NeurIPS Rebuttal for "Reliable training and estimation of variance networks"

We thank the reviewers for their constructive and fair reviews. We here address the key concerns, and note that the paper will be updated accordingly. We will discuss two shared concerns, and then move to individual reviewers.

First, however, we emphasize that the paper provides the first study of training issues related to variance networks, and develops new training methodologies that result in significant gains over current state of the art.

High dimensionality (R2+R3): Two reviewers expresses concern about extending our methodologies to higher dimensions. This is reasonable as both the locality sampler and the variance extrapolation depend on Euclidean distances, which are known to perform poorly in high dimensions. While we acknowledge the issue, we want bring additional perspectives to the discussion:

- One definition of variance is that it is the mean squared deviation from the mean, so the notion of variance itself is dependent on the Euclidean distance. This may indicate that variance networks are inherently difficult in high dimensions, where some regularity assumptions may be needed.
- While Euclidean distances behave poorly in high dimensions, there is plenty of empirical evidence that nearest neighbors are somewhat better behaved. As an example, spectral learning techniques have been very successful in moderately high dimensions. This is also in line with our observation that the proposed methods work well on MNIST, which is 768 dimensional (medium-sized data).
- As R3 mentions, the issue can possibly be circumvented through a different choice of metric; possibly one that is learned. This is an interesting venue for further research.

Computational costs (R2+R3): We agree that performing nearest neighbor search in the inner training loop presents a scalability issue. Two practical comments are, however, in order:

- We can rely on fast approximate nearest neighbor algorithms. This is an established area of research, where, in particular, randomized algorithms are currently showing notable speed gains.
- The variance network is typically trained on a GPU while the nearest neighbor search is performed on the CPU. Thus, the two processes can in principle be performed in parallel. The total training time can then be unaffected as long as the neighbor search is faster than a forward and a backward pass.

That being said, we acknowledge the concern, and view this as a topic for future work.

We consider the above two concerns to be both valid and important, but we also maintain that the paper provides the first working implementation of variance network training, which is a significant contribution.

R1: We agree with the reviewer that the three variance problems discussed in this paper (underestimation, trivialization, extrapolation) could easily each be the subject of its own paper. With this paper we aim at bringing attention to the multiple separate problems within uncertainty estimation, and argue that each problem requires its own solution. We provide initial solutions that significantly improve on the current state of affairs, but do expect that they can all be improved. We, further, would not be surprised if different application areas might benefit from slightly different solutions. We will update the paper with a discussion of these matters in more detail.

R3: We agree that the proposed methodologies do not solve the problem of overfitting. Interestingly, this problem also depends on the flexibility of the mean function. If this perfectly fits the data, then overfitting of the variance becomes more likely. This indicates that the link between variance estimation and regularization of the mean, could use further study; we consider this to be an interesting venue for research.

In practice we do not believe that the bias introduced by the sampling scheme plays much of a role, if the data is dense. However, theoretically we could have regions of data that are under-trained because they are underrepresented in the samples. The reason for the improved result is more likely due to an optimization that avoids the local minima of the usual mini-batching.

Regarding, Eq. 5: Yes, this is a typo. Thanks for catching it!