We would firstly like to thank the reviewers for their time. We are particularly excited to see that even the reviewers were inspired to propose a variety of extensions and improvements in their feedback, which shows the interest this paper may generate at NeurIPS. In addition, we are pleased to see our algorithm is also capable of producing diverse reviews. Given the green field nature of this project, we have some additional insights to provide since the submission.

New RL experiments We extended our RL experiments to the approximate setting where gradients are computed using samples. We use an Actor-Critic algorithm with a tabular policy, and compute Hessians using the DiCE operator (Foerster, 2018). The results show that RR works not only in the exact setting but also when gradients are approximate. We believe this is a key step towards eventually using RR for deep RL. We believe these results improve the potential scalability of our approach, and will be included in the CRC. Below we address individual comments in more detail:

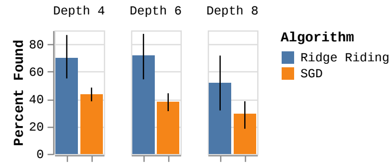

Figure 1: % solutions found per algorithm, by tree depth, each is randomly generated 10 times to produce error estimates shown.

R1: **Saddle vs local maximum** When we use the word 'saddle' we mean a stationary point with at least one negative curvature direction, which includes maxima. **method ..not clearly defined .. even in authors' minds.** We provide a method section, multiple instances of pseudo code and implementations that closely match the pseudo code.

R2: **computation of the Hessian spectrum should be $O(m^3)$..requires a spectral computation EACH timestep..complexity analysis** We addressed this in the section labeled *Approximate Ridge Riding* There, we formulated a version of RR that only requires Hessian-Vector products, which can be computed with $O(m)$ complexity in modern auto-diff frameworks. As explained, we also use an iterative process for updating the EV at each timestep rather than recomputing it. The overall complexity will depend on the search method being used, but each solution will required $O(m) * O(N)$ compute, where N is the number of update steps. The memory is also dependent on the search method, but will be $O(m)$ for depth-first. **Theorem 1 assumption.** One can use Hessian-Vector products to compute $\langle \nabla L(\theta), e_i(\theta) \rangle$ and estimate γ at any θ. This can be used to set the learning rate. **minima not better than SGD.** We show in both the zero-shot coordination problem and the out-of-distribution that RR can obtain better solutions than SGD. Clearly though, in general this will be problem specific and all we can do is provide a method that can obtain more diverse solutions. **cannot see how symmetry and equivalence relate to algorithm.** This is exploited and described in the zero-shot coordination setting and in the out-of-distribution part. In general symmetries can be used to make the search more efficient by only exploring one EV from each equivalence group.

R3: Many thanks for the encouraging, insightful and positive review! **ChooseFromArchive** The best way to search is problem dependent, so we specify ChooseFromArchive separately in the method sections for each of our applications and extensions of RR. **Many pre-print cited** We apologize. That there are 18 papers in the paper which look like 'pre-prints' is an artifact of us using the default Google Scholar bibtex. Out for these 18, 3 are journal papers, 9 are top-tier conference papers, and only 6 are actual pre-prints. Of those, 4 are from 2020, one is the Tensorflow paper with >5k citations, and the last is the well-known and relevant MAP-elite algorithm. We will correct the citations in the Camera Ready Copy (CRC).

R7: Thank you for your interesting comments. We are glad you appreciate the novelty of our work in the ZS setting and note the strength of the diversity in RL experiments. Regarding **RR not working "in generality"**, we feel this is a high bar for any algorithm. While we show that our algorithm is applicable and useful in a wide array of diverse settings, we clearly can't promise that it will find all solutions in all settings (there is probably an impossibility result somewhere to be written down here). **ridges vs. random directions** We tried to address this in Fig 2 (in the paper) with a baseline that follows random (unit) vectors instead of EVs (*Rand. Ridge*). To evaluate your proposal, we include two additional baselines: (1) following ridges, but not updating them (*Fixed-EV*). (2) following random unit vectors with *positive ascent direction* (*Rand-Ridge+*), and compare vs. Ridge Riding. We ran these ablations both on the RL experiment (exact RR) and on MNIST, where we used a fixed budget hyperparameter search for approximate RR and the ablations. As shown in Fig 2, RR on MNIST significantly outperforms all ablations. Fixed-EVs obtain a top accuracy of a linear classifier (92%), compared to the 98% obtained by RR. In contrast, none of the suggestions using random directions exceed 30%. In the low dimensional RL example, Fixed-EVs obtains competitive performance. This clearly illustrates the importance of following EVs rather than random directions. We furthermore observe all differences to be more pronounced in MNIST, which is intuitive since random search is known to scale poorly to high dimensional problems. We expect this effect to be even more pronounced as Approximate RR is applied to harder and higher dimensional tasks in the future. We once again thank the reviewers for these suggestions and will update the paper to include all plots shown here.

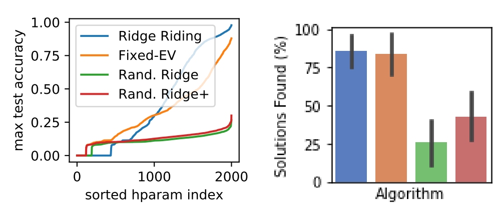

Figure 2: Ablations on ridge riding algorithm. Left: We perform a hyperparameter search for all methods on MNIST and show best performance found. Right: The same four methods for diversity in RL, with tree depth of 12, 5 seeds. Legend applies to both plots.