First of all, we thank all reviewers for their valuable time and feedback. Reflected in this years reviewing process, reproducibility is of central importance to the whole NeurIPS community and was also unanimously identified during the panel discussion at the AutoML workshop @ ICML19 as one of the major challenges the AutoML community has to face. Unfortunately, hyperparameter optimization (HPO) often requires tremendous computational resources which renders reproducibility hard in practice, since one can only afford a few function evaluations. As an important step to enable better reproducibility we provide a principled way to generate cheap-to-evaluate benchmarks which contain the typical characteristics of real HPO problems.

We thank the reviewers for pointing out typos and grammatical errors, which we of course have fixed now. We will not address these any further here and proceed by addressing the reviewers' comments in turn.

**R1:** We are afraid that the reviewer might have misunderstood some parts of the paper. The goal is **not** to speed up Bayesian optimization, such as warm-starting or multi-fidelity optimization, but instead to provide a cheap-to-evaluate and realistic **benchmark suite** for hyperparameter optimization methods. This allows the community to execute exhaustive experiments with a low computational budget and to easily compare to existing methods, which is necessary to make HPO more reproducible. We have made this claim clearer in a revised version of the manuscript to avoid confusion. We strongly believe that methods able to tackle the reproducibility problem are essential in modern machine learning, and we hope our clarifications will help the reviewer to support our contribution in this direction.

1. Figure 1 shows the XGBoost benchmark with 8 hyperparameters described in Section 4.1. Due to space constraints further details can be found in Appendix A (as referenced in the main paper).

2. In a nutshell, we first learn a latent embedding across optimization tasks together with a generative multi-task model that allows us to sample an infinite amount of new optimization tasks which resemble the original ones. We have added pseudocode to make the proposed method more tangible.

3. Figure 3 visualizes the learned latent space and shows that our embedding indeed captures similarities across tasks (see also Section 5.1 for further details).

**R2:** We thank reviewer 2 for the constructive feedback:

About the methodology: The way we learn the latent embedding is straightforward and follows the general GPLVM framework, which, given a matrix with all observed target values across all tasks, learns a latent Gaussian distribution for each task. We refer to the original paper for further details about the approximation of the variational posterior. The subscript n indicates the datapoint (where N is the total number of datapoint) and h indicates the sample drawn from the latent distribution over tasks provided by our embedding. We have made this more clear in the main paper now.

About the experiments: We would love to conduct the same analysis that we did for the Forrester function in Section 5.2 also for real HPO problems. However, this is (i) computationally impossible (and can only be conducted using Profet) and (ii) we do not have access to any HPO problem where 1000 real tasks (or datasets) are available.

The hyperparameters for BOHAMIANN (together with the hyperparameters for all other methods) are, due to space constraints, described in Appendix E and follow the default parameters proposed by Springenberg et al. Note that, consistent with our results, also in the original paper by Springenberg et al. BOHAMIANN was outperformed by BO-GP in low dimensional continuous problems (for example see Figure 1 in Springenberg et al.) and seems to improve upon BO-GP if the dimensionality increases.

About Section 5.2: the reason why results with 1000 generative tasks stick more to the result to 1000 original tasks than the subset of 9 tasks is because our generative model captures the variability of tasks. We added further details.

**R3:** We thank reviewer 3 for the helpful feedback and agree that the benchmarks will be key for researchers working in black-box or hyperparameter optimization. Indeed, it is surprising that the community hasn't yet produced a lot of research in this direction, ML being a discipline that is being applied in such a long list of real applications. Many thanks also to the proposed improvements, which we found very helpful. While we think they are out-of-scope for this paper, we actually plan to include them into future work.

1. Indeed, we also think complex search space are interesting and having a benchmark suite would enable future work to tackle these spaces.

2. We have added a discussion about PBT and Hyperband in the related work section. Note that, we are planning to extend our benchmarks to also model fidelities of the objective function in order to apply multi-fidelity methods, such as Hyperband or BOHB.

3. Having different kinds of noise is indeed a good idea. Since our multi-task model is a Bayesian neural network, it would be possible to adapt the likelihood and the predictive distribution to allow for other noise models, such as Student'T distributions.