document

We thank both reviewers and the editor for their helpful and insightful comments. We respond to each of the comments below. For ease, we italicize the reviewer comments. Quotes from the revision are in red. Note that the reference numbers in this response correspond to the bibliography of this document only; the references are numbered differently in the main text. We are grateful to have the opportunity to resubmit this new and improved version of our manuscript.

Reviewer 1

itemize

The authors rule out many potential methods for this problem based on the disered criteria of "intepretability". However, they conclude by noting a weakness of their method (based on simulated data) is the unreliability of intepreting the results with respect to specific edges. Therefore, there is arguably an inconsistency between what is promised and what has been achieved.

We have tried to clarify in two ways. First, we more clearly explain the desideratum in the Introduction:

For prognostic and diagnostic purposes, merely being able to differentiate groups of brain-graphs from one another is sufficient. However, for treatment, it is desirable to know which vertices and/or edges are malfunctioning, such that therapy can be targeted to those locations. This is the motivating application for our work.

Unfortunately, this desideratum is not achieved with such a small sample size and data distributed according to our synthetic data analysis. We have highlighted this by adding the below line to §5.1.1:

Despite our stated desideratum of interpretability of the resulting classifier in terms of correctly identifying the signal-edges and vertices, for data sampled from this assumed distribution, sample sizes of < 50 seem to be insufficient.

Indeed, with small sample sizes, subspace identification is a difficult problem, even under the true model.

Moreover, from a general point of view, it would greatly enhance the value of the paper if comparison to some "less intepretable" methods were offered; in general, one might imagine practitioners making some trade-off between performance and interpretability, and there is an opportunity here to offer valuable data to aid in that decision. It would be useful to know how the method proposed in the paper compares to some standard baseline other than Naive Bayes, such as those mentioned in the third paragraph in the introduction. Another simple baseline would be logistic regression, using edge/no-edge between vertices as the features, with L1 regularization on the weights, which should be quite similar to the incoherent estimator.

We have now compared the performance of our classifiers both in simulation and on the real data. In the simulations, Figure 3 now demonstrates that the lasso classifier yields errors similar to our incoherent classifier, despite that it requires about 10 times longer to run. We modified the text around Figure 3 to emphasize this point, including adding the below two paragraphs:

An important aspect of any algorithm is compute time, both of training and testing. The signal subgraph classifiers that we developed are very fast. Computations essentially amount to computing a test-statistic for all $|E|$ edges, then sorting them. The parameter estimates of the likelihood and prior terms come directly from the same test-statistics used to obtain the significance of each edge. Thus, obtaining those estimates amounts to essentially computing a mean. On the other hand, the lasso classifier, which yields worse signal detection and misclassification rates than both our classifiers, requires an iterative algorithm for each value on the hyper-parameter path Tibs96. Despite that efficient computational schemes have been developed for searching the whole regularization path Efron2004, such iterative algorithms should be much slower than our classifiers.

Indeed, the lower panel of Figure 3 demonstrates that our MATLAB implementation of the signal subgraph classifiers are approximately 10 times faster than MATLAB's lasso implementation. All the results shown in Figure 3 include errorbars computed from 100 trials, each with 100 held-out samples, demonstrating that for these simulation parameters, the differences are highly significant. Although the quantitative results may vary for different implementations and different parameter settings, our expectation is that the qualitative results should be consistent. Thus, because our classifiers have lower risk, better signal identification, and run an order of magnitude faster than the standard, we do not consider lasso in further simulations.

Moreover, we added Table tab:bakeoff, which compares the performance of a variety of "interpretable" and "less interpretable" classifiers on real data. The table and corresponding text are below:

table[h!] Bake-off comparing a number of different classifiers on the MR connectome sex classification data. Error indicates misclassification error using the best hyper-parameters found for each classifier. P-value indicates the p-value of a one-sided McNemar's test comparing each classifier to the coherent classifier. The coherent classifier is significantly better than all the others, except the incoherent classifier. center tabular—c—c—c— classifier error p-val

We next compare the performance of our classifiers on this MR connectome sex classification data set to several others. First, a standard parametric classifier: lasso. We chose the regularization parameter via a 10-fold cross-validation. Second, a non-parametric (distribution free) classifier: k_n-nearest neighbor (kNN) which operates directly on graphs VP11$_s$$uper.This$k$NNclassifierusestheFrobeniusnormdistancemetric.Wetriedallk$ $\in [n]$ and simply report the best performance. The universal consistency of this kNN classifier is useful in assessing the algorithm complexity supported by this data. In particular, given enough samples, kNN will achieve optimal performance. Less than optimal performance therefore indicates that the sample size is not sufficiently large for this kNN classifier. Third, a graph invariant based classifier. We computed six graph invariants for each graph: size, max degree, scan statistic, number of triangles, clustering coefficient, and average path length PCR10, normalized each to have zero mean and unit variance, and then used a kNN with ℓ_2 distance metric on the invariants.

Despite the small sample size, , Table tab:bakeoff demonstrates that the coherent classifier is significantly better than all the others (except the incoherent classifier), as assessed via a one-sided McNemar's test. Note that although lasso and the incoherent classifier achieved the same misclassification rate, they erred on different subjects, suggesting that further improvements in performance might be possible from combining these two classifiers. We do not pursue that option here.

The hyper-parameter section is unclear, and therefore the experimental results are unconvincing. Specifically, on the non-synthetic experiment for classification of sex based on MR connectome, cross-validated misclassification was used to judge performance.

We have added "§2.4.5 Hyper-Parameter Selection" to clarify this issue.

The signal subgraph estimators require specifying the number of signal-edges s, as well as the number of signal-vertices m for the coherent classifier. In both cases, the number of possible values of finite. In particular, $s \in [d_V]$ and $m \in [V]$. Thus, we implement cross-validation procedures, iterating over all possible settings of the hyper-parameters, to choose the hyper-parameters. For all simulated data, we compare hyper-parameter performance via a training and held-out set. For the real data application, the sample size is unfortunately too low to justify a held-out corpus, we therefore utilize a leave-one-out cross-validation procedure.

(By the way, it is only mentioned in the abstract and Figure 6 caption that the task is actually gender classification; this could be made more clear in the text.)

Thank you, we have added the word "sex" all over the manuscript, in particular, in the title of "§5 MR Connectome Sex Classification", and throughout the following paragraphs.

These numbers only reflect performance when the hyper-parameters are set optimally for the data being tested, therefore it's not surprising that the more complex models did better. To compute how well the model would perform when the hyper-parameters are not known, they must be set on some held-out data, and then applied to the test data.

We agree that the better performance of the more complex (nested) model is expected assuming the data sample size is sufficient. Our simulations and synthetic data analysis are designed to assess the extent to which we can trust which algorithm is doing better with finite sample sizes. Comparing performance with the kNN classifier indicates that the sample size is not large enough to support such a non-parametric classifier. We added the below sentence to clarify this issue. If data sample sizes were larger, we would use a different cross-validation scheme. Also note that the coherent classifier, in some very real sense, does not have more complexity. In particular, the number of edges in the incoherent and coherent classifier can both vary between 1 and $V2$.

The universal consistency of this kNN classifier is useful in assessing the algorithm complexity supported by this data. In particular, given enough samples, kNN will achieve optimal performance. Less than optimal performance therefore indicates that the sample size is not sufficiently large for this kNN classifier.

In the top panel of Figure 4, for small numbers of training samples, the coherent estimator has a larger miss rate than the incoherent estimator, thus it is stated that to pick the right estimator, the number

of samples must be known. However, how many samples were used to generate the plot? Is there a confidence interval, and is the difference between estimators when the number of training samples is small statistically significant?

We now ran this simulation for 200 trials to estimate the mean and standard error in those plots. Both the main text and caption now states this clearly. The errorbars indicate the the differences are likely significant.

In most applications, obtaining subgraph information (connectivity) about specific vertices is easier than obtainining information about all vertices. Can this procedure be used to provide a principled method to estimate important vertices and then focus additional data acquisition selectively to maximize information regarding some class identity?

Thank you for pointing out this omission! We have added a sentence to the discussion mentioning our work on this interesting and very related topic:

Fifth, extension to situations for which none of the vertices are labeled VP11$_Q AP, VP11_u nlabeled, only some subset of vert$

It would be helpful if the authors could comment more on the issue of assuming the vertices are labeled.

In many important applications, vertices will not be labeled and a vertex matching procedure will have to be performed. Is this truly a distinct problem, or would it make more sense in such a situation to think of both matching & classification being performed simultaneously?

Again, thank you for asking this particular question. We have added the following sentences to the Introduction:

The field of connectomics (the study of brain-graphs), however, is ripe with many examples of brain-graphs with vertex labels. In invertebrate brain-graphs, for example, often each neuron is named, such that one can compare neurons across individuals of the same species North2007. In vertebrate neurobiology, while neurons are rarely named, "neuron types" Shepherd2007 and neuroanatomical regions Nolte2002 are named.

Indeed, we have two other manuscript pending at PAMI right now addressing the issue of how to classify when vertex labels are unknown VP11$_Q AP, VP11_u nlabeled$.