Response to Reviewer 2 - 2025-28

I. REVIEWER SUMMARY

This article describes techniques for on-line statistical analysis of climate model output which can reduce the amount of output data without needing to degrade the spatial resolution of the output. In particular a sophisticated, novel t-digest algorithm is described for on-line generation of percentiles and histograms. The publication is timely given the increasingly large volumes of output being created by increasingly complex and high-resolution numerical models, especially for the emerging class of km-scale models many modeling centers are developing. In particular the quantification of memory savings is a powerful selling point. This paper should be published and GMD is the ideal journal for this purpose. However I was somewhat confused by the discussion of the t-digest method and would like to have some more clarification on a number of points.

II. MAJOR COMMENTS:

1. I am unfamiliar with some of the terminology. Does the "scale function" k(q) introduced in equation (6) define the "edge values" for each of the bins/clusters? If so, an illustration (showing the domains for each cluster as a function of delta) would be useful.

   Thank you for the comments relating to the t-digest description. We have taken them all into account and have significantly modified the introduction of this section. The scale-function defines a monotonically increasing function, where the edge values for each cluster are defined by it's slope. Due to it's hyperbolic shape, k has a steeper gradient at the tails, resulting in smaller cluster sizes over these ends. We have now shown this visually in a new Figure 3(a), which shows the shape of the scale function over a range different compression parameters. Note a similar figure is shown in Dunning, T. and Ertl, O.: Computing Extremely Accurate Quantiles Using t-Digests, arXiv[preprint],arXiv:1902.04023, 2019, Figure 1. In the new figure the clusters are shown for $\delta = 20$ and $\delta = 100$ for two different t-digests that represent the same data set X_{500}. We have used the size of the gray dots to indicate the weight of the cluster, showing that fewer clusters with larger weights are used for $\delta = 20$, than 100. Exactly how the edges of clusters are defined depends on the size of the cluster in k space. This is described in detail in Dunning and Ertl [1] and we refer the reader there for technical details of this process as we feel it is outside the scope of this work and we want to focus on the accuracy of the algorithms implementation for different climate variables.

2. Also, should the same scale function be used for each variable? While having a somewhat equal-spaced set of bins is possibly a good choice for wind speed, this would not be as useful for precipitation since (as is shown in Section 5.3) small values are far more common than large ones, and so the small bin will fill up very quickly.

   The reviewer raises an excellent point here and we agree that for precipitation a different scale function would be preferable. The aim of this work is to show the performance of the One_Pass package (v0.7.2) (upgraded version since previous submission) in representing these variables so we have performed the analysis using the scale function currently implemented. We comment at the end of section 6.3 (previously 5.3) that these results would definitely be improved by using a scale function specifically designed for precipitation, however this is outside the scope of the paper. We also note that one easy way to bypass this problem would be to include a cut-off value for the precipitation, so simply ignoring all values below say 1 mm/day to make best use of the available clusters. We have modified this section to make it more conscience and included the NumPy histograms to show more visually how the current implementation represents the full distribution.

3. Also, how the t-digest method goes from its clusters to creating percentiles and histograms is unclear. Could a more concrete demonstration of this be shown? Also, are the numpy percentiles and histograms created by a similar algorithm, or are these the exact percentiles and histograms computed from all of the data without approximation?

   We have addressed both these comments in the new Sec 6.1. Specifically we have added the sentence "These clusters are ultimately converted to a percentile or a histogram (where bin densities may have non-integer values due the underlying cluster representation), based on the closest cluster mean to the required percentile. " We hope that this, along with the new Figure which shows the cluster weights in relation to their quantile will make this conversion clearer. For the NumPy implementation, it does indeed have access to the full data without approximation. We have added a clear statement explaining this in the last paragraph of Sec 6.1.

4. I was very glad to see the careful validation of the simpler one-pass algorithms (mean, stan-

dard deviation) and the striking demonstration that the errors are at rounding level. I had more trouble understanding Figures 4 and 5. It is not immediately clear that the red and orange dots represent the different locations in panels (b)–(e) without a legend; and the histograms inset in panels (b) and (e) are so small as to be almost impossible to interpret. It would be more useful to overplot these on a larger panel to more directly compare the two. I also don't quite understand the main result in panels (b) and (e): the shaded area is the typical wind turbine operating tolerance for these locations, and each dot is another percentile larger (first dot is 1st percentile, last is 99th)? Overall I think splitting out the panels and/or insets into different figures, providing better labels and annotations, and enlarging some plots would make it much easier to read and understand.

We have significantly updated both Figure 4 and 5. They both now contain legends to indicate that the range of colors signify particular locations. We agree this was not so clear before. We have also converted the histogram inserts in Figure 4 into their own panels to allow for a proper comparison between the t-digest and NumPy methods. This has further been done in Figure 5 to allow for continuity between the Figures and also show how the full precipitation distribution is represented. For all histograms, their colors now correspond to the location they represent, marked in the legend.

5. The discussion of convergence of climatologies in Section 6 is interesting and does get at the vital scientific question of how much data is needed to create a useful climatology. However, given that the one-pass algorithms create nearly-exact statistics (for means and standard deviations) or up to the cluster's sampling uncertainty (t-digest) for the model run segment being considered; constructing a climatology for bias correction or other purposes would be a downstream step that would presumably mean aggregating statistics over a large number of segments and/or individual simulations. Would not this analysis then say how long the simulation needs to be to create an accurate climatology, and not have much effect on the algorithms of this paper?

   We have changed the introduction to this section to better explain motivation as it was not clear before. Indeed, this section boils down to "how much data do you need for a climatology/summary", however in this context it falls under the one-pass umbrella as we consider the bias-adjustment of streamed climate data. As we now explain, the One_Pass package supports another package designed for bias-adjustment of streamed data, which, when we were designing, made us investigate this concept of convergence. What we are aiming to do with this section is show other applications for the one-pass algorithms and how the rolling summary of any statistic S_n can be considered a reliable estimate of the statistic of the total distribution after a certain number of samples. This is crucial when designing bias-adjustment of streamed data or other downstream applications. We hope we have addressed the reviewers comment and explained how this is another 'application' of the one-pass algorithms.

Minor comments

1. There is precedent for on-line calculations of statistics such as means, extrema, standard deviations and so on. See the "data reduction" techniques here, for instance: `https://github.com/NOAA-GFDL/FMS/blob/main/diag_manager/diag_yaml_format.md` and `https://github.com/NOAA-GFDL/FMS/blob/main/docs/diag_table.md`

   We have now included a new section (Sect. 2) to discuss the context of the development of this package and why it was necessary to decouple it from online calculations produced by the models. We have used this repo as as a reference for those online calculations.

2. - Line 171: Should "extremely values" be "extremely small values"?

   Fixed.

---

[1] T. Dunning and O. Ertl, Computing extremely accurate quantiles using t-digests, (2019).
1