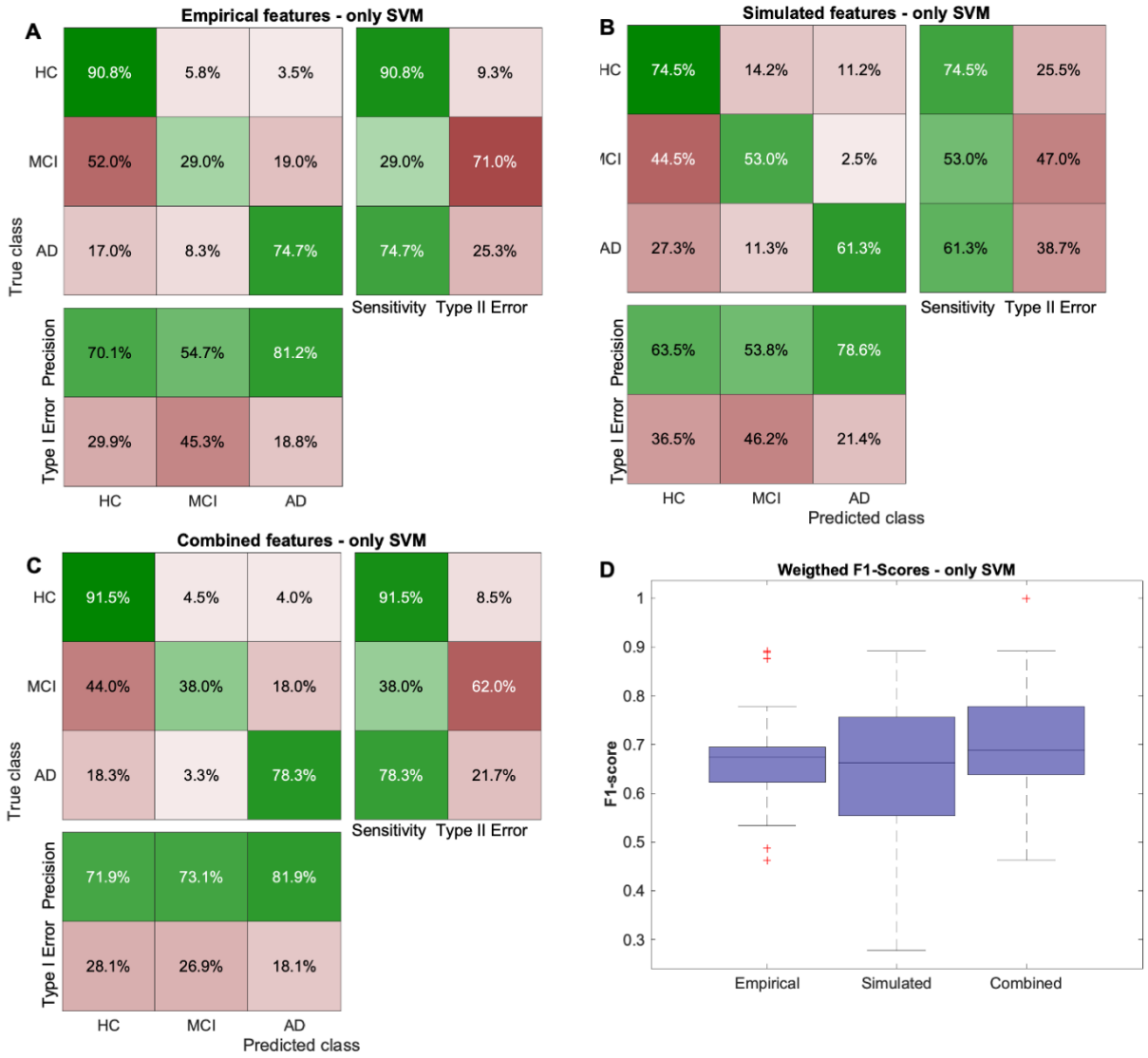
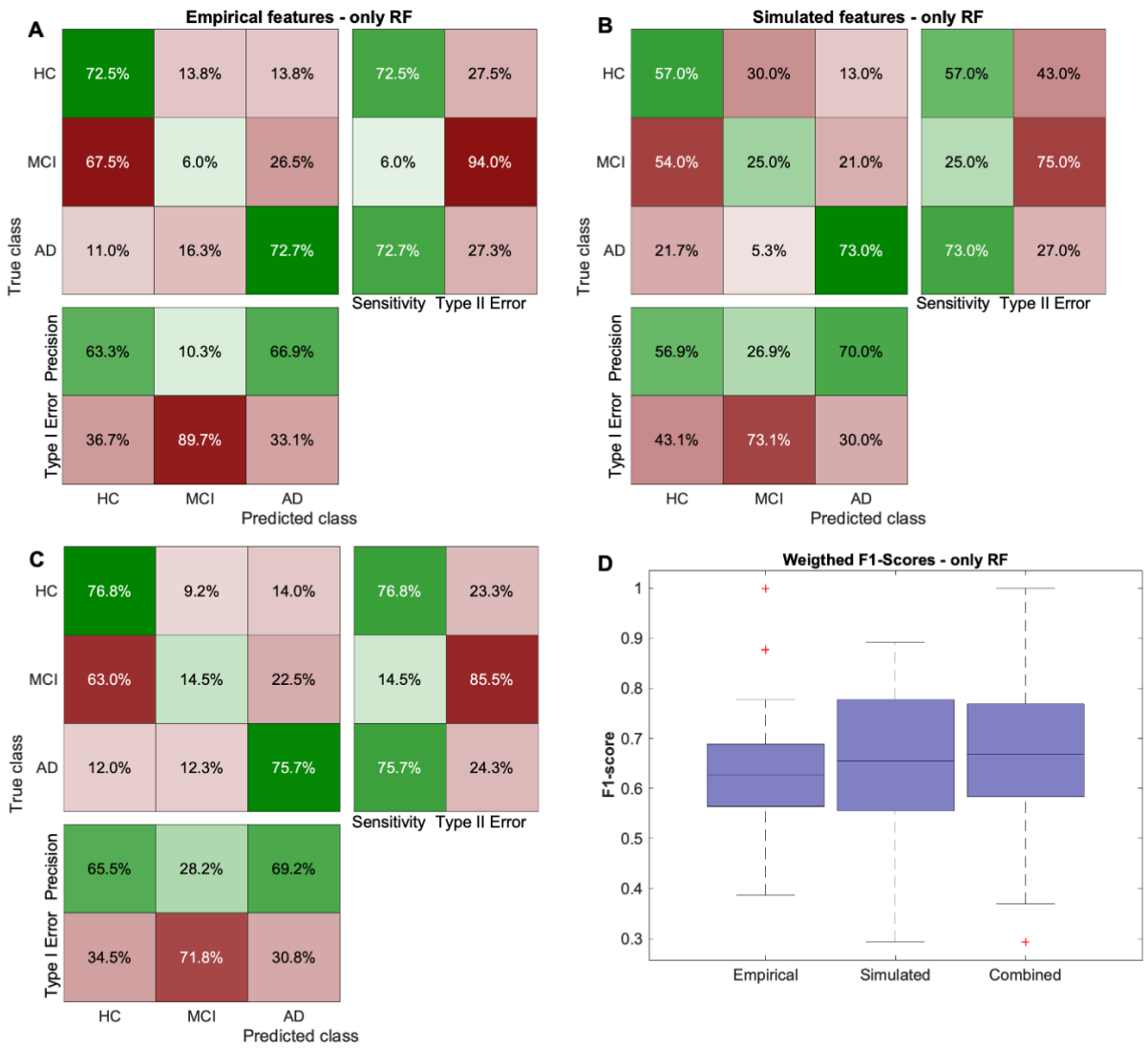
Supplementary Material.

Figures.



Supplementary Figure 1. Results of SVM classification approach. (**A-C**) Confusion matrices are computed by summing the confusion matrices across all 100 cross-validation runs and normalizing per class. As in the (superior) nested approach mentioned in the main text, the combined approach improved prediction of MCI participants. (**D**) Boxplots of mean F1-scores for three different feature spaces.



Supplementary Figure 2. Results of RF classification approach. (**A-C**) Confusion matrices are computed by summing the confusion matrices across all 100 cross-validation runs and normalizing per class. (**D**) Boxplots of mean F1-scores for three different feature spaces.

Tables and detailed methods.

Supplementary Table 1.
Best hyperparameter settings for each feature set for SVM only.

Parameter	Empirical	Simulated	Combined	Searched parameter space
Kernel	RBF	RBF	Polynomial (d=3)	Radial basis function (RBF), polynomial functions
Gamma	0.01	0.1	n.a.	Gamma $\in \{10^{-3}, 10^{-2}, 10^{-1}, 1\}$
C	1000	100	1000	$C \in \{10^{-2}, 10^{-1}, 1, 10, 10^{2}, 10^{3}\}$
Number of features K	30	10	40	$X \in \{5, 10, 15, 20, 25, 30, 35, 40\}$

Explanations of SVM hyperparameters:

Kernel:
The kernel of an SVM is the function that is used for the transformation of the feature space. A kernel is an invertible function $k(x_i, x_j) = r(x_i) \cdot r(x_j)$ used to transform data points $x_i, i \in 1 \dots N$ in such a way as to preserve the relationships between datapoints so that the SVM can learn a linear classification rule in the feature space of $r(x_i)$ that corresponds to a non-linear classification rule in the feature space of x_i. The classification rule is learned in the transformed space and then projected back down into the original data space to obtain a non-linear classification boundary. I.e., the kernel is the function that transforms the original feature space into a higher dimensional (artificial) feature space to separate data points, while it defines also the rule for its back-projection.
We explored two types of kernels. The Gaussian radial basis function (RBF) kernel is the most commonly used kernel for non-linear problems for a variety of reasons (e.g., the kernel is stationary, smooth, and tunable with only a single isotropic parameter). The kernel is defined as follows: $k(x_i, x_j) = exp\left(-\gamma \left|\left|x_i - x_j\right|\right|^2\right)$ for $\gamma > 0$. The polynomial kernel instead raises the degree of the data space using, i.e., quadratic or cubic transforms: $k(x_i, x_j) = (x_i, x_j)^d$. Here we only explored $d \in 2,3,4$ to avoid overly complex decision functions.

Gamma:
The scaling parameter as defined in the RBF kernel (see above). It scales the distance of datapoints to the decision boundary that are used for its calculation. This must typically be tuned empirically using cross validation, as it can take any value over zero, including values less than one to shrink the norms between points, or values much larger than 1, to expand those norms, depending on how the data are clustered.

C:
The soft-margin parameter C controls whether the solution emphasizes a wide margin (i.e., a decision boundary as far away as possible from the closest points), which can lead to some underfitting, versus a narrow margin that can result in some overfitting. E.g., an extremely narrow margin around zero could still perfectly separate datapoints of the training set, but as the decision boundary almost crosses the most similar datapoints between the classes, it will be not very adaptable to new data. Like Gamma, it must be tuned empirically.

Number of Features:
Since our list of candidate features is very high compared to the number of subjects, to find a more robust and interpretable solution (Bellman and Collection 1961, Trunk 1979) , we greatly limit the number of selected features prior to classification.

For the SVM parameters Gamma and C, these are fairly standard search ranges (Chapelle and Zien 2005, Ben-Hur and Weston 2010). Typically a search will span different orders of

magnitude, for example in base 10 or in base 2. We used a coarse grid in our hyperparameter search so as to not overdetermine our model to our relatively small sample.

Supplementary Table 2.
Best hyperparameter settings for each feature set for RF only.

Parameter	Empirical	Simulated	Combined	Searched parameter space
Class weight	balanced	balanced	balanced	None or balanced
Number of Estimators	10	10	10	n ∈ $\{10, 50, 100, 200\}$
Min. samples per split	2	2	4	n ∈ $\{2, 3, 4, 5\}$
Min. samples per leaf	1	3	2	n ∈ $\{1, 2, 3\}$
Max. features	$\sqrt{P}$	none	$\sqrt{P}$	n ∈ $\{P^{-1/2}, \log_2 P\}$

Explanations of RF hyperparameters

Class Weight:
Simply whether or not the model should take into account the imbalanced class representation in the training set. When balanced, incorrect class labels during the training process are penalized more or less depending on whether the correct class is under- or over-represented in the training set. This aims to overcome biases that arrive from different frequencies of classes in the underlying training data.

Number of Estimators:
The number of decision trees to train, i.e. the class estimates of which are aggregated in the final model. In other words, the size of the ensemble.

Min. samples per split:
The minimum number of divergent samples required to split a branch of a tree into two new branches. A lower value means a more detailed tree that can be more precise. E.g., the lowest value would be 1, meaning that even a single subject can be separated by a decision rule of one tree. Lower values are typically required when the number of samples are low, especially compared to the number of features.

Min. samples per leaf:
The minimum number of samples required at the end of each leaf node (the last node of each path down the tree). Similar to min. samples per split, this influences how detailed the tree can become, and lower values are typically required for a low sample setting with many features. When a branch reaches this number, it can no longer be split further.

Max. features:
The maximum number of features to consider when evaluating the best split for each branch. Lower values typically mean better generalization but reduced flexibility.

For the RF parameters, we kept to low values for splitting criteria so that more detailed trees could be learned (as mentioned, this is more important for our N<<P scenario, since generalization is much harder).

References:

Bellman, R. and K. M. R. Collection (1961). Adaptive Control Processes: A Guided Tour, Princeton University Press.

Ben-Hur, A. and J. Weston (2010). A user's guide to support vector machines. Data mining techniques for the life sciences, Springer**:** 223-239.

Chapelle, O. and A. Zien (2005). Semi-supervised classification by low density separation. AISTATS, Citeseer.

Trunk, G. V. (1979). "A Problem of Dimensionality: A Simple Example." IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-1**(3): 306-307.