The authors would like to thank all reviewers for their time and the high-quality replies including thoughtful wording. We apologize for the delay regarding our reply and appreciate the patience on the journal side.

# Changelog

This is only a high-level changelog. Detailed replies can be found in the point-by-point replies.

* Added method spcv\_knndm to the package and manuscript
* Aligned all figures with respect to font and point sizes
* Removed CLUTO method from package and paper (as the method is not available anymore)
* Improved reproducibility script (instructions for installation of python based dependencies for plotly() calls)
* Literature research for uses of knndm method in publications
* Package: Added the option to show omitted points for spcv\_buffer
* Added Figure 1 showing an mlr3 ecosystem overview
* Sanitize capitalization in references
* Increased bundled R package version from v2.0.2 to v2.3.0
* Extended and renamed section “Resampling in hyperparameter tuning” to “Resampling in model optimization”

Reviewer 2

*> One structural comment is that the bulk of the manuscript is spent focusing on the details of the cross-validation methods implemented in the package, with very little attention given to the details of the mlr3 framework and how this package integrates with the broader ecosystem. Given that the bulk of development work in mlr3spatiotempcv appears to be focused on integrating pre-existing methods into the mlr3 ecosystem, it would make sense for the manuscript to give more space to the details of what it means to adapt these methods for mlr3 and why that is desirable. Section 3 begins to address this, but focuses mostly on a broad overview of the ecosystem without giving concrete examples of what advantages mlr3 has in comparison to other frameworks.*

This comment mainly refers to the contents of Section 3, pointing out some open questions that we are addressing in the revised manuscript. We identified the following four questions in the reviewer’s comment:

1. What are the advantages of mlr3 in comparison to other frameworks?

While we have addressed important characteristics of mlr3 in the first two paragraphs of Section 3, we did indeed focus on mlr3 itself. The following sentence has been added to paragraph three to highlight how mlr3 is different (perhaps better) than other frameworks:

“Compared to other existing machine-learning frameworks in R (e.g., \pkg{caret} [@kuhn2020] or \pkg{tidymodels} [@tidymodels]), \pkg{mlr3} is the only one that provides a dedicated object-oriented framework for spatial and spatiotemporal resampling methods.

In addition \pkg{mlr3} uses efficient core dependencies from the \pkg{data.table} [@datatable] and \pkg{R6} [@r6] packages, which are particularly well suited for large datasets.

Through this object-oriented approach, \pkg{mlr3} uses substantially less memory than other frameworks due to the use of pointer references and the avoidance of deep object copies whenever possible [@mlr3book].”

2. What is the structure of mlr3, and how does mlr3spatiotempcv fit into this structure?

We presented this in paragraphs 3-6 of Section 3, but due to the rather technical nature of this presentation we decided to include a new figure (Figure 1), which we believe complements the text very well as it provides a visual overview of mlr3, and mlr3spatiotempcv’s place in the mlr3 ecosystem.

3. Why is it desirable to integrate pre-existing methods (or their implementations) in mlr3?

We have added the following justification to the fourth paragraph of Section 3, which addressed the technical fact that pre-existing methods are either imported into or re-implemented in mlr3spatiotempcv:

“By closely following previously proposed methods and existing implementations, we allow users of these established and tested approaches to transition into \pkg{mlr3} without having to adjust their resampling procedures.

To reduce unnecessary upstream dependencies, some methods were re-implemented instead of importing them from the respective upstream packages.

”

4. How is this done?

We would like to point the reviewer to the third and fourth paragraph of Section 3, which we believe addresses this technical aspect (importing functions and re-implementation of methods as appropriate). In addition, sections 1.5 to 1.8 of the mlr3book address these points in additional detail.

*> Secondly, the manuscript claims (on page 3) that alternative implementations focus exclusively on model assessment and not model selection, and that the integration with the mlr3 ecosystem provides better support for model selection. This claim is, as best as I can tell, never returned to. I'm not sure I understand the distinction being made here, given that model selection typically follows directly from model assessment. I believe this section either needs more explanation, clearly demonstrating what mlr3 provides that is not possible through other methods, or should be cut entirely.*

That claim aims to hint the additional capabilities of an ML framework compared to standalone packages which focus only on specific building blocks of an ML pipeline, in that case model assessment.

By “model selection” we mean the (automatic) selection of the best modeling technique and/or model setup in a model assessment workflow - which is supported by frameworks like mlr3 and others but not by standalone R packages - at least not the ones mentioned here.

We have rephrased the respective section as follows to make it more clear:

“One limitation of all the packages which only provide a subset of the machine learning building blocks, e.g., resampling methods, is the sole focus on model assessment.

Through the integration into the \pkg{mlr3} framework, \pkg{mlr3spatiotempcv} offers model selection capabilities, i.e., applying and evaluating various algorithms across different preprocessing setups (e.g., all the preprocessing ‘pipelining operators’ of \pkg{mlr3pipelines}) while being able to make use of parallel execution and enhanced logging abilities.”

*> I think it would be appropriate to discuss the spatialsample package here, as the official tidymodels implementation of spatial cross-validation which treads a lot of the same ground as mlr3spatiotempcv (though directly implements these methods, rather than importing them from other packages).*

We have updated the paragraph and mentioned spatialsample from the tidymodels framework, which has likely been heavily inspired by mlr3spatiotempcv but never officially linked/mentioned {mlr3spatiotempcv} in any resource.

{mlr3spatiotempcv} has been around earlier (2020-11-11) than {spatialsample} (2021-03) with the former being available on GitHub before for an even longer time.

The brief mentioning of {spatialsample} is based on the motivation not to create a "comparison competition" between the two packages but acknowledge its presence as a welcome additional option for spatial CV in R.

Users should make up their own mind which package/ecosystem to use, based on their needs and preferences.

*> "Since re-using the learning sample L for testing, i.e., T := L, would yield the over-optimistic resubstitution or apparent performance"*

*> Could you either cite or briefly explain this point?*

We rephrased the sentence slightly and added Hand (1997) as a reference for this widely known fact: “...would over-optimistically estimate the model’s predictive performance on new instances (so-called resubstitution or apparent model performance; Hand, 1997)...”

*> "It should be noted that k-fold CV with a large value of k, and LOO CV in particular (k = n),  is not only very time-consuming since the model has to be trained k times; these models  will also be nearly identical since only a tiny fraction of the data is withheld, and therefore  estimation bias increases. ‘Pure’ LOO CV is therefore not recommended for machine-learning  model assessment."*

*> Could you cite this section? My sense from the literature is that this is a point of contention, though it seems like most actual studies find LOO CV to minimize bias and variance. For instance, section 7.1 of Zhang and Yang 2015 (https://doi.org/10.1016/j.jeconom.2015.02.006) demonstrates that LOO CV generally has the smallest bias and variance of CV procedures (and cite a source from 1989 to that effect).*

We would like to thank the reviewer for this hint, as this does not seem to be widely known, and there appear to be subtle differences between model selection and performance assessment situations. As it is beyond the scope of this work to discuss these issues in detail, we have shortened this paragraph, pointing the reader to the ongoing debate, including a reference to Zhang & Yang:

“It should be noted that k-fold CV with a large k, and LOO CV in particular (k=n) is computationally expensive due to the large number of models being fitted. There is ongoing debate regarding the bias-variance trade-off of these methods in model selection and assessment (Kohavi, 1995; Arlot & Celisse, 2010; Zhang & Yang, 2015).”

*> I want to compliment you on this figure! It's very effective. My only nitpick is that the viridis colors in the upper-right panel might seem to imply that the various green blocks are "more connected" to each other than to the purple and yellow blocks, and using a qualitative scale with more distinct hues might be useful to reinforce that the folds are independent.*

Thank you, this is great to hear!

We have changed the color scheme to a qualitative one and hope that this improves the upper right sub-figure.

*> There are some LaTeX comments in the text: "% this is required to not expand the authors in the table citations on first use"*

These were placed to justify/document the use of certain latex commands in the text which might look unusual or unneeded on the first look.

We have removed them.

*> At first glance it's not obvious that the buffering is removing points from either set (it looks like a single fold of standard LOO CV, where the test set happens to be the point furthest from the training data). Is there a way to either show buffered points in a third color, or to display the full data either as its own figure or as a panel next to this figure?*

Indeed, so far, it was not possible to show the omitted points for the spcv\_buffer method. It was only possible for the sptcv\_cstf methods. We have added the ability to show omitted points for spcv\_buffer into the package with v2.2.0.

*> I don't know that the paper as it currently stands makes it clear why users would want to provide their own partitions. It would be good to provide examples at the start of the paragraph of when `custom\_cv` is useful for this purpose. The last sentence of the paragraph suggests why it would be useful, but doesn't explain the benefits (statistical or otherwise) of partitioning using altitudinal zones or other factors*

*I'd remove the reference to the deprecated `mlr` package. It confuses the terminology.*

We have removed the reference.

*It might be appropriate to cite Meyer et al 2018 (https://doi.org/10.1016/j.envsoft.2017.12.001), or any of the citations from the first paragraph on page 5 (Martin et al 2011, Kasurak et al 2011, Anderson et al 2005) to motivate why user-defined partitions would be useful.*

Thank you for the precise suggestion. We have included the mentioned references into the referenced paragraph.

*I don't know that printing `print(rsmp\_cv\_group$instance)` is useful here -- the generated table is rather difficult to understand in the format of the paper, and winds up floating without a caption between two figures.*

We agree with this, especially since there is no further sentence which refers to the printed table. We have removed it.

*Figures 7-8:*

*The text is difficult to read. I'd recommend resizing all the autoplot graphics to match Figure 10.*

We tried to align all autoplot figures in the manuscript with respect to point size, axis font size and overall width/height while increasing font sizes of axes and legend elements.

*Figure 12:*

*This figure is nearly impossible to interpret. I think you need one of the axes to be truly "straight on" for this to make visual sense; the current orientation looks almost as though transects (with multiple observations along each transect) are being assigned to folds.*

We agree, the figure is not easy to interpret, especially due to its 3D nature on a 2D surface. However, it is the only orientation in which the fold grouping based on location transects is visible. And yes, due to the “leave-location-out” characteristic of this resampling method, individual locations (across time) are being left out, making it look like transects are being assigned to folds. If you view the figure with the mentioned birds-eye perspective in mind, the targeted fold groups become visible. We have tested other viewing angles again, but were not able to find one which provides a clearer view.

To aid the interpretation, we added more details to the figure caption.

*“To define a resampling strategy, the rsmp() function is used to generate a resampling object using four folds and two repetitions following a random sampling logic (“cv”).”*

*I'm a little confused about what "cv" represents here. Should this be "repeated\_cv" to match the code?*

Well spotted! As a repeated resampling is created (using the “cv” method) it should be repeated\_cv instead of cv.

*replace "repeated\_cv" by "repeated\_spcv\_coords"*

*Should read:*

*replace "repeated\_cv" with "repeated\_spcv\_coords"*

Thank you - we replaced it.

*The `rsmp\_nsp` plots should be first (before `rsmp\_sp`) to match the rest of the section.*

We agree and switched the order.

Page 26/27:

*The third/fourth paragraph ("In other application scenarios...") suffers from being too long; getting into the details confuses the point. Simplifying this to (for example) "In other scenarios, such as when predicting crop types across monocultural fields (Section 1), it makes sense to group observations into blocks based on previously known boundaries and resample at the polygon level ("cv" method with grouping)." gets the point across much faster, without getting mired in the specifics of one case study.*

Thank you for the precise and well-pointed improvement. We agree and have shortened the paragraph, including additional minor adjustments to make it more precise. Note that there might still be minor differences visible between the single-plot figures and multi-plot figures due to the internal sizing of ggplot2 in these different scenarios.

*Page 28:*

*“For example, buffering, or the use of a spatial or temporal separation distance between training  and test sets, is currently only implemented for some methods ("spcv\_buffer", "spcv\_disc",  and "sptcv\_cstf" with both space\_var and time\_var). Its use should, however, be limited  to use cases involving a prediction distance, as a buffer zone reduces the size of the training  sample and introduces the risk of geographically biased training data.”*

*Do you have a citation for "geographically biased training data", or somewhere that this concept is more thoroughly discussed? I think I understand the risk being described, but this is (as best as I can tell) the first time this concern has been mentioned in the paper.*

We have added the references Meyer2021 “Predicting into unknown space? Estimating the area of applicability of spatial prediction models” and Zhu2015 “Predictive soil mapping with limited sample data” to the paragraph, which discuss the topic in more detail. The topic is too big to be discussed within the scope of this article and within in the context of the presented methods because it highly depends on the available data characteristics. However, we believe it is important to raise awareness for it as it becomes important when applying (spatial) CV methods and interpreting prediction results.

*No seed is set, which means many of the results cannot be reproduced. Please set a seed.*

A seed had been set for some chunks, but not for all. We have now included a seed in the setup chunk that should be honored for all commands executed after it.

*I'd recommend including a comment at the top of the reproducibility script with installation requirements for dependencies that aren't automatically installed alongside mlr3spatiotempcv. The plotly docs suggest this set of commands to resolve this issue:*

*```*

*install.packages('reticulate')*

*reticulate::install\_miniconda()*

*reticulate::conda\_install('r-reticulate', 'python-kaleido')*

*reticulate::conda\_install('r-reticulate', 'plotly', channel = 'plotly')*

*reticulate::use\_miniconda('r-reticulate')*

*```*

*After running those commands, I got the following error:*

*```*

*Error in py\_run\_string\_impl(code, local, convert) :*

*NameError: name 'sys' is not defined*

*```*

*At which point I gave up.*

Thank you for pointing this out. Reproducibility is important to us. Even though installation of third-party interpreters and their packages differs across operating systems, we have now included helpful instructions on how to install a virtual environment for python packages at the top of the reproducibility script. There are many ways to configure python and its packages, hence many users also likely prefer to reuse existing packages instead of installing the miniconda framework in the first place.

We did not encounter the error of the missing ‘sys’ package while reproducing the dependency installation in a fresh dockerized environment.