I appreciate the authors' efforts to build a parallelizable model-agnostic particle filter system, which holds promise for various geoscience applications. Nevertheless, I would like to raise some concerns about the state transition density, or the model error Q, in the following:

1. The difference of the particle weights between using the bootstrap and using the optimal proposal is determined by Q (e.g., see the reference below). Therefore, the way that Q is specified determines how much the optimal proposal outperforms the bootstrap proposal. This could be added into the discussion to emphasize the importance of the choice of Q.

   e.g., see Sections 9.2.2-9.2.3 in
   Evensen, Geir, Femke C. Vossepoel, and Peter Jan van Leeuwen. Data assimilation fundamentals: A unified formulation of the state and parameter estimation problem. Springer Nature, 2022.

2. The particle filter algorithm itself is model-agnostic, while Q is not. Although this manuscript has provided examples illustrating the generation of Q, in general this is not trivial for many geophysical models. For example, not only do we need to consider the smoothness of the state variable, but also the physical constraints across different variable types. For example, the wind and pressure should largely satisfy the geostrophic balance relation in the AGCM. In addition, for many geoscience applications, Q can be state dependent. For example, the model errors for predicting a heat wave can be very different from predicting a hurricane. Probably beyond the scope of this study, the transition density can also be quite non-Gaussian, e.g., for modeling the convection process in weather prediction models.

   The user might need to build Q for their own model, which again, is not trivial for many geophysical models. Since this manuscript is submitted to GMD, I would recommend expand the discussions surrounding the construction of Q for geoscience applications (e.g., in the conclusion section).

3. I would recommend elaborate more on how *get_covariance_state_noise* and *get_covariance_state_observation_given_previous_state* are being evaluated. Are there any computational challenges to evaluate these two functions for a very high-dimensional (e.g., $d_x \sim 10^9$) model?

I have a few other minor comments:

1. For a spatially extended model, like a weather prediction model, the dimension can be as high as 10^9. Will this be an issue when copying a state from one processor to another? How does the overall algorithm scale with the dimension of the model state d_x?

2. I find it somewhat less convincing that the particle filter can work better than any existing linear and Gaussian DA methods in the experiments (e.g., from the results in Figs 8-9). Nevertheless, I do understand the primary goal of this manuscript is to showcase the capabilities of the package, instead of proposing a novel particle filter methodology and conduct comprehensive comparisons with existing methods, etc.

   Therefore, I do not insist but recommend, e.g., add a new experiment with non-linear observation operator, and/or compare the performance of PF and an ensemble Kalman filter against the ground truth in an OSSE (e.g., in Figure 8, you could also add a panel that shows the results from using an ensemble Kalman filter).

3. Figure 8 -> is the unit of the time averaged error correct? An error in surface pressure exceeding 50 hPa seems unrealistically large.