We thank the reviewers for their time and productive comments. However, there is one misunderstanding that might have affected the reviewers' scores: reviewers 1 and 3 have both misunderstood the baseline. When compared like-for-like, our results outperform the baseline by a large margin (details below). While this misunderstanding is surely a shortcoming of our presentation, the reviewers criticised that we did not improve on baseline performance, when we in fact did. Leaving the misunderstanding aside, reviewers found the work to be "really well written and every decision is well motivated" (Reviewer #3), found that the proposed implementation "greatly improves the efficiency for generating random subspace at each training step" (Reviewer #2) and wrote that the contributions are "novel relative to prior work [...] and can inspire future work in the area" (Reviewer #1) without mentioning any other major concern. Given a correct understanding of the baseline, it seems likely that their overall scoring would have been more positive.

To expand further on the misunderstanding, Reviewers #1 and #3 pointed out that our results did not seem to be consistent with the results published by Li et al. [1]. The misunderstanding most likely stems from the fact that Li et al. [1] reported the achieved accuracy as a percent value relative to the SGD baseline while we reported the absolute percent accuracy, without normalising against the SGD baseline. In particular, for a 20x reduced CIFAR-10 LetNet, [1] reported 90% of the original 58% SGD accuracy which amounts to an $0.9 \cdot 57\% = 51.3\%$ accuracy in absolute terms (see Figure S14b in [1] which presents the absolute accuracies). This is consistent with our reported 58.35% accuracy for a 10x reduced Resnet-8-CIFAR-10 in Table 1, where the 7% improvement can be attributed to increased efficiency of the ResNet architecture, as Reviewer #3 expected. Similarly, our MNIST baseline of 80% reported in Figure 1 for d=250 is consistent with the absolute accuracy reported in Figure S6 in [1] (note that the network dimensionality is D=100K, but the subspace dimensionality is d=250 only). We realize now that our reporting should have made this subtle issue of the percent notation in [1] clearer. We will add the relative accuracy levels to the manuscript to ease the direct comparison with prior art.

Minor: All suggested improvements are gratefully received and we will incorporate the feedback into our revision.

* Reviewer #2 asked whether the substantial IPU hardware speedup over CPUs was due to the accelerated PRNG or could be merely explained by the forward-backward pass acceleration.

> While the IPU accelerates the forward-backward pass of the network, we found that the main bottleneck on CPU hardware is indeed the PRNG (particularly for large subspace dimensions d > 1000). To rule out the possibility that the measured speedup can be attributed to the forward-backward acceleration only, we benchmarked the throughput of our implementation on a GPU V100 accelerator that, unlike the IPU, does not have an on-chip PRNG. We found that the GPU provided no throughput improvement relative to the CPU baseline. We will include these additional results in the respective Section 4.2.

* Reviewer #2 noted that the throughput "31 images per second" does not match with the "100 epochs / 67 minutes" statement in Section 4.2.

> Thank you, this is a mistake. We accidentally mixed the images per second throughput for d=10k with the wall-clock time figure for d=1k. The correct throughput for d=1k is 1366 and 112 images per second on IPU and CPU respectively.

* Reviewer #1 asked if there was "any comparison to FPD approach in terms of parallelization?"

> Both approaches can be parallelized in the same way since FPD can be seen as a special case of our algorithm where $\varphi_t \equiv \varphi_0$. Our more efficient distributed implementation can thus be seen as a technical contribution that can also benefit the investigations of intrinsic dimensionality in [1]. We will update the discussion to point this out.

* Reviewer #3 asked whether the low performance of the NES baseline stems from a small number of random samples.

> Indeed, the NES baseline in Figure 2 used the same very low-dimensional number of d=250 samples, while more samples would certainly increase the approximation quality. The low-dimensional comparison at d=250 demonstrates the superiority of gradient based RBD optimization over NES black-box sampling in this setting. We will adjust the caption of Figure 2 to underline this point.

* Reviewer #3 asked about a "potential bottleneck in the distributed version of the training, as all the random numbers need to be generated on the main worker".

> This is an good observation that motivates a trade-off between increased compute through PRNG versus reduced communication between workers. Notably, however, our implementation does not require a central main worker but the PRNG generation can be shared between workers in a decentralised way to load balance potential PRNG bottlenecks (see Algorithm 1, right).

[1] Chunyuan Li et al. "Measuring the Intrinsic Dimension of Objective Landscapes". In: Sixth International Conference on Learning Representations. 2018. URL: https://openreview.net/forum?id=ryup8-WCW.