Supporting Information: A multidimensional ODE-based model of Alzheimer's disease progression

Matías Nicolás Bossa and Hichem Sahli

S1 Supplementary Descriptions

S1.1 IRT model

The ADAS-Cog is discrete and bounded, which points to the binomial distribution as the simplest model. However, this simple model is not the most appropriate because all the rich information from individual sub-items is lost. Therefore, we used an Item Response Theory (IRT) based scoring methodology ([Verma et al., 2015]) that extracts latent traits of three cognitive domains. 12 of the 13 items that ADAS-Cog are divided into three traits: language, memory and praxis. For each of the 12 items, we used ordered logistic models, *a.k.a.* proportional odds ([McCullagh, 1980]) or Samejima's graded response model (GRM) in this context ([Samejima, 1968]), where the three latent scores are shared among a predefined set of items [This model is simpler than that proposed in [Verma et al., 2015] because they used GRMs for graded responses (*e.g.*, mild/moderate/severe) or counts (*e.g.*, number of recalled words) and three-parameter logistic (3PL) models for items composed of dichotomous scores (*e.g.*, Orientation: month, season, place, etc.). For simplicity, we used GRM for all the items.].

Let $A_p \in \{1, \dots, K_p\}$ be the score of the *p-th* ADAS-Cog item, whose possible values are integers from 1 to the number of levels (K_p) of that item, then the likelihood is defined by:

$$\begin{aligned} P(A_p = 1) &= 1 - \text{logit}^{-1}\left(-(x_{\text{Cog},I_p}\alpha_p - d_{p,1}\right) \\ P(A_p > k) &= \text{logit}^{-1}\left(-(x_{\text{Cog},I_p}\alpha_p - d_{p,k}\right) \\ &\vdots \\ P(A_p = K_p) &= \text{logit}^{-1}\left(-(x_{\text{Cog},I_p}\alpha_p - d_{p,K_p-1}\right) \end{aligned}$$

where α_p is called the item discrimination or scale parameter, and $\{d_{p,k}\}_{k=1}^{K_p-1}$ are the item difficulty or location parameters (see [Verma et al., 2015]). There are therefore 3 continuous dynamical variables linked to ADAS-Cog, $x_{\text{Cog,lang.}}$, $x_{\text{Cog,mem.}}$ and $x_{\text{Cog,pra.}}$, each associated with a different cognitive domains. The index variable I_p indicates to which of the three cognitive domains belongs each item p.

The likelihood of an individual score k is known as the item characteristic function

$$P(A_p = k) = P(A_p > k - 1) - P(A_p > k).$$

Figure S1 shows the posterior distribution of the item characteristic functions fitted to the ADNI dataset.

S1.2 Prior distributions and hierarchical structure

In a Bayesian setting, we should choose priors for the parameter Θ_k, Θ_D, $\mathbf{v}(\cdot)$ and $\mathbf{x}_0^s$ (see equations (2), (3) and (4) in the main text), that can be a fixed probability distribution or a parameterized probability distribution. The latter case gives rise to a so-called hierarchical or multilevel structure ([Gelman et al., 2013]) and has different uses related to regularization, shrinkage and smoothing (see [Hodges, 2013], chapter 13).

The following two sets of parameters in the proposed model may benefit from a hierarchical structure. Firstly, the velocity field $\mathbf{v}(\cdot)$ can be modelled parametrically or non-parametrically (*e.g.*, with Gaussian Process priors). The non-parametric models often require a multilevel structure, for example, priors on the covariance parameters. Note that this choice will have consequences on the method options to integrate or approximate the ODE. Secondly, the subject-level parameters (*e.g.*, $\mathbf{x}_0^s$) can be treated as a random effect, which is standard practice for all longitudinal studies, among many others.

One benefit of including a hierarchical structure to subject-level parameters is that it can handle naturally missing observations, which are assumed completely at random in this work. Missing data can be treated as unknown parameters if we want their posterior distribution to impute values on other models, or simply for prediction purposes. If there were known missing data mechanisms in a dataset (*e.g.*, missing not at random), additional likelihood terms could be easily included. These terms should model the distribution of observed and missing values in terms of the appropriate covariates ([Gelman et al., 2013]).

We set the following hierarchical prior on the initial values $\mathbf{x}_0^s$

$$
\begin{aligned}
x_{0,\tau}^s &\sim \mathcal{N}(\mu_\tau, \rho_\tau) && (1)\\
x_{0,A\beta}^s &\sim \mathcal{N}(\mu_{A\beta}, \rho_{A\beta}) && (2)\\
\mathbf{x}_{0,Cog}^s &\sim \mathcal{N}(\mathbf{0}, \Omega), && (3)
\end{aligned}
$$

with normal ($\mathcal{N}(0,1)$) hyperpriors on CSF mean parameters μ_τ and $\mu_{A\beta}$, and half normal prior on CSF variance parameters ρ_τ and $\rho_{A\beta}$. These hyperpriors are very weakly informative since CSF biomarkers were already transformed to take values between 0 and 1.

The cognitive part of $\mathbf{x}_0^s$ requires special attention to avoid identifiability issues associated with the IRT models. Adding an arbitrary constant vector to all $\mathbf{x}_{0,Cog}^s$ is equivalent to subtracting the same vector components to the appropriate location parameters $d_{p,k}$. The same happens with variance and scale parameters α_p. One common approach to achieve identification is to set the latent variable mean to 0 and the variance to 1, as shown in eq. 3, where Ω is a correlation matrix. We let the model have the freedom to learn the correlation terms of the covariance matrix. The Lewandowski-Kurowicka-Joe (LKJ) distribution is usually recommended for correlation matrices ([Gelman et al., 2013])

$$\Omega \sim \text{LkjCorr}(1), \tag{4}$$

where the shape parameter equal to 1 implies that all the correlation terms have a uniform distribution.

S1.3 Calibration

In clinical prediction modelling, calibration refers to the agreement between predictions and observed outcomes. It is critical in clinical decision-making ([Van Calster et al., 2016]) yet it is often overlooked when reporting the results of clinical prediction models.

A common tool for measuring calibration is given by the calibration plots (or reliability diagrams in Machine Learning literature). They consist in plotting the accuracy or proportion of positive outcomes as a function of the predicted probability ([Steyerberg, 2019]). A diagonal line indicates that the predictions are perfectly calibrated. This definition was extended to multi-class prediction, where calibration plots are derived from a recalibration process, either parametric, based on multinomial logistic regression ([Van Hoorde et al., 2014]), or non-parametric, based on vector spline multinomial logistic regression ([Van Hoorde et al., 2015]). More recently, the model-free concepts of *confidence-reliability diagram* and *classwise-reliability diagrams* were defined ([Kull et al., 2019]). The former is the overall accuracy (proportion of observations for which argmax($\hat{\mathbf{p}}$ is the correct class) for each probability $c \in [0,1]$ (*i.e.*, max($\hat{\mathbf{p}} = c$), and the latter is the accuracy for each class as a function of the probability assigned to this class. Model-free calibration plots are made by building histograms with a predefined bin size, for which a large number of validation samples are required. Model-based approaches are more common in medicine, where sample sizes are usually small compared with other prediction applications.

Figure S2 shows the confidence- and classwise-reliability diagrams with different colours (top-left) for the left-out observations of the cross-validation experiment. The shaded regions represent the 95% CI computed as the posterior of a Beta-Binomial model at 0.2 width bins [The larger the bins, the smaller the uncertainty but also the resolution. Both resolution and uncertainty can be reduced by imposing some regularization constraints, *i.e.* using a model-based approach such as a spline-regularized logistic regression.]. Confidence-reliability is only reported for probabilities larger than 0.4 because the 1/3 is the minimum attainable value for the maximum probability for 3 classes. We simulated new labels from predicted probabilities to show how this figure would look if the model were perfectly calibrated (top-right).

The model is overall well-calibrated, but it shows classwise miscalibration, which can be interpreted as an excess of sensitivity to detect AD conversion. This bias may be related to the cross-validation experiment. While the model is trained with data from the complete trajectories of training set subjects, the accuracy is evaluated only in the last parts of the testing set follow-ups (namely, the first two years

of observation are not included). The model can be re-calibrated, for which there is a large set of tools available. Alternatively, the likelihood can be reformulated to better reflect the actual setting where the model is going to be used. For the present exposition, we only wanted to highlight the potential biases, miscalibrations and their likely sources.

The predicted probabilities are shown in the middle and bottom right panels of S2, with colours indicating the correct label. Again, the left panels show results from the cross-validation experiment and on the right are the simulations. Note how similar the left and right panels for this representation of probabilities are compared to the reliability diagrams shown on the top panels, highlighting the need for calibration measures.

S1.4 Decision curve analysis

Decision curve analysis is used to evaluate the utility of clinical predictions through net benefit plots ([Vickers and Elkin, 2006]). The risk-benefit ratio of a given medical action could be reduced to a threshold probability p_{th} such that, if p_{th} were the true probability of the subject having the disease, then the benefits of taking a given action (*e.g.*, recommend a treatment) if the patient is AD would compensate the harms of taking the action if the patient is not AD. The threshold probability is fixed for a given setting (*i.e.*, a disease and a chosen medical action) and take into account the expected medical improvement and side effects in patients, the harm to healthy people, the economical costs, and any other quantifiable factor. The decision curve tells us how better (or worse) is to treat all patients, treat none, or treat only the ones with a risk larger than the threshold ($P(AD) > p_{th}$) according to a prediction model ([Steyerberg, 2019]). Figure S3 shows the net benefit when the model is used to predict the probability of conversion to AD a certain number of years after the first visit. The intersection of the *treat all* curves with y or x axis reflects the increasing prevalence of the number of converters with time. The fact that red and purple curves (6y and 7y) show a high net benefit beyond 0.7 is related to the fact that there are fewer false positives.

S2 Supplementary Figures

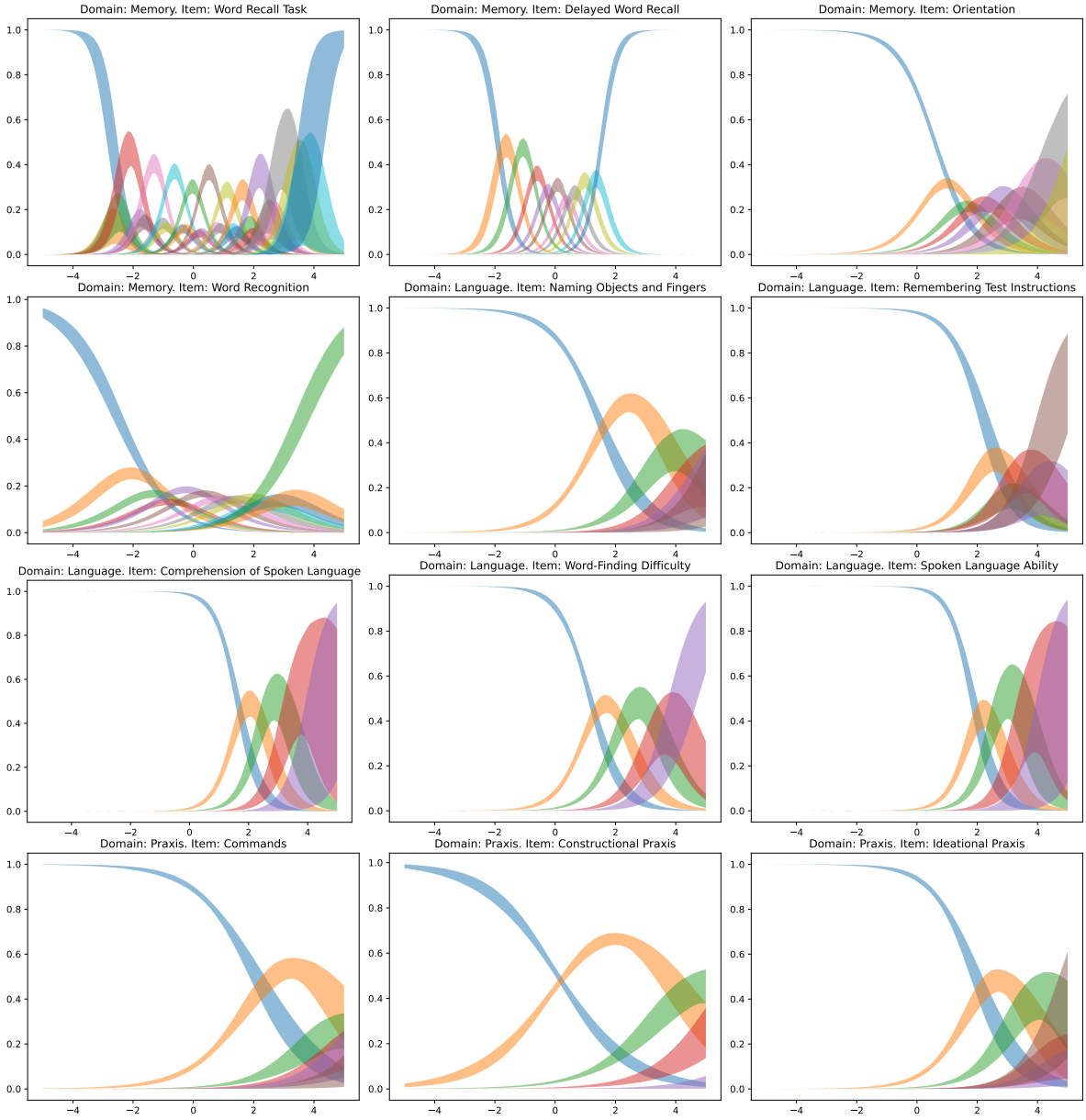

Figure S1: Posterior distribution of the item characteristic functions.

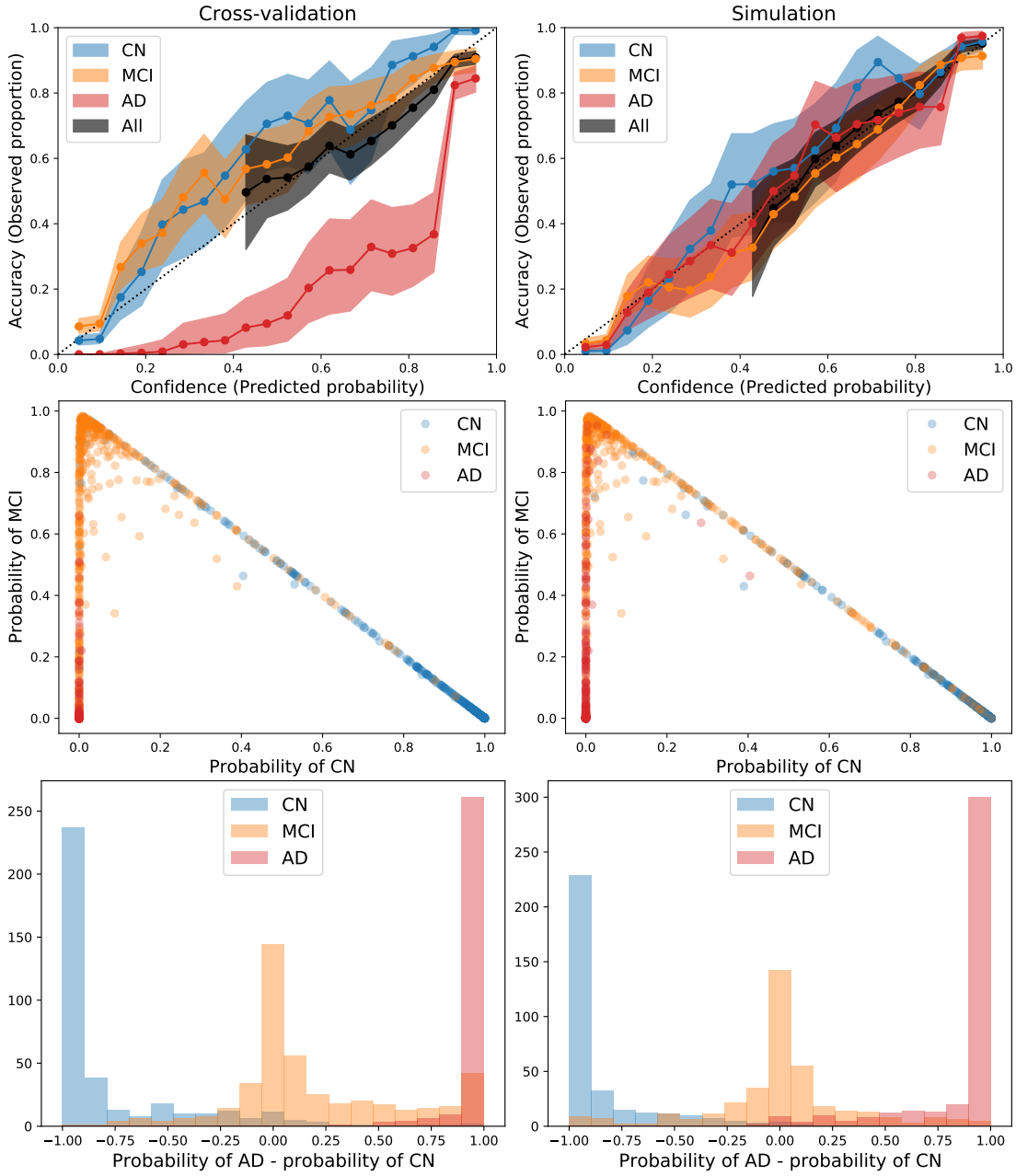

Figure S2: **Top**: Calibration estimation. Confidence- (black) and classwise-reliability diagrams (coloured), dots represent the mean and shaded areas represent the 95% CI. **Middle**: Simplex plot of the probability. Colours indicate the ground truth label. **Bottom**: Probability of AD minus probability of CN. **Left**: Results from cross-validation experiments. **Right**: Labels simulated from predicted probabilities for comparison.

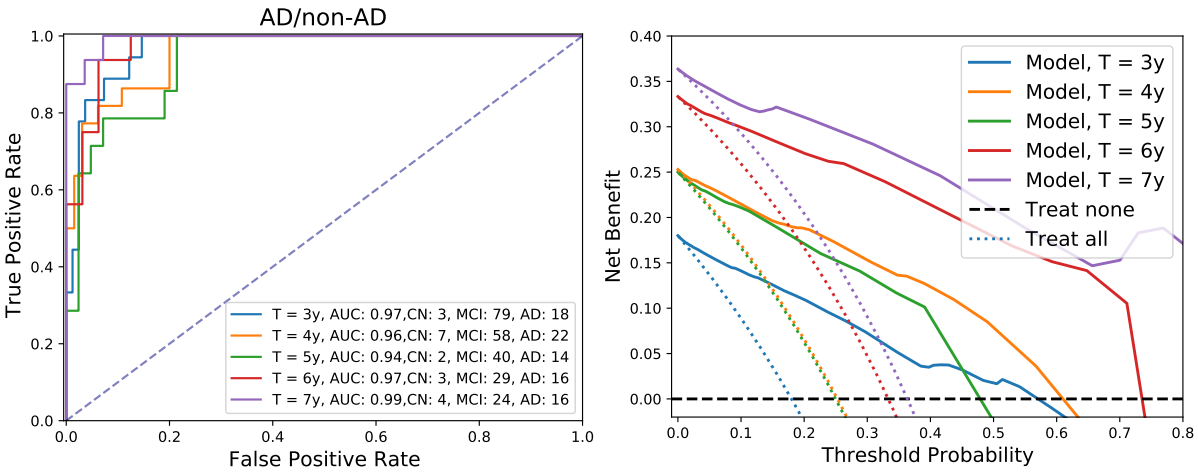

Figure S3: Prediction of conversion from MCI to AD at a given number of years after the first visit. There is only one diagnosis assessment per subject (*i.e.* diagnosis is assessed approximately every year in ADNI study), but the number of subjects is reduced for longer follow-ups. **Left**: ROC curves. The number of subjects in each category is indicated in the legend. **Right**: Decision curves. The "Treat all" strategy has a larger Net Benefit (NB) than "Treat none" for probability thresholds lower than the prevalence. The model NB slopes are much less steep than the "Treat all" strategy for lower thresholds, indicating a higher sensitivity for detecting AD converters, optimal for situations where False Positives have less cost associated than False Negatives. This is consistent with the asymmetries observed in ROC curves, which show that perfect sensitivity (TPR=1) is easier to achieve than perfect specificity (FPR=1).

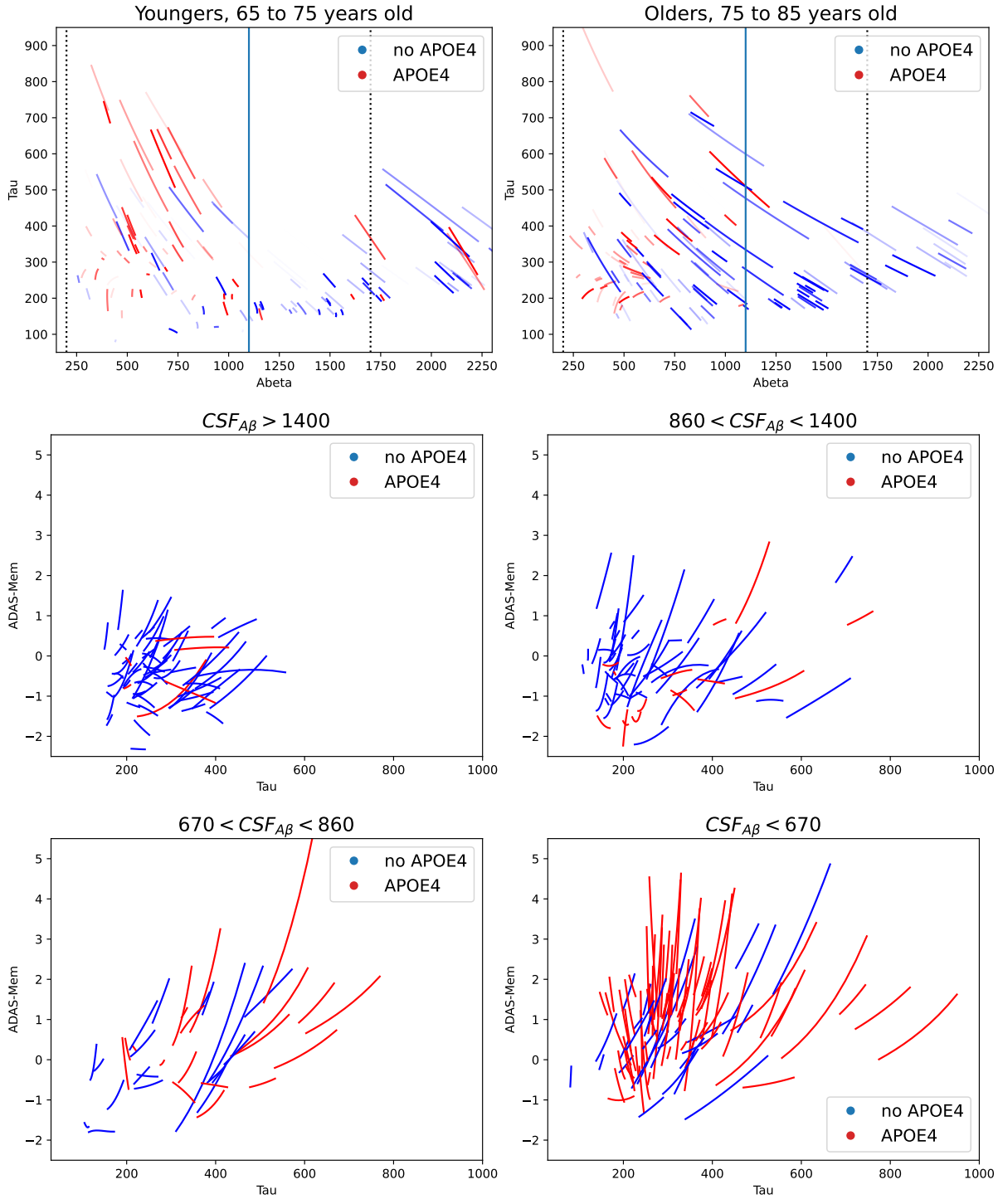

Figure S4: Estimated individual mean trajectories. Note that the curve lengths are not related to the biomarker rate of change, but the subject's follow-up duration.

S3 Model specification

```
data {
  int N;                          // Number of subjects
  int n;                          // Number of observations
  real t[n];                      // Observation times
  int m;                          // Numer of covariates
  matrix[n, m] X;                 // Covariates
  real AGE[N];                    // Subjects' age
  int<lower=0, upper=1> APOE4[N]; // One APOE4 allele minimum
  int<lower=1, upper=N> ID[n];    // Subj. ID for each observation
  int kTau;            // number of elements of Tau vector (1 or 2)

  // Number of valid observations for each type of data
  int nAB;
  int nTau;
  int nADAS;
  int nDX;

  // Indexes to locate time (t) or ID for each measurement
  int<lower=1, upper=n> idxAB[nAB];
  int<lower=1, upper=n> idxTau[nTau];
  int<lower=1, upper=n> idxDX[nDX];
  int<lower=1, upper=n> idxADAS[nADAS];

  // Observations
  matrix[nTau,kTau] Tau;
  vector[nAB] AB;
  int<lower=1, upper=3>   DX[nDX];
  // ABETA censored info
  real<lower=0, upper=1> ABmin;
  int<lower=0, upper=n> NABCens;
  int<lower=1, upper=n> idxABCens[NABCens];
  // ADAS
  int<lower=2> ADASCats[12];        // Number of categories per item
  int<lower=1, upper=ADASCats[ 1]>  Q1[nADAS]; // categorical data
  int<lower=1, upper=ADASCats[ 2]>  Q2[nADAS];
  int<lower=1, upper=ADASCats[ 3]>  Q3[nADAS];
  int<lower=1, upper=ADASCats[ 4]>  Q4[nADAS];
  int<lower=1, upper=ADASCats[ 5]>  Q5[nADAS];
  int<lower=1, upper=ADASCats[ 6]>  Q6[nADAS];
  int<lower=1, upper=ADASCats[ 7]>  Q7[nADAS];
  int<lower=1, upper=ADASCats[ 8]>  Q8[nADAS];
  int<lower=1, upper=ADASCats[ 9]>  Q9[nADAS];
  int<lower=1, upper=ADASCats[10]>  Q10[nADAS];
  int<lower=1, upper=ADASCats[11]>  Q11[nADAS];
  int<lower=1, upper=ADASCats[12]>  Q12[nADAS];
  int<lower=1>  D;                // Number of dimensions/factors
  int<lower=1, upper=D> it_d[12]; // Which factor corresponds to each item
}
transformed data {
  // Auxiliary variable with zeros
  int kCSF = kTau + 1;
  int K = kCSF + D; // Total number of features
  matrix[kCSF, D] zCSF = rep_matrix(0, kCSF, D);
}
parameters {
  vector[N] c0CSF[kCSF];        // Random effects (CSF)
```

```
  real<lower=0> sRECSF[kCSF];   // Variance of CSF RE
  real muRECSF[kCSF];           // Mean CSF

  // Velocity field parameters (main effect and interactions)
  matrix<lower=-0.2, upper=0.2>[kCSF,kCSF] vCSF;
  matrix<lower=-0.5, upper=0.5>[D, K] vADAS;
  row_vector[K] v0;
  matrix<lower=-0.1, upper=0.1>[kCSF, kCSF] wCSFAge;
  matrix<lower=-0.1, upper=0.1>[kCSF, kCSF] wCSFAPOE;
  matrix<lower=-0.2, upper=0.2>[D, K] wADASAge;
  matrix<lower=-0.2, upper=0.2>[D, K] wADASAPOE;

  // Observation noise variances
  real<lower=0> sTau[kTau];
  real<lower=0> sAB;

  // Prediction model parameters
  vector<lower=-20>[K + m] bDX;     // Predictor coefficients
  ordered[2] cDX;                   // Cut-off points

  // ADAS-Cog IRT model parameters
  vector<lower=0, upper=5>[12] alpha; // Slopes for each item
  ordered[ADASCats[ 1]-1] th1;    // Thresholds for each item
  ordered[ADASCats[ 2]-1] th2;
  ordered[ADASCats[ 3]-1] th3;
  ordered[ADASCats[ 4]-1] th4;
  ordered[ADASCats[ 5]-1] th5;
  ordered[ADASCats[ 6]-1] th6;
  ordered[ADASCats[ 7]-1] th7;
  ordered[ADASCats[ 8]-1] th8;
  ordered[ADASCats[ 9]-1] th9;
  ordered[ADASCats[10]-1] th10;
  ordered[ADASCats[11]-1] th11;
  ordered[ADASCats[12]-1] th12;
  row_vector[D] c0ADAS[N];            // Factor scores
  real<lower=0, upper=5> sigma_alpha;
  cholesky_factor_corr[D] L_corr_d;   // Cholesky correlation between factors
}
transformed parameters{
  // Auxiliary varaibles:
  //   velocity field parameters in the CSF+ADAS space
  matrix[K, K] v;
  matrix[K, K] wAge;
  matrix[K, K] wAPOE;
  row_vector[K] c0[N];
  v[1:kCSF, 1:kCSF] = vCSF;
  v[1:kCSF, (kCSF + 1):K] = zCSF;
  v[(kCSF + 1):K, :] = vADAS;
  wAge[1:kCSF, 1:kCSF] = wCSFAge;
  wAge[1:kCSF, (kCSF + 1):K] = zCSF;
  wAge[(kCSF + 1):K, :] = wADASAge;
  wAPOE[1:kCSF, 1:kCSF] = wCSFAPOE;
  wAPOE[1:kCSF, (kCSF + 1):K] = zCSF;
  wAPOE[(kCSF + 1):K, :] = wADASAPOE;

  for (k in 1:kCSF)
    c0[:, k] = to_array_1d(muRECSF[k] + c0CSF[k]*sRECSF[k]);
  c0[:, (1+kCSF):K] = c0ADAS;
```

```
}
model {
  matrix[n, K + m] mu;

  // Priors
  for (k in 1:kTau)
    sTau[k] ~ normal(0, 0.2);
  sAB ~ normal(0, 0.2);
  for (k2 in 1:kCSF) {
    bDX[k2] ~ normal(0, 100);
    for (k1 in 1:kCSF) {
      wCSFAge[k1, k2] ~ normal(0, 0.02);
      wCSFAPOE[k1, k2] ~ normal(0, 0.02);
      vCSF[k1, k2] ~ normal(0, 0.1);
    }
  }

  for (k in 1:K) {
    v0[k] ~ normal(0,0.1);
    for (d in 1:D) {
      wADASAge[d, k] ~ normal(0, 0.05);
      wADASAPOE[d, k] ~ normal(0, 0.05);
      vADAS[d, k] ~ normal(0, 0.1);
    }
  }

  for (k in 1:kCSF) {
    sRECSF[k] ~ std_normal();
    muRECSF[k] ~ std_normal();
  }
  for (i in 1:N)
    for (k in 1:kCSF)
      c0CSF[k, i] ~ std_normal();

  // priors: hierarchical (RE) for ADAS slopes
  sigma_alpha ~ std_normal();
  alpha ~ lognormal(0,sigma_alpha);
  // informative prio for ADAS latent variables
  L_corr_d ~ lkj_corr_cholesky(1);
  c0ADAS ~ multi_normal_cholesky(rep_vector(0, D), L_corr_d);

  // Trajectory
  for (i in 1:n) {
    matrix[K,1] c = scale_matrix_exp_multiply(t[i], (v + AGE[ID[i]]*wAge +
                                                     APOE4[ID[i]]*wAPOE),
                                      to_matrix( ( c0[ID[i]] - v0 ), K, 1));
    mu[i,1:K] = (to_row_vector(c) + v0);
  }
  mu[:,(K + 1):(K + m)] = X;

  // Likelihood
  // ABeta
  for (i in 1:nAB)
    AB[i] ~ normal(mu[idxAB[i],1], sAB);
  for (i in 1:NABCens) // Left censored ABeta
    target +=  normal_lcdf(ABmin | mu[idxABCens[i], 1], sAB);

  // Tau
```

```
  for (k in 1:kTau)
    for (i in 1:nTau)
      Tau[i,k] ~ normal(mu[idxTau[i],k+1], sTau[k]);

  //Diagnosis
  target += ordered_logistic_glm_lpmf(DX | mu[idxDX,:], bDX, cDX);

  //IRT ADAS-Cog
  Q1  ~ ordered_logistic(to_vector(mu[idxADAS, kCSF + it_d[1]] ) * alpha[1],  th1);
  Q2  ~ ordered_logistic(to_vector(mu[idxADAS, kCSF + it_d[2]] ) * alpha[2],  th2);
  Q3  ~ ordered_logistic(to_vector(mu[idxADAS, kCSF + it_d[3]] ) * alpha[3],  th3);
  Q4  ~ ordered_logistic(to_vector(mu[idxADAS, kCSF + it_d[4]] ) * alpha[4],  th4);
  Q5  ~ ordered_logistic(to_vector(mu[idxADAS, kCSF + it_d[5]] ) * alpha[5],  th5);
  Q6  ~ ordered_logistic(to_vector(mu[idxADAS, kCSF + it_d[6]] ) * alpha[6],  th6);
  Q7  ~ ordered_logistic(to_vector(mu[idxADAS, kCSF + it_d[7]] ) * alpha[7],  th7);
  Q8  ~ ordered_logistic(to_vector(mu[idxADAS, kCSF + it_d[8]] ) * alpha[8],  th8);
  Q9  ~ ordered_logistic(to_vector(mu[idxADAS, kCSF + it_d[9]] ) * alpha[9],  th9);
  Q10 ~ ordered_logistic(to_vector(mu[idxADAS, kCSF + it_d[10]]) * alpha[10], th10);
  Q11 ~ ordered_logistic(to_vector(mu[idxADAS, kCSF + it_d[11]]) * alpha[11], th11);
  Q12 ~ ordered_logistic(to_vector(mu[idxADAS, kCSF + it_d[12]]) * alpha[12], th12);
}
```

Figure S5: Stan code

References

[Gelman et al., 2013] Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian Data Analysis.* Chapman and Hall/CRC, third edition.

[Hodges, 2013] Hodges, J. S. (2013). *Richly Parameterized Linear Models: Additive, Time Series, and Spatial Models Using Random Effects.* Chapman and Hall/CRC, first edition.

[Kull et al., 2019] Kull, M., Nieto, M. P., Kängsepp, M., Silva Filho, T., Song, H., and Flach, P. (2019). Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with Dirichlet calibration. In *Advances in Neural Information Processing Systems*, pages 12295–12305.

[McCullagh, 1980] McCullagh, P. (1980). Regression models for ordinal data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 42(2):109–142.

[Samejima, 1968] Samejima, F. (1968). Estimation of latent ability using a response pattern of graded scores. *ETS Research Bulletin Series*, 1968(1):i–169.

[Steyerberg, 2019] Steyerberg, E. W. (2019). *Clinical Prediction Models. A Practical Approach to Development, Validation, and Updating.* Statistics for Biology and Health. Springer Nature Switzerland AG, Cham, Switzerland, second edition.

[Van Calster et al., 2016] Van Calster, B., Nieboer, D., Vergouwe, Y., De Cock, B., Pencina, M. J., and Steyerberg, E. W. (2016). A calibration hierarchy for risk models was defined: from utopia to empirical data. *Journal of Clinical Epidemiology*, 74:167–176.

[Van Hoorde et al., 2015] Van Hoorde, K., Van Huffel, S., Timmerman, D., Bourne, T., and Van Calster, B. (2015). A spline-based tool to assess and visualize the calibration of multiclass risk predictions. *Journal of Biomedical Informatics*, 54:283–293.

[Van Hoorde et al., 2014] Van Hoorde, K., Vergouwe, Y., Timmerman, D., Van Huffel, S., Steyerberg, E. W., and Van Calster, B. (2014). Assessing calibration of multinomial risk prediction models. *Statistics in Medicine*, 33(15):2585–2596.

[Verma et al., 2015] Verma, N., Beretvas, S. N., Pascual, B., Masdeu, J. C., Markey, M. K., and Initiative, T. A. D. N. (2015). New scoring methodology improves the sensitivity of the Alzheimer's Disease Assessment Scale-Cognitive subscale (ADAS-Cog) in clinical trials. *Alzheimer's Research & Therapy*, 7(1):64.

[Vickers and Elkin, 2006] Vickers, A. J. and Elkin, E. B. (2006). Decision curve analysis: A novel method for evaluating prediction models. *Medical Decision Making*, 26(6):565–574.