We thank the reviewers for their comments and suggestions. Many of the comments are quite good and will improve the quality of the paper. Minor comments and typos have now been fixed in the text, and we thank the reviewers for pointing them out. Our point-by-point response to the reviewers' major comments follows, with their comments italicized.

Reviewer 1: *. . . not much is discussed about the large sample consistency of the method.* As far as we know, the identifiability of the Mondrian process in the infinite data limit is still an open problem. If such a theory were discovered, it would likely generalise to random tessellation processes. There is however work showing that Mondrian forests achieve minimax convergence rates for regression (Mourtada et al. 2018), and in future work those proofs may be adapted to random tessellation processes. We have added this discussion to the manuscript.

Reviewer 2: *. . . I'm curious about the comparison of runtimes against various methods.* We display below the mean running time (in minutes) across different methods for the largest dataset *SCZ93* (left table) and runtimes for the wuRTF on all datasets (right table). The experiments were run on an Intel Xeon CPU E5-2683v4@2.10GHz.

Dataset	LR	SVM	RF	MRTF.i	uRTF.i	MRTF	uRTF	wMRTF	wuRTF
SCZ93	0.006	0.012	0.006	0.005	89.106	43.534	35.802	41.759	41.068

Dataset	*SCZ42*	*SCZ51*	*GL85*	*SCZ93*
Runtime	5.066	8.641	14.679	41.068

While the runtimes are not the focus of the paper, we will include a complete version of these tables in the supplement. Further, since submitting this work, we've developed new inference based on pseudomarginals allowing the spherical approximation and rejection sampling to be replaced by a scheme allowing exact samples without computation of $\lambda^{d-1}([a])$. Finding the radius of the sphere and rejection sampling are the bottlenecks for our methods, and so this advance will considerably improve the runtimes of RTF methods.

Reviewer 3

Comment 2: The approach would only seem to work in a bounded domain... W is compact, and so we should have used a strict subset. This is now corrected in the text. We note that projective processes can be extended to unbounded domains using Kolmogorov extension theorems. This is done for example in the Nagel and Weiss reference, but we do not consider it as it is not relevant for inference.

Comment 3: I was skeptical that a rejection sampler would work as written in a space of even moderately high dimension . . . does the hyperplane . . . still intersect a with reasonable probability? & Comment 4: Similarly the authors state eg on line 168/169 that explicit computation of the polytopes a is not required . . . Is this approximation not increasingly poor in higher dimensions? Rejection sampling and computing the radius of the approximating sphere are computational bottlenecks (due to the reasons raised by the reviewer). This leads to longer runtimes. For $D = 85$, we are able to conduct inference with the rejection sampling, indicating that the interaction is still possible at this dimensionality. Since submitting this work, we've improved inference using a pseudomarginal method in which, instead of choosing a polytope to cut with probability proportional to the radius of a sphere, we instead sample a hyperplane cutting the whole domain and choose a polytope to cut uniformly among all polytopes that the cut intersects. This obviates the need for approximations and rejection sampling. We will discuss or report on this new method in the camera ready copy, should this work be accepted.

Comment 5: What if lots of the test data is outside the collection of convex polytopes? When we form convex hulls, we consider a version of the training data that includes the predictors of the testing data. Testing data lying outside of the convex hulls formed by the training data will be 'snapped to the nearest' polytope. Further, when data are missing-at-random, test data will not generally lie outside of the convex polytopes formed by the training data and so not much data will be snapped in this way. Data affected by test/train shifts may suffer from this approach. However, test/train shifts tend to confound any machine learning method and so this analysis is outside the scope of the paper.

Comment 6: The tessellations in T will not be equal wp1. . . . what is the mode of T? For each random tessellation process in a forest, we predict a label for each test point. By 'mode', we meant to refer to the mode of the distribution of the predicted label, and not the of the tessellations themselves. We've now clarified this in the text.

Comment 7: What is the effect of setting the Dirichlet parameter α as you have in line 219... Our setting of α provides a weak prior matched to the empirical label frequencies. We do not use any higher level features. This method is popular for likelihoods in Bayesian nonparametrics. An exploration of this likelihood and also hierarchical likelihoods and Polya trees are an area of future work.

Minor comment 6: Is it not sufficient to sample u from [0,r]? This was a typo. It's indeed sufficient and correct to sample u from $[0, r]$, and we do that in our implementation.

Minor comment 14: If the budget is infinite, when does the partitioning procedure stop? We use a 'pausing condition' that is described in Section 2.2.2 and originally proposed in Breiman 2001: if the training points in a polytope all have the same label then no further cuts are performed on the polytope.