

Response to Reviewer #2

Specific comments

1. *Lines 157-164 and Eqs. (1)-(2): I may be misunderstanding the percentile convention, but the ordering in Eqs. (1)-(2) appears inconsistent with Table 2. The text describes P100%, P99%, and P95% as percentiles of the ensemble spread data; under the usual percentile convention, $P100\% \geq P99\% \geq P95\%$. This would make b_{low} the largest bound and b_{high} the smallest, whereas Table 2 presents low, mid, and high as increasing error bounds. Please clarify the percentile convention or check whether the labels in Eqs. (1)-(2) are reversed.*

Response: This was a mistake in the script, our apologies. The low, mid, and high bounds are the 0%, 1%, and 5% percentile in the data, where the 0% percentile is the minimum according to the standard percentile convention. We have now corrected this.

2. *Definition 1 and Table 2: The manuscript does define a pointwise relative error bound, and it notes that this differs from the normalized ratio $|X_i - \hat{X}_i| / |X_i|$. I still think this distinction deserves more emphasis for readers from the compression community, because different papers and compressors use different relative-error conventions near zero. In this benchmark, the definition $|X_i - \hat{X}_i| \leq b_{rel} |X_i|$, together with the exact-equality condition, makes the bound zero-preserving. That matters for interpreting failures on biomass, precipitation, and other zero-heavy or near-zero fields.*

Response: We have added additional discussion, in the paragraph following Definition 1, on the behaviour of the relative error and also mention that there are sometimes alternative choices such as the range relative error.

3. *Lines 179-187 and Table 2: The text says that the expert bounds mostly lie between the low and high computed bounds, with exceptions only for sea-level pressure and precipitation. However, the expert bound for sea-surface temperature is 0.01 degrees C, which is much larger than all three computed SST bounds in Table 2. Please check either the table or the statement in the text.*

Response: We have adjusted the main text to highlight the discrepancy for the sea surface temperature.

4. *Table 2 and lines 179-187: The caption notes that some expert bounds are average rather than pointwise bounds. Since the benchmark bounds themselves are pointwise, I would also mention this in the text where the comparison to expert bounds is discussed. Otherwise the comparison can sound more like-for-like than it really is.*

Response: We have adjusted the main text to more strongly emphasize that some of the expert bounds are not directly comparable to the derived error bounds/

5. *Section 2.2.1: The spectral error metric is useful, but I could not tell exactly how it is computed for global longitude-latitude data. Is the spectrum computed directly on the native lon-lat grid? Is any latitude weighting used? Is there any reprojection? How is the radial averaging defined? A few implementation details would make this easier to reproduce.*

Response: We've expanded the text with more implementation details. In general, all the code to reproduce the benchmark results, including the metrics, is open sourced to ensure reproducibility. To compute the spectra we followed the conventions of the pysteps library, we computed the spectra on the native lon-lat grid without any latitude weightings or reprojections.

6. *Section 2.2.1: Relatedly, please clarify how DSSIM and spectral error are aggregated for variables with multiple time steps and vertical levels. Are they computed per 2-D slice and then averaged, or on higher-dimensional fields directly?*

Response: For the spectral error, they are computed over each 2D slice and then averaged. For the DSSIM metric, we take the minimum over vertical levels and the average in the time dimension. We have made this explicit now in the main text.

7. *Lines 313-324 and 347-358: The conversion from relative to absolute bounds using the smallest non-zero right-hand side is intentionally conservative, but it can be dominated by near-zero or subnormal values. The precipitation example later in the paper shows this well. I would make this point more explicit in the main text, especially for compressors that do not natively support the benchmark's relative-error definition or use a different relative-error convention.*

Response: See response to point 8.

8. *Figure 2 and Table 3: I found Table 3 helpful, especially the bracketed entries for partial relative-error support. In the scorecards, however, it is still easy to conflate two cases: compressors that do not natively support the tested pointwise constraint, and compressors that nominally support it but fail in particular cases. Some additional cue in the Figure 2 caption or surrounding text would help readers separate these cases.*

Joint response to points 7 and 8: We have now further emphasized that the relative to absolute error bound conversion can lead to very conservative absolute error bounds at the end of Section 3.1. Furthermore, we have updated the text in Section 3.2 and the scorecards in Figure 2 to highlight all the compressor-variable combinations for which we applied this transformation while emphasizing that higher compression ratios can be achieved by using a more permissive error bound transformation.

9. *Lines 227-234, Figure 4, and Figure E1: The instruction-count metric is a nice reproducibility feature. The paper already notes that wall-clock measurements are noisy and that WebAssembly trades execution speed for reproducibility and portability. I would still bring a little more of this practical interpretation into the main text, especially how the single-threaded WebAssembly measurements should be read relative to native or parallel implementations.*

Response: We have added multiple clarifications regarding our focus on the single-threaded WebAssembly measurements: highlighted in Section 2.2 that other performance metrics such as the ability to exploit hardware accelerators are often relevant in practice, mentioned in the discussion in Section 4 that scaling compression pipelines to terabytes of data comes with its own distinct challenges, and highlighted in Figure E1 that the throughput measurements should be read as unoptimized, lower bound estimates of what might be achieved in practice.

10. *Lines 392-405: The chunking sensitivity analysis is valuable, especially because SZ3 behaves differently under relative-error settings. Since chunking is often important in practice, please report the actual chunk shapes used, perhaps in an appendix or supplement.*

Response: We have added Table F1 in the appendix which specifies the actual chunk shapes used. It is also referenced in the main text.

11. *Figure 3 and Appendix D: The normalized rate-distortion summaries are helpful, but they should not be read as a single global Pareto ranking. Because the values are normalized per metric and variable and then averaged, I would add a short caution in the main text about over-interpreting the aggregate ordering.*

Response: We have added more clarifications in the main text to highlight that the rate-distortion plots show averaged quantities and should always be read in combination with the scorecards because of the variation in performance between different datasets and variables.

12. *Figure 8 and lines 426-434: Please explain in the Figure 8 caption why only a subset of compressors is shown. The text gives the context, but a reader looking directly at the figure may wonder why SPERR and JPEG2000 are absent, or why the two lossless backends are not separated.*

Response: We have clarified this now. SPERR and JPEG2000 are excluded because they fail on this variable. We only show one of the lossless backends because by definition the lossless compressors do not change the error patterns.

13. *Section 2.1 / Figure 2 caption / lines 336-339: SST is described earlier as containing NaNs, but I only learned later that the air-temperature field also contains NaNs. Please mention this earlier in the dataset description.*

Response: We now mention in Section 2.1 when the dataset is originally introduced that the air temperature data also contains NaNs.

14. *Introduction and Section 3.1: The introduction motivates the broader area partly by mentioning neural compression, but the baseline suite only includes conventional codecs. A short explanation of why learned methods are not included in the current baseline would help frame the benchmark and future extensions.*

Response: We have added the following justification in Section 3.1: “For now, we have not included any neural compressors in the benchmark because they are either developed only for a specific dataset, generally ERA5, or do not come with any built-in mechanisms to control their error bound. We hope this benchmark encourages development in neural compressors that overcome these shortcomings.”

15. *Line 101: "Linear packing" may not be familiar to all readers. A short definition or reference would help.*

Response: We now instead use the term “linear quantization” which is more common and provide references that explain the concept in more detail.

16. *Conclusion, lines 448-449, versus Table 1: The phrase "below a couple of gigabytes" seems inconsistent with Table 1, which gives the evaluation set size as about 2.95 GB. "Around 3 GB" would be more accurate.*

Response: Adjusted.

Technical corrections

Line 99: There is extra punctuation around the footnote marker in "outputs.3."

Definition 1, line 136: The index $0 \leq v \leq V$ implies $V+1$ variables. I assume this should be $0 \leq v < V$, or an equivalent 1-based convention.

Line 318: "This ensure" should be "This ensures".

Line 413: "e.g Lindstrom (2017)" should read "e.g., Lindstrom (2017)".

Figure 3 caption: The final averaging expression appears to drop the error-bound index. I suspect the intended notation is something like $n_{\{c,b\}} = \text{mean}_v(\{n_{\{v,b,c\}}\})$.

Figures 5-6: "Not chunked" sounds a little awkward; "unchunked" would be more idiomatic.

Figures 3 and 6: Figure 3 would be easier to read with distinct marker symbols in addition to colors and line styles. Figure 6 already uses crosses and squares to distinguish chunked from unchunked runs, but compressor identity still relies heavily on color.

Response: We have corrected all the technical comments.