Thank you for all your comments and corrections; they will be incorporated into our revised version.

All reviewers. The paper focuses on theoretical guarantees, however we agree providing more experiments can be helpful. We plan to add (a) An example using general retraction—to show the generality of our approach (see an illustration below); (b) to compare the algorithm's rate on two manifolds with different curvature—to illustrate how curvature affects the rate empirically.

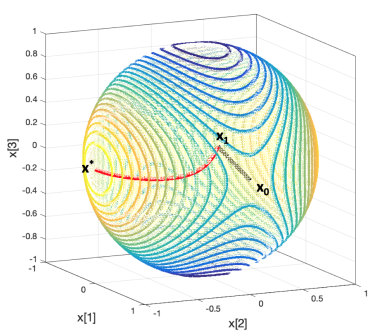
Our existing experiments illustrate that the iterates escape from saddle points and converge fast to an approximate second order stationary point. Figure 1 in the paper shows that, after perturbation, the iterates escape the saddle point x_0 and converges to x^*.

Reviewer 1.

1. The ML community has shown growing interest in manifold optimization and rates of escape from saddle points; papers on both non-convex optimization (e.g., Jin et al., 2017) and Riemannian optimization (e.g., Bécigneul & Ganea, 2019) are published in top ML conferences.

2. If we agree about the importance of Grassmannian and Stiefel manifolds, numerous other interesting manifolds also have closed-form exponential maps: the *Minkowski space, the hyperbolic space,* $SE(n)$, $SO(n)$...(see, Miolane et al. (2018); Boumal et al. (2014) and their open-source packages and Bécigneul & Ganea (2019, Sec 5) for an example in NLP). Focusing on the Grassmanian and Stiefel manifolds would be too restrictive and would leave out all these other example applications.

3. We stress that a major contribution is to derive convergence rate for the Riemannian gradient expressed in terms of *the manifold curvature*. As one may already see from the "proof strategy", this is made possible by involved results in differential geometry which control the evolution of the exponential map in a curvature dependent manner (Rauch comparison theorem on Jacobi fields).



As far as we are aware, there is no such results for general retraction, and convergence results for general retractions are therefore not curvature dependent and consequently *not* truly geometric. Thereby convergence rate in Riemannian optimization when retraction is used, (see, e.g., Criscitiello & Boumal, 2019; Lee et al., 2017; Zhang & Zhang, 2018) is based on the Lipschitz constants of the pullback function instead, which are hard to quantify and mix the properties of the function and the manifold.
However, as explained in Section 3, our algorithm also works with a general retraction. We agree about providing additional experiments to illustrate this feature. We re-did Figure 1 in the paper by replacing the *exponential map* with the *retraction* $R_x(v) = (x+v)/\|x+v\|_2$; we show the resulting figure here. We observe that the behavior is very similar as when the exponential map is used.

4. In the theoretical part of the paper, we state the key technical challenges, main lemmas and the roadmap/proof sketch to the final result, in order to give the reader the intuition behind the proofs. This certainly interests a part of the community, and also strikes a balance between giving full details and completely skipping all proof in the main body.

Reviewer 2. We thank the reviewer for their helpful comments. We will correct the small typos (thanks!). We will expand the discussion on our examples and empirical results to (a) explore the dependence of the rate on the geometric and function-dependent constants for the specific examples studied, and (b) empirically investigate whether for a manifold more curved than a sphere (large $|K|$), the convergence rate is slower than on the sphere. We will also post our code that implements the algorithm in Matlab for the examples in our paper later.

References

Bécigneul, G. and Ganea, O.-E. Riemannian adaptive optimization methods. In *ICLR*, 2019.

Boumal, N., Mishra, B., Absil, P.-A., and Sepulchre, R. Manopt, a Matlab toolbox for optimization on manifolds. *JMLR*, 2014.

Criscitiello, C. and Boumal, N. Efficiently escaping saddle points on manifolds. *ArXiv:1906.04321*, 2019.

Jin, C., Ge, R., Netrapalli, P., Kakade, S., and Jordan, M. I. How to escape saddle points efficiently. In *ICML*, 2017.

Lee, J. D., Panageas, I., Piliouras, G., Simchowitz, M., Jordan, M. I., and Recht, B. First-order methods almost always avoid saddle points. *ArXiv:1710.07406*, 2017.

Miolane, N., Mathe, J., Donnat, C., Jorda, M., and Pennec, X. Geomstats: a python package for Riemannian geometry in machine learning. *ArXiv:1805.08308*, 2018.

Zhang, J. and Zhang, S. A cubic regularized Newton's method over Riemannian manifolds. *ArXiv:1805.05565*, 2018.