Dear editors of Behavior Research Methods,

Thank you for considering our paper "Subgroup detection in linear growth curve models with generalized linear mixed model (GLMM) trees" (BR-Org-23-622) for publication. We were happy to receive helpful reviews and have revised the manuscript. We hope the editor and reviewer’s agree the changes are improvements indeed. Below, we address the reviewer’s comments in a point-by-point manner.

Sincerely,

Marjolein Fokkema

**Reviewer 1**

**Comment 1:** This article investigates two potential improvements of GLMM trees: 1) initializing the random effects with the root node model instead of using 0; 2) using clustered covariances for the stability tests used in the splitting process. Simulations studies with numerous variants of GLMM trees with and without these modifications and with other competing methods are presented, as well as a real data example. The article is interesting, well written and easy to follow. The simulation study is limited but sufficient to offer practical recommendations.

No new methods are proposed, and the improvements are very specialized ones. Hence, from a methodological point of view, this is a niche paper. However, from a user point of view, the findings are interesting and useful.

**Response:** We thank the reviewer for investing time to read the manuscript and are happy to see the practical relevance of this work acknowledged.

**Comment 2:** I only have a small comment. The description on page 6 is confusing at first. One might think that there are no covariates except for time. It should be stated clearly that the setting studied in this paper is the case where we have time-invariant covariates, and only time as the time-varying one. Model (1) (2) are for the within-node (within-subgroup) model.

**Response:** Thanks for pointing this out, te distinction between the regressor(s) in the node-specific model and the partitioning variables is critical indeed, and we want to avoid confusion. We adjusted the text as follows (page 6, changes underlined):

“In the general notation above, *Xi* is the *ni × (p + 1)* fixed-effects design matrix for subject *i (i = 1, …, N)*, comprising *p* time-varying covariates plus one column of 1s for the intercept. In this paper, we assume that time is the only time-varying covariate of interest (i.e., *p = 1*), but other time-varying covariates can also be included in *Xi*. The number *ni* and spacing of observed timepoints may differ between subjects. Time-constant covariates are not contained in *Xi*, but enter the model through the subgroup indicator *j*. The value of the fixed-effects parameters *β* (here, intercept and time slope) in GLMM trees are subgroup-specific, their value depending on the subgroup/node *j* into which subject *i* falls. In the current paper, subgroup membership is defined by the values of the time-constant covariatesǂ.

ǂ In principle, it is also possible to use time-varying covariates for partitioning with GLMM trees, but this might not be appropriate for partitioning growth curves, as participants changing subgroups over time yields discontinuous growth curves.”

We hope these changes also help to clarify that Model (1) and (2) are not node-specific, but give the full GLMM-tree model.

**Reviewer 2**

The authors present GLMM trees as a statistical approach for detecting heterogeneity in growth curve models. The topic of this article is timely and of great interest to a larger audience. I could follow the article very well and enjoyed reading about this new approach much. I have researched similar approaches (SEM trees) in my own work, so I may be positively biased towards trees in my evaluation. Here, I would like to share a couple of thoughts:

We thank the reviewer for investing time in reading our manuscript, and the positive evaluation.

**Comment 1:** "when these covariates are not known in advance" (p.2) I'd rather say that the covariates are often known but not their relationship to the response (in lack of a good psychological theory).

**Response:** We agree and have changed the wording on page 2 accordingly (changes underlined):

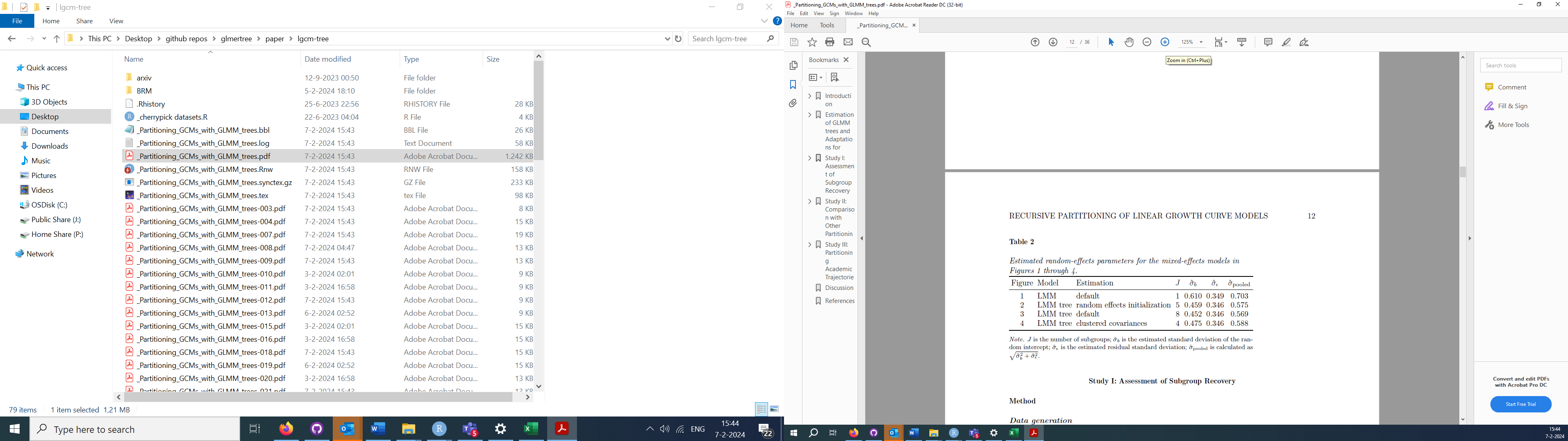
“However, when these covariates or their shape of association with the response are not known in advance, methods for identifying them are needed.”

**Comment 2:** I would rather move the Table from Figure 2 to a proper Table and also include estimates of uncertainty to the point estimates, such as confidence intervals. In SEM trees, I would usually report all parameters of interest including the intercept-slope-covariance and most importantly, residual error variances. I think that one virtue in SEM trees is the fact that you could discover differences only in the residual error variances, that is, even though all "true" change parameters were identical across a set of groups, they may only differ in measurement error (ie., if a certain questionaire works less reliable in one group of people). I am convinced that the proposed local-global approach makes a lot of sense and there is already some discussion on the trade-offs (complexity, generalizability); however, I think that differences in RE are particularly interesting for many research questions as a main outcome and I would love to see it if authors briefly touched upon this issue. For example, in a study where gene expressions were the predictors, we would assume that different combinations of genes would lead to different mean change profiles but also to limited or pronounced individual differences, which in itself is important to note. Would this induce a bias in the current GLLM tree approach? (after reading the entire ms, I found that the point is actually taken up in the discussion again - thanks!).

**Response:** We agree on the relevance of testing for instability of random-effects parameters, which cannot be done with GLMM trees. We opted not to present standard errors or confidence intervals, because these are difficult to validly estimate with trees. Although standard errors can be extracted from the final fitted mixed-effects model, these do not account for the searching of the subgroups / tree structure and are thus overly optimistic.

Instead, we presented estimates of variability:

* The fixed-effects coefficient estimates from Figure 2 are now presented separately in Table 1.
* We agree that variability estimates are relevant for interpretation, these are now presented in Table 2 (page 12):



* Page 8 introduces Table 2 for the first time:

“Table 2 presents the estimated random-effects parameters: Compared to the global LMM in Figure 1, the trees in Figure 2 and 3 have very similar residual variances, but clearly lower variance of the random intercept. Part of the inter-individual variation captured by the random effects in the global LMM, is explained by subgroup-specific fixed effects in the trees.”

* The pooled SDs from Table 2 allows for quantifying the coefficients in Table 1 as effect sizes. We added a discussion on page 8/9:

“The pooled standard deviations in Table 2 can also be used to compute effect sizes for the subgroup differences (Westfall, Kenny, & Judd, 2014). The effect sizes for Figure 2 and Table 1 range from *d* = 0.35 for nodes 7 and 8, indicating a small to medium negative effect of internalizing symptoms; to *d* = 2.26 for nodes 3 and 9, indicating a huge effect of socio-economic status and gross motor skills combined. Yet, these estimates are based on training data only, so can be overly optimistic. In light of the small number of respondents for some of the subgroups in Figure 2, these effects may not generalize well to other samples. Therefore, generalizability of the tree structures obtained will be empirically evaluated using cross validation in Study III.”

**Comment 3:** The log display of the growth curves makes sense but I wonder whether it isn't worth elaborating a little more on how people would choose this in practice and why not instead model log(response).

**Response:** Indeed, choosing the timing metric can be challenging. We assume the timing metric will not affect predictive performance of the tree algorithms much, but for interpretation of growth curves it can indeed be critical. We chose the transformations based on visual inspection before fitting and evaluating the trees in the Application (although one might argue: preregistration, or it did not happen).

We preferred fractional exponential over log transformations to avoid negative values of time and additional tinkering with a constant to be added to avoid log(0). Further, the fraction of 2/3 was chosen for Science abilities to make the distances between the consecutive assessments at 0, 12 and 36 months more similar than in the original time metric, while also not skewing the original time metric too strongly. We believe this timing metric would fit most perspectives on academic skill development. A comparison of the chosen fractional exponential with two log transformations:

log(c(0, 24, 60) + 1)

## [1] 0.000 3.219 4.111 ## too small distance from T2 to T3 compared to T1 to T2

> log(c(0, 24, 60) + .1)

[1] -2.302585 3.182212 4.096010 ## negative time, distance T1-T2 inflated

> c(0, 24, 60)^(2/3)

## [1] 0.000000 8.320335 15.326189 ## seems reasonable

We employed a different exponential fraction for Math and Reading than for Science, because Reading and Math were also assessed in Kindergarten and 1st grade, in addition to 3rd, 5th and 8th grade . The original assessment times thus were at 0, and roughly 12, 36, 60 and 96 months after baseline assessment. We expected rapid gains in Reading and Math to typically occur earlier than for Science, and expected strongest gains during the first assessments, so we chose a somewhat stronger exponential transformation, by using a fraction of 1/2 instead of 2/3:

> c(0, 12, 36, 60, 96)^(1/2)

[1] 0.000000 3.464102 6.000000 7.745967 9.797959

We have made some changes in the manuscript to signal and clarify these choices in the manuscript:

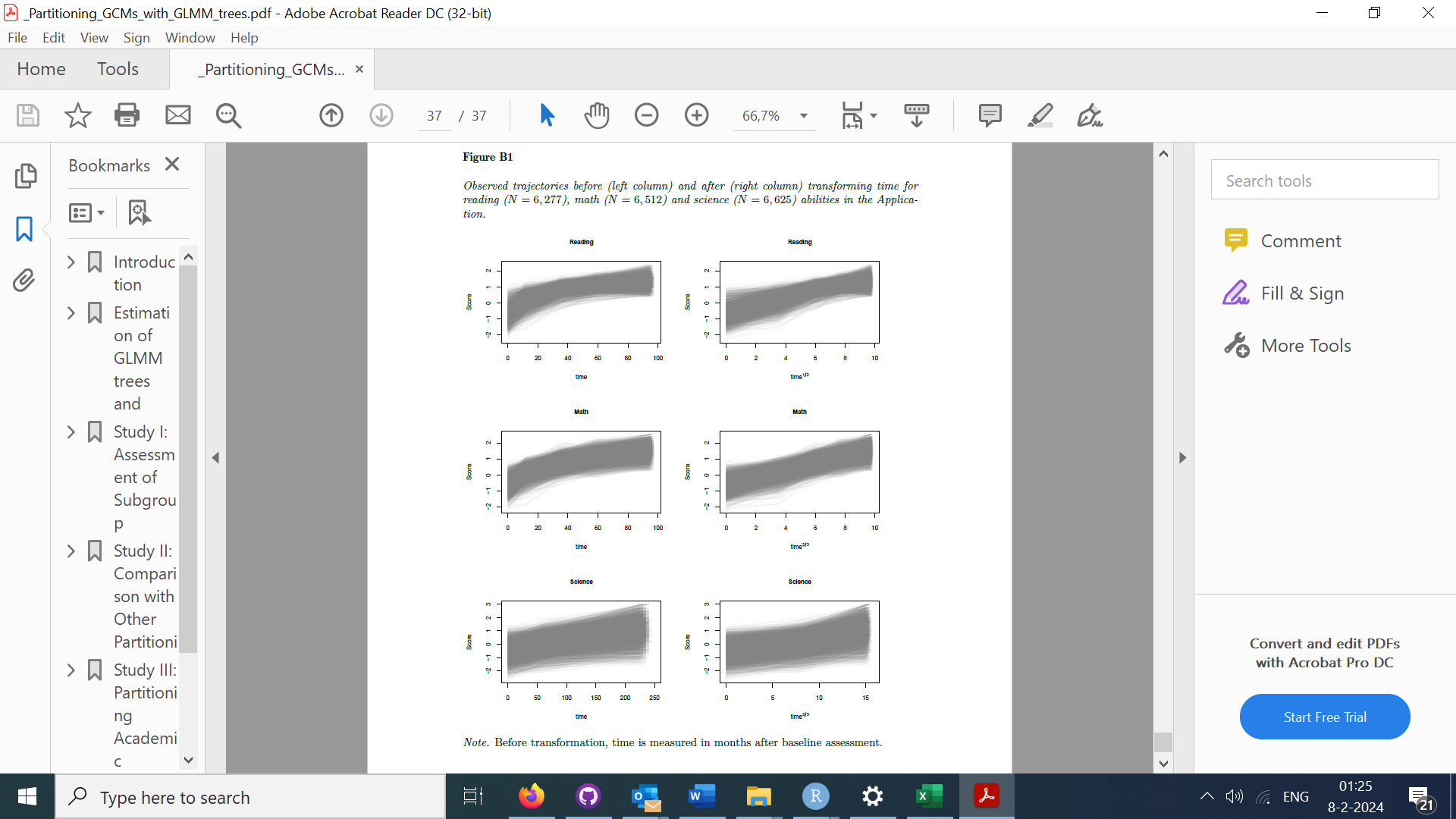
* One page 2, we added a footnote:

“\*Further details on the source of these data and the scaling of time are provided in Study III and Appendix B.”

* Page 22/23 now writes (changes underlined):

“Time was measured as the number of months since the baseline assessment. In order to obtain approximately linear trajectories, we transformed the timing metric based on visual inspection of the observed data: months1*/*2 was used as the timing metric for reading and math trajectories, and months2*/*3 for science trajectories. Figure B1 in Appendix B shows the trajectories before and after transformation. We chose a fractional exponential rather than a log transform, in order for the baseline assessment to retain a value of 0 and to avoid negative time values. Either transformation allows to increase distances between earlier timepoints, and decrease distances between later timepoints.”

* Appendix B, Figure B1 (page 37) features plots of all observed trajectories, before and after transformation:



**Comment 4:** "Considering the relatively large number of subgroups in Figure 2, some with relatively small sample sizes, this tree may overfit and not generalize well to other samples." (p.8) wouldn't it be possible to get an estimate of this using 10-fold or LOO cross validation on these very data?

**Response:** We agree that cross-validated estimates of predictive accuracy are important. These are extensively assessed in Study III, but this might not yet be obvious for readers at this point.

* We therefore now point at this on page 9:

“In light of the small number of respondents for some of the subgroups in Figure 2, these effects may not generalize well to other samples. Therefore, generalizability of the tree structures obtained will be empirically evaluated using cross validation in Study III.”

* And at the very end of the Introduction (page 11):

“… the cluster-level parameter stability tests provided the most parsimonious tree structure thus far. The random-effects parameters in Table 2 show that this smaller tree size also results in slightly higher variance of the random intercept: With more subgroups or terminal nodes, less variance is captured by the random effects and more variance is captured by the tree structure. Whether cluster-level parameter stability tests and/or random-effects initialization yields more accurate or better generalizable trees will be assessed next, in Studies I, II and III. “

**Comment 5:** Now I am trying to sneak in some more of my own research (please feel free to ignore). I argued earlier (Brandmaier et al., 2018; https://psycnet.apa.org/record/2018-18815-001) that it is useful to report LGCM parametrizations in terms of latent variable reliabilities, particularly reliability of the latent slope. This might help to better gauge how reliable the chosen design was to find individual differences in change in the first place. We proposed effective curve reliability (ECR) as such a measure that can be interpreted similarly as reliability estimates in cross-sectional data. You can compute ECR with the semper R package (free to install from here: https://github.com/brandmaier/semper) like this

model = lgcm(timepoints=0:4, slope.variance = 0.1, intercept.slope.covariance = 0,

intercept.variance = 0.1, residual.variance = 5)

This yields a rather low to medium slope reliability for the model with the small random slope:

> ecr(model)

[1] 0.3604651

and a medium to good slope reliability for the condition with large slope variance:

model = lgcm(timepoints=0:4, slope.variance = 0.4, intercept.slope.covariance = 0,

intercept.variance = 0.1, residual.variance = 5)

> ecr(model)

[1] 0.6927374

This may simply serve as a vehicle to drive home the point that you made great choices for your simulation.

**Response:** Thanks! We earlier guestimated appropriate levels for the variances of the random slope in the simulation. The suggested effect size / reliability measure provides a useful quantification of the strength of variation introduced. We have added the following to the section describing the data-generating design (page 14):

“Expressed as effect sizes, the values chosen for σb0 correspond to intra-class correlations of 0.167 and 0.444, respectively. Values chosen for σb1 correspond to effective curve reliabilities of 0.219 and 0.286 for σb0 = 1, and 0.528 and 0.615 for σb1 = 4, as computed per Brandmaier, Von Oertzen, Ghisletta, Lindenberger and Hertzog (2018).”

**Comment 6:** "We evaluated tree accuracy by counting the number of splits in each tree" (p.13) While I find this a very interesting measure to report, I wonder whether this shouldn't also be about the predictive accuracy of the trees. Often, we are not interested in finding the one right model with trees and forests but really in finding a useful representation of constellations of predictors that help us with theory building or prediction. Particularly, by their inherent regularization, trees may find a better predictive model than the true model (e.g., as argued by Galit Shmueli's "To Predict or To Explain" article). So, wouldn't it make sense to compute log-likelihoods of hold-out data given the respective trees as a measure or predictive accuracy?

**Response:** We agree that in addition to accuracy of subgroup recovery, accuracy of predictions on new observations are also important, especially for real-world data, where the true subgroups are not known. For the simulations (Studies I and II), we focused on accuracy of subgroup recovery. We tried evaluating predictive accuracy, but it turned out cumbersome, if not impossible:

* The predict method for R objects of class semtree return only node memberships (although we would certainly like to hear if we overlooked something). We could potentially use those node memberships to compute predictions from the node-specific SEMs, but this requires a higher level of familiarity with the semtree infrastructure than we have yet obtained.
* We believe this approach would also require use of the predict method for models of class lavaan, which only computes factor scores, instead of observed values of the response variable (see also ?lavaan::`predict,lavaan-method`). For LGCMs, the factor scores would represent the subject-level latent slopes and intercepts, computed based on observed variable values and the model parameters. Surely, we could compare these factor-score predictions to the values we used for simulating the response, instead of the observed response variable values. Yet, one could argue this amounts to evaluating model accuracy, not predictive accuracy.
* Further, the factor-score predictions of SEM trees would not be comparable to predictions of LM(M) trees and LongCART, which compute predictions of observed values of the response. To obtain comparable predictions of the subject-level intercept and slope from LM(M) trees and LongCART, we require to combine the estimated fixed and random-effects coefficients from these fitted models. Yet, mixed-effects models (and thereby LMM trees and LongCART) do not return random-effects predictions for new observations; these are always zero. The predictions would thus only evaluate the accuracy of the tree structure, which in effect was already evaluated in Studies I and II.

The reviewer’s comment points at some fascinating differences between prediction with SEMs and LMMs, which are perhaps beyond the scope of the current paper. We have therefore stuck to computing predictive accuracy only in Study III. We expect performance of SEM trees to be comparable, but it is difficult to empirically evaluate this as SEMs do not seem to allow for the same type of predictions as LM(M) trees do, and vice versa.

**Comment 7:** I could imagine that readers may be interested in how well a standard non-tree approach would fare against the tree approaches; for example, what would be the mean MSE if one added all linear predictors to the model as in a standard GLM/SEM approach?

**Response:** As a benchmark, we have added both a very simple and very complex LMM to the comparisons on math, reading and science trajectories:

* Method section of Study III (page 24) now reads:

“Although LRT- and score-based SEM trees performed very well in the simulations, theycould not be used in this study because growth-curve SEMs do not allow for incorporating continuous time. For comparison, we therefore fitted two linear mixed models, both with random intercept and slope of time freely estimated:

* + A simple LMM with a linear fixed effect of time, reflecting a situation where no subgroups are detected or used for prediction.
  + A complex LMM with a linear fixed effect of time, as well as main effects and interactions with time of all possible partitioning variables.

[…]

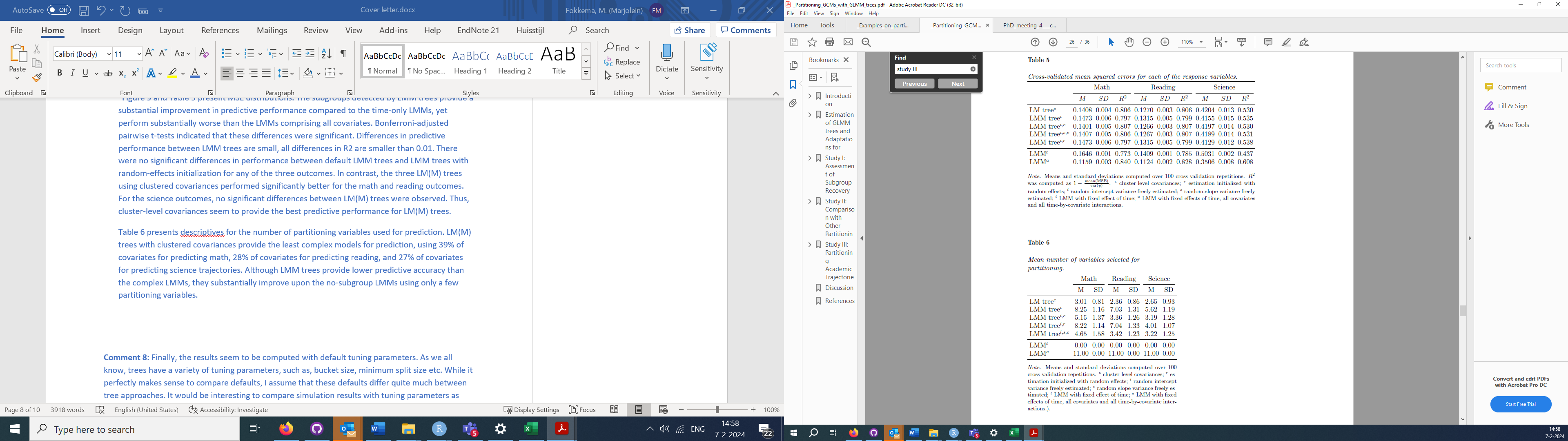
“Model complexity was measured by two indicators: Tree size as quantified by the number of splits in each tree, and the number of partitioning variables included in the final model.”

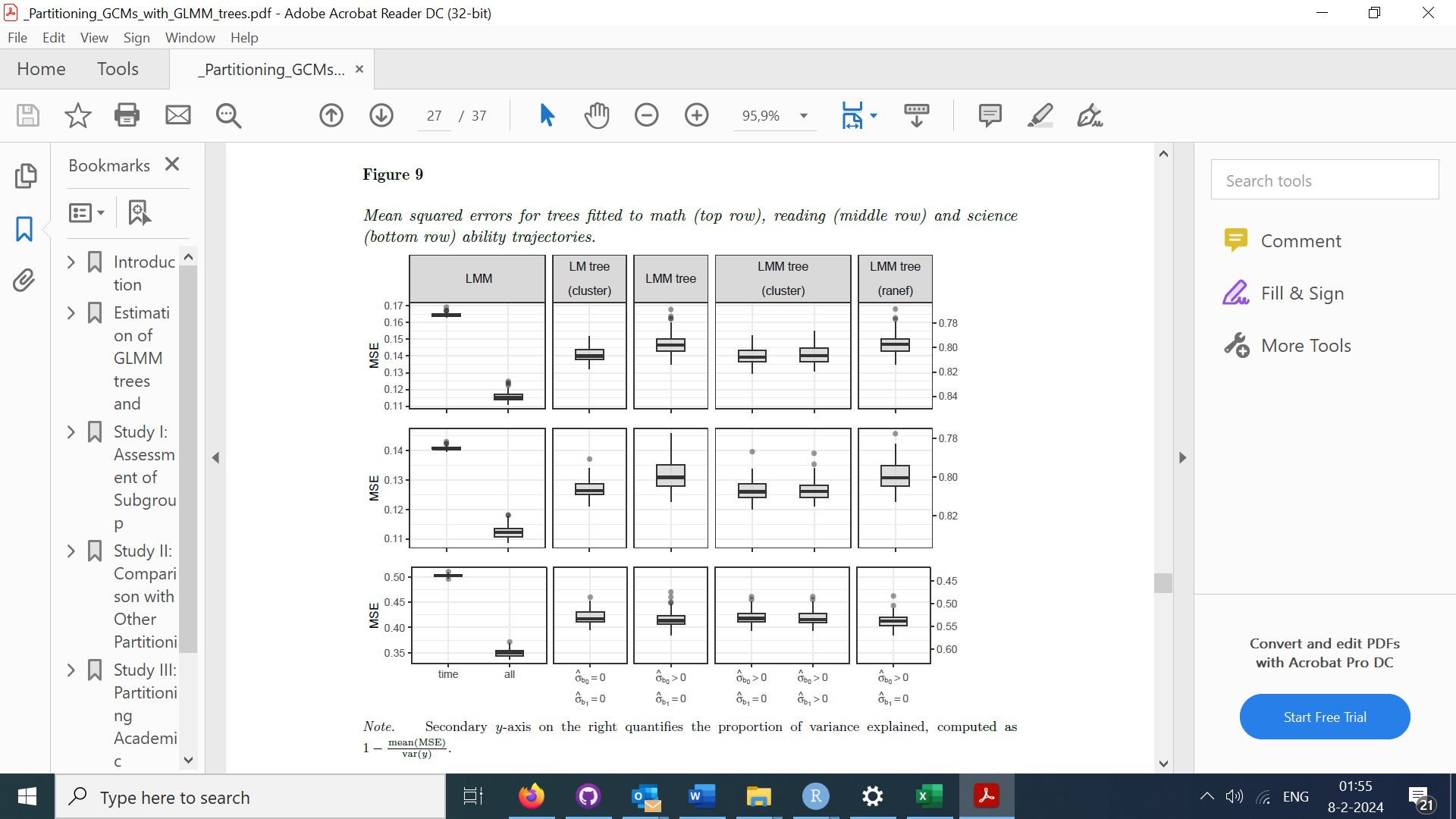
* The Results section of Study III (page 25) has changed substantially:

“Table 5 and Figure 9 present MSE and R2 distributions. Simple LMMs comprising only a fixed effect of time performed worst, but their R2 values ranging from 0.43 to 0.79 illustrate that time is the major predictor of math, reading and science abilities. The complex LMMs, comprising fixed effects of time and all covariates as well as their interactions, showed best performance, with R2 values ranging from 0.58 to 0.85. Table 6 shows the number of covariates used for prediction, and indicates this performance gain comes at a considerable cost in complexity.

Predictive performance of LM(M) trees strikes a balance between the simple and complex LMMs. As shown in Table 5, LM(M) trees with clustered covariances provide predictive accuracy closer to the complex than the simple LMMs, while using only a fraction of the covariates: 39% of covariates for predicting math, 28% of covariates for reading, and 27% of covariates for science trajectories, on average. As such, LMM trees substantially improve upon the simple, no-subgroup LMMs using only few partitioning variables.”

* The two benchmark LMMs have been added to Table 5 and Figure 9, and Table 6 was added to the manuscript (pages 26-27):





**Comment 8:** Finally, the results seem to be computed with default tuning parameters. As we all know, trees have a variety of tuning parameters, such as, bucket size, minimum split size etc. While it perfectly makes sense to compare defaults, I assume that these defaults differ quite much between tree approaches. It would be interesting to compare simulation results with tuning parameters as similar as possible (at least w.r.t bucket size) - could some of the observed differences be explained away by differences in tuning parameters? At least, one could discuss the differences if you don't want to rerun the entire simulation.

**Response:** The following default values are employed for controlling tree depth by each of the tree methods:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **minbucket** | **min.N or minsplit** | **alpha** | **bonferroni** |
| **SEM tree** | 10 | 20 | 0.05 | FALSE |
| **LM(M) tree** | 20\* | 20\* | 0.05 | TRUE |
| **LongCART** | 20 | 40 | 0.05 | FALSE |
| **\***10 times the number of parameters to be estimated; for growth curves the latter would be the intercept and linear slope. | | | | |

Note that:

* Given the value of minbucket, the min.N or minsplit arguments will not affect final tree size, so we focus on minbucket.
* SEM trees employ a wide data format, while LM(M) trees and LongCART employ a long data format for the analysis of longitudinal data, so these values are not directly comparable. With 5 timepoints, the sample size is 5 times larger for LM(M) trees and LongCART, compared to SEM trees.

Thus, with the sample sizes employed in the simulation, the maximum number of terminal nodes and splits that can be implemented based on the minbucket criterion is:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Max. number of terminal nodes** | | **Max. number of splits** | |
| **Method** | N=100 (N=500)\* | N=250 (N=1250)\* | N=100 (N=500)\* | N=250 (N=1250)\* |
| **SEM tree** | 10 | 25 | 9 | 24 |
| **LM(M) tree** | 25 | 62 | 24 | 61 |
| **LongCART** | 25 | 62 | 24 | 61 |

\* Sample size *N* is given for SEM trees, sample size for LM(M) trees and LongCART is given in parentheses.

Figure 7 and the top row of Figure A2 show that these maxima are never reached, so the differences between the tree methods in the defaults do not seem to have any effect on tree size.

Given that the number of splits implemented substantially increases with sample size (Figure A2, top row; page 36), we conclude that the number of implemented splits is mostly a function of the alpha parameter, which is identical between the methods, so does not contribute to differences in methods.

However, the table above shows that the different tree methods do differ in application of a Bonferroni correction. In effect, LM(M) trees employ a more conservative alpha than SEM trees and LongCART. The middle row of Figure A2 depicts the effect of the number of possible partitioning variables, which reveals a very small effect, but also that with an increasing number of predictors, LMM trees implement less splits, while SEM trees and LongCART implement more splits. Yet, employing the Bonferroni correction would likely bring down all boxplots of SEM tree and LongCART somewhat, which might improve performance for trees that were too large, but also worsen performance of trees that were right-sized or too small.

We made the following changes in the manuscript:

* Study II, Method section (page 19) now writes:

“For all model-fitting parameters not discussed above, default settings were employed. The tree-fitting algorithms employ the same value of *α* = 0.05 for evaluating significance of potential splits and deciding whether splitting should continue. Yet, they differ in application of the Bonferroni correction: By default, **semtree** and **LongCART** do not employ a Bonferroni correction for the number of splitting candidates tried at every split, while **glmertree** does. Further, the algorithms have different defaults for the minimum number of observations in a terminal node (argument minbucket). In the current experiments, the default setting for **semtree** implements at most 9 (*N* = 100) and 24 (*N* = 250) splits. Further, **LongCART** and **glmertree** both implement at most 24 (*N* = 100) and 61 (*N* = 250) splits. To evaluate the effects of these differences, we will compare the number of implemented splits against these maxima.”

* Study II, Results section (page 21):

“None of the algorithms reached the maximum number of splits retained as specified by the minbucket argument in any of the simulated datasets, suggesting that this parameter does not affect tree size. The strong effect of sample size suggests the significance level *α* for continuation of splitting is much more important. With *N* = 100, underfitting is more likely, while with *N* = 250, overfitting is more likely, suggesting that the default *α* performs well in these simulations. Combined with the very small effect of the number of potential partitioning variables, the default (lack of) Bonferroni correction also performs well.”

**Comment 9:** In sum, I very much enjoyed reading this manuscript and I am looking forward to use the method in my own research. PS: My heartfelt condolences that you had to endure simulations with LRT SEM trees

**Response:** Thanks, we are happy to hear that! Considering the differences in performance between LRT- and score-based SEM trees, it was worth the wait.