Denote $T(\sigma) = \frac{1}{N}\sum_{i=1}^{N}(Z_{\sigma(i)} - Z_{\sigma(n+i)})$. Since $P = Q$, for any $\sigma, \sigma' \in \mathfrak{S}_{2N}$ we have $T(\sigma) \overset{d}{=} T(\sigma')$. Then, because B_N does not depend on the permutation σ (but it depends on the values of Z_1^{2N}, we have, for any $\sigma \in \mathfrak{S}_{2N}$

$$\mathbb{P}\left(T(\mathrm{id}) > B_N\right) = \mathbb{P}\left(T(\sigma) > B_N\right)$$

Now, take the sum over all the permutations,

$$\begin{aligned}\mathbb{P}\left(T(\mathrm{id}) > B_N\right) &= \frac{1}{(2N)!}\sum_{\sigma\in\mathfrak{S}_{2N}} \mathbb{E}\left[\mathbb{1}\{T(\sigma) > B_N\}\right] \\ &= \mathbb{E}\left[\frac{1}{(2N)!}\sum_{\sigma\in\mathfrak{S}_{2N}} \mathbb{1}\{T(\sigma) > B_N\}\right] \leq \alpha\end{aligned}$$

which proves the result. □ □

Next, we prove weak control in the general case.

C.2 Proof of Theorem 1

In this section, we use the shorthand $\mathbb{P}$ instead of $\mathbb{P}_{H_j, j\in\mathbf{I}}$ and omit $H_j, j \in \mathbf{I}$ because $\mathbf{I}$ will always be the set of true hypotheses and the meaning should be clear from the context.

Weak control on FWE: First, we prove weak control on the FWE. This means that we suppose that $\mathbf{I} = \{1, \dots, J\}$: all the hypotheses are true, and we control the probability to make at least one rejection. We have,

$$\mathrm{FWE} = \mathbb{P}\left(\exists j \in \mathbf{I}: \quad H_j \text{ is rejected}\right).$$

We decompose the FWE on the diverse interims.

$$\mathrm{FWE} = \sum_{k=1}^{K} \mathbb{P}\left(\overline{T}_{N,k}^{(\mathbf{I})}(\mathrm{id}) > B_{N,k}^{(\mathbf{I})}, \mathrm{NR}_k(\mathrm{id})\right), \tag{3}$$

where $\mathrm{NR}_k(\sigma_{1:k}) = \{\forall m < k, \overline{T}_{N,m}^{(\mathbf{I})}(\sigma_{1:k}) \leq B_{N,m}^{(\mathbf{I})}\}$ is the event on which we did not reject before. We use $\sigma_{1:k}$ and not only id as this will be useful later on (See Equation (4)).

Then, similarly as in the proof of Lemma 1, we want to use the invariance by permutation to make the link with the definition of $B_{N,k}^{(\mathbf{I})}$. For this purpose, we introduce the following lemma, that we prove in Appendix D.

Lemma 2. *We have that for $k \leq K$, for any $\sigma_{1:k}$ concatenation of k permutations,*

$$(\overline{T}_{N,l}^{(\mathbf{I})}(\mathrm{id}), B_{N,l}^{(\mathbf{I})})_{l\leq k} \overset{d}{=} (\overline{T}_{N,l}^{(\mathbf{I})}(\sigma_{1:l}), B_{N,l}^{(\mathbf{I})})_{l\leq k}.$$

Using Lemma 2, we have for any $\sigma_{1:k}$

$$\mathbb{P}\left(\overline{T}_{N,k}^{(\mathbf{I})}(\mathrm{id}) > B_{N,k}^{(\mathbf{I})}, \mathrm{NR}_k(\mathrm{id})\right) = \mathbb{P}\left(\overline{T}_{N,k}^{(\mathbf{I})}(\sigma_{1:k}) > B_{N,k}^{(\mathbf{I})}, \mathrm{NR}_k(\sigma_{1:k})\right) \tag{4}$$

Hence, injecting this in Equation (3),

$$\begin{aligned}\mathrm{FWE} &\leq \sum_{k=1}^{K} \frac{1}{m_k} \sum_{\sigma_{1:k}\in\mathcal{S}_k} \mathbb{P}\left(\overline{T}_{N,k}^{(\mathbf{I})}(\sigma_{1:k}) > B_{N,k}^{(\mathbf{I})}, \mathrm{NR}_k(\sigma_{1:k})\right) \\ &= \sum_{k=1}^{K} \mathbb{E}\left[\frac{1}{m_k} \sum_{\sigma_{1:k}\in\mathcal{S}_k} \mathbb{1}\left\{\overline{T}_{N,k}^{(\mathbf{I})}(\sigma_{1:k}) > B_{N,k}^{(\mathbf{I})}, \mathrm{NR}_k(\sigma_{1:k})\right\}\right]\end{aligned}$$

Then, use that $\sigma_{1:k} \in \widehat{S}_k$ if and only if $\sigma_{1:k} \in \mathcal{S}_k$ and $\mathrm{NR}_k(\sigma_{1:k})$ is true. Hence,

$$\mathrm{FWE} \leq \sum_{k=1}^{K} \mathbb{E}\left[\frac{1}{m_k} \sum_{\sigma_{1:k}\in\widehat{S}_k} \mathbb{1}\left\{\overline{T}_{N,k}^{(\mathbf{I})}(\sigma_{1:k}) > B_{N,k}^{(\mathbf{I})}\right\}\right] \leq \sum_{k=1}^{K} q_k \leq \alpha$$

where we used the definition of $B_{N,k}^{(\mathbf{I})}$ to make the link with α

Strong control of FWE: To prove strong control, it is sufficient to show the following Lemma (see Appendix D for a proof), which is an adaptation of the proof of step-down multiple-test strong control of FWE from [27].

Lemma 3. *Suppose that* $\mathbf{I} \subset \{1, \dots, J\}$ *is the set of true hypotheses. We have*

$$\mathrm{FWE} = \mathbb{P}\left(\exists j \in \mathbf{I} : \ H_j \text{ is rejected}\right) \leq \mathbb{P}\left(\exists k \leq K : \quad \overline{T}^{(\mathbf{I})}_{N,k}(\mathrm{id}) > B^{(\mathbf{I})}_{N,k}\right).$$

Lemma 3 shows that to control the FWE, it is sufficient to control the probability to reject on $\mathbf{I}$ given by $\mathbb{P}\left(\exists k \leq K : \quad \overline{T}^{(\mathbf{I})}_{N,k}(\sigma_{1:k}) > B^{(\mathbf{I})}_{N,k}\right)$ and this quantity, in turns, is exactly the FWE of the restricted problem of testing $(H_j)_{j\in\mathbf{I}}$ against $(H'_j)_{j\in\mathbf{I}}$. In other words, Lemma 3 says that to prove strong FWE control for our algorithm, it is sufficient to prove weak FWE control, and we already did that in the first part of the proof.

D Proof of Lemmas

D.1 Proof of Lemma 2

In this section, for an easier understanding, we change the notation for the performance measure $e^{(j)}_{n,k}(\sigma)$ and denote by $e_{n,k}(A_i)$ the nth performance value of agent A_i at interim k. In effect, this means that for the comparison j of agent A_i versus agent A_l, we have the equality $e_{n,k}(A_i) = e^{(j)}_{n,k}(\mathrm{id})$ for $n \leq N$ and $e_{n,k}(A_l) = e^{(j)}_{N+n,k}(\mathrm{id})$.

We denote the comparisons by $(c_i)_{i\in\mathbf{I}}$, they describe a graph with the nodes being the agents denoted $1, \dots, L$ and (j_1, j_2) has an edge if $(j_1, j_2) \in (c_i)_{i\in\mathbf{I}}$ is one of the comparisons that corresponds to a true hypothesis. This graph is not necessarily connected, we denote $C(i)$ the connected component to which node a (e.g. agent a) belongs, i.e. for any $a_1, a_2 \in C(a)$ there exists a path going from a_1 to a_2. Remark that $C(a)$ cannot be equal to the singleton $\{a\}$, because it would mean that all the comparisons with a are in fact false hypotheses, and then a would not belong to a couple in $\mathbf{I}$.

Then, it follows from the construction of permutation test that jointly on $k \leq K$ and $a_1, a_2 \in C(i)$, we have $T^{(a_1,a_2)}_{N,k}(\mathrm{id}) \overset{d}{=} T^{(a_1,a_2)}_{N,1}(\sigma_{1:k})$ for any $\sigma_1, \dots, \sigma_k \in \mathfrak{S}_{2N}$.

Let us illustrate that on an example. Suppose that $N = 2$ and $J = 3$. Consider the permutation

$$\sigma_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 2 & 4 \end{pmatrix}$$

Because all the evaluations are i.i.d., we have the joint equality in distribution

$$\begin{pmatrix} |e_{1,1}(A_1) + e_{2,1}(A_1) - e_{1,1}(A_2) - e_{2,1}(A_2)| \\ |e_{1,1}(A_3) + e_{2,1}(A_3) - e_{1,1}(A_2) - e_{2,1}(A_2)| \\ |e_{1,1}(A_1) + e_{2,1}(A_1) - e_{1,1}(A_3) - e_{2,1}(A_3)| \end{pmatrix} \overset{d}{=} \begin{pmatrix} |e_{1,1}(A_1) + e_{2,1}(A_2) - e_{1,1}(A_1) - e_{2,1}(A_2)| \\ |e_{1,1}(A_3) + e_{2,1}(A_2) - e_{1,1}(A_3) - e_{2,1}(A_2)| \\ |e_{1,1}(A_1) + e_{2,1}(A_3) - e_{1,1}(A_1) - e_{2,1}(A_3)| \end{pmatrix}$$

and hence,

$$(T^{(j)}_{N,1}(\mathrm{id}))_{1\leq j\leq 3} \overset{d}{=} (T^{(j)}_{N,1}(\sigma_1))_{1\leq j\leq 3}.$$

For $k = 2$, we have for $\sigma_2 = \sigma_1$,

$$\begin{pmatrix} |e_{1,1}(A_1) + e_{2,1}(A_1) - e_{1,1}(A_2) - e_{2,1}(A_2)| \\ |e_{1,1}(A_3) + e_{2,1}(A_3) - e_{1,1}(A_2) - e_{2,1}(A_2)| \\ |e_{1,1}(A_1) + e_{2,1}(A_1) - e_{1,1}(A_3) - e_{2,1}(A_3)| \\ |e_{1,1}(A_1) + e_{2,1}(A_1) - e_{1,1}(A_2) - e_{2,1}(A_2) + e_{1,2}(A_1) + e_{2,2}(A_1) - e_{1,2}(A_2) - e_{2,2}(A_2)| \\ |e_{1,1}(A_3) + e_{2,1}(A_3) - e_{1,1}(A_2) - e_{2,1}(A_2) + e_{1,2}(A_3) + e_{2,2}(A_3) - e_{1,2}(A_2) - e_{2,2}(A_2)| \\ |e_{1,1}(A_1) + e_{2,1}(A_1) - e_{1,1}(A_3) - e_{2,1}(A_3) + e_{1,2}(A_1) + e_{2,2}(A_1) - e_{1,2}(A_3) - e_{2,2}(A_3)| \end{pmatrix}$$

$$\overset{d}{=} \begin{pmatrix} |e_{1,1}(A_1) + e_{2,1}(A_2) - e_{1,1}(A_1) - e_{2,1}(A_2)| \\ |e_{1,1}(A_3) + e_{2,1}(A_2) - e_{1,1}(A_3) - e_{2,1}(A_2)| \\ |e_{1,1}(A_1) + e_{2,1}(A_3) - e_{1,1}(A_1) - e_{2,1}(A_3)| \\ |e_{1,1}(A_1) + e_{2,1}(A_2) - e_{1,1}(A_1) - e_{2,1}(A_2) + e_{1,2}(A_1) + e_{2,2}(A_2) - e_{1,2}(A_1) - e_{2,2}(A_2)| \\ |e_{1,1}(A_3) + e_{2,1}(A_2) - e_{1,1}(A_3) - e_{2,1}(A_2) + e_{1,2}(A_3) + e_{2,2}(A_2) - e_{1,2}(A_3) - e_{2,2}(A_2)| \\ |e_{1,1}(A_1) + e_{2,1}(A_3) - e_{1,1}(A_1) - e_{2,1}(A_3) + e_{1,2}(A_1) + e_{2,2}(A_3) - e_{1,2}(A_1) - e_{2,2}(A_3)| \end{pmatrix}$$

and then, we get jointly

$$(T_{N,k}^{(j)}(\mathrm{id}))_{1\leq j\leq 3,k\leq 2} \overset{d}{=} (T_{N,k}^{(j)}(\sigma_1\cdot\sigma_2))_{1\leq j\leq 3,k\leq 2}.$$

This reasoning can be generalized to general N, J and K:

$$(T_{N,k}^{(a_1,a_2)}(\mathrm{id}))_{k\leq K,a_1\in C(i),a_2\in C(i)} \overset{d}{=} (T_{N,k}^{(a_1,a_2)}(\sigma_{1:k}))_{k\leq K,a_1\in C(i),a_2\in C(i)}.$$

Then, use that by construction, the different connected component $C(i)$ are independent of one another and hence,

$$(T_{N,k}^{(c_i)}(\mathrm{id}))_{k\leq K,c_i\in\mathbf{I}} \overset{d}{=} (T_{N,k}^{(c_i)}(\sigma_{1:k}))_{k\leq K,c_i\in\mathbf{I}}.$$

The result follows from taking the maximum on all the comparisons, and because the boundaries do not depend on the permutation.

D.2 Proof of Lemma 3

Denote by $\mathbf{C}_k$ the (random) value of $\mathbf{C}$ at the beginning of interim k. We have,

$$\begin{aligned}\mathrm{FWE} &= \mathbb{P}\left(\exists j\in\mathbf{I}: \quad H_j \text{ is rejected}\right)\\ &= \mathbb{P}\left(\exists k\leq K: \quad \overline{T}_{N,k}^{(\mathbf{C}_k)}(\mathrm{id}) > B_{N,k}^{(\mathbf{C}_k)}, \arg\max_{j\in\mathbf{C}_k}\overline{T}_{N,k}^{(j)}(\mathrm{id})\in\mathbf{I}\right). \end{aligned} \tag{5}$$

Then, let k_0 correspond to the very first rejection (if any) in the algorithm. Having that the argmax is attained in $\mathbf{I}$,

$$\overline{T}_{N,k_0}^{(\mathbf{C}_{k_0})}(\mathrm{id}) = \max\{T_{N,k_0}^{(j)}(\mathrm{id}), j\in\mathbf{C}_{k_0}\} = \max\{T_{N,k_0}^{(j)}(\mathrm{id}), j\in\mathbf{I}\} = \overline{T}_{N,k_0}^{(\mathbf{I})}(\mathrm{id})$$

Moreover, having $\mathbf{C}_{k_0}\supset\mathbf{I}$, we have $B_{N,k_0}^{(\mathbf{C}_{k_0})}\geq B_{N,k_0}^{(\mathbf{I})}$. Injecting these two relations in Equation (5), we obtain

$$\begin{aligned}\mathrm{FWE} &\leq \mathbb{P}\left(\exists k\leq K: \quad \overline{T}_{N,k}^{(\mathbf{I})}(\mathrm{id}) > B_{N,k}^{(\mathbf{I})}, \arg\max_{j\in\mathbf{C}_k}\overline{T}_{N,k}^{(j)}(\mathrm{id})\in\mathbf{I}\right)\\ &\leq \mathbb{P}\left(\exists k\leq K: \quad \overline{T}_{N,k}^{(\mathbf{I})}(\mathrm{id}) > B_{N,k}^{(\mathbf{I})}\right).\end{aligned}$$

This proves the desired result.

E Asymptotic results for two agents

E.1 Convergence of boundaries and comparing the means

Because there are only two agents and no early stopping, we simplify the notations and denote

$$t_{N,i}(\sigma_i) = \sum_{n=1}^{N} e_{\sigma_i(n),i}(2) - \sum_{n=N+1}^{2N} e_{\sigma_i(n),i}(1)$$

and

$$\begin{aligned}T_{N,k}(\sigma_{1:k}) &= \left|\sum_{i=1}^{k}\left(\sum_{n=1}^{N} e_{\sigma_i(n),i}(2) - \sum_{n=N+1}^{2N} e_{\sigma_i(n),i}(1)\right)\right|\\ &= \left|\sum_{i=1}^{k} t_{N,i}(\sigma_i)\right|\end{aligned}$$

and

$$B_{N,k} = \inf\left\{b>0: \frac{1}{((2N)!)^k}\sum_{\sigma_1,\ldots,\sigma_k\in\mathfrak{S}_{2N}^k}\mathbb{1}\{T_{N,k}(\sigma_{1:k})\geq b\}\leq q_k\right\}$$

When there is only one interim ($K=1$), we have the following convergence of the randomization law of $T_{N,1}(\sigma)$.

Proposition 1 (Theorem 17.3.1 in [21]). *Suppose $e_{1,1}(1), \dots, e_{N,1}(1)$ are i.i.d from P and $e_{1,1}(2), \dots, e_{N,1}(2)$ are i.i.d from Q and both P and Q has finite variance. Then, we have*

$$\sup_t \left| \frac{1}{(2N)!} \sum_{\sigma \in \mathfrak{S}_{2N}} \mathbb{1}\left\{ \frac{1}{\sqrt{N}} T_{N,1}(\sigma) \leq t \right\} - \Phi\left(t/\tau(P,Q)\right) \right| \xrightarrow[N\to\infty]{P} 0$$

where Φ is the standard normal c.d.f. and $\tau(P,Q)^2 = \sigma_P^2 + \sigma_Q^2 + \frac{(\mu_P - \mu_Q)^2}{2}$.

Using the non-sequential result from proposition 1, we can show the following theorem that controls the asymptotic law of the sequential test.

Theorem 2. *We have that for any $1 \leq k \leq K$, $\frac{1}{\sqrt{N}} B_{N,k} \xrightarrow[N\to\infty]{} b_k$ where the real numbers b_k are defined as follows. Let $W_1, \dots, W_K$ be i.i.d random variable with law $\mathcal{N}(0,1)$, then b_1 is the solution of the following equation:*

$$\mathbb{P}\left(|W_1| \geq \frac{b_1}{\tau(P,Q)} \right) = \frac{\alpha}{K},$$

and for any $1 < k \leq K$, b_k is the solution of

$$\mathbb{P}\left(\left| \frac{1}{k} \sum_{j=1}^{k} W_j \right| > \frac{b_l}{\tau(P,Q)}, \quad \forall j < k, \left| \frac{1}{j} \sum_{i=1}^{j} W_i \right| \leq \frac{b_j}{\tau(P,Q)} \right) = \frac{\alpha}{K}.$$

Remark that the test we do corresponds to testing

$$\mathbb{1}\{\exists k \leq K : \frac{1}{\sqrt{N}} T_{N,k}(\mathrm{id}) > \frac{1}{\sqrt{N}} B_{N,k}\}$$

and from Theorem 2 and central-limit theorem $\frac{1}{\sqrt{N}} T_{N,k}(\mathrm{id})$ converges to $\sum_{j=1}^{k} W_j \sqrt{\sigma_P^2 + \sigma_Q^2}$ and $B_{N,k}/\sqrt{N}$ converges to b_k, hence the test is asymptotically equivalent to

$$\mathbb{1}\left\{ \exists k \leq K : \sum_{j=1}^{k} W_j \sqrt{\sigma_P^2 + \sigma_Q^2} > b_k \right\}.$$

Then, in the case in which $\mu_P = \mu_Q$, we have $\tau(P,Q) = \sqrt{\sigma_P^2 + \sigma_Q^2}$ and

$$\begin{aligned}
\mathrm{FWE} &= \mathbb{P}\left(\exists k \leq K : \sum_{j=1}^{k} W_j \sqrt{\sigma_P^2 + \sigma_Q^2} > b_k \right) \\
&= \sum_{k=1}^{K} \mathbb{P}\left(\left| \frac{1}{k} \sum_{j=1}^{k} W_j \right| > \frac{b_l}{\tau(P,Q)}, \quad \forall j < k, \left| \frac{1}{j} \sum_{i=1}^{j} W_i \right| \leq \frac{b_j}{\tau(P,Q)} \right) \\
&= \sum_{k=1}^{K} \frac{\alpha}{K} = \alpha.
\end{aligned}$$

Hence, for the test $H_0 : \quad \mu_P = \mu_Q$ versus $H_1 : \quad \mu_P \neq \mu_Q$, our test is asymptotically of level α.

E.2 Proof of Theorem 2

We denote for $x \in \mathbb{R}$,

$$R_{N,k}(x) = \frac{1}{(2N)!} \sum_{\sigma_k \in \mathfrak{S}_{2N}} \mathbb{1}\{t_{N,k}(\sigma_k) \leq x\}.$$

$R_{N,k}$ is the c.d.f of the randomization law of $t_{N,k}(\sigma_k)$, and by Proposition 1, it converges uniformly to a Gaussian c.d.f when N goes to infinity.

Convergence of $B_{N,1}$

$$\frac{1}{\sqrt{N}}B_{N,1} = \frac{1}{\sqrt{N}}\min\left\{b>0 : \frac{1}{(2N)!}\sum_{\sigma_1\in\mathfrak{S}_{2N}} \mathbb{1}\{|T_{N,1}(\sigma_1)|>b\} \leq \frac{\alpha}{K}\right\}$$

This implies

$$\begin{aligned}\frac{1}{(2N)!}\sum_{\sigma_1\in\mathfrak{S}_{2N}} \mathbb{1}\{|T_{N,1}(\sigma_1)|\leq B_{N,1}\} &= \widehat{R}_{N,1}\left(\frac{1}{\sqrt{N}}B_{N,1}\right) - \widehat{R}_{N,1}\left(-\frac{1}{\sqrt{N}}B_{N,1}\right)\\ &\geq 1-\frac{\alpha}{K}\end{aligned}$$

and for any $b < B_{N,1}$, we have

$$\frac{1}{(2N)!}\sum_{\sigma_1\in\mathfrak{S}_{2N}} \mathbb{1}\{|T_{N,1}(\sigma_1)|\leq b\} = \widehat{R}_{N,1}\left(\frac{b}{\sqrt{N}}\right) - \widehat{R}_{N,1}\left(-\frac{b}{\sqrt{N}}\right) < 1-\frac{\alpha}{K}$$

Then,

$$\begin{aligned}&\Phi\left(\frac{B_{N,1}}{\tau(P,Q)\sqrt{N}}\right) - \Phi\left(-\frac{B_{N,1}}{\tau(P,Q)\sqrt{N}}\right)\\ \geq&\widehat{R}_{N,1}\left(\frac{B_{N,1}}{\sqrt{N}}\right) - \widehat{R}_{N,1}\left(-\frac{B_{N,1}}{\sqrt{N}}\right) - \left|\Phi\left(\frac{B_{N,1}}{\tau(P,Q)\sqrt{N}}\right) - \widehat{R}_{N,1}\left(\frac{B_{N,1}}{\sqrt{N}}\right)\right|\\ &- \left|\Phi\left(-\frac{B_{N,1}}{\tau(P,Q)\sqrt{N}}\right) - \widehat{R}_{N,1}\left(-\frac{B_{N,1}}{\sqrt{N}}\right)\right|\\ \geq&1-\frac{\alpha}{K} - 2\sup_t\left|\Phi\left(\frac{t}{\tau(P,Q)}\right) - \widehat{R}_{N,1}(t)\right|\end{aligned}$$

Hence, by taking N to infinity, we have from Proposition 1,

$$\liminf_{N\to\infty}\Phi\left(\frac{B_{N,1}}{\tau(P,Q)\sqrt{N}}\right) - \Phi\left(-\frac{B_{N,1}}{\tau(P,Q)\sqrt{N}}\right) \geq 1-\frac{\alpha}{K}.$$

and for any $\varepsilon > 0$, because of the definition of $B_{N,1}$ as a supremum, we have

$$\limsup_{N\to\infty}\Phi\left(\frac{B_{N,1}+\varepsilon}{\tau(P,Q)\sqrt{N}}\right) - \Phi\left(-\frac{B_{N,1}+\varepsilon}{\tau(P,Q)\sqrt{N}}\right) < 1-\frac{\alpha}{K}.$$

By continuity of Φ, this implies that $\frac{1}{\sqrt{N}}B_{N,1}$ converges almost surely and its limit is such that

$$\Phi\left(\frac{\lim_{N\to\infty}B_{N,1}/\sqrt{N}}{\tau(P,Q)}\right) - \Phi\left(-\frac{\lim_{N\to\infty}B_{N,1}/\sqrt{N}}{\tau(P,Q)}\right) = 1-\frac{\alpha}{K}.$$

Or said differently, let $W\sim\mathcal{N}(0,1)$, then we have the almost sure convergence $\lim_{N\to\infty}\frac{1}{\sqrt{N}}B_{N,1} = b_1$ where b_1 is the real number defined by

$$\mathbb{P}\left(|W|\geq\frac{b_1}{\tau(P,Q)}\right) = \frac{\alpha}{K}.$$

Convergence of $B_{N,k}$ for $k>1$. We proceed by induction. Suppose that $\frac{1}{\sqrt{N}}B_{N,k-1}$ converges to some $b_{k-1}>0$ and that for any $d_1,\ldots,d_{k-1}$, the randomization probability

$$\begin{aligned}\sup_{d_1,\ldots,d_{k-1}}\Bigg|\frac{1}{((2N)!)^{k-1}}\sum_{\sigma_1,\ldots,\sigma_{k-1}\in\mathfrak{S}_{2N}}&\mathbb{1}\left\{\forall j\leq k-1,\ \sum_{i=1}^{j}t_{N,i}(\sigma_i)\leq d_j\sqrt{N}\right\}\\ &-\mathbb{P}\left(\forall j\leq k-1,\ \sum_{i=1}^{j}W_i\leq\frac{d_j}{\tau(P,Q)}\right)\Bigg| \xrightarrow[a.s.]{N\to\infty} 0, \quad (6)\end{aligned}$$

where $W_1, \dots, W_{k-1}$ are i.i.d $\mathcal{N}(0,1)$ random variables. In other words, the randomization law converges uniformly to the joint law described above with the sum of Gaussian random variables.

Then, by uniform convergence and by convergence of the $B_{N,j}$, we have

$$\frac{1}{((2N)!)^{k-1}} \sum_{\sigma_1,\dots,\sigma_{k-1}\in\mathfrak{S}_{2N}} \mathbb{1}\left\{T_{N,j}\left(\sigma_{1:j}\right) > B_{N,k-1}, \quad \forall j<k-1,\, T_{N,j}\left(\sigma_{1:j}\right) \le B_{N,j}\right\}$$

converges to

$$\mathbb{P}\left(\left|\sum_{i=1}^{l} W_i\right| > \frac{b_l}{\tau(P,Q)}, \quad \forall j<l,\, \left|\sum_{i=1}^{j} W_i\right| \le \frac{b_j}{\tau(P,Q)}\right). \tag{7}$$

which is equal to $\frac{\alpha}{K}$ by construction of $B_{N,j}$ for $j<k$,

We have

$$B_{N,k} = \min\left\{b>0 : \frac{1}{((2N)!^k)} \sum_{\sigma_1,\dots,\sigma_k\in\mathfrak{S}_{2N}} \mathbb{1}\left\{\begin{smallmatrix}\left|\sum_{j=0}^{k} t_{N,j}(\sigma_j)\right|\ge b,\\ \forall j<k,\, \left|\sum_{i=0}^{j} t_{N,i}(\sigma_i)\right|\le B_{N,j}\end{smallmatrix}\right\} + \sum_{i=1}^{k-1} q_i \le \frac{k\alpha}{K}\right\}.$$

By the induction hypothesis, we have $q_i \xrightarrow[n\to\infty]{} \alpha/K$ for any $i<k$.

Let $W_1, \dots, W_k$ be i.i.d $\mathcal{N}(0,1)$ random variables. We show the following lemma that prove part of the step k of the induction hypothesis, and proved in Section E.3.

Lemma 4. *Suppose Equation* (7) *is true. Then,*

$$\sup_{d_1,\dots,d_k}\left|\frac{1}{((2N)!)^k} \sum_{\sigma_1,\dots,\sigma_k\in\mathfrak{S}_{2N}} \mathbb{1}\left\{\forall j\le k,\, \sum_{i=1}^{j} t_{N,i}\left(\sigma_i\right) \le d_j\sqrt{N}\right\} - \mathbb{P}\left(\forall j\le k,\, \sum_{i=1}^{j} W_i \le \frac{d_j}{\tau(P,Q)}\right)\right| \xrightarrow[a.s.]{N\to\infty} 0,$$

Then, what remains is to prove the convergence of $B_{N,k}$. Denote

$$\Psi_k(d_k) = \mathbb{P}\left(\left|\sum_{i=1}^{k} W_i\right| > \frac{d_k}{\tau(P,Q)}, \forall j\le k-1,\, \left|\sum_{i=1}^{j} W_i\right| \le \frac{b_j}{\tau(P,Q)}\right),$$

we have, from Lemma 4, that

$$\left|\Psi_k\left(\frac{B_{N,k}}{\sqrt{N}}\right) - \frac{1}{((2N)!)^k} \sum_{\sigma_1,\dots,\sigma_k\in\mathfrak{S}_{2N}} \mathbb{1}\left\{T_{N,k}\left(\sigma_{1:k}\right) > B_{N,k}, \quad \forall j<k,\, T_{N,j}\left(\sigma_{1:j}\right) \le B_{N,j}\right\}\right|$$

converges to 0 as N goes to infinity. Hence,

$$\lim\sup_{N\to\infty} \Psi_k\left(\frac{B_{N,k}}{\sqrt{N}}\right) \le \alpha/K.$$

Then, similarly to the case $k=1$, we also have for any $\varepsilon>0$,

$$\lim\inf_{N\to\infty} \Psi_k\left(\frac{B_{N,k}-\varepsilon}{\sqrt{N}}\right) \ge \alpha/K$$

and by continuity of Ψ_k (which is a consequence of the continuity of the joint c.d.f. of Gaussian random variables) we conclude that $B_{N,k}/\sqrt{N}$ converges almost surely to b_k.

E.3 Proof of Lemma 4

In this proof, we denote by $E_{\sigma_{1:k}}(x)$ the expectation of the randomization law defined for some function $f : \mathfrak{S}_{2N}^k \to \mathbb{R}$ by

$$E_{\sigma_{1:k}}[f(\sigma_{1:k})] = \frac{1}{((2N)!)^k} \sum_{\sigma_1,\dots,\sigma_k\in\mathfrak{S}_{2N}} f(\sigma_{1:k}).$$

Remark that this is still random and should be differentiated from the usual expectation $\mathbb{E}$.

First, let us first handle the convergence of step k. We have,

$$\begin{aligned}
&\frac{1}{(2N)!}\sum_{\sigma_k\in\mathfrak{S}_{2N}}\mathbb{1}\left\{\sum_{j=1}^{k}t_{N,j}\left(\sigma_j\right)\leq d_k\sqrt{N}\right\}\\
&\quad=\frac{1}{(2N)!}\sum_{\sigma_k\in\mathfrak{S}_{2N}}\mathbb{1}\left\{\frac{1}{\sqrt{N}}t_{N,k}\left(\sigma_k\right)\leq d_k-\frac{1}{\sqrt{N}}\sum_{j=1}^{k-1}t_{N,j}\left(\sigma_j\right)\right\}\\
&\quad=\widehat{R}_{n,k}\left(d_k-\frac{1}{\sqrt{N}}\sum_{j=1}^{k-1}t_{N,j}\left(\sigma_j\right)\right)
\end{aligned}$$

We have, because the convergence in Proposition 1 is uniform,

$$\begin{aligned}
&\left|\widehat{R}_{n,k}\left(d_k-\frac{1}{\sqrt{N}}\sum_{j=1}^{k-1}t_{N,j}\left(\sigma_j\right)\right)-\Phi\left(\frac{1}{\tau(P,Q)}\left(d_k-\frac{1}{\sqrt{N}}\sum_{j=1}^{k-1}t_{N,j}\left(\sigma_j\right)\right)\right)\right|\\
&\leq\sup_t\left|\widehat{R}_n(t)-\Phi\left(\frac{t}{\tau(P,Q)}\right)\right|\xrightarrow[n\to\infty]{}0
\end{aligned}$$

Then, using this convergence we have that

$$E_{\sigma_{1:k}}\left[\mathbb{1}\left\{\forall j<k,\sum_{i=1}^{j}t_{N,i}\left(\sigma_i\right)\leq d_j\sqrt{N}\right\}\right]$$

converges uniformly on $d_1,\ldots,d_k$ when N goes to infinity to

$$\begin{aligned}
&E_{\sigma_{1:k}}\left[\mathbb{1}\left\{\forall j<k-1,\sum_{i=1}^{j}t_{N,i}\left(\sigma_i\right)\leq d_j\sqrt{N}\right\}\mathbb{P}\left(W_k\leq\frac{1}{\tau(P,Q)}\left(d_k-\frac{1}{\sqrt{N}}\sum_{i=1}^{k-1}t_{N,i}\left(\sigma_i\right)\right)\right)\right]\\
&=\mathbb{E}\left[E_{\sigma_{1:k-1}}\left[\mathbb{1}\left\{\begin{smallmatrix}\forall j<k-2,\sum_{i=1}^{j}t_{N,i}(\sigma_i)\leq d_j\sqrt{N},\\ \frac{1}{\sqrt{N}}\sum_{i=1}^{k-1}t_{N,j}(\sigma_i)\leq\min(d_k-\tau(P,Q)W_k,d_{k-1})\end{smallmatrix}\right\}\right]\right]
\end{aligned}$$

Then, using the induction hypothesis, this converges to Equation (6),

$$\begin{aligned}
\mathbb{E}\left[\mathbb{1}\left\{\begin{smallmatrix}\forall j<k-2,\sum_{i=1}^{j}W_i\leq\frac{d_j}{\tau(P,Q)},\\ \frac{1}{\sqrt{N}}\sum_{i=1}^{k-1}W_i\leq\frac{1}{\tau(P,Q)}\min(d_k-W_k,d_{k-1})\end{smallmatrix}\right\}\right]&=\mathbb{E}\left[\mathbb{1}\left\{\forall j<k,\sum_{i=1}^{j}W_i\leq\frac{d_j}{\tau(P,Q)}\right\}\right]\\
&=\mathbb{P}\left(\forall j\leq k,\sum_{i=1}^{j}W_i\leq\frac{d_j}{\tau(P,Q)}\right).
\end{aligned}$$

F Understanding AdaStop through simplified algorithms – 2 agents, non sequential

F.1 Comparing two agents

In this section, we present the simpler case of two agents, we call them A_1 and A_2. We denote by $\mathfrak{S}_{2N}$ the set of permutations of $\{1,\ldots,2N\}$ and for $\sigma_1,\sigma_2,\ldots,\sigma_k\in\mathfrak{S}_{2N}$, we denote $\sigma_{1:k}=\sigma_1\cdot\sigma_2\cdot\ldots\cdot\sigma_k$ the concatenation of the permutation σ_1 done in interim 1 with σ_2 done on interim 2,..., and σ_k on interim k. At interim i, we denote the concatenation of the $2N$ evaluations obtained from the two agents with $e_{1,k},\ldots,e_{2N,k}$. Then, we indicate with $e_{\sigma_i(n),i}$ the permutation of the n-th evaluation using permutation σ_i for interim i.

In the case where only two agents are compared, we use the following algorithm (see Section 4 for the multi-agent and fully developed version of the algorithm). We denote

$$T_{N,k}(\sigma_{1:k})=\left|\sum_{i=1}^{k}\left(\sum_{n=1}^{N}e_{\sigma_i(n),i}-\sum_{n=N+1}^{2N}e_{\sigma_i(n),i}\right)\right|,$$

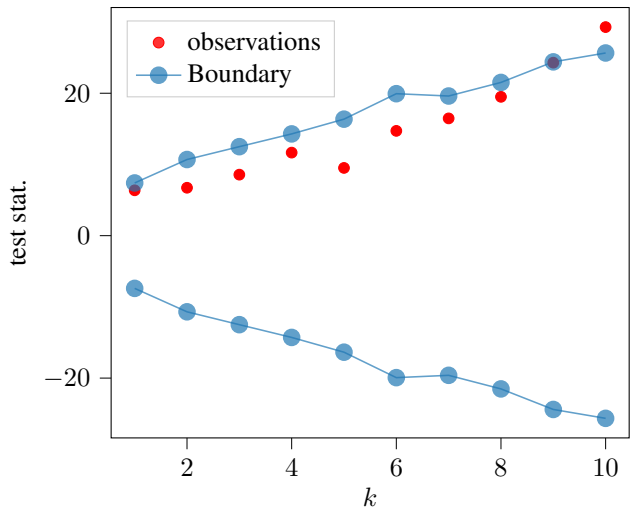

Figure 3: Illustration of the boundary for two agents.

and the boundary

$$B_{N,k} \in \inf \left\{ b > 0 : \frac{1}{((2N)!)^k} \sum_{\sigma_{1:k} \in S_k} \mathbb{1}\{T_{N,k}(\sigma_{1:m}) \geq b\} \leq \frac{\alpha}{K} \right\}, \tag{8}$$

where S_k is the set of permutations $\sigma_{1:k} \in (\mathfrak{S}_{2N})^k$ such that it would not have rejected before

$$\forall m < k, \quad T_{N,m}(\sigma_{1:k}) \leq B_{N,m} \,.$$

The algorithm is summarized in Algorithm

Algorithm 2: Adaptive stopping to compare two agents.

Parameters: Agents A_1, A_2, environment $\mathcal{E}$, number of blocks $K \in \mathbb{N}^*$, size of a block N, level of the test $\alpha \in (0, 1)$.

```
Define 2NK different seeds s_{1,1}, ..., s_{1,N}, s_{2,1}, ..., s_{2,N}.
for k = 1, ..., K do
    for i = 1, 2 do
        Train agent A_i on environment E with the seeds s_{i,kN}, ..., s_{i,(k+1)N}.
        Collect evaluations e_{1,k}(A_i), ..., e_{N,k}(A_i) using this trained agent.
    end
    Compute the boundary B_{N,k} from Equation (8).
    if T_{N,k}(id) ≥ B_{N,k} then
        Reject the equality of the agents' evaluation, break the loop.
    end
    else
        If k = K then accept, otherwise continue.
    end
end
If the test was never rejected, return accept. Else return reject.
```

An illustration of the group sequential test is given in Fig. 3. The boundary in blue is computed sequentially to have a final level $1 - \alpha$ for the test, the red points are the observed values of the test statistic (denoted w_i). The algorithm stops at the third iteration $k = 10$ because the observed value of w_i is outside the boundary, which was computed using Equation (8).

F.2 Step-down procedure

Proposed by [27], the step-down procedure is defined as follows: let $\mathfrak{S}_{2N}$ be the set of all the permutations of $\{1, \dots, 2N\}$, for a permutation $\sigma \in \mathfrak{S}_{2N}$ we define the concatenation $(e_n(j))_{1 \leq n \leq 2N}$ of the random variables being compared in hypothesis j, and the permuted test statistics of hypothesis j is

$$T_N^{(j)}(\sigma) = \left| \sum_{n=1}^{N} e_{\sigma(n)}(j) - \sum_{n=N+1}^{2N} e_{\sigma(n)}(j) \right| .$$

This test statistics is extended to any subset of hypothesis $\mathbf{C} \subset \{1, \dots, J\}$, as follows

$$\overline{T}_N^{(\mathbf{C})}(\sigma) = \max_{j \in \mathbf{C}} T_N^{(j)}(\sigma). \tag{9}$$

To specify the test, one compares $\overline{T}_N^{(\mathbf{C})}(\mathrm{id})$ to some threshold value $B_N^{(\mathbf{C})}$, that is: accept all hypotheses in $\mathbf{C}$ if $\overline{T}_N^{(\mathbf{C})}(\mathrm{id}) \leq B_N^{(\mathbf{C})}$. The threshold of the test $B_N^{(\mathbf{C})}$ is defined as the quantile of order $1 - \alpha$ of the permutation law of $\overline{T}_N^{(\mathbf{C})}(\sigma)$:

$$B_N^{(\mathbf{C})} = \inf \left\{ b > 0 : \left(\frac{1}{(2N)!} \sum_{\sigma \in \mathfrak{S}_{2N}} \mathbb{1}\{\overline{T}_N^{(\mathbf{C})}(\sigma) \geq b\} \right) \leq \alpha \right\}. \tag{10}$$

In other words, $B_N^{(\mathbf{C})}$ is the real number such that an α proportion of the values of $\overline{T}_N^{(\mathbf{C})}(\sigma)$ exceeds it, when σ enumerates all the permutations of $\{1, \dots, 2N\}$.

Algorithm 3: Multiple testing by step-down permutation test.

Parameters: $\alpha \in (0, 1)$
Input: $e_n(j)$ for $1 \leq n \leq 2N$ and $j \in \mathbf{C}_0 = \{1, \dots, J\}$.

```
Initialize C ← C_0.
while C ≠ ∅ do
    Compute T_N^(C)(σ) for every j and every σ using Equation (9).
    Compute B_n^(C) using Equation (10).
    if T̄_N^(C)(id) ≤ B_N^(C) then
        Accept all the hypotheses H_j, j ∈ C and break the loop.
    else
        Reject H_{j_max} where j_max = arg max_{j∈C} T_N^(j)(id).
        Define C = C \ {j_max}
    end
end
```

The permutation test is summarized in Algorithm 3. It is initialized with all the hypotheses to test in $\mathbf{C} = \mathbf{C}_0$. Then, it enters a loop where it decides to reject or not the most extreme hypothesis in $\mathbf{C}$, the set of not yet discarded, nor accepted hypotheses. If the test statistic $T_N^{(j)}(\mathrm{id})$, for the most extreme hypothesis in $\mathbf{C}$ (i.e. $\overline{T}_N^{\mathbf{C}}(\mathrm{id})$), does not exceed the given threshold $B_n^{(\mathbf{C})}$, then it accepts all the hypotheses in $\mathbf{C}$ and breaks the loop. Otherwise, it just rejects the most extreme hypothesis and it discards that from the set $\mathbf{C}$. Then, it enters the loop again until the set of remaining hypotheses is empty.

The maximum of the statistics in Equation (9) for $\sigma = \mathrm{id}$ allows to test an intersection of hypotheses, while, the threshold $B_n^{(\mathbf{C})}$, because of the equality of distribution hypotheses, allows to have strong control on the FWE (i.e. $\mathrm{FWE} \leq \alpha$). In fact, this procedure is not specific to permutation tests, and it can be used for other tests provided some properties on the thresholds $B_n^{(\mathbf{C})}$.

Remark (Non-independent hypothesis)**.** *The acute reader may have noticed that as the hypotheses are not assumed to be independent, we can not resort to Benjamini-Hochberg or similar procedure [3] here. We adapt this procedure to the case of group sequential testing later in Section 4*

G On early accept in AdaStop

In this section, we present in details the early-accept heuristic proposed to speed up computation of AdaStop. Let $\mathbf{C} \subset \{1, \dots, J\}$ be a subset of the set of comparisons that we want to do, denote

$$\overline{T}^{(\mathbf{C})}_{N,k}(\sigma_1^k) = \max\left(T^{(j)}_{N,k}(\sigma_1^k), \quad j \in \mathbf{C}\right) \quad \text{and} \quad \underline{T}^{(\mathbf{C})}_{N,k}(\sigma_1^k) = \min\left(T^{(j)}_{N,k}(\sigma_1^k), \quad j \in \mathbf{C}\right)$$

$$\overline{B}^{(\mathbf{C})}_{N,k} = \inf\left\{ b > 0 : \frac{1}{m_k} \sum_{\sigma \in \widehat{S}_k} \mathbb{1}\{\overline{T}^{(\mathbf{C})}_{N,k}(\sigma_1^k) \geq b\} \leq \overline{q}_k \right\} \tag{11}$$

and

$$\underline{B}^{(\mathbf{C})}_{N,k} = \sup\left\{ b > 0 : \frac{1}{m_k} \sum_{\sigma \in \widehat{S}_k} \mathbb{1}\{\underline{T}^{(\mathbf{C})}_{N,k}(\sigma_1^k) \leq b\} \leq \underline{q}_k \right\}. \tag{12}$$

where $\sum_{j=1}^k \underline{q}_j \leq \frac{k\alpha}{K}$ and $\sum_{j=1}^k \overline{q}_j \leq \frac{k\beta}{K}$ and where $\widehat{S}_k$ is the subset of $\mathcal{S}_k$ such that it would not have accepted or rejected before: for each $\sigma_1^k \in \widehat{S}_k$, we have the following property

$$\forall m < k, \quad \overline{T}^{(\mathbf{C})}_{N,m}(\sigma_1^k) \leq \overline{B}^{(\mathbf{C})}_{N,m} \text{ and } \underline{T}^{(\mathbf{C})}_{N,m}(\sigma_1^k) \geq \underline{B}^{(\mathbf{C})}_{N,m}.$$

In AdaStop, modify the decision step (line 10 to 15 in Algorithm 1) to The resulting algorithm have a

Algorithm 4: Early accept

if $\overline{T}^{(\boldsymbol{C})}_{N,k}(\mathrm{id}) > \overline{B}^{(\boldsymbol{C})}_{N,k}$ **then**
  Reject $H_{j_{\max}}$ where $j_{\max} = \arg\max\left(T^{(j)}_{N,k}(\mathrm{id}), \quad j \in \mathbf{C}\right)$.
  Update $\mathbf{C} = \mathbf{C} \setminus \{j_{\max}\}$
else if $\underline{T}^{(\boldsymbol{C})}_{n,k}(\mathrm{id}) < \underline{B}^{(\boldsymbol{C})}_{N,k}$ **then**
  Accept $H_{j_{\min}}$ where $j_{\min} = \arg\min\left(T^{(j)}_{N,k}(\mathrm{id}), \quad j \in \mathbf{C}\right)$.
  Update $\mathbf{C} = \mathbf{C} \setminus \{j_{\min}\}$

small probability to accept a decision early, and as a consequence it may be unnecessary to compute some of the agent in the subsequent steps.

As illustration of the performance of early accept, if one was to execute AdaStop with early parameter $\beta = 0.01$ for the Walked2D-v3 experiment from Section 5.3, the experiment would stop at interim 2 and 10 seeds would have been used for each agent. By comparison, in Section 5.3 we showed that without early accept, Adastop uses 30 seeds for DDPG and TRPO. Early stopping give in this instance a consequent speed-up without affecting the final decisions.

H Implementation details and additional plots

H.1 Plot of the census on the number of seeds in ICML 2022 RL articles

In Fig 4, we present the results of the census we did on the number of seeds used in the RL articles from ICML 2022. We plot only the articles having experiment on environments using Mujoco as this represent the majority of the articles.

H.2 The distribution of evaluation performances in Deep RL problems

In Fig. 5, we represent the densities of the evaluations (cumulative rewards) of some RL agents on the Hopper environment (see Section 5.3 for the details on the experiment). The p-values for the Shapiro normality test on these data are: 0.0032 for PPO[5], 0.0003 for SAC, 0.0393 for DDPG and 0.015 for

[5]Here, PPO, SAC, DDPG, and TRPO should be read as "an agent that implements xxx".

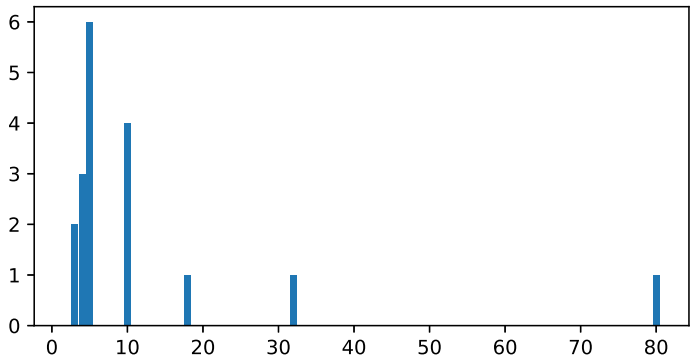

Figure 4: Census of the number of seeds used in RL articles published in ICML 2022 proceedings to study environments using Mujoco.

TRPO. As all p-values are less than $\alpha = 0.05$, we reject the null hypothesis of normality for at least one of them. This confirms what our eyes already told us: the performance of Deep-RL agents is not normally distributed (at least on this example).

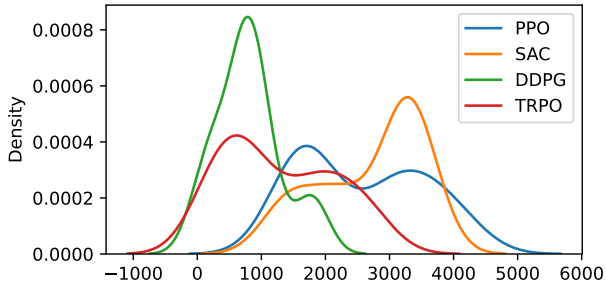

Figure 5: Empirical density of the performances of various agents on Hopper Mujoco environment using 15 seeds for each agent.

H.3 Complementary experiment for Section 5.1

In this third example, we suppose we have 10 agents whose performance distributions are listed in Fig. 6, where the first column indicates the labels of the agents as they are shown in Fig. 7.

Similarly to Cases 1 and 2 (see Section 5.1), we execute AdaStop with $K = 5$, $N = 5$, $\alpha = 0.05$ and the maximum number of permutations $B = 10\,000$. In contrast to Cases 1 and 2, in Case 3 we use early accept (with $\beta = 0.01$) to avoid situations when all agents are run with all NK seeds, which may occur when each agent has a similar distribution to at least one another agent in comparison.

We show the performance of AdaStop for multiple agents' comparison in Fig. 7, which corresponds to the output of one execution of AdaStop (with labels summarized in Fig. 6). The table (left) summarizes the decisions of the algorithm for every pair of comparisons, and violin plots (right) reflect empirically measured distributions in the comparison. From this figure, we can see that almost all agents are grouped in clusters of distributions with equal means, except for *MG3 that is assigned to two different groups at the same time. Interestingly, these clusters except for *MG3 are correctly formed. Moreover, similarly to two previous cases, we have executed AdaStop $M = 5\,000$ times to measure FWE of the test. The empirical measurements are 0.0178 of rejection rate of at least one correct hypothesis when comparing distributions and 0.0472 of rejection rate when comparing means, both are below 0.05. Thus, AdaStop can be efficiently used to compare performances of several agents simultaneously.

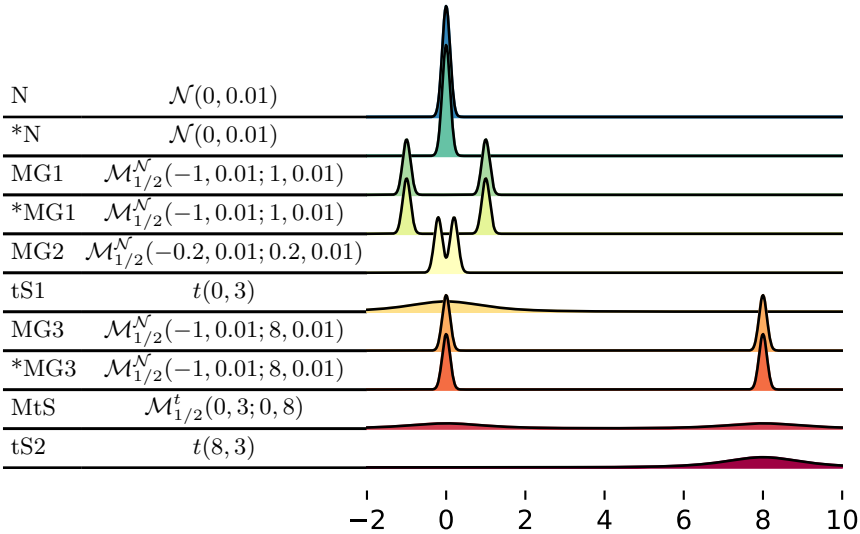

Figure 6: Toy example 3, with an illustration of the involved distributions.

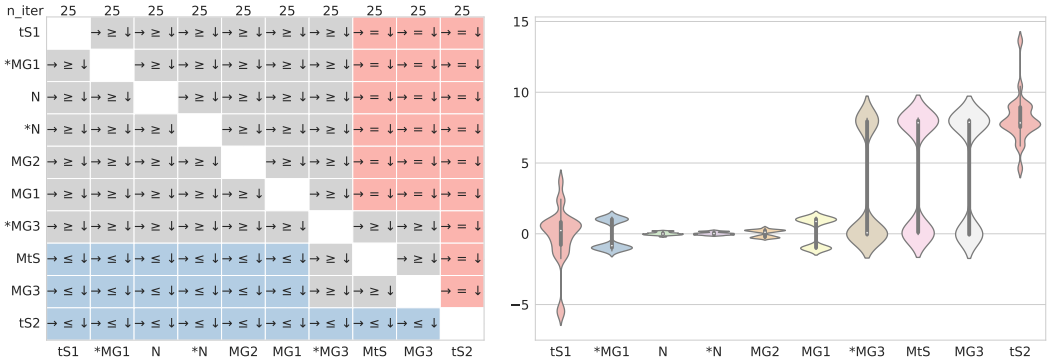

Figure 7: Case 3. AdaStop decision table (left) and measured empirical distributions (right).

H.4 Additional plot for Section 5.2

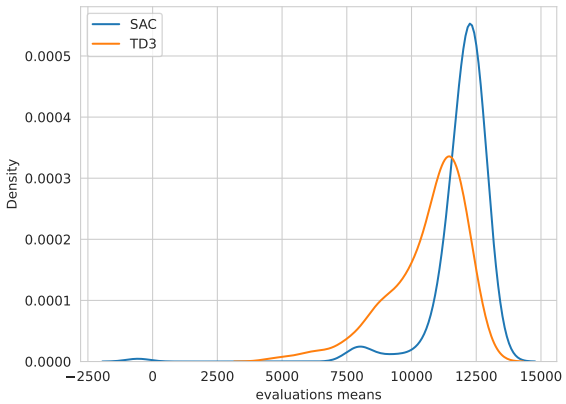

Figure 8: Evaluations distributions for a SAC and a TD3 agents on HalfCheetah obtained with 192 independent seeds, each made of 2 million steps.

H.5 MuJoCo Experiments

	DDPG	TRPO	PPO	SAC
γ	0.99	0.99	0.99	0.99
Learning Rate	1×10^{-3}	1×10^{-3}	3×10^{-4}	3×10^{-4}
Batch Size	128	64	64	256
Buffer Size	10^6	1024	2048	10^6
Value Loss	MSE	MSE	AVEC [12]	MSE
Use gSDE	No	No	No	Yes
Entropy Coef.	-	0	0	`auto`
GAE λ	-	0.95	0.95	-
Advantage Norm.	-	Yes	Yes	-
Target Smoothing	0.005	-	-	0.005
Learning Starts	10^4	-	-	10^4
Policy Frequency	32	-	-	-
Exploration Noise	0.1	-	-	-
Noise Clip	0.5	-	-	-
Max KL	-	10^{-2}	-	-
Line Search Steps	-	10	-	-
CG Steps	-	100	-	-
CG Damping	-	10^{-2}	-	-
CG Tolerance	-	10^{-10}	-	-
LR Schedule	-	-	Linear to 0	-
Clip ϵ	-	-	0.2	-
PPO Epochs	-	-	10	-
Value Coef.	-	-	0.5	-
Train Freq.	-	-	-	1 step
Gradient Steps	-	-	-	1

Table 3: Hyperparameters used for the MuJoCo experiments.

In this section, we go into details about the experimental setup of the MuJoCo experiments, as well as present additional plots.

Basic properties of chosen algorithms. On-policy algorithms, such as PPO and TRPO, update their policies based on the current data they collect during training, while off-policy algorithms, such as SAC and DDPG, can learn from any data, regardless of how it was collected. This difference may make off-policy algorithms more sample-efficient but less stable than on-policy algorithms. Furthermore, SAC typically outperforms DDPG in continuous control robotics tasks due to its ability to handle stochastic policies, while DDPG restricts itself to deterministic policies [15]. Finally, PPO is generally accepted to perform better than TRPO in terms of cumulative reward [10].

Hyperparameters. Table 3 details the hyperparameters used with each Deep RL on the MuJoCo benchmark. For all agents, we use a budget of one million time steps for HalfCheetah-v3, Hopper-v3, and Walker2d-v3, and a budget of two million time steps for Ant-v3 and Humanoid-v3. Finally, we use a maximum horizon of one thousand steps for all environments.

Evaluation. Agents are evaluated by stopping the training procedure on predetermined time steps and averaging the results of 50 evaluation episodes.

Learning Curves. Fig. 10 presents sample efficiency curves for all algorithms in each environment. The shaded areas represent 95% bootstrapped confidence intervals, computed using `rliable` [1]. Note that each curve may be an aggregation of a different number of runs, which can be found in the bottom right of Fig. 9.

Additional Comparison Plots. Fig. 9 expands upon the comparisons given in the main text (in Fig. 2) by also plotting the evaluation distributions of each agent using boxplots.

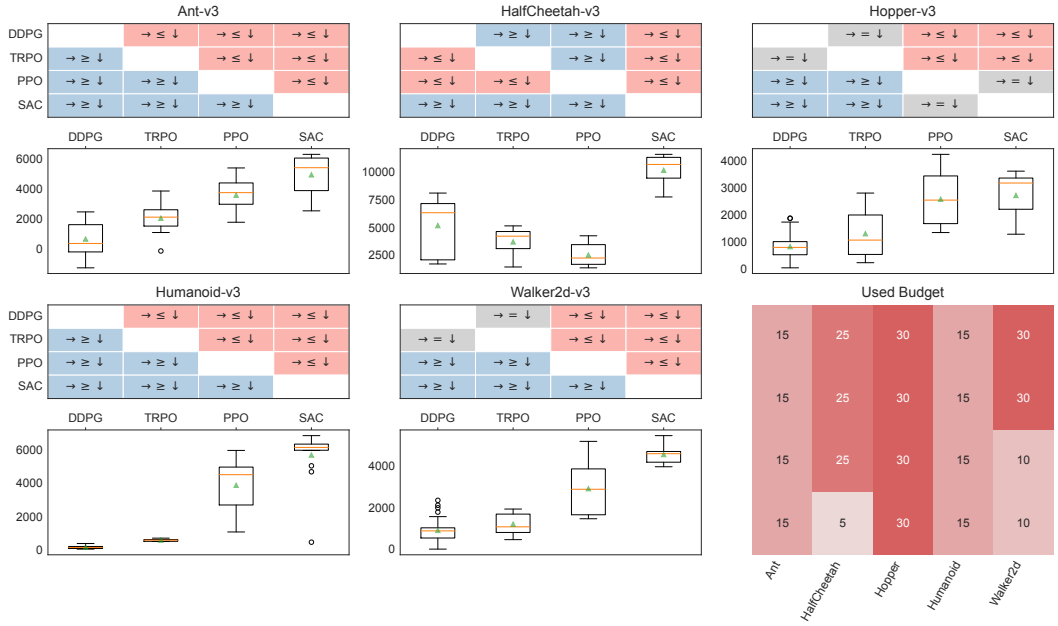

Figure 9: AdaStop decision tables (top) and evaluation distributions (bottom) for each MuJoCo environment, and the budget used to make these decisions (bottom right). The medians are represented as the green triangles and the means as the horizontal orange lines.

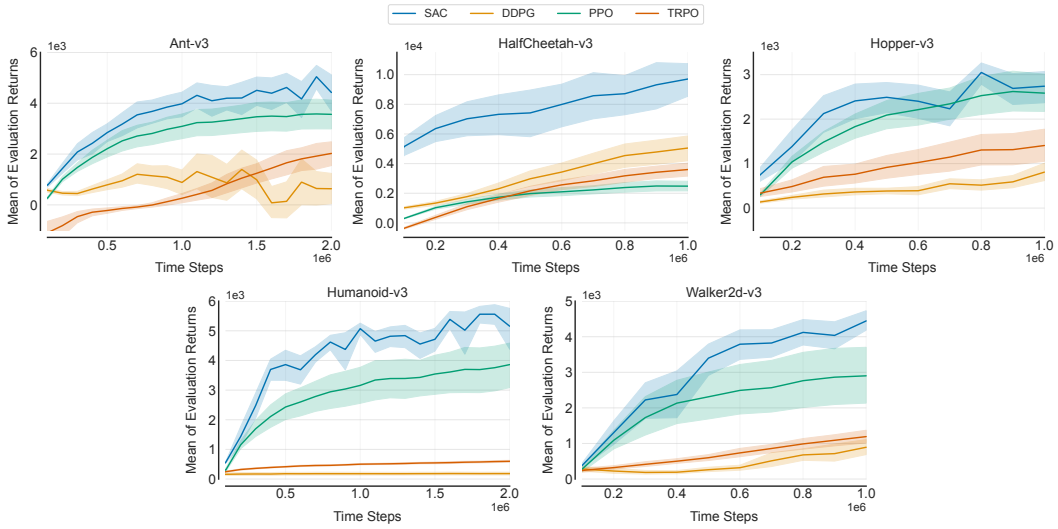

Figure 10: Mean of Evaluation Returns with 95% stratified bootstrap CIs. Note that curves in the same figure may use a different number of random seeds, depending on when AdaStop made the decisions.