Manuscript

Responses to Reviewers

Dear editor and reviewers,

Thanks very much for taking your time to review this manuscript. We really appreciate all your comments and suggestions, which are very helpful for us to improve the manuscript. We have made a thorough revision to address the comments and questions. Please find our point-to-point responses below.

Reviewers' Comments to the Authors:

Reviewer 2

1. *The statistical model used for model calibration in Section 2.4 "Step: Execution of data assimilation" is not well defined. It's quite unclear to me what the authors are using for model calibration therefore it is difficult to evaluate the calibration exercises themselves. I'm not convinced that you are really showing posterior distributions in the figures because of the description of the algorithm in the methods. What is the actual statistical model used for model calibration?*

Thanks for the comments and the question. The statistical method for model calibration in Section 2.4 is Metropolis-Hasting algorithm. It is a sampling-based algorithm including a proposing step and a moving step. The proposing step (L231-237) is to generate new set of parameter values and the moving step (L238-248) is to decide whether to accept these new parameter values or not. The posterior distribution is generated from all accepted parameters (L250). The significant peak in the posterior distribution indicates the parameters are well constrained (L360-366).

2. *My second major comment is that Fer 2018 and PEcAn were entirely left out of this manuscript but should certainly be included in several places.*

We appreciate this constructive comment. Including Fer 2018 and PEcAn is a great addition and We have added Fer et al. (2018) and PEcAn in L101.

3. *Line 26: what kinds of states?*

We are sorry for the unclear description. We changed "state" to "states of ecosystems" in L26.

4. *Line 38: I think there's a word missing "model … the land component"*

Thanks for your question. The Energy, Exascale, Earth System Model (E3SM) is an Earth system modeling project sponsored by the US Department of Energy (E3SM Project, 2018). E3SM land model (ELM) is the land and energy component of the earth system model. To be clearer, we added a colon between 'model' and 'the land component' in L39.

5. *Line 42: Doesn't the easy implementation potentially make this more 'black box'?*

Thanks for pointing it out. We removed "black-box" to avoid misleading in L42.

6. *Line 58: Citation for invasive coding?*

Thanks for the great suggestion. Invasive coding, a concept from Java programming language, means to modify current code to incorporate new algorithms or features. In this study, invasive coding means programming the DA algorithm into the model source code. We added a citation for invasive coding in L58.

7. *Line 75: Missing link sentence between ", 2009). … DA was"*

Yes, the link is missing. We added "In the study by Liang et al. (2018)" to better link the two sentences in L75.

8. *Line 78: Not sure about "data-worth"*

We apologize for the vagueness. To avoid confusion, we removed the redundant phrase "for data-worth analysis" in L78.

9. *Line 100: Include Predictive Ecosystem Analyzer (PEcAn)*

This is a great suggestion. We added PEcAn and the citation of Fer et al., (2018) in L101.

10. *Line 120-121: Not sure what is meant by this sentence*

Sorry for the unclear statement. This sentence is to express a point that changes in estimated parameter values by EnKF each time when a data point is assimilated usually do not reflect a reality of biogeochemical cycles in the real world. That is, parameter values of biogeochemical cycles in the real world do not suddenly change at a time point when data is assimilated by EnKF. We have modified the sentence to clarify this point.

11. Line 131: Nice!

Thanks.

12. Line 176: Hinders?

Thanks for pointing it out. We have corrected the typo to "hides" in L191.

13. Line 178: How does MIDA know how to write out model specific configurations?

We hope we have understood your question correctly. Generally, MIDA does not write out configurations for a specific model. Instead, MIDA uses a 'call' function written in Python to run the model executable first and does not require model-specific forcing or other configurations (L253). Then, the data exchanges in the communication between MIDA and the model are realized by file I/O operations and MIDA does not write out model-specific configurations, either (L196-203). Taking parameter values as an example, MIDA reads the parameter range from a file "namelist.txt" that is provided by users. According to the parameter range, MIDA gets the number, maximum limit and minimum limit of the parameters. Based on this information, MIDA generates new parameter values and writes them to a file for the model executable to read. Third, all model-specific information is provided by users. For example, users need to indicate the file names of parameter range, observations, and model outputs in the "namelist.txt" or via GUI. Users also need to prepare a model-specific output configuration file to instruct how to map model outputs with each observation. Section 2.7 describes such information in details. Appendix B and Appendix C provides an example of the output configuration file and an example of the namelist.txt file, respectively.

14. Line 183 transfers -> transfer

We have corrected the typo to “transfer” in L198 as suggested.

15. Line 184: organize -> organizes

The typo has been corrected to “organizes” in L199 as suggested.

16. Line 184: So the “different files” are like lookup tables?

Thanks for the great question. The ‘different files’ in the original L184 are to save parameter ranges, parameter values, simulation outputs, observations and their covariances, and an output configuration to match between observations and simulation outputs. Different from PEcAn, there are no lookup tables in MIDA. PEcAn uses a lookup table to find the right data (e.g., PFT) for a specific model. MIDA defines a prototype class for data (i.e., parameter, observation or simulation output). These classes are initialized to adapt a specific model according to the data files loaded when MIDA is running. These coding methods are object-orient programming and dynamic initialization, which are commonly used in Java programming language. All this information is available in L204-215.

*17. Line 194: Would be cool to have an illustration of dynamic initialization **

Thanks for the suggestion. Object-orient programming and dynamic initialization in L207 are concepts from Java programming. As readers of this manuscript are most likely ecologists who may not have advanced knowledge about programming, it would make it easier if we avoid such technical details.

18. Line 209: In think you mean “inference”

We have corrected the typo to “inference” in L224 as suggested.

19. Line 210: Before talking about the sampling algorithm, it would be good for the reader to know about the model formulation like the likelihood, prior, etc.

We thank the reviewer for the valuable suggestion. In Section 2.1, we introduce general concepts about data assimilation methods. We revised to include description of likelihood, prior, and posterior distribution in L141-154.

20. *Line 245: What kind of MLE? What's the model formulation? What are you maximizing over here? Why do you get maximum likelihoods and posteriors?*

Thanks for the great questions. All these questions are related to data assimilation (DA), which is to estimate parameter posterior distributions and constrain parameter values through assimilating observations into the model. Based on Bayes' theorem, estimating parameter posterior distribution is transferred to maximize the likelihood function (Eq. 1,2), which is negatively proportional to the mismatch between simulation outputs and observations (Eq. 3). Meanwhile, maximum likelihood estimator of Eq. 2 can also help estimate the optimal parameter values. The distinctive mode of the posterior distributions indicates whether the parameter uncertainty is well constrained. All these information has been added to L141-154 in Section 2.1 to clarify the relationship between likelihood, posterior, and MLE in DA.

21. *Line 250: At this point, I'm still confused how you define your drivers and model settings in general? For a particular model you usually have a parameter file that gets read by the executable. So are you rewriting those parameter files in step 2?*

We apologize for the confusions. For the first question, model simulation is independent of MIDA and MIDA uses a 'call' function from the subprocess package in Python to execute model simulation. Taking DALEC model in the first case study as an example, MIDA calls executable file of DALEC, which has already defined the directory where forcings (i.e., temperature, vapor pressure deficit, carbon dioxide concentration, etc.) are read to execute. The data exchange between MIDA and the model is model-specific (i.e., parameter values, simulation outputs, observations and their variances) and users only need to provide the paths and names of these data files in the 'namelist.txt' file or via GUI. Appendix C is an example of a 'namelist.txt' file. Section 2.3 in the manuscript has introduced how to realize these model-specific data exchanges in MIDA. Section 2.7 has introduced how users prepare the 'namelist.txt' file.

For the second question, MIDA reads parameter range from a file indicated in the 'namelist.txt' file, which is provides by users. Based on the parameter range, MIDA repeatedly generates new set of parameter values during the DA and writes these parameter values to the 'ParameterValue.txt' in the work path that users chose in the 'namelist.txt'. Model executable needs to read this file to get new parameter values. Users can also change the name of 'ParameterValue.txt' in the 'namelist.txt' according to the need of a specific model. L259-261

and L320-324 described how MIDA writes parameter values to a file, from which the model reads these parameter values to run simulation.

22. Line 263: Make sure to cite all software packages.

Thanks for the kind reminder. We added citations of all Python packages used in MIDA in L287.

23. Line 314: Helpful examples. Really curious where 302 years of leaf area data come from!

We first generated simulation results of the BiomeE model with forest ages ranging from 0 to 800 years, then compared the simulated forest height or vertical structure to GEDI-derived observations and determined the forest age is 302 yrs. According to the empirical knowledge, the forest is around 300~350 years old and a variation of 50 years is acceptable as because forests are almost equilibrated.

24. Line 343: "complex reasons" is somewhat vague

We are sorry about the vague description. We added specific descriptions "such as improper prior parameter range" for the reasons leading to edge-hitting posterior distribution in L363. In addition, we included other possible reasons to explain the results.

25. Line 499: Citation?

Thanks for the suggestion. We added a citation of Gao et al. (2011) in L520.

26. Line 504: Not sure about "first" unless you make your definition of model agnostic more specific

We are sorry about the unclear statement. We changed the sentence to "Compared to the model-independent DA tools mentioned above, MIDA is the first that uses the MCMC method for DA.'" in L524. It echoes with L514-515.

27. Figure 1: Very nice!

Thanks.

28. *Figure 6: the Cis appear homogenous. Can MIRA deal with heteroskedasticity?*

Thanks for your question. MIDA uses MCMC algorithms which requires homogenous variance. MIDA is also flexible enough to incorporate other algorithms that can deal with data of heteroskedasticity.

29. Figure 7: the colors in the legend have an extra character that should be removed

As suggested, we removed the extra character in the legend of model no. in Fig. 7.

30. Figure 4, 5, and 8: these posterior distributions do not look converged to me. Could you also include the mcmc chains as well here or in the supplements?

Thanks for your question. These cases study are to repeat published DA results to demonstrate the capability of MIDA. Taking Fig. 4 as an example, the constrained posterior distributions are similar to those from the original study in Lu et al. (2017). In addition, MIDA has already incorporated algorithms to support Gelman-Robin convergence test of multiple MCMC chains as described in L335.