

1 We are glad that our reviewers agree on the merits and relevance of our work. We are grateful for the constructive
2 comments and will incorporate them into the final version of our submission. We would also like to reassure our
3 reviewers, that we will make an implementation of our work available upon publication.

4 **R3/R4: Applying Freeze-Thaw BO in the settings considered.** While both the exponential decay in Freeze-Thaw (FT)
5 and our compression function encode preferences regarding training development, there is an important distinction
6 between the two approaches. FT utilises the exponential decay property to *terminate* the training curve, while BOIL
7 only uses the sigmoid curve to *guide* the search. See Fig. 1 for further illustration of why FT struggles in DRL settings.

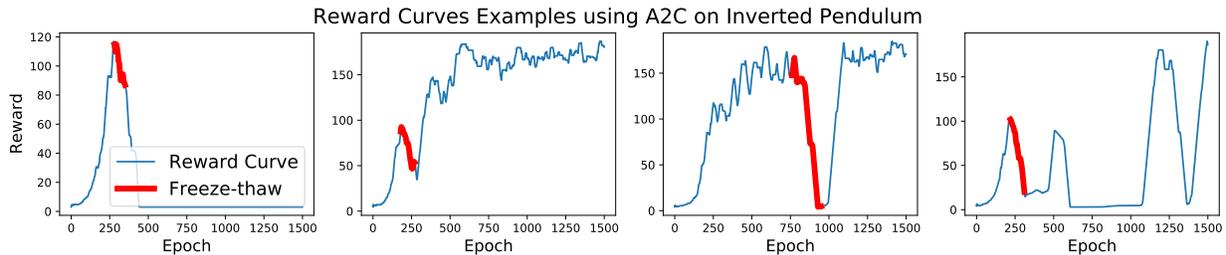


Figure 1: Illustration of Freeze-thaw (FT) in DRL. FT will terminate training processes when training performance (in blue) significantly drops (i.e. at the red locations) as the *exponential decay* model will predict low final performance. In most RL environments noisy training curves are unavoidable. Thus, FT will dismiss all curves including good settings, never completing a single training run before the final epoch.

8 **R3: Comparison to Fobolas.** Fobolas uses a different way of obtaining low-fidelity information. In particular, it
9 uses subsets of the training data to estimate performance on a full data set. This is *orthogonal* to our approach—a
10 combination of the two might make for interesting future work. In the context of DRL, however, it is unclear how it
11 would apply, as no fixed set of training data exists. The CM-T/F-BO work which we compare against can be seen as an
12 adaptation of Fobolas to run-time fidelity.

13 **R3: Sec 3.2 and 3.3 should be reversed as Sec 3.2 makes reference to Eq (7).**

14 We will revise the presentation to improve the cohesion across sections in the final version.

15 **R3: Different versions of Hyperband.** We have used the original implementation of Hyperband [Li et al 2018]. We
16 can consider the variant of [Falkner et al 2018]. However, we would highlight that (1) our curve compression idea is
17 unique and (2) we can extend their approaches using our compression strategy in future work.

18 **R1/R3: Regarding the issue of ill-conditioning and its connection to kernel choice** The ill-conditioning problem can
19 occur whenever sampled points give near-identical rows in the kernel Gram matrix. For stationary kernels, this is
20 usually the case for tightly-clustered data points. While the SE kernel does pose particular conditioning problems, we
21 found that the problem persisted across all popular kernel choices, including the Matérn class. Different solutions, such
22 as the addition of artificial noise or altering the kernel’s lengthscales, have been proposed. The active learning approach
23 was chosen as sampled data points are expected to contain a lot of redundant information. As a consequence, the loss of
24 information from sub-sampling the data should be minimal and information-eroding modification of the kernel matrix
25 itself can be avoided. As an added benefit, the reduced number of sampled points speeds up inference in our GP models.

26 **R3: How is the reward curve considered in the CNN case? Was it also compressed using a logistic curve?**

27 Training performance in the CNN experiments is measured as the predictive performance on test data after a number of
28 SGD training steps. The resulting performances curves are compressed using a logistic curve whose parameters are
29 estimated as described and shown in Fig. 5 of the supplement.

30 **R2: In Fig 6, the performance of BOIL is marginally better than other methods.**

31 While other algorithms achieve similar final performance to BOIL, they typically require more time to do so and are
32 less robust across varying problem settings as shown in the right column of Fig 6. (Note that the left column shows
33 performance over training iterations - not real time.)

34 **R4: Computational complexity of the BOIL algorithm.** In the settings considered, the overhead of BOIL’s computation
35 is significantly outweighed by the cost of executing the actual learning algorithm. For instance, a full training run on
36 the CartPole environment takes about 600 secs, whereas the matching BOIL iteration required about 1 sec of real time.

37 The highest computational cost of BOIL is incurred by inference for the GP models. This cost scales cubically in the
38 number of observed data points. However, due to the high cost of collecting data, we would not expect this to become a
39 problem. Alternatively, sparse GP approximations can alleviate such problems and scale to millions of data points.