We thank the reviewers for their detailed reviews and constructive feedback. Below we respond to each review as much as space allows, to provide clarification on points of confusion and answer the questions raised. We are grateful to see that the responses are unanimously positive and we hope this work will be beneficial to the field as a whole.

Reviewer #1

Improvement of ALEBO: Thanks for raising this. On stationary problems with low-d structure, the magnitude of improvement is large (Fig. 5). Now, as we show, real-world problems can be more complicated and while the improvement over REMBO remained large, local-search methods were highly competitive. However, there are still settings where linear embedding BO (and thus ALEBO) would be the best choice. The promise of embedding BO is that all of the BO machinery developed over the years can be applied directly to HDBO. For instance, BO can maintain high sample efficiency with high parallelism (e.g., 100 total iterations spread across 25 workers, where iterations take hours or days). The same is not true for local search methods, including TuRBO, which requires sequential iterations to move the trust region. Other settings where BO is not matched by local search include cost-aware, multi-task, and multi-fidelity, to name a few. We will add discussion of this in the extra page.

Selecting d_e: This is a great point we will discuss in more detail. In some problems (e.g. robot locomotion) there is domain knowledge. Practically, the evaluation budget will be an important factor: 500 function evaluations will support a higher embedding dimension than 25. Sensitivity is explored in S9, and ALEBO is shown to be better than prior work.

Constraint on NASBench: See R3. *Supplemental*: Thanks for the suggestion, we will update to improve clarity!

Reviewer #2

Clarifications: Thanks for pointing these out, we will clarify them. k=4 is the recommendation made in the REMBO paper, which does some sensitivity analysis. L213: the random subspace will not be axis aligned w.p. 1.

Selecting d_e: This was also brought up by R1 and is clearly a topic of importance, which has not been thoroughly explored by the embedding BO literature. The Mahalanobis kernel can be sample-efficient despite the quadratic number of hyperparameters parameters because of the posterior sampling, which avoids overfitting (Fig. S2). The optimization in Fig. 5(center) used d_e=12, yet had excellent performance already at 25 iterations. We will add discussion of this.

Kernel evaluation: Prop. 1 gives a generative model for the kernel starting from a d-dim ARD RBF. We will add the requested comparison; from the theoretical result in Prop. 1 there is little reason to doubt its performance.

Reviewer #3

On performance: Thanks for the review, we agree that one conclusion of the paper is that linear embedding BO is not appropriate in every case. But we do want to highlight that there are other reasons why one might still favor linear embedding BO over methods like local search (CMA-ES, TuRBO) that performed strongly in our results; see the response to R1, which describes high parallelism and multi-fidelity optimization as two such settings.

NASBench: Real problems of interest to us have constraints, and CMA-ES and TuRBO do not guarantee constraint satisfaction. We added results where we apply them via low objective for infeasibility, and ALEBO remained best.

Constrained BO: The biggest benefit of linear embedding BO is the ability to directly apply existing BO techniques. In Fig. 6, we actually did use the constrained EI of Gardner et al. 2014; this is described in Sec. S5. Random embeddings are especially useful for constrained BO because we can maintain the same embedding for all outcomes. The method is agnostic to the acquisition function, and cPES or cMVES could be used just as easily. We'll move this to the main text.

Nonlinear embeddings: Thanks for the suggestion, we will add discussion of this in the extra page. In short, the main findings all apply to the nonlinear case. A GP must be able to fit well in the embedding. End-to-end training a VAE to include GP likelihood is an important first step, but then the same considerations apply for handling box bounds and maintaining optima in the embedding. We will discuss potential extensions of our solutions.

Reviewer #4

MOO: Thanks for the suggestion. As discussed above, a benefit of linear embedding BO is that techniques like MOO can be directly applied. Similar to how constraints are handled in Sec. S5, we would evaluate multiple objectives in the embedding and use a MOO acquisition function. We will add discussion of this.

Popt: Thanks for raising this, we can increase clarity around this. The constrained space is not guaranteed to contain an optimum; this is the Popt evaluated in Sec. 5. Under the problem prior used there, Popt for ALEBO is higher than for HeSBO. REMBO can use a larger space by clipping to the boundaries, but this makes the function harder to model, and so it is harder to find the optimum even if it is in the space.

Prop 1: The Mahalanobis kernel is specific for ARD RBF, but the corresponding result for a stationary kernel is that stationary in the true space implies stationary in the embedding (a result that does not hold with clipping to box bounds).
CIFAR: See R3; it will be added.