



## Functional Mixtures-of-Experts

Faïcel Chamroukhi, Nhat Thien Pham, van Hà Hoang, Geoffrey Mclachlan

### ► To cite this version:

Faïcel Chamroukhi, Nhat Thien Pham, van Hà Hoang, Geoffrey Mclachlan. Functional Mixtures-of-Experts. *Statistics and Computing*, 2024, 34 (3), pp.98. 10.1007/s11222-023-10379-0 . hal-04360919v1

**HAL Id: hal-04360919**

**<https://hal.science/hal-04360919v1>**

Submitted on 22 Dec 2023 (v1), last revised 22 Mar 2024 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Public Domain

# Functional Mixtures-of-Experts

Faïcel Chamroukhi<sup>1,2</sup>, Nhat Thien Pham<sup>2</sup>  
Van Hà Hoang<sup>3</sup>, Geoffrey J. McLachlan<sup>4</sup>

<sup>1</sup>IRT SystemX, 2 boulevard Thomas Gobert, 91120 Palaiseau, France

<sup>2</sup>Normandie Univ, UNICAEN, CNRS, LMNO, 14000 Caen, France

<sup>3</sup>University of Science, Vietnam National University, Ho Chi Minh City, Vietnam

<sup>4</sup>School of Mathematics and Physics, University of Queensland, Brisbane, Australia.

December 21, 2023

## Abstract

We consider the statistical analysis of heterogeneous data for prediction in situations where the observations include functions, typically time series. We extend the modeling with Mixtures-of-Experts (ME), as a framework of choice in modeling heterogeneity in data for prediction with vectorial observations, to this functional data analysis context. We first present a new family of ME models, named functional ME (FME) in which the predictors are potentially noisy observations, from entire functions. Furthermore, the data generating process of the predictor and the real response, is governed by a hidden discrete variable representing an unknown partition. Second, by imposing sparsity on derivatives of the underlying functional parameters via Lasso-like regularizations, we provide sparse and interpretable functional representations of the FME models called iFME. We develop dedicated expectation–maximization algorithms for Lasso-like (EM-Lasso) regularized maximum-likelihood parameter estimation strategies to fit the models. The proposed models and algorithms are studied in simulated scenarios and in applications to two real data sets, and the obtained results demonstrate their performance in accurately capturing complex nonlinear relationships and in clustering the heterogeneous regression data.

**Keywords:** Mixtures-of-Experts; Functional Data Analysis; EM algorithms; Maximum likelihood Estimation; Lasso regularization; Clustering

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Functional Mixtures-of-Experts (FME)</b>	<b>5</b>
2.1	ME with functional predictor and scalar response . . . . .	5
2.2	Smoothing representation of the functional experts . . . . .	6
2.3	Smoothing representation of the functional gating network . . . . .	7
2.4	The FME model conditional density . . . . .	8
2.5	Maximum likelihood estimation via the EM algorithm . . . . .	8
<b>3</b>	<b>Regularized maximum likelihood estimation</b>	<b>10</b>
3.1	$\ell_1$ -regularization and the EM-Lasso algorithm . . . . .	10
3.2	Interpretable Functional Mixture of Experts (iFME) . . . . .	13
3.2.1	Interpretable sparse regularization . . . . .	13
3.2.2	The iFME model . . . . .	15
3.3	Regularized MLE via the EM-iFME algorithm . . . . .	16
3.4	Non-linear regression and clustering with FME models . . . . .	18
3.5	Tuning parameters and model selection . . . . .	19
<b>4</b>	<b>Experimental study</b>	<b>19</b>
4.1	Evaluation criteria . . . . .	19
4.2	Simulation studies . . . . .	20
4.2.1	Simulation parameters and experimental protocol . . . . .	20
4.2.2	Some implementation details . . . . .	20
4.2.3	Simulation results . . . . .	21
4.3	Application to real data . . . . .	27
4.3.1	Canadian weather data . . . . .	27
4.3.2	Diffusion tensor imaging data for multiple sclerosis subjects . . . . .	31
<b>5</b>	<b>Conclusion and discussion</b>	<b>35</b>
<b>6</b>	<b>Acknowledgements</b>	<b>36</b>
<b>A</b>	<b>EM for the FME model</b>	<b>40</b>
<b>B</b>	<b>EM-Lasso for <math>\ell_1</math>-regularized MLE of the FME model</b>	<b>41</b>
B.1	Updating the gating network parameters . . . . .	42
B.2	Updating the experts network parameters . . . . .	43
<b>C</b>	<b>EM-iFME for updating iFME model parameters</b>	<b>45</b>
C.1	Updating the gating network parameters . . . . .	45
C.2	Updating the expert network parameters . . . . .	45
<b>D</b>	<b>Detailed Data generating protocol, simulation parameters and experimental protocol</b>	<b>47</b>
D.1	Data generating protocol . . . . .	47
D.2	Simulation parameters and experimental protocol . . . . .	47

# 1 Introduction

Mixture-of-experts (ME), introduced by (Jacobs et al., 1991), is a successful and flexible supervised learning architecture that allows one to efficiently represent complex non-linear relationships in observed pairs of heterogeneous data  $(\mathbf{X}, Y)$ . The ME model relies on the divide and conquer principle, so that the response  $Y$  is gathered from soft-association of several expert responses, each targeted to a homogeneous sub-population of the heterogeneous population, given the input covariates (predictors or features)  $\mathbf{X}$ . From the statistical modeling point of view, a ME model is an extension of the finite mixture model (McLachlan and Peel, 2000) which explores the unconditional (mixture) distribution of a given set of the features  $\mathbf{X}$ . It is thus more tailored to unsupervised learning than to supervised learning and it has the fully conditional mixture model of the form

$$\text{ME}(y|\mathbf{x}) = \sum_{k=1}^K G_k(\mathbf{x}) E_k(y|\mathbf{x}). \quad (1)$$

In model (1) the ME distribution of the response  $y$  given the predictors  $\mathbf{x}$  is a conditional mixture distribution with predictor-dependent mixing weights, referred to as gating functions,  $G_k(\mathbf{x})$ , and conditional mixture components, referred to as experts  $E_k(y|\mathbf{x})$ ,  $K$  being the number of experts.

Mixture of experts (ME) models thus allow one to better capture more complex relationships between  $y$  and  $\mathbf{x}$  in heterogeneous situations in non-linear regression  $y \in \mathbb{R}$ , classification  $y \in \{1, \dots, G\}$ , and in clustering the data by associating each expert component to a cluster. The richness of the class of ME models in terms of conditional density approximation capabilities has been recently demonstrated by proving denseness results (Nguyen et al., 2021a, 2019).

They have been investigated in their simple form, as well as in their hierarchical form (Jordan and Jacobs, 1994), for non-linear regression and model-based cluster and discriminant analyses and in different application domains. The inference in this case can be performed by maximum likelihood estimation (MLE) via the expectation–maximization (EM) algorithm (Jordan and Jacobs, 1994; McLachlan and Krishnan, 2008; Dempster et al., 1977) or, when  $p$  is possibly larger than the sample size  $n$ , by regularized MLE via dedicated EM-Lasso algorithms as in Khalili (2010); Montuelle et al. (2014); Chamroukhi et al. (2019); Chamroukhi and Huynh (2019); Huynh and Chamroukhi (2019); Nguyen et al. (2020) which include Lasso-like penalties (Tibshirani, 1996). For a more complete review of ME models, the reader is referred to Yuksel et al. (2012) and Nguyen and Chamroukhi (2018). More recent theoretical results about the ME estimation and model selection for different families of ME models, can be found in Nguyen et al. (2021b,c, 2020).

To the best of our knowledge, ME models have been exclusively studied in multivariate analysis when the inputs are vectors, i.e.  $\mathbf{X} \in \mathcal{X} = \mathbb{R}^p$ . However, in many problems, the predictors and/or the responses are observed from smooth functions. Indeed, in many situations, unlike in predictive and cluster analyses of multivariate and potentially high-dimensional heterogeneous data, which have been studied with the ME modeling in (1), the observed data may arise from continuously observed processes, e.g. time series. Thus, a multivariate (vectorial) analysis does not allow one to enough capture the inherent functional

structure of the data. In such situations, classical multivariate models are not adapted as they ignore the underlying intrinsic nature and structure of the data. Functional Data Analysis (FDA) (Ramsay and Silverman, 2005; Ferraty and Vieu, 2006) in which the individual data units are assumed to be functions, rather than vectors, offers an adapted framework to deal with continuously observed data, including in regression, classification and clustering. FDA considers the observed data as (discretized) values of smooth functions, rather than multivariate observations represented in the form of “simple” vectors.

The study of functional data has been considered in most of the statistical modeling and inference problems including regression, classification, clustering, functional graphical models (Qiao et al., 2019), among others. In regression, functional linear models have been introduced including penalized functional regression (Élodie Brunel et al., 2016; Goldsmith et al., 2011) and in particular the FLiRTI approach, a functional linear regression constructed upon interpretable regularization (James et al., 2009), and more generally generalized linear models with functional predictors (Müller et al., 2005; James, 2002), which cover functional logistic regression for classification. In classification, we can also cite functional linear discriminant analysis (James and Hastie, 2001), and, as a penalized model, Lasso-regularized functional logistic regression (Mousavi and Sørensen, 2017). To deal with heterogeneous functional data, the construction of mixture models with functional data analytic aspects have been introduced for model-based clustering (Liu and Yang (2009), Jacques and Preda (2014a)) including Lasso-regularized mixtures for functional data (Devijver, 2017; James and Sugar, 2003; Jacques and Preda, 2014b; Chamroukhi and Nguyen, 2019). The resulting functional mixture models are better able to handle functional data structures compared to standard multivariate mixtures.

The problem of clustering and prediction in presence of functional observations from heterogeneous populations, leading to complex distributions, is still however less investigated. In this paper, we investigate the framework of Mixtures-of-Experts (ME) models, as models of choice in modeling heterogeneity in data for prediction and clustering with vectorial observations, and extend it to the functional data framework. The main novelty of our paper is to get interpretable results for functional ME (FME). First, the ME framework is extended to a functional data setting to learn from functional predictors, and the statistical inference in the resulting setting (ie. of the FME model) looks for estimating a sparse and interpretable FME model parameters. The key technical challenges brought by this setting are addressed by considering sparsity on the derivatives of the underlying functional parameters of the FME model, thanks to Lasso-like regularizations, solved by a dedicated EM-Lasso algorithm. To the best of our knowledge, this is the first time the ME model is constructed upon functional predictors, and provides interpretable sparse estimates.

Firstly, we introduce in Section 2 a new family of ME models, referred to as FME, to relate a functional predictor to a scalar response, and develop a dedicated EM algorithm for the maximum-likelihood parameter estimation. Secondly, to deal with potential high-dimensional setting of the introduced FME model, we develop in Section 3 a Lasso-regularized approach, which consists of a penalized MLE estimation via an hybrid EM-Lasso algorithm, which integrates an optimized coordinate ascent procedure to efficiently implement the M-Step. Thirdly, we in particular present in Section 3.2 an extended FME model, which is constructed upon a sparse and highly-interpretable regularization of the functional expert and gating parameters. The resulting model, abbreviated as iFME, is fitted by

regularized MLE via an dedicated EM algorithm. The developed algorithms for the two introduced ME models are applied and evaluated in Section 4 on several simulated scenarios and on real data sets, in both clustering and non-linear regression.

## 2 Functional Mixtures-of-Experts (FME)

We wish to derive and fit new mixture-of-experts (ME) models in presence of functional predictors, and potentially, functional responses. In this paper, we first consider ME models with a functional predictor  $X(\cdot)$  and a real response  $Y$  where the pair arises from heterogeneous population composed of unknown  $K$  homogeneous sub-populations. To the best of our knowledge, this is the first time ME models are considered for functional data.

### 2.1 ME with functional predictor and scalar response

Let  $\{X_i(t), Y_i\}_{i=1}^n$  be a sample of  $n$  independently and identically distributed (i.i.d.) data pairs where  $Y_i$  is a real-valued response and  $X_i(t)$  is a functional predictor with  $t \in \mathcal{T} \subset \mathbb{R}$ , for example the time in time series. First, to model the conditional relationships between the continuous response  $Y$  and the functional predictor  $X(\cdot)$ , given an expert  $z$ , we formulate each expert component  $E_z(y|\mathbf{x})$  in (1) as a functional regression model (cf. Müller et al. (2005), James et al. (2009)). The resulting functional expert regression model for the  $i$ th observation takes the following stochastic representation

$$Y_i = \beta_{z_i,0} + \int_{\mathcal{T}} X_i(t) \beta_{z_i}(t) dt + \varepsilon_i, \quad i \in [n], \quad (2)$$

where  $\beta_{z_i,0}$  is an unknown constant intercept,  $\beta_{z_i}(t)$ ,  $t \in \mathcal{T}$  is the function of unknown coefficients of functional expert  $z_i$ , and  $\varepsilon_i \sim \mathcal{N}(0, \sigma_{z_i}^2)$  are independent Gaussian errors,  $z_i \in [K]$  being the unknown label of the expert generating the  $i$ th observation. The notation  $z_i \in [K]$  means  $z_i = 1, \dots, K$  which is used throughout this paper. In this context, the response  $Y$  is related to the entire trajectory of  $X(\cdot)$ . Let  $\boldsymbol{\beta} = \{\beta_{z,0}, \beta_z(t), t \in \mathcal{T}\}_{z=1}^K$  represents the set of unknown functional parameters for the experts network.

Now consider the modeling of the gating network in the proposed FME model. As in the context of ME for vectorial data, different choices are possible to model the gating network function, typically softmax-gated or Gaussian-gated ME (e.g. see Nguyen and Chamroukhi (2018), Xu et al. (1994), Chamroukhi et al. (2019)). A standard choice as in Jacobs et al. (1991) to model the gating network  $G_z(\mathbf{x})$  in (1) is to use the multinomial logistic (softmax) function as a distribution of the latent variable  $Z$ . In this functional data modeling context with  $K \geq 2$  experts, we use a multinomial logistic function as an extension of the functional logistic regression presented in Mousavi and Sørensen (2018) for linear classification. The resulting functional softmax gating network then takes the following form

$$\begin{aligned} \pi_z(X(t), t \in \mathcal{T}; \boldsymbol{\alpha}) &= \mathbb{P}(Z = z | X(t), t \in \mathcal{T}; \boldsymbol{\alpha}) \\ &= \frac{\exp\{\alpha_{z,0} + \int_{\mathcal{T}} X(t) \alpha_z(t) dt\}}{1 + \sum_{z'=1}^{K-1} \exp\{\alpha_{z',0} + \int_{\mathcal{T}} X(t) \alpha_{z'}(t) dt\}}, \end{aligned} \quad (3)$$

where  $\boldsymbol{\alpha} = \{\alpha_{z,0}, \alpha_z(t), t \in \mathcal{T}\}_{z=1}^K$  is the set of unknown constant intercept coefficients  $\alpha_{z,0}$  and functional parameters  $\alpha_z(t), t \in \mathcal{T}$  for each expert  $z \in [K]$ . Note that model (3) is equivalent to assuming that each expert  $z$  is related to the entire trajectory  $X(\cdot)$  via the following functional linear predictor for the gating network

$$\begin{aligned} h_z(X(t), t \in \mathcal{T}; \boldsymbol{\alpha}) &= \log \left\{ \frac{\pi_z(X(t), t \in \mathcal{T}; \boldsymbol{\alpha})}{\pi_K(X(t), t \in \mathcal{T}; \boldsymbol{\alpha})} \right\} \\ &= \alpha_{z,0} + \int_{\mathcal{T}} X(t) \alpha_z(t) dt. \end{aligned} \quad (4)$$

The objective is to estimate the functional parameters  $\boldsymbol{\alpha}$  and  $\boldsymbol{\beta}$  of the FME model defined by (2)-(3), from an observed sample. In this setting with functional predictors, this requires estimating a possibly infinite number of coefficients (as many as the number of temporal observations for the predictor). In order to reduce the complexity of the problem, the observed functional predictor can be projected onto a fixed number of basis functions so that we sufficiently capture enough the functional structure of the data, and sufficiently reduce enough the number of coefficients to estimate.

## 2.2 Smoothing representation of the functional experts

Here we consider the case of fixed design, that is, the covariates  $X_i(t)$  are non-random functions. We suppose that the  $X_i(\cdot)$ 's are measured with error at any given time  $t$ . Hence, instead of observing directly  $X_i(t)$ , one has a noisy version of it  $U_i(t)$ , defined as

$$U_i(t) = X_i(t) + \delta_i(t), \quad i \in [n],$$

where  $\delta_i(\cdot) \sim \mathcal{N}(0, \sigma_\delta^2)$  are measurement errors assumed to be independent of the  $X_i(\cdot)$ 's and the  $Y_i$ 's. Since the functional predictors  $X_i(t)$  are not directly observed, we first construct an approximation of  $X_i(t)$  from the noisy predictors  $U_i(t)$  by projecting the latter onto a set of continuous basis functions. Let  $\mathbf{b}_r(t) = [b_1(t), \dots, b_r(t)]^\top$  be a  $r$ -dimensional (B-spline, Fourier, Wavelet) basis, then with  $r$  sufficiently large  $X_i(t)$  can be represented as

$$X_i(t) = \sum_{j=1}^r x_{ij} b_j(t) = \mathbf{x}_i^\top \mathbf{b}_r(t), \quad (5)$$

where  $x_{ij} = \int_{\mathcal{T}} X_i(t) b_j(t) dt$  for  $j \in [r]$  and  $\mathbf{x}_i = (x_{i1}, \dots, x_{ir})^\top$ . Since  $X_i(t)$  is not observed, the representation coefficients  $x_{ij}$ 's are unknown. Hence we propose an unbiased estimator of  $x_{ij}$  defined as

$$\hat{x}_{ij} := \int_{\mathcal{T}} U_i(t) b_j(t) dt.$$

Thus, an estimate  $\hat{X}_i(t)$  of  $X_i(t)$  is given by

$$\hat{X}_i(t) = \hat{\mathbf{x}}_i^\top \mathbf{b}_r(t), \quad i \in [n], \quad (6)$$

with  $\hat{\mathbf{x}}_i = (\hat{x}_{i1}, \dots, \hat{x}_{ir})^\top$ .

Similarly, to represent the regression coefficient functions  $\beta_z(\cdot)$ , consider a  $p$ -dimensional basis  $\mathbf{b}_p(t) = [b_1(t), b_2(t), \dots, b_p(t)]^\top$ . Then the function  $\beta_z(t)$  can be represented as

$$\beta_z(t) = \boldsymbol{\eta}_z^\top \mathbf{b}_p(t) \quad (7)$$

where  $\boldsymbol{\eta}_z = (\eta_{z,1}, \eta_{z,2}, \dots, \eta_{z,p})^\top$  is the vector of unknown coefficients and the choice of  $p$  should ensure the tradeoff between smoothness of the functional predictor and complexity of the estimation problem. We select  $r \geq p$  to satisfy the identifiability constraint (see for instance Goldsmith et al. (2011), Ramsay and Silverman (2002)). Furthermore, rather than assuming a perfect fit of  $\beta_z(t)$  by  $\mathbf{b}_p(t)$  as in (7), we use for each Gaussian expert regressor  $z$ , the following error model as proposed by James et al. (2009) for functional linear regression

$$\beta_z(t) = \boldsymbol{\eta}_z^\top \mathbf{b}_p(t) + e(t)$$

where  $e(t)$  represents the approximation error of  $\beta_z(t)$  by the linear projection (7). As we choose  $p \gg n$ ,  $|e(t)|$  can be assumed to be small.

## 2.3 Smoothing representation of the functional gating network

Since here we are examining functional predictors, an appropriate representation has also to be given for the gating network (3) with functional parameters  $\{\alpha_z(t), t \in \mathcal{T}\}_{z=1}^K$ . Due to the infinite number of these parameters, we also represent the gating network by a finite set of basis functions similarly as for the experts network. For the representation of the functional predictors  $X_i(t)$ ,  $i \in [n]$ , we use  $\hat{X}_i(t)$  established in (6). The coefficients function  $\alpha_z(t)$  is represented similarly as for the  $\beta$  coefficients function of the experts network, by using a  $q$ -dimensional basis  $\mathbf{b}_q(t) = [b_1(t), b_2(t), \dots, b_q(t)]^\top$ ,  $q \leq r$ , via the projection

$$\alpha_z(t) = \boldsymbol{\zeta}_z^\top \mathbf{b}_q(t), \quad (8)$$

where  $\boldsymbol{\zeta}_z = (\zeta_{z1}, \zeta_{z2}, \dots, \zeta_{zq})^\top$  is the vector of softmax coefficients function. Note that here we use the same type of basis functions for both representations of  $\beta_z$  and  $\alpha_z$ , but one can use different types of bases if needed. Then, by using the representations (6) and (8) of  $X(t)$  and  $\alpha_z(t)$ , respectively, in the linear predictor  $h_z(\cdot)$  defined in (4) for  $i \in [n]$ , the latter is thus approximated as

$$h_z(U_i(t), t \in \mathcal{T}; \boldsymbol{\alpha}) = \alpha_{z_i,0} + \boldsymbol{\zeta}_{z_i}^\top \mathbf{r}_i, \quad (9)$$

where  $\mathbf{r}_i = [\int_{\mathcal{T}} \mathbf{b}_r(t) \mathbf{b}_q(t)^\top dt]^\top \hat{\mathbf{x}}_i$ . Thus, following its definition in (3), the functional softmax gating network is approximated as

$$\pi_k(\mathbf{r}_i; \boldsymbol{\xi}) = \frac{\exp\{\alpha_{k,0} + \boldsymbol{\zeta}_k^\top \mathbf{r}_i\}}{1 + \sum_{k'=1}^{K-1} \exp\{\alpha_{k',0} + \boldsymbol{\zeta}_{k'}^\top \mathbf{r}_i\}}, \quad (10)$$

where  $\boldsymbol{\xi} = ((\alpha_{1,0}, \boldsymbol{\zeta}_1^\top), \dots, (\alpha_{K-1,0}, \boldsymbol{\zeta}_{K-1}^\top))^\top \in \mathbb{R}^{(q+1)(K-1)}$  is the unknown parameter vector of the functional gating network to be estimated.

## 2.4 The FME model conditional density

We now have appropriate representations for the functional predictors, as well as for both the functional gating network and the functional experts network, involved in the construction of the functional ME (FME) model (2)-(3). Gathering (6) and (7), the stochastic representation (2) of the FME model can thus be defined as follows,

$$Y_i|u_i(\cdot) = \beta_{z_i,0} + \boldsymbol{\eta}_{z_i}^\top \mathbf{x}_i + \varepsilon_i^*, \quad i \in [n], \quad (11)$$

where  $\mathbf{x}_i = \left[ \int_{\mathcal{T}} \mathbf{b}_r(t) \mathbf{b}_p(t)^\top dt \right]^\top \hat{\mathbf{x}}_i$  and  $\varepsilon_i^* = \varepsilon_i + \hat{\mathbf{x}}_i^\top \int_{\mathcal{T}} \mathbf{b}_r(t) e(t) dt$ . We can see that the problem is now reduced at the standard version of the ME model. From this stochastic representation under the Gaussian assumption for the error variable  $\varepsilon_i$ , the conditional density of each approximated functional expert  $z_i = k$  is thus given by

$$f(y_i|u_i(\cdot), z_i = k; \boldsymbol{\theta}_k) = \phi(y_i; \beta_{k,0} + \boldsymbol{\eta}_k^\top \mathbf{x}_i, \sigma_k^2), \quad (12)$$

where  $\phi(\cdot; \mu, v)$  is the Gaussian probability density function with mean  $\mu$  and variance  $v$ ,  $\beta_{k,0} + \boldsymbol{\eta}_k^\top \mathbf{x}_i$  is the mean of the approximated functional regression expert,  $\sigma_k^2$  its variance, and  $\boldsymbol{\theta}_k = (\beta_{k,0}, \boldsymbol{\eta}_k^\top, \sigma_k^2)^\top \in \mathbb{R}^{p+2}$  the unknown parameter vector of expert density  $k$ ,  $k \in [K]$  to be estimated. Finally, combining (12) and (10) in the ME model (1), leads to the the following conditional density defining the FME model,

$$f(y_i|u_i(\cdot); \boldsymbol{\Psi}) = \sum_{k=1}^K \pi_k(\mathbf{r}_i; \boldsymbol{\xi}) \phi(y_i; \beta_{k,0} + \boldsymbol{\eta}_k^\top \mathbf{x}_i, \sigma_k^2), \quad (13)$$

where  $\boldsymbol{\Psi} = (\boldsymbol{\xi}^\top, \boldsymbol{\theta}_1^\top, \dots, \boldsymbol{\theta}_K^\top)^\top$  is the parameter vector of the model to be estimated.

## 2.5 Maximum likelihood estimation via the EM algorithm

The FME model (13) is now defined upon an adapted finite representation of the functional predictors, and its parameter estimation can then be performed given an observed data sample. We first consider the maximum likelihood estimation framework via the EM algorithm (Dempster et al., 1977; Jacobs et al., 1991) which has many desirable properties including stability and convergence guarantees (e.g. see McLachlan and Krishnan (2008) for more details). Note that here we use the term maximum likelihood estimation to not unduly clutter the clarity of the text, while as it will be specified later, we refer to the conditional maximum likelihood estimator.

In practice, the data are available in the form of discretized values of functions. The noisy functional predictors  $U_i(t)$  are usually observed at discrete sampling points  $t_{i1} < \dots < t_{im_i}$  with  $t_{ij} \in \mathcal{T}$  for  $j \in [m_i]$ . We suppose that  $U_i(t)$  is scaled such that  $0 \leq t \leq 1$  and divide the time period  $[0, 1]$  up into a fine grid of  $m_i$  points  $t_{i1}, \dots, t_{im_i}$ . Thus, in (11) we have  $\mathbf{x}_i = \left[ \sum_{j=1}^{m_i} \mathbf{b}_r(t_{ij}) \mathbf{b}_p(t_{ij})^\top \right]^\top \hat{\mathbf{x}}_i$ ,  $\mathbf{r}_i = \left[ \sum_{j=1}^{m_i} \mathbf{b}_r(t_{ij}) \mathbf{b}_q(t_{ij})^\top \right]^\top \hat{\mathbf{x}}_i$ , where  $\hat{\mathbf{x}}_{ij} = \sum_{j=1}^{m_i} U_i(t_{ij}) \mathbf{b}_j(t_{ij})$ . Note that if we choose  $p = q = r$ , then  $\mathbf{x}_i = \mathbf{r}_i = \hat{\mathbf{x}}_i$ . Let  $\mathcal{D} = \{(\mathbf{u}_1, y_1), \dots, (\mathbf{u}_n, y_n)\}$  be an i.i.d. sample of  $n$  observed data pairs where  $\mathbf{u}_i = (u_{i,1}, \dots, u_{i,m_i})$  is the observed functional predictor for the  $i$ th response  $y_i$ .

We use  $\mathcal{D}$  to estimate the parameter vector  $\Psi$  by iteratively maximizing the observed data log-likelihood,

$$\log L(\Psi) = \sum_{i=1}^n \log \sum_{k=1}^K \pi_k(\mathbf{r}_i; \xi) \phi(y_i; \beta_{k,0} + \boldsymbol{\eta}_k^\top \mathbf{x}_i, \sigma_k^2), \quad (14)$$

via the EM algorithm. As detailed in Appendix, the EM algorithm for the FME model is implemented as follows. After starting with an initial solution  $\Psi^{(0)}$ , it alternates, at each iteration  $s$ , between the two following steps, until convergence (when there is no longer a significant change in the values of the log-likelihood (14)).

**E-step.** Calculate the following conditional probability memberships  $\tau_{ik}^{(s)}$  (for all  $i \in [n]$ ), that the observed pair  $(u_i, y_i)$  originates from the  $k$ th expert, given the observed data and the current parameter estimate  $\Psi^{(s)}$ ,

$$\tau_{ik}^{(s)} = \mathbb{P}(Z_i = k | y_i, u_i(\cdot); \Psi^{(s)}) = \frac{\pi_k(\mathbf{r}_i; \xi^{(s)}) \phi(y_i; \beta_{k,0}^{(s)} + \mathbf{x}_i^\top \boldsymbol{\eta}_k^{(s)}, \sigma_k^{2(s)})}{f(y_i | u_i(\cdot); \Psi^{(s)})}. \quad (15)$$

**M-step.** Update the value of the parameter vector  $\Psi$  by maximizing the  $Q$ -function (43) with respect to  $\Psi$ . The maximization is performed by separate maximizations with respect to (w.r.t.) the gating network parameters  $\xi$  and, for each expert  $k$ , w.r.t. the expert network parameters  $\theta_k$ , for each of the  $K$  experts.

*Updating the gating network's parameters  $\xi$*  consists of maximizing w.r.t.  $\xi$  the part of (43) that is a function of  $\xi$ . Since we use a softmax-gated expert network in (10), this maximization problem consists of a weighted multinomial logistic problem for which there is no a closed-form solution. We then use a Newton-Raphson (NR) procedure, which iteratively maximizes (44) after starting from an initial parameter vector  $\xi^{(0)}$ , by updating, at each NR iteration  $t$ , the values of the parameter vector  $\xi$  according to the following updating formula:

$$\xi^{(t+1)} = \xi^{(t)} - \left[ H(\xi; \Psi^{(s)}) \right]_{\xi=\xi^{(t)}}^{-1} g(\xi; \Psi^{(s)}) \Big|_{\xi=\xi^{(t)}} \quad (16)$$

where  $H(\xi; \Psi^{(s)})$  and  $g(\xi; \Psi^{(s)})$  are, respectively, the Hessian matrix and the gradient vector of  $Q(\xi; \Psi^{(s)})$ , and are provided in Appendix A. At each NR iteration, the Hessian matrix and gradient vector are evaluated at the current value of  $\xi$ . We keep updating the gating network parameter  $\xi$  according to (16) until there is no significant change in  $Q(\xi; \Psi)$ . The maximization then provides  $\xi^{(s+1)}$  for the next EM iteration. Alternatively, instead of using the NR procedure, one can employ the Minorize-Maximization (MM) algorithm to update the parameter  $\xi$  of the gating network. In this approach, the Hessian matrix in (16) is approximated by a square matrix that is independent of  $\xi$ . This can offer an improved computational efficiency, although it comes at the cost of requiring a greater number of iterations. For further details, refer to Gormley and Murphy (2008).

*Updating the experts network parameters  $\theta_k$*  consists of solving  $K$  independent weighted regression problems where the weights are the conditional expert memberships  $\tau_{ik}^{(s)}$  given by

(15). The updating formulas for the regression parameters  $(\beta_{k,0}, \boldsymbol{\eta}_k)$  and the noise variances  $\sigma_k^2$  for each expert  $k$  are straightforward and correspond to weighted versions of those of standard Gaussian linear regression, i.e., weighted ordinary least squares. The updating rules for the experts network parameters are given by the following formulas:

$$\begin{aligned}\beta_{k,0}^{(s+1)} &= \frac{1}{n_k^{(s)}} \sum_{i=1}^n \tau_{ik}^{(s)} (y_i - \mathbf{x}_i^\top \boldsymbol{\eta}_k^{(s)}), \quad \boldsymbol{\eta}_k^{(s+1)} = \frac{1}{\sum_{i=1}^n \tau_{ik}^{(s)} \mathbf{x}_i \mathbf{x}_i^\top} \sum_{i=1}^n \tau_{ik}^{(s)} (y_i - \beta_{k,0}^{(s+1)}) \mathbf{x}_i, \\ \sigma_k^{2(s+1)} &= \frac{1}{n_k^{(s)}} \sum_{i=1}^n \tau_{ik}^{(s)} \left[ y_i - (\beta_{k,0}^{(s+1)} + \mathbf{x}_i^\top \boldsymbol{\eta}_k^{(s+1)}) \right]^2,\end{aligned}\tag{17}$$

where  $n_k^{(s)} = \sum_{i=1}^n \tau_{ik}^{(s)}$  represents the expected cardinal number of component  $k$ .

This EM algorithm, designed here for the FME that is constructed upon smoothing of the functional data, can be seen as a direct extension of the vectorized version the ME model. While it can hence be expected to provide accurate estimations as in the vector predictors setting, the number of parameters to estimate here in the case of the FME can still be high, for example when a big number of basis functions is used to have more accurate approximation of the functional predictors. In that case, it is better to regularize the maximum likelihood estimator in order to establish a compromise between the quality of fit and complexity.

### 3 Regularized maximum likelihood estimation

We rely on the LASSO (Tibshirani, 1996) as a successful procedure to encourage sparse models in high-dimensional linear regression based on an  $\ell_1$ -penalty, and include it in this mixture of experts modeling framework for functional data. The  $\ell_1$ -regularized ME models have demonstrated their performance from a computational point of view (Chamroukhi and Huynh, 2019; Huynh and Chamroukhi, 2019) and enjoy good theoretical properties of universal approximation and consistent model selection in the high-dimensional setting. (Nguyen et al., 2020, 2021d).

#### 3.1 $\ell_1$ -regularization and the EM-Lasso algorithm

We propose an  $\ell_1$ -regularization of the observed-data log-likelihood (18) to be maximized, along with coordinate ascent algorithms to deal with the high-dimensional setting when updating the parameters within the resulting EM-Lasso algorithm. The objective function in this case is given by the following  $\ell_1$ -regularized observed-data log-likelihood,

$$\mathcal{L}(\boldsymbol{\Psi}) = \log L(\boldsymbol{\Psi}) - \text{Pen}_{\lambda, \chi}(\boldsymbol{\Psi}),\tag{18}$$

where  $\log L(\boldsymbol{\Psi})$  is the observed-data log-likelihood of  $\boldsymbol{\Psi}$  defined by (14), and  $\text{Pen}_{\lambda, \chi}(\boldsymbol{\Psi})$  is a LASSO regularization term encouraging sparsity for the expert and the gating network parameters, defined by

$$\text{Pen}_{\lambda, \chi}(\boldsymbol{\Psi}) = \lambda \sum_{k=1}^K \sum_{j=1}^p |\eta_{k,j}| + \chi \sum_{k=1}^{K-1} \sum_{j=1}^q |\zeta_{k,j}|,\tag{19}$$

where  $\lambda$  and  $\chi$  are positive real values representing tuning parameters. The maximization of (18) cannot be performed in a closed form but again the EM algorithm can be adapted to iteratively solve the maximization problem. The resulting algorithm for the FME model, called EM-Lasso, takes the following form, as detailed in Appendix B. After starting with an initial solution  $\Psi^{(0)}$ , it alternates between the two following steps, until convergence, i.e., when there is no longer a significant change in the values of the  $\ell_1$ -penalized log-likelihood (18).

**E-step.** The E-Step in this EM-Lasso algorithm is unchanged compared to the previously presented EM algorithm, and only requires the computation of the conditional expert memberships  $\tau_{ik}^{(s)}$  according to (15).

**M-step.** In this regularized MLE context, the parameter vector  $\Psi$  is now updated by maximizing the regularized  $Q$ -function (46), i.e.,  $\Psi^{(s+1)} = \arg \max_{\Psi} \{Q(\Psi; \Psi^{(s)}) - \text{Pen}_{\lambda, \chi}(\Psi)\}$ . This is performed by separate maximizations w.r.t. the gating network parameters  $\xi$  and, for each expert  $k$ , w.r.t. the expert network parameters  $\theta_k$ ,  $k \in [K]$ .

*Updating the gating network parameters* at iteration  $s$  of the EM-Lasso algorithm consists of maximizing the following regularized  $Q$ -function w.r.t.  $\xi$ ,

$$\mathcal{Q}_{\chi}(\xi; \Psi^{(s)}) = Q(\xi; \Psi^{(s)}) - \chi \sum_{k=1}^{K-1} \|\zeta_k\|_1, \quad (20)$$

where  $Q(\xi; \Psi^{(s)}) = \sum_{i=1}^n \left[ \sum_{k=1}^{K-1} \tau_{ik}^{(s)} (\alpha_{k,0} + \zeta_k^{\top} \mathbf{r}_i) - \log \left( 1 + \sum_{k'=1}^{K-1} \exp\{\alpha_{k',0} + \zeta_{k'}^{\top} \mathbf{r}_i\} \right) \right]$ . One can see this is equivalent to solving a weighted regularized multinomial logistic regression problem for which  $\mathcal{Q}_{\chi}(\xi; \Psi^{(s)})$  is its penalized log-likelihood, the weights being the conditional probabilities  $\tau_{ik}^{(s)}$ . There is no closed-form solution for this kind of problem. We then use an iterative optimization algorithm to seek for a maximizer of  $\mathcal{Q}_{\chi}(\xi; \Psi^{(s)})$ , i.e., an update for the parameters of the gating network. To be effective when the number of parameters to estimate is high, we propose a coordinate ascent algorithm to update the softmax gating network coefficients in this regularized context.

*Coordinate ascent for updating the gating network.* The idea of the coordinate ascent algorithm (e.g. see [Hastie et al. \(2015\)](#), [Huynh and Chamroukhi \(2019\)](#)), implemented in our context at the  $s$ th EM-Lasso iteration to maximize  $\mathcal{Q}_{\chi}(\xi; \Psi^{(s)})$  at the M-Step, is as follows. The gating function parameter vectors  $\xi_k = (\alpha_{k,0}, \zeta_k^{\top})^{\top}$  as components of the whole gating network parameters  $\xi = (\xi_1^{\top}, \dots, \xi_{K-1}^{\top})^{\top}$ , are updated one at a time, while fixing the other gate's parameters to their previous estimates. Furthermore, to update a single gating parameter vector  $\xi_k$ , we only update its coefficients  $\xi_{kj}$  one at a time, while fixing the other coefficients to their previous values. More precisely, for each single gating function  $k$ , we partially approximate the smooth part of  $\mathcal{Q}_{\chi}(\xi; \Psi^{(s)})$  with respect to  $\xi_k$  at the current EM-Lasso estimate, say  $\xi^{(t)}$ , then optimize the resulting objective function (with respect to  $\xi_k$ ). This corresponds to solving penalized weighted least squares problems using coordinate ascent. Thus, this results into an inner loop, indexed by  $m$ , that cycles over the components

of  $\xi_k$  and updates them one by one, while the others are kept fixed to their previous values, i.e.,  $\xi_{kh}^{(m+1)} = \xi_{kh}^{(m)}$  for all  $h \neq j$ , until the objective function (48) is not significantly increased.

The obtained closed form updates for each coefficient  $\zeta_{kj}$ ,  $j \in [q]$ , and for the intercept  $\alpha_{k,0}$ , are as follows

$$\zeta_{kj}^{(m+1)} = \frac{\mathcal{S}_\chi\left(\sum_{i=1}^n w_{ik} r_{ij}(c_{ik} - \tilde{c}_{ikj}^{(m)})\right)}{\sum_{i=1}^n w_{ik} r_{ij}^2} \text{ for } j \in [q], \quad \alpha_{k,0}^{(m+1)} = \frac{\sum_{i=1}^n w_{ik}(c_{ik} - \mathbf{r}_i^\top \boldsymbol{\zeta}_k^{(m+1)})}{\sum_{i=1}^n w_{ik}},$$

where  $\tilde{c}_{ikj}^{(m)} = \alpha_{k,0}^{(m)} + \mathbf{r}_i^\top \boldsymbol{\zeta}_k^{(m)} - \zeta_{kj}^{(m)} r_{ij}$  is the fitted value excluding the contribution from the  $j$ th component of the  $i$ th vector  $\mathbf{r}_{ij}$  in the design matrix of the gating network and  $\mathcal{S}_\chi(\cdot)$  is a soft-thresholding operator defined by  $\mathcal{S}_\chi(u) = \text{sign}(u)(|u| - \chi)_+$  with  $(v)_+$  is a shorthand for  $\max\{v, 0\}$ . The values  $(\alpha_{k,0}^{(m+1)}, \boldsymbol{\zeta}_k^{(m+1)})$  obtained at convergence of the coordinate ascent inner loop for the  $k$ th gating function are taken as the fixed values of that gating function, in the procedure of updating the next parameter vector  $\xi_{k+1}$ . Finally, when all the gating functions have their values updated, i.e., after  $K - 1$  inner coordinate ascent loops, to avoid numerical instability, we perform a backtracking line search, before actually updating the gating network's parameters for the next EM-Lasso iteration. More precisely, the update is  $\xi^{(t+1)} = (1 - \nu)\xi^{(t)} + \nu\bar{\xi}^{(t)}$ , where  $\bar{\xi}^{(t)}$  is the output after  $K - 1$  inner loops and  $\nu$  is backtrackingly determined to ensure  $\mathcal{Q}_\chi(\xi^{(t+1)}; \Psi^{(s)}) \geq \mathcal{Q}_\chi(\xi^{(t)}; \Psi^{(s)})$ .

We keep cycling these updated iterates for the parameter vectors  $\xi_k$ , until convergence of the whole coordinate ascent (CA) procedure inside the M-Step, i.e., when the relative increase in the Lasso-regularized objective  $\mathcal{Q}_\chi(\xi; \Psi^{(s)})$  related to the softmax gating network is not significant, e.g., less than a small tolerance. Then, the next EM-Lasso iteration is calculated with the updated gating network's parameters  $\xi^{(s+1)} = (\tilde{\alpha}_{1,0}, \tilde{\zeta}_1^\top, \dots, \tilde{\alpha}_{K-1,0}, \tilde{\zeta}_{K-1}^\top)^\top$  where the values  $\tilde{\alpha}_{k,0}$  and  $\tilde{\zeta}_{kj}$  for all  $k \in [K - 1], j \in [q]$  are those obtained for the  $\alpha_{k,0}$ 's and the  $\zeta_{kj}$ 's at convergence of the CA algorithm.

*Updating the experts network parameters* The maximization step for updating the expert parameters  $\theta_k$  consists of maximizing the function  $\mathcal{Q}_\lambda(\theta_k; \Psi^{(s)})$  given by

$$\mathcal{Q}_\lambda(\theta_k; \Psi^{(s)}) = Q(\theta_k; \Psi^{(s)}) - \lambda \|\eta_k\|_1, \quad (21)$$

where  $Q(\theta_k; \Psi^{(s)}) = -\frac{1}{2\sigma_k^2} \sum_{i=1}^n \tau_{ik}^{(s)} (y_i - (\beta_{k,0} + \eta_k^\top \mathbf{x}_i))^2 - \frac{n}{2} \log(2\pi\sigma_k^2)$ . This corresponds to solving a weighted LASSO problem where the weights are the conditional experts memberships  $\tau_{ik}^{(s)}$ . We then solve it by an iterative optimization algorithm similarly to the previous case of updating the gating network parameters. As it can be seen in Appendix B.2, updating  $(\beta_{k,0}, \eta_k)$  according to (21) is obtained by coordinate ascent as follows. For each  $j \in [p]$ , the closed-form update for  $\eta_{kj}$  at the  $m$ th iteration of the coordinate ascent algorithm within the M-Step of EM-Lasso, is given by

$$\eta_{kj}^{(m+1)} = \frac{\mathcal{S}_{\lambda\sigma_k^2(s)}\left(\sum_{i=1}^n \tau_{ik}^{(s)} x_{ij}(y_i - \tilde{y}_{ij}^{(m)})\right)}{\sum_{i=1}^n \tau_{ik}^{(s)} x_{ij}^2}, \quad \beta_{k,0}^{(m+1)} = \frac{\sum_{i=1}^n \tau_{ik}^{(s)} (y_i - \mathbf{x}_i^\top \eta_k^{(m+1)})}{\sum_{i=1}^n \tau_{ik}^{(s)}},$$

in which  $\tilde{y}_{ij}^{(m)} = \beta_{k,0}^{(m)} + \mathbf{x}_i^\top \eta_k^{(m)} - \eta_{kj}^{(m)} x_{ij}$  is the fitted value excluding the contribution from  $x_{ij}$  and  $\mathcal{S}_{\lambda\sigma_k^2(s)}(\cdot)$  is the soft-thresholding operator. We keep updating the components of

$(\beta_{k,0}, \boldsymbol{\eta}_k)$  cyclically until no enough increase in objective function (21). Then, once  $(\beta_{k,0}, \boldsymbol{\eta}_k)$  are updated while fixing the variance  $\sigma_k^2$ , the latter is then updated straightforwardly as in the case of standard weighted Gaussian regression, and its update is given by

$$\sigma_k^{2(s+1)} = \frac{\sum_{i=1}^n \tau_{ik}^{(s)} \left( y_i - \beta_{k,0}^{(s+1)} - \mathbf{x}_i^\top \boldsymbol{\eta}_k^{(s+1)} \right)^2}{\sum_{i=1}^n \tau_{ik}^{(s)}},$$

where  $(\beta_{k,0}^{(s+1)}, \boldsymbol{\eta}_k^{(s+1)}) = (\tilde{\beta}_{k,0}, \tilde{\boldsymbol{\eta}}_k)$  is the solution obtained at convergence of the CA algorithm, which is taken as the update in the next EM-Lasso iteration.

This completes the parameter vector update  $\boldsymbol{\Psi}^{(s+1)} = (\boldsymbol{\xi}^{(s+1)}, \boldsymbol{\theta}_1^{(s+1)}, \dots, \boldsymbol{\theta}_K^{(s+1)})$  of the regularized FME model, where  $\boldsymbol{\xi}^{(s+1)}$  and  $\boldsymbol{\theta}_k^{(s+1)}, k \in [K]$ , are, respectively, the updates of the gating network parameters and the experts network parameters, calculated by the coordinate ascent algorithms.

The EM-Lasso algorithm provides an estimate of the FME parameters with sparsity constraints on the values of the parameter vectors  $\boldsymbol{\xi}$  and  $\boldsymbol{\theta}_k, k \in [K]$ . Actually, since here these parameter vectors do not relate directly the original functional inputs, to the output, assuming some of their values is zero is not necessarily justified, as there is no indeed any reason that these values are zero, nor easily interpretable, compared to the sparsity in vectorial generalized linear models, mixture of regressions and ME models.

From now on, we refer to FME and FME-Lasso, respectively, the FME model fitted by EM algorithm in Section 2.5 and the regularized FME model fitted by EM-Lasso algorithm, in Section 3.1. In the following Section, we introduce a regularization that is interpretable and encourages sparsity in the FME model.

## 3.2 Interpretable Functional Mixture of Experts (iFME)

In this section, we provide a sparse and, especially highly-interpretable fit, for the coefficient functions  $\{\beta_k(t), t \in \mathcal{T}\}$  and  $\{\alpha_k(t), t \in \mathcal{T}\}$  representing each of the  $K$  functional experts and gating network. We call our approach Interpretable Functional Mixture of Experts (iFME). The presented iFME allows us to control the expert and gating parameter functions while still providing performance as with the standard FME model presented previously.

### 3.2.1 Interpretable sparse regularization

We rely on the methodology of Functional Linear Regression That's Interpretable (FLiRTI) presented in James et al. (2009). The idea of the FLiRTI methodology is as follows. We use variable selection with sparsity assumption on appropriate chosen derivatives of the coefficient function, say  $\beta_{z_i}(t)$  here, in the case of the functional expert network, to produce a highly interpretable estimate for the coefficient functions  $\beta_{z_i}(t)$ . For instance,  $\beta_{z_i}^{(0)}(t) = 0$  implies that the predictor  $X_i(t)$  has no effect on the response  $Y_i$  at  $t$ ,  $\beta_{z_i}^{(1)}(t) = 0$  means that  $\beta_{z_i}(t)$  is constant in  $t$ ,  $\beta_{z_i}^{(2)}(t) = 0$  shows that  $\beta_{z_i}(t)$  is linear in  $t$ , etc. Assuming sparsity in higher-order derivatives of  $\beta_{z_i}(t)$ , for instance when  $d = 3$  or  $d = 4$ , will however give us a less easily interpretable fit. Hence, for example, if one believes that the expert parameter function  $\beta_{z_i}(t)$  is exactly zero over a certain region and exactly linear over other region of

$t$ , then it is necessary to estimate  $\beta_{z_i}(t)$  such that  $\beta_{z_i}^{(0)}(t) = 0$  and  $\beta_{z_i}^{(2)}(t) = 0$  over those regions, respectively. In this situation, we need to model  $\beta_{z_i}(t)$  assuming that its zeroth and second derivatives are sparse. However, with the EM-Lasso method derived above via the Lasso regularization, there is no actually any reason that we could obtain those desired properties, which may result in an estimate for  $\beta_{z_i}(t)$  that is rarely exactly zeros (and/or linear), and making the sparsity and coefficient curves hard to interpret. The same situation may occur with the gating parameter functions. To handle this, we describe in what follows the construction of our iFME model that produces flexible-shape and highly-interpretable estimates for the expert and gating coefficient functions, by simultaneously constraining any two of their derivatives to be sparse.

We start by selecting a  $p$ -dimensional basis  $\mathbf{b}_p(t)$  and a  $q$ -dimensional basis  $\mathbf{b}_q(t)$  for approximating the experts and gating networks, respectively. For the expert network, if we divide the time domain into a grid of  $p$  evenly spaced points, and let  $D^d$  be the  $d$ th finite difference operator defined recursively as

$$\begin{aligned} D\mathbf{b}_p(t_j) &= p[\mathbf{b}_p(t_j) - \mathbf{b}_p(t_{j-1})], \\ D^2\mathbf{b}_p(t_j) &= D[D\mathbf{b}_p(t_j)] = p^2[\mathbf{b}_p(t_j) - 2\mathbf{b}_p(t_{j-1}) + \mathbf{b}_p(t_{j-2})], \\ &\vdots \\ D^d\mathbf{b}_p(t_j) &= D[D^{d-1}\mathbf{b}_p(t_j)], \end{aligned}$$

then  $D^d\mathbf{b}_p(t_j)$  provides an approximation for  $\mathbf{b}_p^{(d)}(t_j) = [b_1^{(d)}(t_j), \dots, b_p^{(d)}(t_j)]^\top$ ,  $j \in [p]$ . Let

$$\mathbf{A}_p = \left[ \underbrace{D^{d_1}\mathbf{b}_p(t_1), D^{d_1}\mathbf{b}_p(t_2), \dots, D^{d_1}\mathbf{b}_p(t_p)}_{\mathbf{A}_p^{[d_1]}}, \underbrace{D^{d_2}\mathbf{b}_p(t_1), D^{d_2}\mathbf{b}_p(t_2), \dots, D^{d_2}\mathbf{b}_p(t_p)}_{\mathbf{A}_p^{[d_2]}} \right]^\top$$

be the matrix of approximate  $d_1$ th and  $d_2$ th derivative of the basis  $\mathbf{b}_p(t)$ . We denote  $\mathbf{A}_p^{[d_1]}$  the first  $p$  rows of  $\mathbf{A}_p$  and  $\mathbf{A}_p^{[d_2]}$  the remainder, i.e.,  $\mathbf{A}_p = [\mathbf{A}_p^{[d_1]}, \mathbf{A}_p^{[d_2]}]^\top$ . One can see such matrix  $\mathbf{A}_p$  is in  $\mathbb{R}^{2p \times p}$  and  $\mathbf{A}_p^{[d_1]}$  is a square invertible matrix.

Now, if we consider the following representation for the expert network coefficient function

$$\gamma_{z_i} = \mathbf{A}_p \boldsymbol{\eta}_{z_i} \quad (22)$$

with  $\gamma_{z_i} = (\gamma_{z_i}^{[d_1]^\top}, \gamma_{z_i}^{[d_2]^\top})^\top$ , where  $\gamma_{z_i}^{[d_1]} = (\gamma_{1,z_i}^{[d_1]}, \dots, \gamma_{p,z_i}^{[d_1]})^\top$ ,  $\gamma_{z_i}^{[d_2]} = (\gamma_{1,z_i}^{[d_2]}, \dots, \gamma_{p,z_i}^{[d_2]})^\top$ , and  $\boldsymbol{\eta}_{z_i}$  defined as in relation to  $\beta_{z_i}(t)$  as in (7), then one can see that  $\gamma_{z_i}^{[d_1]}$  provides an approximation to  $\beta_{z_i}^{(d_1)}(t)$ , i.e. the  $d_1$ th derivative of  $\beta_{z_i}(t)$ . Respectively,  $\gamma_{z_i}^{[d_2]}$  provides an approximation to  $\beta_{z_i}^{(d_2)}(t)$ , the  $d_2$ th derivative of  $\beta_{z_i}(t)$ . Therefore, enforcing sparsity in  $\gamma_{z_i}$  will constrain  $\beta_{z_i}^{(d_1)}(t)$  and  $\beta_{z_i}^{(d_2)}(t)$  to be zero at most time points.

Similarly, let  $\mathbf{A}_q = [\mathbf{A}_q^{[d_1]}, \mathbf{A}_q^{[d_2]}]^\top \in \mathbb{R}^{2q \times q}$  be the corresponding matrix defined for the gating network. If we consider the following representation for the gating network coefficient function

$$\omega_{z_i} = \mathbf{A}_q \boldsymbol{\zeta}_{z_i} \quad (23)$$

with  $\omega_{z_i} = (\omega_{z_i}^{[d_1]^\top}, \omega_{z_i}^{[d_2]^\top})^\top$ , where  $\omega_{z_i}^{[d_1]} = (\omega_{1,z_i}^{[d_1]}, \dots, \omega_{q,z_i}^{[d_1]})^\top$ ,  $\omega_{z_i}^{[d_2]} = (\omega_{1,z_i}^{[d_2]}, \dots, \omega_{q,z_i}^{[d_2]})^\top$ , and  $\boldsymbol{\zeta}_{z_i}$  defined as in relation to  $\alpha_{z_i}(t)$  as in (8), then we can derive the same interpretation as above for the coefficient vector  $\omega_{z_i}$  and the gating parameter function  $\alpha_{z_i}(t)$ .

Using simple calculations, the representations (22) and (23) imply the following relations that subsequently used in the iFME model:

$$\boldsymbol{\eta}_{z_i} = (\mathbf{A}_p^{[d_1]})^{-1} \boldsymbol{\gamma}_{z_i}^{[d_1]}, \quad \boldsymbol{\gamma}_{z_i}^{[d_2]} = \mathbf{A}_p^{[d_2]} (\mathbf{A}_p^{[d_1]})^{-1} \boldsymbol{\gamma}_{z_i}^{[d_1]}, \quad (24)$$

and

$$\boldsymbol{\zeta}_{z_i} = (\mathbf{A}_q^{[d_1]})^{-1} \boldsymbol{\omega}_{z_i}^{[d_1]}, \quad \boldsymbol{\omega}_{z_i}^{[d_2]} = \mathbf{A}_q^{[d_2]} (\mathbf{A}_q^{[d_1]})^{-1} \boldsymbol{\omega}_{z_i}^{[d_1]}, \quad (25)$$

respectively.

In fact, one can construct  $\mathbf{A}_p$  and  $\mathbf{A}_q$  with only one derivative. Then the constraints involved to the  $d_2$ th derivative will be eliminated making the estimation easier, but also limiting the flexibility in the shapes of the functions. That is why in this construction and in our experimental studies,  $\mathbf{A}_p$  and  $\mathbf{A}_q$  are constructed with multiple derivatives in order to produce curves of  $\beta_{z_i}(\cdot)$  and  $\alpha_{z_i}(\cdot)$  with such many desired properties.

### 3.2.2 The iFME model

Combining the stochastic representation of the FME model in (11) for the experts model, the linear predictor definition in (9), and the relations (24)-(25), we obtain the following iFME model construction

$$Y_i | u_i(\cdot) = \beta_{z_i,0} + \boldsymbol{\gamma}_{z_i}^{[d_1]\top} \mathbf{v}_i + \varepsilon_i^*, \quad (26)$$

$$h_{z_i}(u_i(\cdot); \boldsymbol{\alpha}) = \alpha_{z_i,0} + \boldsymbol{\omega}_{z_i}^{[d_1]\top} \mathbf{s}_i, \quad (27)$$

subject to

$$\boldsymbol{\gamma}_{z_i}^{[d_2]} = \mathbf{A}_p^{[d_2]} \mathbf{A}_p^{[d_1]-1} \boldsymbol{\gamma}_{z_i}^{[d_1]} \quad \text{and} \quad \boldsymbol{\omega}_{z_i}^{[d_2]} = \mathbf{A}_q^{[d_2]} \mathbf{A}_q^{[d_1]-1} \boldsymbol{\omega}_{z_i}^{[d_1]}, \quad (28)$$

where  $\mathbf{v}_i = (\mathbf{A}_p^{[d_1]-1})^\top \mathbf{x}_i$  is the new design vector for the experts and  $\mathbf{s}_i = (\mathbf{A}_q^{[d_1]-1})^\top \mathbf{r}_i$  the new one for the gating network. Hence, from (26) and (27), the conditional density of each expert and the gating network are now written as

$$f(y_i | u_i(\cdot); \boldsymbol{\psi}_k) = \phi(y_i; \beta_{k,0} + \boldsymbol{\gamma}_k^{[d_1]\top} \mathbf{v}_i, \sigma_k^2) \quad (29)$$

and

$$\pi_k(\mathbf{s}_i; \mathbf{w}) = \frac{\exp \{ \alpha_{k,0} + \boldsymbol{\omega}_k^{[d_1]\top} \mathbf{s}_i \}}{1 + \sum_{k'=1}^{K-1} \exp \{ \alpha_{k',0} + \boldsymbol{\omega}_{k'}^{[d_1]\top} \mathbf{s}_i \}}, \quad (30)$$

where  $\boldsymbol{\psi}_k = (\beta_{k,0}, \boldsymbol{\gamma}_k^{[d_1]\top}, \sigma_k^2)^\top$  is the unknown parameter vector of expert component density  $k$  and  $\mathbf{w} = (\alpha_{1,0}, \boldsymbol{\omega}_1^{[d_1]\top}, \dots, \alpha_{K-1,0}, \boldsymbol{\omega}_{K-1}^{[d_1]\top})^\top$ , with  $(\alpha_{K,0}, \boldsymbol{\omega}_K^{[d_1]\top})^\top$  a null vector, is the unknown parameter vector of the gating network. Finally, gathering (29) and (30) as for (13), the iFME model density is now given by

$$f(y_i | u_i(\cdot); \boldsymbol{\Psi}) = \sum_{k=1}^K \pi_k(\mathbf{s}_i; \mathbf{w}) \phi(y_i; \beta_{k,0} + \boldsymbol{\gamma}_k^{[d_1]\top} \mathbf{v}_i, \sigma_k^2), \quad (31)$$

where  $\boldsymbol{\Psi} = (\mathbf{w}^\top, \boldsymbol{\psi}_1^\top, \dots, \boldsymbol{\psi}_K^\top)^\top$  is the parameter vector of the model to be estimated. Thus, the iFME model constructed upon the parameter vectors  $\boldsymbol{\gamma}_k$ 's and  $\boldsymbol{\omega}_k$ 's, for which the sparsity is assumed to obtain interpretable estimates.

### 3.3 Regularized MLE via the EM-iFME algorithm

In order to fit the iFME model and to maintain the sparsity in  $\gamma_k$  and  $\omega_k$ , the following EM-iFME algorithm is then developed to maximize the penalized log-likelihood function

$$\mathcal{L}(\Psi) = \sum_{i=1}^n \log f(y_i|u_i(\cdot); \Psi) + \text{Pen}_{\lambda, \chi}(\Psi) \quad (32)$$

with the conditional iFME density  $f(y_i|u_i(\cdot); \Psi)$  is defined in (31) and the new sparse and interpretable regularization term is given by

$$\text{Pen}_{\lambda, \chi}(\Psi) = \lambda \sum_{k=1}^K \left( \|\gamma_k^{[d_1]}\|_1 + \rho \|\gamma_k^{[d_2]}\|_1 \right) + \chi \sum_{k=1}^{K-1} \left( \|\omega_k^{[d_1]}\|_1 + \varrho \|\omega_k^{[d_2]}\|_1 \right), \quad (33)$$

where  $\rho$  and  $\varrho$  are, respectively, the weights associated to the  $d_2$ th derivative of the expert and the gating parameter function. The appearance of the weighting parameters  $\rho$  and  $\varrho$ , besides the usual regularization parameters  $\lambda$  and  $\chi$ , is motivated by the fact that one may wish to place a greater emphasis on sparsity in the  $d_2$ th derivative than in the  $d_1$ th derivative of the parameter functions, or vice versa (we will see the usage of them in the subsequent section of experimental study). In practice, the selection of  $\rho$  and  $\varrho$  is more about whether they are greater than or less than one (i.e., the emphasis on  $d_2$ th) rather than select an exact value.

Note that, firstly, unlike the previous FME-Lasso, in iFME model the regularization operates on the functional derivative  $\gamma_k$ 's rather than the functional coefficients  $\eta_k$ 's for the experts, and on the functional derivatives  $\omega_k$ 's rather than the functional coefficients  $\zeta_k$ 's for the gating network. Secondly, maximizing the penalized log-likelihood function (32) with penalization in (33) in iFME model must be coupled with the constraints (28). Follows are the two steps of the proposed EM-iFME algorithm.

**E-Step.** The E-Step for the new iFME model calculates for each observation the conditional probability memberships of each expert  $k$

$$\tau_{ik}^{(s)} = \mathbb{P}(Z_i = k | y_i, u_i(\cdot); \Psi^{(s)}) = \frac{\pi_k(\mathbf{s}_i; \mathbf{w}^{(s)}) \phi(y_i; \beta_{k,0}^{(s)} + \mathbf{v}_i^\top \gamma_k^{[d_1]^{(s)}}, \sigma_k^{2(s)})}{f(y_i|u_i(\cdot); \Psi^{(s)})}, \quad (34)$$

where  $f(y_i|u_i(\cdot); \Psi^{(s)})$  is now calculated according to the iFME density given by (31).

**M-Step.** The maximization is performed by separate maximizations w.r.t. the gating network parameters  $\mathbf{w}$  and the experts network parameters  $\psi_k$ 's.

*Updating the gating network parameters.* The maximization step for updating the gating network parameters  $\mathbf{w} = \left( \alpha_{1,0}, \omega_1^{[d_1]^\top}, \dots, \alpha_{K-1,0}, \omega_{K-1}^{[d_1]^\top} \right)^\top$  consists of maximizing the function  $\mathcal{Q}_\chi(\mathbf{w}; \Psi^{(s)})$  given by

$$\mathcal{Q}_\chi(\mathbf{w}; \Psi^{(s)}) = Q(\mathbf{w}; \Psi^{(s)}) - \chi \sum_{k=1}^{K-1} \left( \|\omega_k^{[d_1]}\|_1 + \varrho \|\omega_k^{[d_2]}\|_1 \right), \quad (35)$$

subject to

$$\boldsymbol{\omega}_k^{[d_2]} = \mathbf{A}_q^{[d_2]} \mathbf{A}_q^{[d_1]-1} \boldsymbol{\omega}_k^{[d_1]}, \quad \forall k \in [K-1], \quad (36)$$

where  $Q(\mathbf{w}; \boldsymbol{\Psi}^{(s)}) = \sum_{i=1}^n \left[ \sum_{k=1}^{K-1} \tau_{ik}^{(s)} \left( \alpha_{k,0} + \boldsymbol{\omega}_k^{[d_1]\top} \mathbf{s}_i \right) - \log \left( 1 + \sum_{k'=1}^{K-1} \exp \{ \alpha_{k',0} + \boldsymbol{\omega}_{k'}^{[d_1]\top} \mathbf{s}_i \} \right) \right]$ . This is a constrained version of the weighted regularized multinomial logistic regression problem, where the weights are the conditional probabilities  $\tau_{ik}^{(s)}$ .

To solve it, in the same spirit as when updating the gating network in the previous EM-Lasso algorithm, we use an outer loop that cycles over the gating function parameters to update them one by one. However, to update a single gating function parameter  $\mathbf{w}_k = (\alpha_{k,0}, \boldsymbol{\omega}_k^{[d_1]\top})^\top$ ,  $k \in [K-1]$ , since the maximization problem (35) is now coupled with an additional constraint (36), rather than using a coordinate ascent algorithm as in EM-Lasso, we use the following alternative approach. For each single gating network  $k$ , using a Taylor series expansion, we partially approximate the smooth part of  $\mathcal{Q}_\chi(\mathbf{w}; \boldsymbol{\Psi}^{(s)})$  defined in (35) w.r.t.  $\mathbf{w}_k$  at the current estimate  $\mathbf{w}^{(t)}$ , then maximize the resulting objective function (w.r.t.  $\mathbf{w}_k$ ), subject to the corresponding constraint (w.r.t.  $k$ ) in (36). It corresponds to solving the following penalized weighted least squares problem with constraints,

$$\begin{aligned} \max_{(\alpha_{k,0}, \boldsymbol{\omega}_k^{[d_1]}, \boldsymbol{\omega}_k^{[d_2]})} & -\frac{1}{2} \sum_{i=1}^n w_{ik} \left( c_{ik} - \alpha_{k,0} - \mathbf{s}_i^\top \boldsymbol{\omega}_k^{[d_1]} \right)^2 - \chi \left( \|\boldsymbol{\omega}_k^{[d_1]}\|_1 + \varrho \|\boldsymbol{\omega}_k^{[d_2]}\|_1 \right) \\ \text{subject to} & \quad \boldsymbol{\omega}_k^{[d_2]} = \mathbf{A}_q^{[d_2]} \mathbf{A}_q^{[d_1]-1} \boldsymbol{\omega}_k^{[d_1]}, \end{aligned} \quad (37)$$

where  $w_{ik} = \pi_k(\mathbf{w}^{(t)}; \mathbf{s}_i) (1 - \pi_k(\mathbf{w}^{(t)}; \mathbf{s}_i))$  are the weights and  $c_{ik} = \alpha_{k,0}^{(t)} + \mathbf{s}_i^\top \boldsymbol{\omega}_k^{[d_1](t)} + \frac{\tau_{ik}^{(s)} - \pi_k(\mathbf{w}^{(t)}; \mathbf{s}_i)}{w_{ik}}$  are the working responses computed given the current estimate  $\mathbf{w}^{(t)}$ . This problem can be solved by quadratic programming (see, e.g., Gaines et al. (2018)) or by using the Dantzig selector (Candes et al., 2007), which we opt to use in our experimental studies. The details of using Dantzig selector to solve problem (37) are given in Appendix C.1.

Therefore, if  $(\tilde{\alpha}_{k,0}, \tilde{\boldsymbol{\omega}}_k^{[d_1]}, \tilde{\boldsymbol{\omega}}_k^{[d_2]})$  is an optimal solution to problem (37), then  $\tilde{\mathbf{w}}_k = (\tilde{\alpha}_{k,0}, \tilde{\boldsymbol{\omega}}_k^{[d_1]\top})^\top$  is taken as an update for the gating parameter vector  $\mathbf{w}_k$ . We keep cycling over  $k \in [K-1]$  until there is no significant increase in the regularized  $Q$ -function (35).

*Updating the experts network parameters.* The maximization step for updating the expert parameter vector  $\boldsymbol{\psi}_k = (\beta_{k,0}, \boldsymbol{\gamma}_k^{[d_1]\top}, \sigma_k^2)^\top$  consists of solving the following problem:

$$\begin{aligned} \max_{(\beta_{k,0}, \boldsymbol{\gamma}_k^{[d_1]}, \boldsymbol{\gamma}_k^{[d_2]}, \sigma_k^2)} & \sum_{i=1}^n \tau_{ik}^{(s)} \log \phi \left( y_i; \beta_{k,0} + \mathbf{v}_i^\top \boldsymbol{\gamma}_k^{[d_1]}, \sigma_k^2 \right) - \lambda \left( \|\boldsymbol{\gamma}_k^{[d_1]}\|_1 + \rho \|\boldsymbol{\gamma}_k^{[d_2]}\|_1 \right) \\ \text{subject to} & \quad \boldsymbol{\gamma}_k^{[d_2]} = \mathbf{A}_p^{[d_2]} \mathbf{A}_p^{[d_1]-1} \boldsymbol{\gamma}_k^{[d_1]}. \end{aligned} \quad (38)$$

As in the previous EM-Lasso algorithm, we first fix  $\sigma_k^2$  to its previous estimate and perform the update for  $(\beta_{k,0}, \boldsymbol{\gamma}_k^{[d_1]})$ , which corresponds to solving a penalized weighted least squares problem with constraints. This is performed by using the Dantzig selector, in the same manner as previously for solving problem (37). The corresponding technical details can be found in Appendix C.2.

Once the  $(\beta_{k,0}, \gamma_k^{[d_1]})$  are updated, the straightforward update for the variance  $\sigma_k^2$  is given by the standard estimate of a weighted Gaussian regression. More specifically, let  $(\tilde{\beta}_{k,0}, \tilde{\gamma}_k^{[d_1]}, \tilde{\gamma}_k^{[d_2]})$  be the solution to the problem (38) (with  $\sigma_k^2$  fixed to  $\sigma_k^{2(s)}$ ), the updates for expert parameter vector  $\boldsymbol{\psi}_k$  are given by

$$\begin{aligned} (\beta_{k,0}^{(s+1)}, \gamma_k^{[d_1](s+1)}) &= (\tilde{\beta}_{k,0}, \tilde{\gamma}_k^{[d_1]}), \\ \sigma_k^{2(s+1)} &= \frac{\sum_{i=1}^n \tau_{ik}^{(s)} \left( y_i - \beta_{k,0}^{(s+1)} - \mathbf{v}_i^\top \gamma_k^{[d_1](s+1)} \right)^2}{\sum_{i=1}^n \tau_{ik}^{(s)}}. \end{aligned}$$

Thus, at the end of the M-Step, we obtain a parameter vector update  $\boldsymbol{\Psi}^{(s+1)} = (\mathbf{w}^{(s+1)}, \boldsymbol{\psi}_1^{(s+1)}, \dots, \boldsymbol{\psi}_K^{(s+1)})$  for the next EM iteration, where  $\mathbf{w}^{(s+1)}$  and  $\boldsymbol{\psi}_k^{(s+1)}$ ,  $k \in [K]$ , are, respectively, the updates of the gating network parameters and the experts network parameters, calculated by the two procedures described above. Alternating the E-Step and M-Step until convergence, i.e., when there is no longer a significant change in the values of the penalized log-likelihood (32), leads to a penalized maximum likelihood estimate  $\hat{\boldsymbol{\Psi}}$  for  $\boldsymbol{\Psi}$ .

*Estimating the coefficient functions:* Finally, once the parameter vector of iFME model has been estimated, the coefficient functions of the gating network  $\alpha_k(t)$ ,  $k \in [K-1]$  and the ones of the experts network  $\beta_k(t)$ ,  $k \in [K]$ , can be reconstructed by the following formulas

$$\begin{aligned} \hat{\alpha}_k(t) &= \mathbf{b}_q(t)^\top \mathbf{A}_q^{-1} \hat{\boldsymbol{\omega}}_k^{[d_1]}, \\ \hat{\beta}_k(t) &= \mathbf{b}_p(t)^\top \mathbf{A}_p^{-1} \hat{\boldsymbol{\gamma}}_k^{[d_1]}, \end{aligned} \tag{39}$$

where  $\hat{\boldsymbol{\omega}}_k^{[d_1]}$  and  $\hat{\boldsymbol{\gamma}}_k^{[d_1]}$  are respectively the regularized maximum likelihood estimates for  $\boldsymbol{\omega}_k^{[d_1]}$  and  $\boldsymbol{\gamma}_k^{[d_1]}$ .

### 3.4 Non-linear regression and clustering with FME models

Once the model parameters have been estimated, one can then construct an estimate of the unknown functional non-linear regression function, as well as a clustering of the data into groups of similar pairs of observations.

For the aim of functional non-linear regression, the unknown non-linear regression function with functional predictors is given by the following conditional expectation  $\hat{y}|u(\cdot) = \mathbb{E}[Y|U(\cdot); \hat{\boldsymbol{\Psi}}]$ , which is defined by  $\hat{y}_i|u_i(\cdot) = \sum_{k=1}^K \pi_k(\mathbf{r}_i; \hat{\boldsymbol{\xi}})(\hat{\beta}_{k,0} + \hat{\boldsymbol{\eta}}_k^\top \mathbf{x}_i)$  for the FME model (13), and by  $\hat{y}_i|u_i(\cdot) = \sum_{k=1}^K \pi_k(\mathbf{s}_i; \hat{\mathbf{w}})(\hat{\beta}_{k,0} + \gamma_k^{[d_1]\top} \mathbf{v}_i)$  for the iFME model (31).

For clustering, a soft partition of the data into  $K$  clusters, represented by the estimated posterior probabilities  $\hat{\tau}_{ik} = \mathbb{P}(Z_i = k|u_i, y_i; \hat{\boldsymbol{\Psi}})$ , is obtained. A hard partition can also be computed according to the Bayes' optimal allocation rule. That is, by assigning each curve to the component having the highest estimated posterior probability  $\tau_{ik}$ , defined by (15) for FME or by (34) for the iFME model, using the MLE  $\hat{\boldsymbol{\Psi}}$  of  $\boldsymbol{\Psi}$ :

$$\hat{z}_i = \arg \max_{1 \leq k \leq K} \tau_{ik}(\hat{\boldsymbol{\Psi}}), \quad i \in [n],$$

where  $\hat{z}_i$  denotes the estimated cluster label for the  $i$ th curve.

### 3.5 Tuning parameters and model selection

In practice, appropriate values of the tuning parameters should be chosen. In using FME, this cover the selection of  $K$ , the number of experts, and  $r$ ,  $p$ , and  $q$ , the dimensions of B-spline bases used to approximate, respectively, the predictors, the experts, and the gating network functions, although they can be chosen to be equal. For the FME-Lasso, additionally the  $\ell_1$  penalty constants  $\chi$  and  $\lambda$  in (19) should be chosen. For the iFME model, the tuning parameters include also  $d_1$ ,  $d_2$ , the two derivatives related to the sparsity constraints, and  $\rho$ ,  $\varrho$ , the weights associated to the  $d_2$ th derivative of the expert and gating functions (see (33)).

The selection of the tuning parameters can be performed by a cross-validation procedure with a grid search scheme to select the best combination. An alternative is to use the well-known BIC criterion (Schwarz, 1978) or, in our paper, its extension, called modified BIC (Städler et al., 2010) defined as

$$\text{mBIC} = L(\hat{\Psi}) - \text{df}(\hat{\Psi}) \frac{\log n}{2}, \quad (40)$$

where  $\hat{\Psi}$  is the obtained log-likelihood estimator (for the FME model) or penalized log-likelihood estimator (for the FME-Lasso and iFME models), and the number of degrees of freedom  $\text{df}(\hat{\Psi})$  is the effective number of parameters of the model, given by

$$\text{df}(\hat{\Psi}) = \begin{cases} \text{df}(\zeta) + (K - 1) + \text{df}(\eta) + K + K & \text{for the FME and FME-Lasso models,} \\ \text{df}(\omega) + (K - 1) + \text{df}(\gamma) + K + K & \text{for the iFME model,} \end{cases}$$

in which the quantities  $\text{df}(\zeta)$ ,  $\text{df}(\eta)$ ,  $\text{df}(\omega)$  and  $\text{df}(\gamma)$  are, respectively, the counts for non-zero free coefficients in the vectors  $\zeta$ ,  $\eta$ ,  $\omega$ , and  $\gamma$ . Note that, because of the constraints (28) for the iFME model, free coefficients in  $\omega$  and  $\gamma$  consist of only the part related to the  $d_1$  derivative. That is,  $\text{df}(\zeta) = \sum_{k=1}^{K-1} \sum_{j=1}^q \mathbf{1}_{\{\zeta_{kj} \neq 0\}}$ ,  $\text{df}(\eta) = \sum_{k=1}^K \sum_{j=1}^p \mathbf{1}_{\{\eta_{kj} \neq 0\}}$ ,  $\text{df}(\omega) = \sum_{k=1}^{K-1} \sum_{j=1}^q \mathbf{1}_{\{\omega_{kj}^{[d_1]} \neq 0\}}$  and  $\text{df}(\gamma) = \sum_{k=1}^K \sum_{j=1}^p \mathbf{1}_{\{\gamma_{kj}^{[d_1]} \neq 0\}}$ . We apply both the BIC and the modified BIC in our experimental study.

## 4 Experimental study

We study the performances of the FME, FME-Lasso, and iFME models in regression and clustering problems by considering simulated scenarios and real data with functional predictors and scalar responses. The interests of this study consist of the prediction performance as well as the estimation of the functional parameters, i.e., the expert and gating functions in the ME model, along with the clustering partition of the considered heterogeneous data.

### 4.1 Evaluation criteria

We will use the following criteria, for where applicable, to assess and compare the performances of the models and the related algorithms. For regression evaluation, we use the *relative prediction error* (RPE) and the *correlation* (Corr) index as standard criteria to evaluate the prediction performance, i.e., to quantify the relationship between the true and the

predicted values of the scalar outputs. The RPE is defined by  $\text{RPE} = \sum_{i=1}^n (y_i - \hat{y}_i)^2 / \sum_{i=1}^n y_i^2$ , where  $y_i$  and  $\hat{y}_i$  are, respectively, the true and the predicted response of the  $i$ th observation in the testing set. For clustering evaluation, we use the *Rand index* (RI), the *adjusted Rand index* (ARI), and the *clustering error* (ClusErr), as standard criteria to evaluate the clustering performance, i.e., to quantify how similar the testing observations are presented in the true partition compared to the predicted partition. To evaluate the parameters estimation performance, we compute the *mean squared error* (MSE) between the true and the estimated functional parameters. The MSE between a true function  $g(\cdot)$  and its estimate  $\hat{g}(\cdot)$  is defined by

$$\text{MSE}(\hat{g}(\cdot)) = \frac{1}{m} \sum_{j=1}^m (g(t_j) - \hat{g}(t_j))^2, \quad (41)$$

with  $m$  being the number of time points taken into account. The function  $g(\cdot)$  here corresponds to an expert function  $\beta(\cdot)$ , or a gating function  $\alpha(\cdot)$ . The parameter functions are reconstructed from the model parameters using (7) and (8) for both FME and FME-Lasso models, and using (39) for iFME model.

The values of these criteria are averaged over  $N$  Monte Carlo runs ( $N = 100$  for simulation, for the real data, see the corresponding section). Note that, the average over  $N$  sample replicates of  $\text{MSE}(\hat{g}(\cdot))$  is equivalent to the usual Mean Integrated Squared Error (MISE) defined by  $\text{MISE}(\hat{g}(\cdot)) = \mathbb{E} \left[ \int_{\mathcal{T}} (\hat{g}(t) - g(t))^2 dt \right]$ , where the integral here is calculated numerically by a Riemann sum over the grid  $t_1, \dots, t_m$ .

## 4.2 Simulation studies

### 4.2.1 Simulation parameters and experimental protocol

We generate data from a  $K$ -component functional mixture of Gaussian experts model that relates a scalar response  $y \in \mathbb{R}$  to a univariate functional predictor  $X(t), t \in \mathcal{T}$  defined on a domain  $\mathcal{T} \subset \mathbb{R}$ . The data generation protocol is detailed in Appendix D.1. The parameters that were used in this data generating process (53) are described and provided in the simulation parameters and experimental protocol section in Appendix D.2. We study different simulation scenarios (sample sizes  $n$ , number of observations per time-series input  $m$ , noise levels  $\sigma_\delta^2, \dots$ ) summarised in Section D.2 and Table. 9.

Figure 1 displays, for each scenario, 10 randomly taken predictors colored according to their corresponding clusters. For each run, the concerned dataset is randomly split into a training set and testing set of equal size, the model parameters are estimated using training set, with the tuning parameters selected by maximizing the modified BIC (40). The evaluation criteria are computed on testing set and reported for each model accordingly.

### 4.2.2 Some implementation details

For all scenarios, for all datasets, we implemented the three proposed models with 10 EM runs and with a tolerance of  $10^{-6}$ . For the iFME model, in principle, for each parameter function the two derivatives  $d_1$  and  $d_2$  to be penalized, and the weights for the latter (i.e.,  $\rho$  and  $\varrho$ ) can

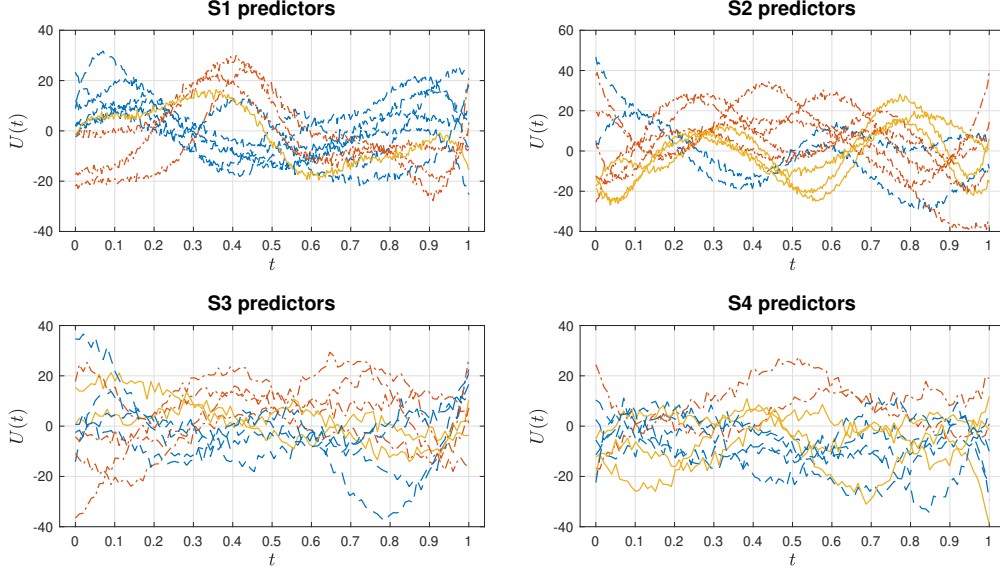


Figure 1: 10 randomly taken predictors in scenarios  $S1$  (large  $m$  and small  $\sigma_\delta$ ),  $S2$  (small  $m$  and small  $\sigma_\delta$ ),  $S3$  (large  $m$  and large  $\sigma_\delta$ ), and  $S4$  (small  $m$  and large  $\sigma_\delta$ ). Here, the noisy predictors  $U_i(\cdot)$  are colored (blue, red, yellow) according to their true cluster labels  $Z_i \in \{1, 2, 3\}$ .

be seen as tuning parameters. However, such an implementation could be computationally expensive in this simulation study with 400 datasets in total and 10 EM runs for each dataset. Therefore, we opted to fix  $d_1$  and  $d_2$  for all implementations ( $d_1 = 0$  and  $d_2 = 3$  for both expert and gating networks.) and left  $\rho$  and  $\varrho$  to be selected in some sets of targeted values. The choices of the targeted values were made by the following straightforward arguments. Since  $\beta_1(\cdot)$  and  $\beta_2(\cdot)$  have zero-valued regions, the weight for their zeroth derivative in penalization term should be large, equivalently, the weight for the third derivative should be small, so  $\rho$  is selected in a set of small values:  $\rho \in \{10^{-2}, 10^{-3}, 10^{-4}\}$ . On the other hand, for  $\alpha_1(t)$  and  $\alpha_2(t)$ , we select  $\varrho \in \{10, 10^2, 10^3\}$  as we should emphasize sparsity in their third derivative.

### 4.2.3 Simulation results

**Clustering and prediction performances.** We report in Table 1 the results of regression and clustering tasks on simulated datasets in the four considered scenarios. The mean and standard error of the relative predictions error (RPE) and correlation (Corr) summarize the regression performance, while the mean and standard error of the Rand index (RI), adjusted Rand index (ARI) and clustering error (ClusErr) summarize the clustering performance.

As we can see from Table 1, all the models have very good performances on both regression and clustering tasks, with high correlation, RI, ARI, and small RPE and clustering error. The iFME appears to slightly have a better performance than the others in all scenarios. The low standard errors confirm the stability of the algorithms.

Figure 2 shows the clustering results obtained by the models with highest values of the modified BIC criterion, on a dataset selected in scenario  $S1$ . Here, we plotted the responses

	RPE	Corr	ARI	ClusErr
$S1$ ( $m = 300, \sigma_\delta^2 = 1$ )				
FME	.1552 <sub>(.1282)</sub>	.9188 <sub>(.0602)</sub>	.7852 <sub>(.0934)</sub>	.0899 <sub>(.0434)</sub>
FME-Lasso	.1390 <sub>(.1074)</sub>	.9224 <sub>(.0603)</sub>	.7670 <sub>(.1436)</sub>	.1112 <sub>(.0891)</sub>
iFME	<b>.1334</b> <sub>(.0977)</sub>	<b>.9287</b> <sub>(.0504)</sub>	<b>.7997</b> <sub>(.0796)</sub>	<b>.0792</b> <sub>(.0340)</sub>
$S2$ ( $m = 100, \sigma_\delta^2 = 1$ )				
FME	.1600 <sub>(.1698)</sub>	.9164 <sub>(.0712)</sub>	<b>.7966</b> <sub>(.0881)</sub>	.0852 <sub>(.0427)</sub>
FME-Lasso	.1656 <sub>(.1316)</sub>	.9071 <sub>(.0733)</sub>	.7455 <sub>(.1520)</sub>	.1236 <sub>(.0961)</sub>
iFME	<b>.1540</b> <sub>(.0950)</sub>	<b>.9192</b> <sub>(.0476)</sub>	.7955 <sub>(.0822)</sub>	<b>.0820</b> <sub>(.0341)</sub>
$S3$ ( $m = 300, \sigma_\delta^2 = 4$ )				
FME	.1724 <sub>(.1807)</sub>	.9110 <sub>(.0778)</sub>	.7798 <sub>(.0838)</sub>	.0918 <sub>(.0415)</sub>
FME-Lasso	.1457 <sub>(.1504)</sub>	.9197 <sub>(.0785)</sub>	.7629 <sub>(.1524)</sub>	.1115 <sub>(.0928)</sub>
iFME	<b>.1432</b> <sub>(.0860)</sub>	<b>.9228</b> <sub>(.0448)</sub>	<b>.7987</b> <sub>(.0899)</sub>	<b>.0783</b> <sub>(.0361)</sub>
$S4$ ( $m = 100, \sigma_\delta^2 = 4$ )				
FME	.2251 <sub>(.4462)</sub>	.9048 <sub>(.0822)</sub>	<b>.7816</b> <sub>(.1040)</sub>	.0927 <sub>(.0497)</sub>
FME-Lasso	<b>.1432</b> <sub>(.1229)</sub>	<b>.9188</b> <sub>(.0683)</sub>	.7526 <sub>(.1654)</sub>	.1215 <sub>(.1055)</sub>
iFME	.1639 <sub>(.0978)</sub>	.9125 <sub>(.0521)</sub>	.7798 <sub>(.0848)</sub>	<b>.0864</b> <sub>(.0325)</sub>

Table 1: Evaluation criteria of FME, FME-Lasso and iFME models for test data in scenarios  $S1, \dots, S4$ . The reported values are the averages of 100 Monte Carlo runs with standard errors in parentheses. The bold values correspond to the best solution.

against the predictors at two specific time points:  $t_1 = 0$  and  $t_{50} = 0.5$ . The highly accurate predictions (in both regression and clustering) can be seen visually through Figure 2. This figure also shows that it is difficult to cluster these data according to a few number of time observations, for example in  $\mathbb{R}^2$ , according to  $\{(U_i(t_0), y_i)\}_{i=1}^n$  or  $\{(U_i(t_{50}), y_i)\}_{i=1}^n$ , which suggests using functional approaches.

*Comparison with functional regression mixtures (FMR):* Finally, we compare our proposed models with the functional mixture regression (FMR) model proposed in Yao et al. (2010). In their approach, the functional predictors are first projected onto its eigenspace, then the obtained new coordinates are fed to the standard mixture regression model to estimate the weights and the coefficients of the  $\hat{\beta}(\cdot)$  in that eigenspace. They performed functional principal component analysis (FPCA) to obtain estimates for the eigenfunctions and the principal component scores (which serve as predictors). The number of relevant FPCA components are chosen automatically for each dataset by selecting the minimum number of components that explain 90% of the total variation of the predictors. It is noticed that, in their simulation studies, the authors computed the scalar responses by using conditional prediction, i.e., the true  $y_i$  were used to determine which cluster the observation belongs to. Then the predicted  $\hat{y}_i$  is calculated as the conditional mean of the density of the corresponding cluster. For comparison with that approach, we also used this strategy to make predictions in our models. We further employed the FMR model with the B-spline functional representation, instead of the functional PCA, the number of B-spline functions is set to be the same as the number of basis functions used in our models. Table 2 shows the evaluation criteria corresponding to the considered models evaluated on 100 datasets in scenario  $S3$ . Here, FMR-PC is the original model of Yao et al. (2010) and FMR-B is the modified one with B-spline bases. As expected, FME, FME-Lasso, and iFME, which are more flexible compared to the the FMR model, allows to capture more complexity in the

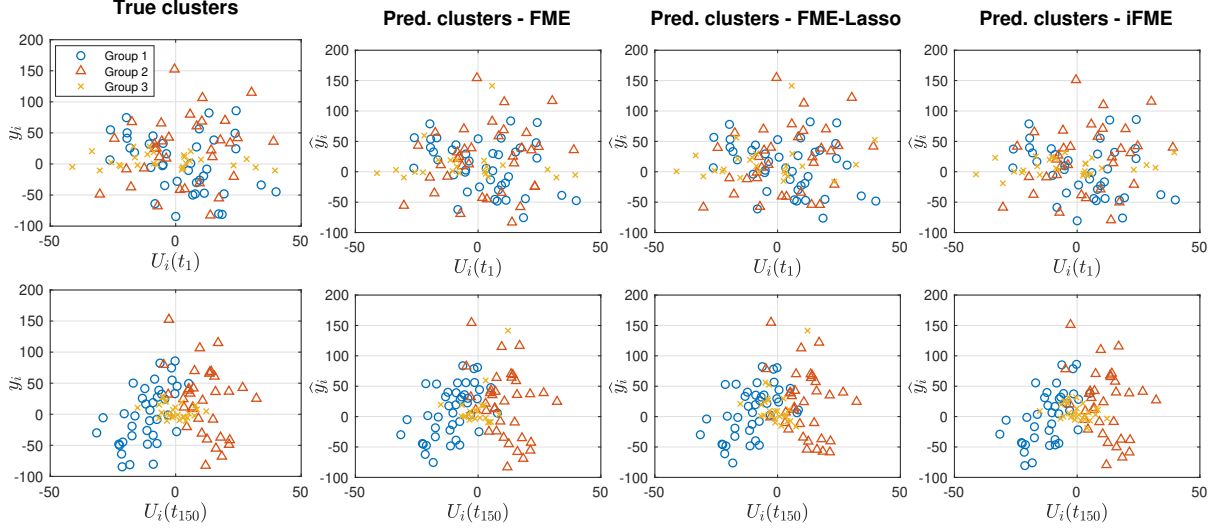


Figure 2: Scatter plots of  $\hat{y}_i$  against  $U_i(t_1)$  (top panels) and  $U_i(t_{50})$  (bottom panels) on a randomly selected dataset. Here, the clustering errors are 5.5%, 4.75% and 5.0% for FME, FME-Lasso and iFME models, respectively.

data, thanks to the predictor-depending mixture weights, and provides clearly better results than the FMR alternatives.

	RPE	Corr	ARI	ClusErr
FME	.0614 <sub>(.0896)</sub>	.9681 <sub>(.0469)</sub>	.8523 <sub>(.0934)</sub>	.0618 <sub>(.0490)</sub>
FME-Lasso	.0145 <sub>(.0254)</sub>	.9924 <sub>(.0133)</sub>	.9582 <sub>(.0641)</sub>	.0174 <sub>(.0325)</sub>
iFME	<b>.0139<sub>(.0086)</sub></b>	<b>.9928<sub>(.0046)</sub></b>	<b>.9594<sub>(.0430)</sub></b>	<b>.0151<sub>(.0166)</sub></b>
FMR-B	.0460 <sub>(.0399)</sub>	.9768 <sub>(.0201)</sub>	.7064 <sub>(.1044)</sub>	.1124 <sub>(.0434)</sub>
FMR-PC	.0191 <sub>(.0331)</sub>	.9902 <sub>(.0164)</sub>	.8345 <sub>(.0794)</sub>	.0612 <sub>(.0361)</sub>

Table 2: Performance comparison of the models for datasets in scenarios  $S3$ . The reported values are the averages of 100 Monte Carlo samples with standard errors in parentheses.

**Parameter estimation performance.** To evaluate the parameter estimation performance, we consider the functional parameter functions estimation error as defined by (41). This error between the true function and the estimated one, provides an evaluation of how well the proposed models reconstruct the hidden functional gating and expert networks. In this evaluation, we considered scenario  $S1$ , i.e.,  $m = 300$ ,  $\sigma_g^2 = 1$ . Moreover, in order to provide an idea of the impact of training size on parameter estimation, we run the models with different training sizes (share the same scenario  $S1$ ) and report the MSE for each function, for each model in Table 3. It shows that there are significant improvements, even with small training size, when using iFME model in estimating the gating network, compared to the FME and FME-Lasso model.

	$\widehat{\beta}_1(\cdot)$	$\widehat{\beta}_2(\cdot)$	$\widehat{\beta}_3(\cdot)$	$\widehat{\alpha}_1(\cdot)$	$\widehat{\alpha}_2(\cdot)$
Training size: 100					
FME	11.68 <sub>(56.63)</sub>	6.21 <sub>(24.34)</sub>	8.57 <sub>(27.60)</sub>	5.15e+04 <sub>(7.23e+04)</sub>	5.69e+04 <sub>(5.90e+06)</sub>
FME-Lasso	<b>0.53</b> <sub>(0.19)</sub>	<b>0.66</b> <sub>(0.55)</sub>	<b>0.62</b> <sub>(0.79)</sub>	17.06 <sub>(28.68)</sub>	19.50 <sub>(37.07)</sub>
iFME	1.09 <sub>(0.78)</sub>	1.05 <sub>(0.82)</sub>	2.44 <sub>(3.06)</sub>	<b>5.76</b> <sub>(7.78)</sub>	<b>4.25</b> <sub>(3.58)</sub>
Training size: 200					
FME	0.77 <sub>(0.48)</sub>	0.61 <sub>(0.30)</sub>	<b>0.19</b> <sub>(0.20)</sub>	8.55 <sub>(11.99)</sub>	9.57 <sub>(8.29)</sub>
FME-Lasso	<b>0.54</b> <sub>(0.18)</sub>	0.54 <sub>(0.20)</sub>	0.31 <sub>(0.81)</sub>	10.71 <sub>(16.63)</sub>	12.64 <sub>(16.33)</sub>
iFME	0.55 <sub>(0.50)</sub>	<b>0.48</b> <sub>(0.38)</sub>	1.26 <sub>(2.91)</sub>	<b>5.34</b> <sub>(6.21)</sub>	<b>2.81</b> <sub>(2.82)</sub>
Training size: 300					
FME	0.62 <sub>(0.56)</sub>	0.59 <sub>(0.50)</sub>	<b>0.16</b> <sub>(0.18)</sub>	12.58 <sub>(19.83)</sub>	16.49 <sub>(30.97)</sub>
FME-Lasso	0.60 <sub>(0.52)</sub>	<b>0.57</b> <sub>(0.44)</sub>	0.17 <sub>(0.19)</sub>	11.26 <sub>(16.56)</sub>	13.15 <sub>(24.80)</sub>
iFME	<b>0.56</b> <sub>(0.43)</sub>	0.59 <sub>(0.50)</sub>	0.49 <sub>(0.47)</sub>	<b>2.98</b> <sub>(3.12)</sub>	<b>3.16</b> <sub>(3.02)</sub>
Training size: 400					
FME	0.54 <sub>(0.23)</sub>	0.53 <sub>(0.19)</sub>	<b>0.11</b> <sub>(0.12)</sub>	8.27 <sub>(11.73)</sub>	7.08 <sub>(8.53)</sub>
FME-Lasso	0.52 <sub>(0.18)</sub>	0.51 <sub>(0.16)</sub>	0.15 <sub>(0.18)</sub>	6.09 <sub>(8.98)</sub>	5.14 <sub>(6.51)</sub>
iFME	<b>0.34</b> <sub>(0.19)</sub>	<b>0.38</b> <sub>(0.25)</sub>	0.74 <sub>(0.62)</sub>	<b>3.06</b> <sub>(4.21)</sub>	<b>2.77</b> <sub>(3.21)</sub>
Training size: 500					
FME	0.49 <sub>(0.22)</sub>	0.50 <sub>(0.28)</sub>	<b>0.09</b> <sub>(0.11)</sub>	5.17 <sub>(7.52)</sub>	3.90 <sub>(5.69)</sub>
FME-Lasso	0.49 <sub>(0.18)</sub>	0.49 <sub>(0.21)</sub>	<b>0.09</b> <sub>(0.11)</sub>	4.62 <sub>(7.28)</sub>	7.63 <sub>(16.27)</sub>
iFME	<b>0.35</b> <sub>(0.23)</sub>	<b>0.40</b> <sub>(0.25)</sub>	0.55 <sub>(0.63)</sub>	<b>2.11</b> <sub>(2.82)</sub>	<b>2.68</b> <sub>(4.70)</sub>

Table 3: Average of 100 Monte Carlo runs of MSE between the estimated functions resulted by FME, FME-Lasso and iFME models in  $S1$  scenario.

Now, in order to evaluate the designed sparsity of zeroth and third derivatives of the reconstructed functions, we compute the MSEs versus their true values, i.e., zeros, on each designed intervals. In particular, we divide the domain  $\mathcal{T} = [0, 1]$  into three parts:  $\mathcal{T}_1 = [0, 0.3)$ ,  $\mathcal{T}_2 = [0.3, 0.7)$ , and  $\mathcal{T}_3 = [0.7, 1]$ , then for each model the MSEs on the corresponding intervals are reported in Table 4.

	on $\mathcal{T}_2$		on $\mathcal{T}_1 \cup \mathcal{T}_3$		on $\mathcal{T}$		
	$\widehat{\beta}_1^{(0)}(\cdot)$	$\widehat{\beta}_2^{(0)}(\cdot)$	$\widehat{\beta}_1^{(3)}(\cdot)$	$\widehat{\beta}_2^{(3)}(\cdot)$	$\widehat{\beta}_3^{(3)}(\cdot)$	$\widehat{\alpha}_1^{(3)}(\cdot)$	$\widehat{\alpha}_2^{(3)}(\cdot)$
Training size: 100							
FME	<b>0.26</b> (0.15)	0.48 (0.91)	0.15 (0.07)	<b>0.12</b> (0.08)	3.96e-08 (1.37e-07)	9.24e-06 (3.32e-05)	1.08e-05 (2.62e-05)
FME-Lasso	0.27 (0.19)	<b>0.26</b> (0.16)	<b>0.14</b> (0.07)	0.15 (0.09)	<b>1.64e-09</b> (7.64e-09)	7.37e-09 (2.64e-08)	5.12e-09 (1.70e-08)
iFME	0.40 (0.26)	0.37 (0.30)	0.20 (0.10)	0.18 (0.11)	6.48e-09 (2.44e-08)	<b>4.05e-09</b> (1.33e-08)	<b>2.46e-09</b> (4.51e-09)
Training size: 200							
FME	0.43 (0.56)	0.22 (0.09)	0.16 (0.10)	0.14 (0.05)	<b>2.23e-11</b> (2.69e-11)	<b>1.64e-10</b> (2.33e-10)	3.83e-10 (8.90e-10)
FME-Lasso	0.21 (0.11)	0.20 (0.09)	0.15 (0.05)	0.15 (0.05)	3.62e-11 (6.04e-11)	2.36e-10 (3.38e-10)	<b>2.96e-10</b> (6.41e-10)
iFME	<b>0.18</b> (0.08)	<b>0.17</b> (0.09)	<b>0.12</b> (0.05)	<b>0.12</b> (0.05)	9.95e-10 (2.00e-09)	1.44e-09 (2.72e-09)	8.52e-10 (1.22e-09)
Training size: 300							
FME	0.25 (0.30)	0.22 (0.25)	0.16 (0.10)	0.15 (0.09)	<b>1.72e-11</b> (2.30e-11)	2.55e-10 (4.61e-10)	1.01e-10 (1.42e-10)
FME-Lasso	0.25 (0.28)	<b>0.21</b> (0.23)	0.16 (0.09)	<b>0.15</b> (0.08)	1.73e-11 (2.52e-11)	<b>1.41e-10</b> (2.00e-10)	<b>1.64e-10</b> (3.77e-10)
iFME	<b>0.25</b> (0.19)	0.24 (0.19)	<b>0.16</b> (0.08)	<b>0.15</b> (0.08)	5.94e-10 (1.26e-09)	5.41e-10 (7.26e-10)	7.15e-10 (1.05e-09)
Training size: 400							
FME	0.23 (0.12)	0.23 (0.10)	0.16 (0.05)	0.16 (0.06)	<b>1.62e-11</b> (3.53e-11)	1.29e-10 (3.59e-10)	<b>8.14e-11</b> (1.02e-10)
FME-Lasso	0.22 (0.11)	0.21 (0.10)	0.16 (0.05)	0.16 (0.05)	1.90e-11 (3.55e-11)	<b>9.71e-11</b> (1.63e-10)	1.03e-10 (1.54e-10)
iFME	<b>0.14</b> (0.08)	<b>0.16</b> (0.09)	<b>0.08</b> (0.03)	<b>0.08</b> (0.04)	1.87e-08 (1.42e-08)	1.13e-08 (1.77e-08)	9.91e-09 (1.05e-08)
Training size: 500							
FME	0.22 (0.11)	0.21 (0.10)	0.17 (0.05)	0.17 (0.05)	2.12e-11 (4.66e-11)	8.98e-11 (2.70e-10)	1.03e-10 (2.83e-10)
FME-Lasso	0.21 (0.09)	0.20 (0.09)	0.17 (0.05)	0.17 (0.05)	<b>1.90e-11</b> (3.64e-11)	<b>7.02e-11</b> (1.44e-10)	<b>8.91e-11</b> (1.69e-10)
iFME	<b>0.14</b> (0.07)	<b>0.16</b> (0.08)	<b>0.08</b> (0.04)	<b>0.08</b> (0.04)	1.55e-08 (1.34e-08)	1.03e-08 (1.58e-08)	8.34e-09 (1.20e-08)

Table 4: MSE of the derivatives of reconstructed functions on the corresponding interested intervals, in which,  $\mathcal{T}_1 = [0, 0.3)$ ,  $\mathcal{T}_2 = [0.3, 0.7)$ ,  $\mathcal{T}_3 = [0.7, 1]$  and  $\mathcal{T} = [0, 1]$ . The reported values are the averaged of 100 Monte Carlo samples with standard errors in parentheses.

The reported values show that, as expected, iFME model is better than both FME and FME-Lasso in providing sparse solutions with respect to the derivatives of the parameter functions.

Table 5 shows the means and the standard errors of the estimated intercepts and variances. Note that these coefficients are not considered in the penalization. For the intercepts  $\beta_{k,0}$ , all the models estimated them very well, while for the intercepts  $\alpha_{k,0}$ , iFME is slightly

better than the others. For the variances, FME-Lasso gave the estimated values closest with the true values on average.

	$\hat{\beta}_{1,0}$	$\hat{\beta}_{2,0}$	$\hat{\beta}_{3,0}$	$\hat{\alpha}_{1,0}$	$\hat{\alpha}_{2,0}$	$\hat{\sigma}_1$	$\hat{\sigma}_2$	$\hat{\sigma}_3$
True value	-5	0	5	-10	-10	5	5	5
Training size: 100								
FME	-16.90 (18.91)	-3.18 (3.25)	4.17 (8.87)	-31.37 (30.74)	-25.20 (41.27)	32.38 (40.05)	15.19 (25.15)	21.47 (24.71)
FME-Lasso	-6.48 (9.37)	<b>-0.07</b> (1.95)	5.51 (4.63)	-12.53 (8.99)	-12.69 (10.99)	6.86 (10.98)	6.73 (8.08)	10.01 (20.60)
iFME	<b>-4.95</b> (1.14)	0.21 (1.08)	<b>5.05</b> (0.90)	<b>-9.44</b> (3.85)	<b>-9.39</b> (3.31)	<b>5.70</b> (1.46)	<b>5.75</b> (1.49)	<b>4.70</b> (1.51)
Training size: 200								
FME	-4.83 (0.78)	0.14 (0.54)	4.89 (0.38)	-23.77 (26.36)	-16.10 (6.65)	5.77 (1.26)	<b>5.63</b> (0.95)	5.24 (0.79)
FME-Lasso	<b>-4.97</b> (0.66)	<b>0.07</b> (0.60)	4.99 (0.45)	-12.70 (4.45)	-13.72 (4.38)	5.94 (1.23)	5.99 (1.12)	<b>4.92</b> (1.30)
iFME	-4.85 (0.72)	0.19 (0.68)	<b>4.99</b> (0.42)	<b>-9.30</b> (3.15)	<b>-9.21</b> (2.83)	<b>5.77</b> (0.90)	5.72 (1.02)	5.28 (1.27)
Training size: 300								
FME	<b>-4.99</b> (0.55)	-0.06 (0.54)	4.98 (0.26)	-16.82 (13.09)	-17.18 (14.89)	5.92 (1.14)	5.87 (0.94)	<b>5.34</b> (0.84)
FME-Lasso	-5.03 (0.54)	<b>-0.03</b> (0.53)	4.98 (0.27)	-13.06 (4.54)	-12.96 (4.96)	6.08 (1.10)	5.99 (1.07)	5.37 (0.89)
iFME	-4.93 (0.61)	0.04 (0.56)	<b>5.00</b> (0.29)	<b>-8.96</b> (2.48)	<b>-8.85</b> (2.57)	<b>5.54</b> (0.75)	<b>5.53</b> (0.72)	5.53 (0.87)
Training size: 400								
FME	<b>-5.02</b> (0.43)	<b>0.01</b> (0.40)	5.04 (0.28)	-16.37 (9.93)	-15.39 (7.83)	5.99 (1.01)	6.20 (1.09)	5.47 (0.77)
FME-Lasso	-5.03 (0.43)	-0.02 (0.38)	5.02 (0.27)	-13.18 (4.61)	-12.65 (4.57)	5.99 (0.94)	6.13 (1.01)	<b>5.45</b> (0.84)
iFME	-4.89 (0.45)	0.05 (0.35)	<b>5.01</b> (0.27)	<b>-10.17</b> (2.79)	<b>-9.82</b> (2.61)	<b>5.08</b> (0.60)	<b>5.22</b> (0.58)	5.57 (0.85)
Training size: 500								
FME	<b>-5.00</b> (0.41)	-0.01 (0.38)	<b>5.01</b> (0.25)	-14.09 (7.26)	-15.42 (9.64)	6.09 (0.98)	6.06 (0.93)	5.44 (0.69)
FME-Lasso	-4.99 (0.40)	<b>-0.01</b> (0.37)	5.01 (0.26)	-11.35 (3.27)	-12.38 (4.15)	6.13 (0.98)	6.06 (0.90)	<b>5.44</b> (0.68)
iFME	-4.92 (0.44)	0.06 (0.34)	5.02 (0.27)	<b>-9.23</b> (1.93)	<b>-9.61</b> (2.68)	<b>5.23</b> (0.55)	<b>5.13</b> (0.55)	5.60 (0.75)

Table 5: Intercepts and variances obtained by FME, FME-Lasso and iFME models in scenario  $S1$ . The reported values are the averages of 100 Monte Carlo samples with standard errors in parentheses.

The initialization is crucial for the EM algorithm. In all of our experimental studies, the model parameter vector  $\Psi$  of the FME model was initialized as follows (similar for the FME-Lasso and iFME models). Firstly, we perform a  $K$ -means algorithm on the predictors  $\{\mathbf{x}_i\}_{i=1}^n$ , then for each estimated group  $k$ ,  $(\beta_{k,0}^{(0)}, \boldsymbol{\eta}_k^{(0)})$  is initialized as the solution to the linear regression problem  $y_i = \beta_{k,0} + \mathbf{x}_i^\top \boldsymbol{\eta}_k$ , for  $i$  belongs to group  $k$ th. The parameter  $\sigma_k^{(0)}$  is then

	log likelihood			RPE			Time (s)		
	rand.	zeros	LR	rand.	zeros	LR	rand.	zeros	LR
FME	<b>-215.52</b> (24.77)	-221.03 (18.51)	-285.23 (69.30)	<b>0.19</b> (0.16)	0.20 (0.13)	0.38 (0.29)	0.21 (0.50)	<b>0.16</b> (0.12)	0.52 (1.99)
FME-Lasso	<b>-235.35</b> (21.98)	-239.47 (18.54)	-244.99 (63.09)	0.14 (0.11)	<b>0.13</b> (0.10)	0.20 (0.16)	1.13 (1.23)	<b>1.08</b> (2.48)	4.63 (15.51)
iFME	<b>-254.65</b> (7.83)	-256.26 (13.76)	-336.87 (80.38)	<b>0.13</b> (0.09)	<b>0.13</b> (0.09)	0.35 (0.30)	6.58 (4.45)	4.17 (3.65)	<b>3.91</b> (2.08)

Table 6: Comparison of different initialization strategies. The reported values are the mean and standard error (in parentheses) over 100 Monte Carlo runs.

initialized as the estimated variance within group  $k$ th. For the gating parameter, we simply drawn each component of  $\xi^{(0)}$  randomly from the uniform distribution  $\mathcal{U}(0, 1)$ .

However, to see the impact of different initialization strategies, we performed a side experiment with data taken from  $S1$  scenario. In particular, the initialization for expert parameters is fixed as described before, while that for gating parameter varies in “rand.”, “zeros”, and “LR”. Here, “rand.” is the random strategy described above, “zeros” is the strategy where all coefficients of  $\xi$  are initialized as zeros, i.e., we are putting equal weights for the experts; and “LR” is the strategy where we perform a logistic regression with the predictors are  $\mathbf{r}_i$  (for FME, FME-Lasso) or  $\mathbf{s}_i$  (for iFME), and the responses are the labels resulted by  $K$ -means on them. Table 6 shows the time to convergence, and the log-likelihood, RPE on testing sets corresponding with the three strategies. We can see that, in terms of log-likelihood and RPE, simple strategies as “rand.” and “zeros” perform quite well, while in terms of time to convergence, LR is a good choice for the iFME model.

Finally, to illustrate the selection of the number of expert components using BIC and/or modified BIC in this simulation study, we provide, in Figure 3, the plots of these criteria against the number of experts for each model. Here, we implemented the models with all fixed tuning parameters, except  $K$  which varies in the set  $\{1, \dots, 6\}$ . We can observe that BIC selects the correct number  $K = 3$  for both FME-Lasso and iFME, while it selects  $K = 4$  for FME. However, the modified BIC selects  $K = 4$  for both FME and FME-Lasso, and selects the true number of components  $K = 3$  for iFME.

### 4.3 Application to real data

In this section, we apply the FME, FME-Lasso and iFME models to two well-known real datasets, Canadian weather and Diffusion Tensor Imaging (DTI). For each dataset, we perform clustering and investigate the prediction performance, estimate the functional mixture of experts models with different number of experts  $K$  and perform the selection of  $K$  using modified BIC, and discuss the obtained results.

#### 4.3.1 Canadian weather data

Canadian weather data has been introduced in Ramsay and Silverman (2005) and is also available in the R package `fda`. This dataset consists of  $m = 365$  daily temperature measure-

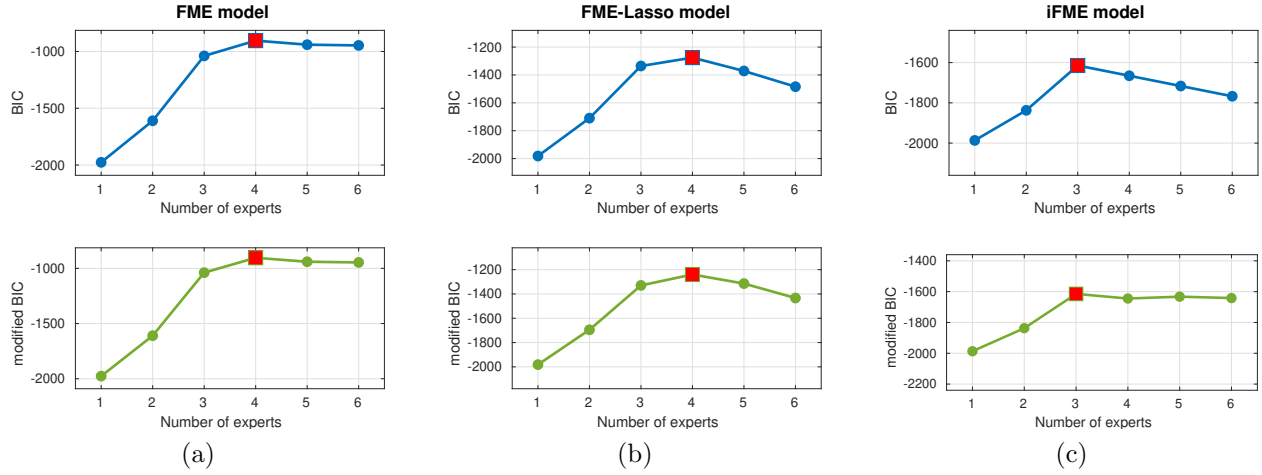


Figure 3: Values of BIC (top) and modified BIC (bottom) for (a) FME, (b) FME-Lasso and (c) iFME model versus the number of experts  $K$ , fitted on a randomly taken dataset in scenario  $S1$ . Here, the square points correspond to the highest values.

ments (averaged over the year 1961 to 1994) at  $n = 35$  weather stations in Canada, and their corresponding average annual precipitation (in log scale). The weather stations are located in 4 climate zones: Atlantic, Pacific, Continental and Arctic (Figure 4b). In this dataset, presented in Figure 4a, the noisy functional predictors  $U_i(\cdot)$  are the curves of 365 averaged daily temperature measurements, the scalar responses  $Y_i$  are the corresponding total precipitation at each station  $i$ , during the year, for 35 stations. Its station climate zone is taken as a cluster label (the cluster label  $Z_i \in \{1, \dots, 4\}$ ). The aim here is to use the daily temperature

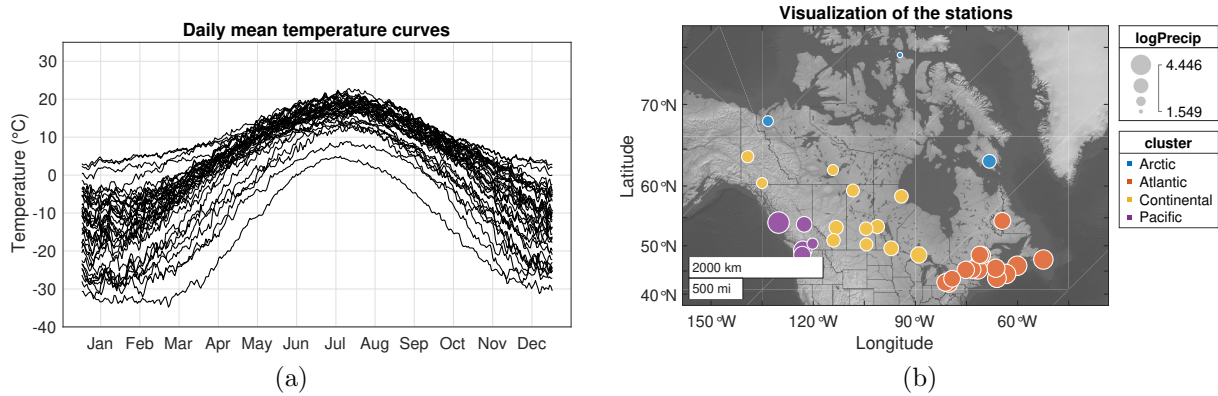


Figure 4: (a) 35 daily mean temperature measurement curves; (b) Geographical visualization of the stations, in which the sizes of the bubbles corresponds to their log of precipitation values and the colors correspond to their climate regions.

curves (functional predictors) to predict the precipitations (scalar responses) at each station. Moreover, in addition to predicting the precipitation values, we are interested in clustering the temperature curves (therefore the stations), as well as identifying the periods of time of the year that have effect on prediction for each group of curves.

	Corr	SSE	RPE
FME	0.690	14.410	0.290
FME-Lasso	0.632	2.011	0.045
iFME	<b>0.944</b>	<b>0.582</b>	<b>0.012</b>

Table 7: 7-fold cross-validated correlation (Corr), sum of squared errors (SSE) and relative prediction error (RPE) of predictions on Canadian weather data.

Firstly, in order to assess the prediction performance of the FME, FME-Lasso and iFME models on this dataset, we implement the models by selecting the tuning parameters, including the number of expert components  $K$  in the set  $\{1, 2, 3, 4, 5, 6\}$ , by maximizing the modified BIC criterion, given its performance as shown in the simulation study. We report in Table 7 the results in terms of correlations, *sum of squared errors* (SSE) and relative prediction errors. According to the obtained results, iFME provides the best results w.r.t. all the criteria. The cross-validated RPE provided by iFME is only of 1.2%, the next is FME-Lasso with 4.5%, while the FME model has the worst RPE value. Note that in [James et al. \(2009\)](#), the authors applied their proposed model to Canadian weather data and obtained a 10-fold cross-validated SSE of 4.77. Clearly, with the smaller cross-validated SSEs, the FME-Lasso and iFME models significantly improve the prediction.

Figure 5 shows the obtained results with the FME, FME-Lasso and iFME models, with  $K = 4$ , and with the derivatives  $d_1 = 0$ ,  $d_2 = 3$  for the iFME model. The estimated experts functions and gating functions are presented in the two top panels of the curve, while the clustering for the temperature curves and the stations are shown in the two bottom panels. As we can see, all models provide reasonable clustering for the curves which may be corresponding to different complicated underlying meteorological forecasting mechanisms. Particularly, although not using any spatial information, merely temperature information, the obtained clustering for the stations is also comparable with the original labels of the stations. For example, the FME and FME-Lasso models identify exactly the Arctic stations, while iFME identifies exactly the Pacific stations, and all of the models provide reasonable spatially organized clusters. However, what is interesting here is the shape of the expert and gating functions  $\hat{\alpha}(\cdot)$ 's and  $\hat{\beta}(\cdot)$ 's obtained by the models. While FME and FME-Lasso gave less interpretable estimations, iFME appears to give, as it can be seen in the two top-right panels, piece-wise zero-valued and possibly quadratic estimated functions, which have a wide range of flat relationship from January to February and from June to September.

Motivated by the above results, on direction of identifying the periods of time of the year that truly have an effect on prediction, we implement the iFME model with  $K = 2$ , and the derivative levels  $d_1$  and  $d_2$  are set to be the zeroth and the third derivatives. The reason for the choices of  $d_1$  and  $d_2$  is that the penalization on the zeroth derivative would take into account zero ranges in the expert and gating functions, while the penalization on the third derivative, would take into account the smoothness for the changes between the periods of times in the functions. The obtained results are shown in Figure 6. As we can see, there are differences in the prediction mechanisms of the models between the northern stations and the southern stations. At southern stations, the obtained  $\hat{\beta}_2(t)$  shows that there is a negative relationship in the spring and a positive relationship in the late fall, but no

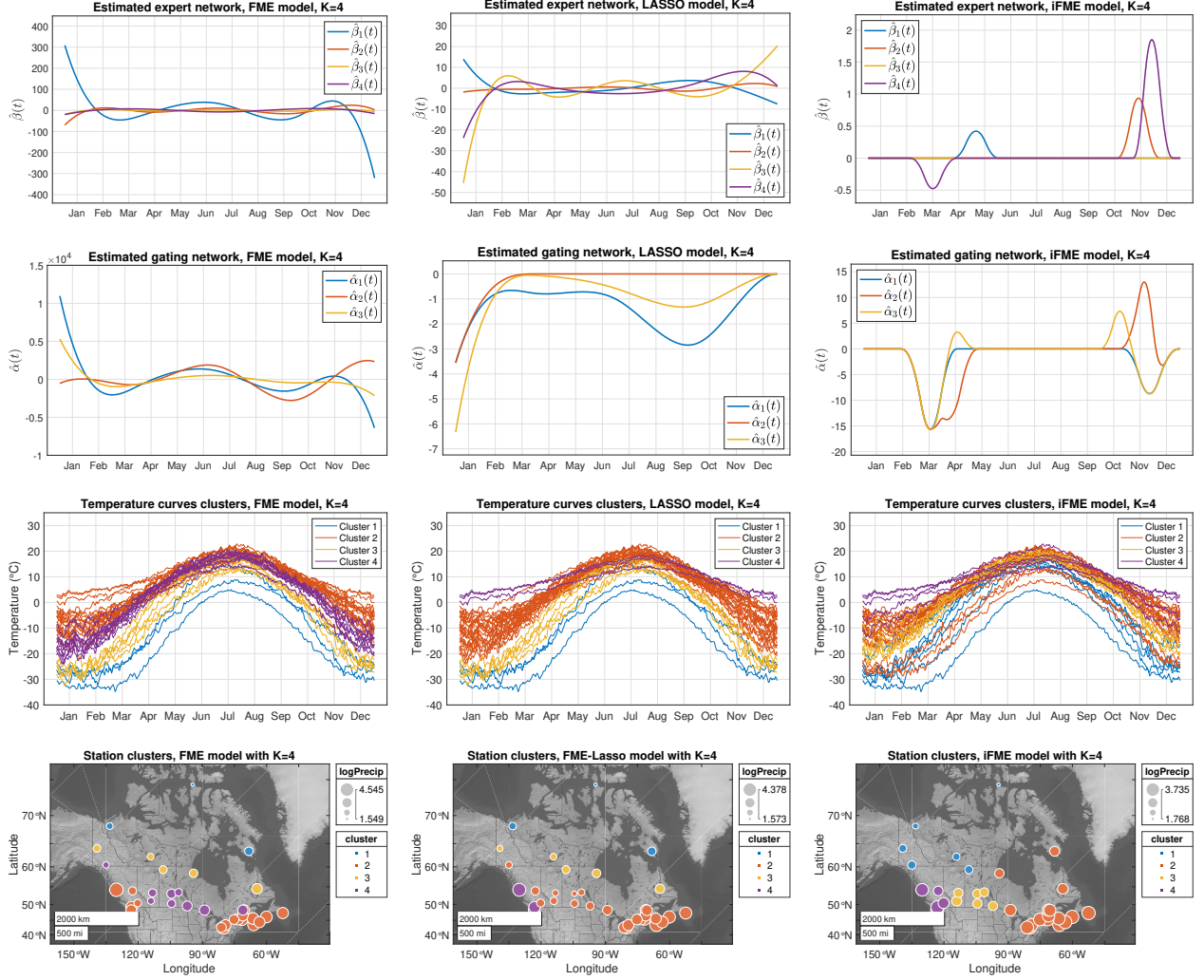


Figure 5: Results obtained by FME (left panels), FME-Lasso (middle panels) and iFME (right panels) on Canadian weather data with  $K = 4$ . For each column, the panels are respectively the estimated functional experts network, estimated functional gating network, estimated clusters of the temperature curves and estimated clusters of the stations.

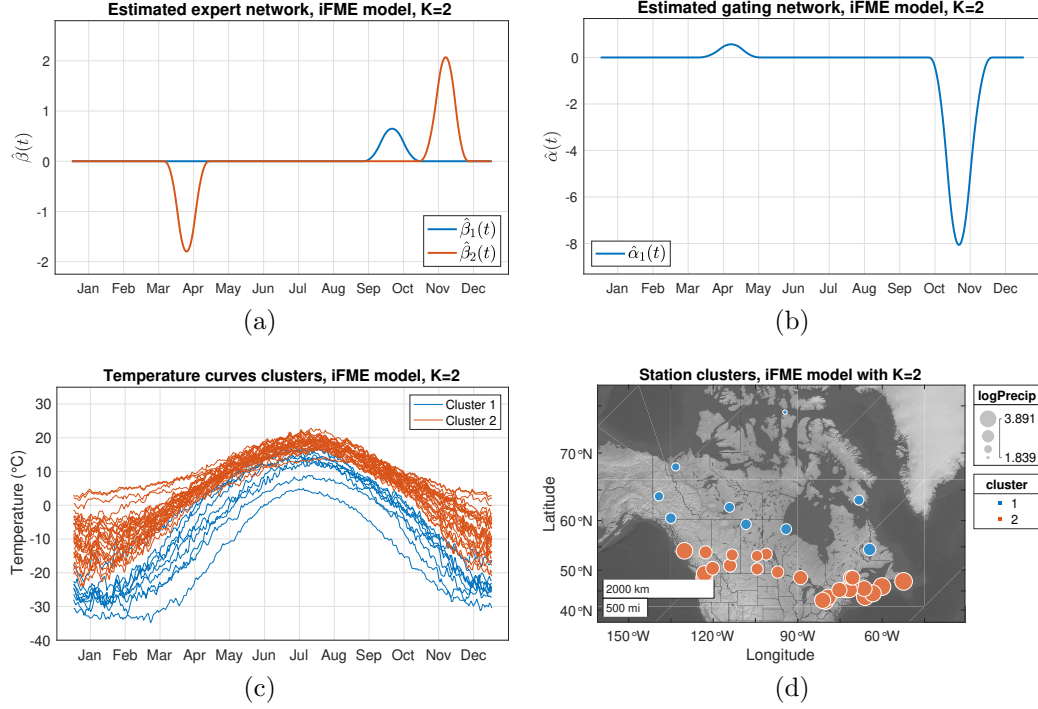


Figure 6: Results obtained by iFME model with  $K = 2$ ,  $d_1 = 0$ ,  $d_2 = 3$ : (a) Estimated functional expert network, (b) Estimated functional gating network, (c) Estimated clusters of the temperature curves, and (d) Estimated clusters of the stations.

relationship in the remaining period of the year. This phenomenon is concordant with the result obtained in [James et al. \(2009\)](#), where the authors obtained the same relationships in the same periods of time. However, our iFME model additionally suggests that, at the northern stations, the relationship between temperature and precipitation may differ from that of southern stations. This may be explained by the differences in mean temperatures and climatic characteristics between the two regions.

Finally, Figure 7 displays the values of modified BIC for varying number of expert component for the proposed models on Canadian weather data. According to these values, FME-Lasso and iFME select  $K = 2$ , while FME selects  $K = 3$ .

#### 4.3.2 Diffusion tensor imaging data for multiple sclerosis subjects

We now apply our proposed models to the diffusion tensor imaging (DTI) data for subjects with multiple sclerosis (MS), discussed in [Goldsmith et al. \(2012\)](#). The data come from a longitudinal study investigating the cerebral white matter tracts of subjects with multiple sclerosis, recruited from an outpatient neurology clinic and healthy controls. We are interested in the underlying relationship between the fractional anisotropy profile (FAP) from the corpus callosum and the paced auditory serial addition test (PASAT) score, which is a commonly used examination of cognitive function affected by MS. The FAP curves are derived from DTI data, which are obtained by a Magnetic Resonance Imaging (MRI) scanner. Each curve is recorded at 93 locations along the corpus callosum. The PASAT score is the

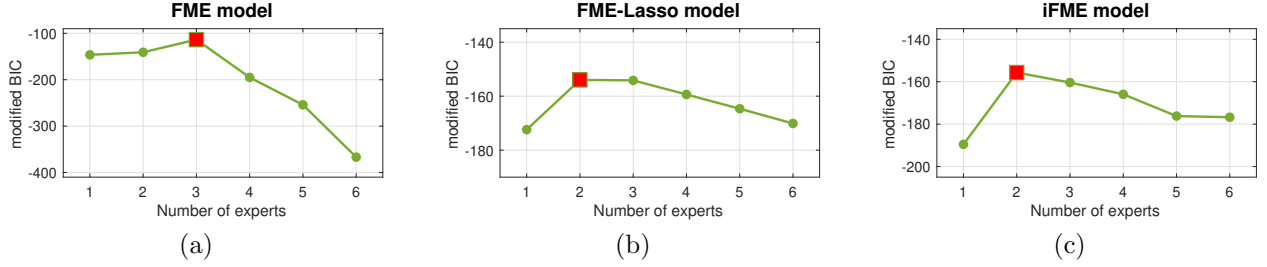


Figure 7: Values of modified BIC for (a) FME, (b) FME-Lasso and (c) iFME model, versus the number of experts  $K$ , fitted on Canadian weather data. The square points correspond to highest values. Here, iFME is implemented with  $d_1 = 0$ ,  $d_2 = 3$  and  $\rho = \varrho = 100$ .

number of correct answers out of 60 questions, and thus ranges from 0 to 60. In our context, the FAP curves serve as the noisy functional predictors  $U_i(\cdot)$  and the PASAT scores serve as the scalar responses  $Y_i$ . So this dataset consists of  $n = 99$  pairs  $(U_i(\cdot), Y_i)$ ,  $i \in [n]$ , with each  $U_i(\cdot)$  contains  $m = 93$  fractional anisotropy values. Figure 9a shows all the predictors (FAP curves), and Figure 8 (most-right panels) shows the responses (PASAT Scores).

In Ciarleglio and Ogden (2016), the authors applied their Wavelet-based functional mixture regression (WBFMR) model with two components to this dataset, and observed that there is one group in which there is no association between the FAP and the PASAT score for those subjects belonging to it. We accordingly fix  $K = 2$  in our models. Figure 8 displays the obtained results for each of the three models, and Figure 9a shows the functional predictors FAP curves clustered with the iFME model. In this implementation, we tried iFME model with two different combinations of  $d_1$  and  $d_2$ :  $(d_1, d_2) = (0, 2)$  and  $(d_1, d_2) = (0, 3)$ . As expected, when  $d_2$  is the second derivative, the reconstructed parameter functions are piecewise zero and linear, while when  $d_2$  is the third derivative, the reconstructed functions have smooth changes along the tract location.

In Figure 8, we have the three following observations. First, as it can be observed in Figure 8 right-panel, all models give a threshold of 50 that clusters the PASAT scores, this is the same as the threshold observed in Ciarleglio and Ogden (2016). Second, the absolute values of the coefficient functions  $\hat{\beta}_2(t)$ 's are significantly smaller than those of  $\hat{\beta}_1(t)$ 's, this is again the same with the result obtained by the WBFMR model. Third, when  $\hat{\beta}_2(t)$  is estimated as zero in the FME-Lasso and iFME models, the shape of  $\hat{\beta}_1(t)$  is almost the same as the shape obtained in Ciarleglio and Ogden (2016), particularly, the peak at around the tract location of 0.42. These confirm the underlying relationship between the fractional anisotropy and the cognitive function: higher fractional anisotropy values between the locations about 0.2 to 0.7 results in higher PASAT scores for subjects in Group 1. The clustering of the FAP curves, resulted by the iFME model with  $d_1 = 0$ ,  $d_2 = 2$ , is shown in Figure 9 (b)-(d).

Next, to compare with Ciarleglio and Ogden (2016), we investigate the prediction performance of the proposed models with respect to the leave-one-out cross validated relative prediction errors defined by  $\text{CVRPE} = \sum_{i=1}^n (y_i - \hat{y}_i^{(-i)})^2 / \sum_{i=1}^n y_i^2$ , where  $y_i$  is the true score for subject  $i$  and  $\hat{y}_i^{(-i)}$  is the score predicted by the model fit on data without subject  $i$ . In this implementation, we keep fixing  $K = 2$  and select the other tuning parameters by maximizing the modified BIC. The CVRPEs corresponding to the models are provided in

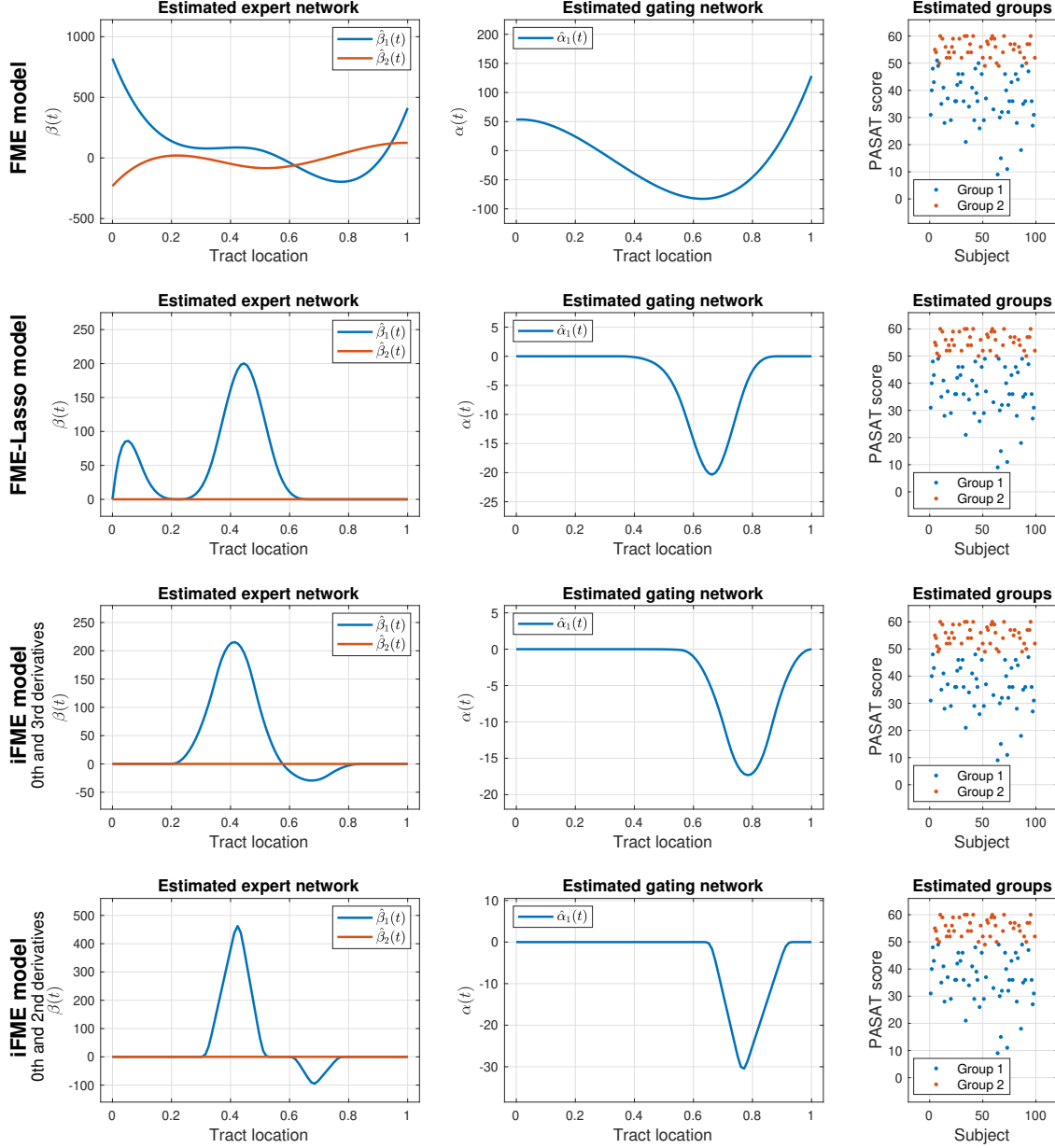


Figure 8: The estimated expert and gating coefficient functions, the estimated groups of the PASAT scores, resulted by FME, FME-Lasso and iFME models with  $K = 2$  for the DTI dataset. For iFME model, the upper is implemented with penalization on the zeroth and third derivatives, while the lower is with penalized zeroth and second derivatives.

Table 8. Note that, for comparison, in [Ciarleglio and Ogden \(2016\)](#), the CVRPE of their WBFMR model is 0.0315 and of the wavelet based functional linear model (FLM) is 0.0723.

Finally, we present in Figure 10 the selection of the number of experts  $K$  with modified BIC. In this case, FME and FME-Lasso select  $K = 2$ , and iFME selects  $K = 4$ .

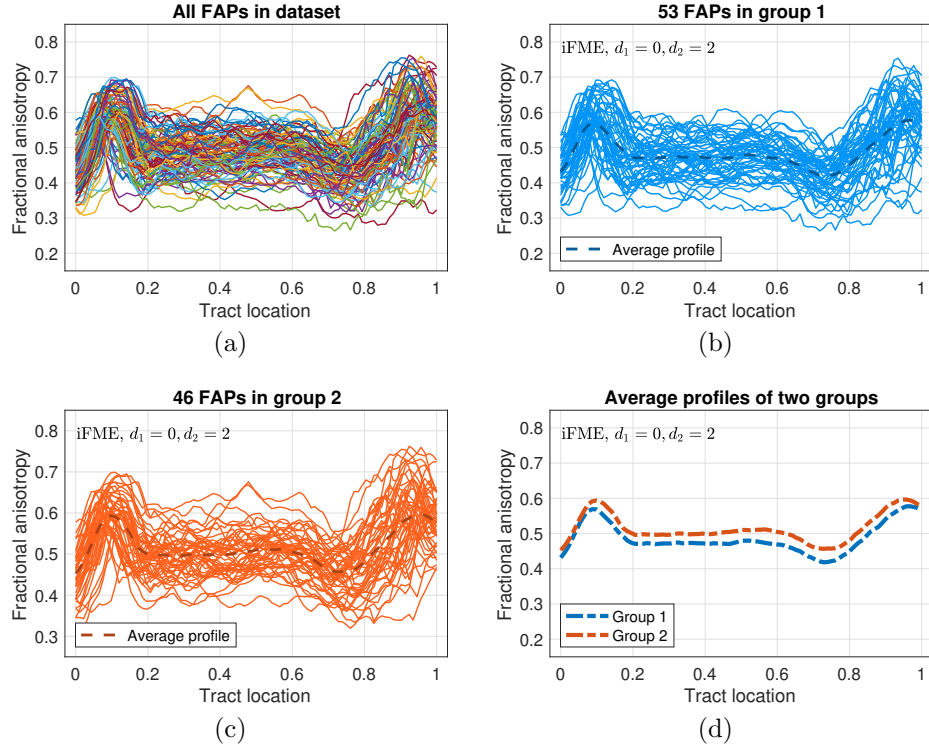


Figure 9: DTI data with (a) FAP curves of all subjects, (b) Cluster 1 and (c) Cluster 2, obtained by iFME model with  $d_1 = 0$ ,  $d_2 = 2$ , and (d) the point-wise average of the curves in each of the two clusters.

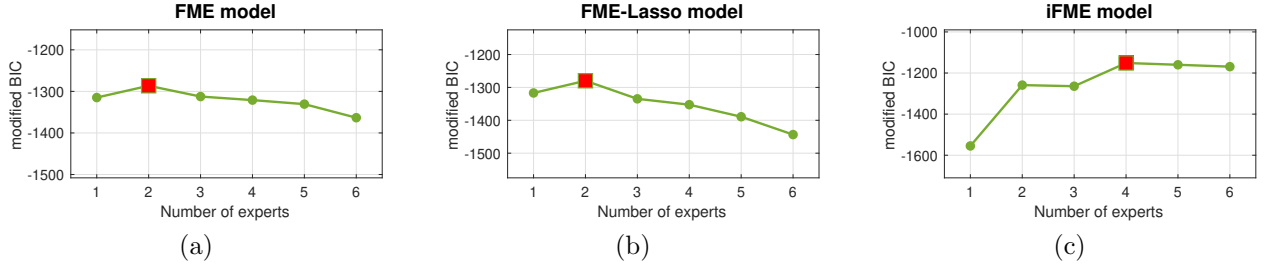


Figure 10: Values of modified BIC for (a) FME, (b) FME-Lasso and (c) iFME, versus the number of experts  $K$ , fitted on DTI data. The square points correspond to highest values.

	CVRPE
FME	0.0273
FME-Lasso	0.0280
iFME (with 0th and 3rd derivatives)	0.0271
iFME (with 0th and 2nd derivatives)	<b>0.0267</b>

Table 8: CVRPEs of the models on the DTI data.

## 5 Conclusion and discussion

The first algorithm for mixtures-of-experts constructed upon functional predictors is presented in this paper. Beside the classic maximum likelihood parameter estimation, we proposed two other regularized versions that allow sparse and notably interpretable solutions, by regularizing in particular the derivatives of the underlying functional parameters of the model, after projecting onto a set of continuous basis functions. The performances of the proposed approaches are evaluated in data prediction and clustering via experiments involving simulated and two real-world functional data.

The presented FME models can be extended in different ways. First direct extensions of the modeling framework presented here can be considered with categorical response, to perform supervised classification with functional predictors, or with vector response, to perform multivariate functional regression. Then, it may be interesting to consider the extension of the FME model to setting involving vector (or scalar) predictors and functional responses (Chiou et al., 2004).

Another extension, which we intend also to investigate in the future, concerns the case when we observe pairs of functional data, i.e., a sample of  $n$  functional data pairs  $\{X_i(t), Y_i(u)\}_{i=1}^n$ ,  $t \in \mathcal{T} \subset \mathbb{R}$ ,  $u \in \mathcal{U} \subset \mathbb{R}$ , where  $Y_i(\cdot)$  is a functional response, explained by a functional predictor  $X_i(\cdot)$ . The modeling with such FME extension then takes the form  $Y_i(u) = \beta_{z_i,0}(u) + \int_t X_i(t)\beta_{z_i}(t, u)dt + \varepsilon_i(u)$ , to explain the functional response  $Y$  by the functional predictor  $X$  via the unknown discrete variable  $z$ . The particularity with this model is that, for the clustering, as well as for the prediction, we model the relation between  $Y$  at any time  $u$  and the entire curve of  $X$ , or the entire curve of each variable  $X_{ij}$  in the case of multivariate functional predictor  $\mathbf{X}_i$ .

Since we look for sparse estimates in the space of the derivatives of the functional parameters, and provided the resulting estimates with highly structured shapes, we believe that using the fused LASSO (Tibshirani et al., 2005) rather than the LASSO could be expected to take accommodate more the targeted sparsity. The choice of the regularization constants for the derivatives is more about the targeted penalization magnitude (i.e., very large or very small) rather than the exact values. Typically, in the case one has no idea about the shape of the functional experts, but still want them to be interpretable, then it is practical to consider the zeroth and third derivatives (since they produce smooth changes), with the grid for  $\rho$  containing both very small and very large values. The same for the functional gating network. Furthermore, if we believe that the relation of the response on the predictor is very sparse, except over some small regions where they are linearly related, then we will penalize the zeroth and second derivatives of the functional experts, and we will put very small regularization on the second derivative. This could produce curves such as those in bottom panels of Figure 8. For a more general tuning of the regularization constant values, we recommend performing a complete cross-validation study to select the best values of the regularization constants.

## 6 Acknowledgements

This research was partly supported by an Ethel Raybould Fellowship, University of Queensland (FC), the Australian Research Council grant number DP180101192 (GJM), the French National Research Agency ANR grant SMILES ANR-18-CE40-0014 (FC and NTP), Région Normandie grant RIN AStERiCs (FC and VHH), Vietnam National University – Ho Chi Minh City (VNU- HCM) grant number C2022-18-01 (VHH), and the exploratory research facility “EXPLO” at SystemX (FC).

## References

- Candes, E., Tao, T., et al. (2007). The dantzig selector: Statistical estimation when  $p$  is much larger than  $n$ . *Annals of Statistics*, 35(6):2313–2351.
- Chamroukhi, F. and Huynh, B.-T. (2019). Regularized maximum likelihood estimation and feature selection in mixtures-of-experts models. *Journal de la Société Française de Statistique*, 160(1):57–85.
- Chamroukhi, F., Lecocq, F., and Nguyen, H. D. (2019). Regularized Estimation and Feature Selection in Mixtures of Gaussian-Gated Experts Models. In *Research School on Statistics and Data Science*, pages 42–56. Springer.
- Chamroukhi, F. and Nguyen, H. D. (2019). Model-based clustering and classification of functional data. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(4):e1298.
- Chiou, J.-M., Müller, H.-G., and Wang, J.-L. (2004). Functional response models. *Statistica Sinica*, 14(3):675–693.
- Ciarleglio, A. and Ogden, R. T. (2016). Wavelet-based scalar-on-function finite mixture regression models. *Computational Statistics & Data Analysis*, 93:86 – 96.
- Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of The Royal Statistical Society, B*, 39(1):1–38.
- Devijver, E. (2017). Model-based clustering for high-dimensional data. Application to functional data. *Advances in Data Analysis and Classification*, 11:243–279.
- Ferraty, F. and Vieu, P. (2006). *Nonparametric Functional Data Analysis: Theory and Practice (Springer Series in Statistics)*. Springer-Verlag, Berlin, Heidelberg.
- Gaines, B. R., Kim, J., and Zhou, H. (2018). Algorithms for fitting the constrained lasso. *Journal of Computational and Graphical Statistics*, 27(4):861–871. PMID: 30618485.
- Goldsmith, J., Bobb, J., Crainiceanu, C. M., Caffo, B., and Reich, D. (2011). Penalized functional regression. *Journal of Computational and Graphical Statistics*, 20(4):830–851.

- Goldsmith, J., Crainiceanu, C. M., Caffo, B., and Reich, D. (2012). Longitudinal penalized functional regression for cognitive outcomes on neuronal tract measurements. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 61(3):453–469.
- Gormley, I. C. and Murphy, T. B. (2008). A mixture of experts model for rank data with applications in election studies. *The Annals of Applied Statistics*, 2(4):1452 – 1477.
- Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical learning with sparsity: the lasso and generalizations*. Chapman and Hall/CRC.
- Huynh, T. and Chamroukhi, F. (2019). Estimation and feature selection in mixtures of generalized linear experts models. *arXiv preprint arXiv:1810.12161*.
- Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.
- Jacques, J. and Preda, C. (2014a). Functional data clustering: a survey. *Advances in Data Analysis and Classification*, 8(3):231–255.
- Jacques, J. and Preda, C. (2014b). Model-based clustering for multivariate functional data. *Computational Statistics & Data Analysis*, 71:92–106.
- James, G. M. (2002). Generalized linear models with functional predictor variables. *Journal of the Royal Statistical Society Series B*, 64:411–432.
- James, G. M. and Hastie, T. J. (2001). Functional linear discriminant analysis for irregularly sampled curves. *Journal of the Royal Statistical Society Series B*, 63:533–550.
- James, G. M. and Sugar, C. (2003). Clustering for sparsely sampled functional data. *Journal of the American Statistical Association*, 98(462).
- James, G. M., Wang, J., and Zhu, J. (2009). Functional linear regression that’s interpretable. *Annals of Statistics*, 37(5A):2083–2108.
- Jiang, W. and Tanner, M. (1999). On the identifiability of mixtures-of-experts. *Neural Networks*, 12(9):1253–1258.
- Jordan, M. I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214.
- Khalili, A. (2010). New estimation and feature selection methods in mixture-of-experts models. *Canadian Journal of Statistics*, 38(4):519–539.
- Liu, X. and Yang, M. (2009). Simultaneous curve registration and clustering for functional data. *Computational Statistics and Data Analysis*, 53(4):1361–1376.
- McLachlan, G. J. and Krishnan, T. (2008). *The EM algorithm and extensions*. New York: Wiley, second edition.
- McLachlan, G. J. and Peel, D. (2000). *Finite Mixture Models*. New York: Wiley.

- Montuelle, L., Le Pennec, E., et al. (2014). Mixture of gaussian regressions model with logistic weights, a penalized maximum likelihood approach. *Electronic Journal of Statistics*, 8(1):1661–1695.
- Mousavi, S. and Sørensen, H. (2017). Multinomial functional regression with wavelets and lasso penalization. *Econometrics and Statistics*, 150–166.
- Mousavi, S. N. and Sørensen, H. (2018). Functional logistic regression: a comparison of three methods. *Journal of Statistical Computation and Simulation*, 88(2):250–268.
- Müller, H.-G., Stadtmüller, U., et al. (2005). Generalized functional linear models. *Annals of Statistics*, 33(2):774–805.
- Nguyen, H. D. and Chamroukhi, F. (2018). Practical and theoretical aspects of mixture-of-experts modeling: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, pages e1246–n/a.
- Nguyen, H. D., Chamroukhi, F., and Forbes, F. (2019). Approximation results regarding the multiple-output Gaussian gated mixture of linear experts model. *Neurocomputing*, 366:208–214.
- Nguyen, H. D., Nguyen, T., Chamroukhi, F., and McLachlan, G. J. (2021a). Approximations of conditional probability density functions in Lebesgue spaces via mixture of experts models. *Journal of Statistical Distributions and Applications*, 8(1):13.
- Nguyen, T., Chamroukhi, F., Nguyen, H. D., and Forbes, F. (2021b). A non-asymptotic model selection in block-diagonal mixture of polynomial experts models. *arXiv:2104.08959*.
- Nguyen, T., Nguyen, H. D., Chamroukhi, F., and Forbes, F. (2021c). A non-asymptotic penalization criterion for model selection in mixture of experts models. *arXiv preprint arXiv:2104.02640*.
- Nguyen, T., Nguyen, H. D., Chamroukhi, F., and Forbes, F. (2021d). A non-asymptotic penalization criterion for model selection in mixture of experts models. *arXiv preprint arXiv:2104.02640*.
- Nguyen, T., Nguyen, H. D., Chamroukhi, F., and McLachlan, G. J. (2020). An  $l_1$ -oracle inequality for the lasso in mixture-of-experts regression models. *arXiv preprint arXiv:2009.10622*.
- Qiao, X., Guo, S., and James, G. M. (2019). Functional graphical models. *Journal of the American Statistical Association*, 114(525):211–222.
- Ramsay, J. O. and Silverman, B. W. (2002). *Applied Functional Data Analysis: Methods and Case Studies*. Springer Series in Statistics. Springer, Berlin, Heidelberg.

- Ramsay, J. O. and Silverman, B. W. (2005). *Functional Data Analysis*. Springer Series in Statistics. Springer.
- Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.
- Städler, N., Bühlmann, P., and Van De Geer, S. (2010).  $\ell_1$ -penalization for mixture regression models. *Test*, 19(2):209–256.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288.
- Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108.
- Wu, C. F. J. (1983). On the convergence properties of the em algorithm. *The Annals of Statistics*, 11(1):95–103.
- Xu, L., Jordan, M., and Hinton, G. E. (1994). An alternative model for mixtures of experts. *Advances in neural information processing systems*, 7:633–640.
- Yao, F., Fu, Y., and Lee, T. C. M. (2010). Functional mixture regression. *Biostatistics*, 12(2):341–353.
- Yuksel, S. E., Wilson, J. N., and Gader, P. D. (2012). Twenty years of mixture of experts. *IEEE Trans. Neural Netw. Learning Syst.*, 23(8):1177–1193.
- Élodie Brunel, Mas, A., and Roche, A. (2016). Non-asymptotic adaptive prediction in functional linear models. *Journal of Multivariate Analysis*, 143:208–232.

## A EM for the FME model

The FME model can be fitted by iteratively maximizing the observed-data log-likelihood (14) iteratively via the EM algorithm. For FME, the EM takes the following form. The complete-data log-likelihood upon which the EM principle is constructed is defined by

$$\log L_c(\Psi) = \sum_{i=1}^n \sum_{k=1}^K Z_{ik} \log [\pi_k(\mathbf{r}_i; \xi) \phi(y_i; \beta_{k,0} + \boldsymbol{\eta}_k^\top \mathbf{x}_i, \sigma_k^2)], \quad (42)$$

$Z_{ik}$  being an indicator binary-valued variable such that  $Z_{ik} = 1$  if  $Z_i = k$  (i.e., if the  $i$ th pair  $(\mathbf{x}_i, y_i)$  is generated from the  $k$ th expert component) and  $Z_{ik} = 0$ , otherwise.

**E-step** This step computes at each EM iteration  $s$  the expectation of the complete-data log-likelihood (42), given the observed data  $\mathcal{D}$ , and the current parameter vector  $\Psi^{(s)}$ :

$$\begin{aligned} Q(\Psi; \Psi^{(s)}) &= \mathbb{E} [\log L_c(\Psi) | \mathcal{D}; \Psi^{(s)}] \\ &= \sum_{i=1}^n \sum_{k=1}^K \tau_{ik}^{(s)} \log [\pi_k(\mathbf{r}_i; \xi) \phi(y_i; \beta_{k,0} + \boldsymbol{\eta}_k^\top \mathbf{x}_i, \sigma_k^2)], \end{aligned} \quad (43)$$

where  $\tau_{ik}^{(s)} = \mathbb{P}(Z_i = k | y_i, u_i(\cdot); \Psi^{(s)})$  is the conditional probability that the observed pair  $(u_i(\cdot), y_i)$  is generated by the  $k$ th expert. This step therefore only requires the computation of the conditional probabilities  $\tau_{ik}^{(s)}$  as defined by (15).

**M-step** This step updates the value of the parameter vector  $\Psi$  by maximizing the  $Q$ -function (43) with respect to  $\Psi$ , that is  $\Psi^{(s+1)} = \arg \max_{\Psi} Q(\Psi; \Psi^{(s)})$ , via separate maximizations w.r.t. the gating network parameters, and the experts network parameters as follows.

**Updating the the gating network parameters** Updating the the gating network's parameters  $\xi$  consists of maximizing w.r.t.  $\xi$  the following function

$$\begin{aligned} Q(\xi; \Psi^{(s)}) &= \sum_{i=1}^n \sum_{k=1}^K \tau_{ik}^{(s)} \log \pi_k(\mathbf{r}_i; \xi) \\ &= \sum_{i=1}^n \left[ \sum_{k=1}^{K-1} \tau_{ik}^{(s)} (\alpha_{k,0} + \zeta_k^\top \mathbf{r}_i) - \log \left( 1 + \sum_{k'=1}^{K-1} \exp \{ \alpha_{k',0} + \zeta_{k'}^\top \mathbf{r}_i \} \right) \right]. \end{aligned} \quad (44)$$

This consists of a weighted multinomial logistic problem for which there is no closed-form solution. This can be performed by the Newton-Raphson (NR) algorithm which iteratively maximizes (44) according to the procedure (16).

Let us denote by  $\xi_1, \dots, \xi_{K-1}$  the parameter vectors  $(\alpha_{1,0}, \zeta_1^\top)^\top, \dots, (\alpha_{K-1,0}, \zeta_{K-1}^\top)^\top$ . Since there are  $K - 1$  parameter vectors to be estimated, the Hessian matrix  $H(\xi; \Psi^{(s)})$  is

a block-matrix, consists of  $(K - 1) \times (K - 1)$  blocks, in which each block  $H_{k\ell}(\boldsymbol{\xi}; \boldsymbol{\Psi}^{(s)})$ , for  $k, \ell \in [K - 1]$ , is given by:

$$H_{k\ell}(\boldsymbol{\xi}; \boldsymbol{\Psi}^{(s)}) = \frac{\partial^2 Q(\boldsymbol{\xi}; \boldsymbol{\Psi}^{(s)})}{\partial \boldsymbol{\xi}_k \partial \boldsymbol{\xi}_\ell^\top} = - \sum_{i=1}^n \pi_k(\mathbf{r}_i; \boldsymbol{\xi}^{(t)}) [\delta_{k\ell} - \pi_\ell(\mathbf{r}_i; \boldsymbol{\xi}^{(t)})] \mathbf{r}_i \mathbf{r}_i^\top,$$

where  $\delta_{k\ell}$  is the Kronecker symbol ( $\delta_{k\ell} = 1$  if  $k = \ell$ , 0 otherwise). The gradient vector consists of  $K - 1$  gradients corresponding to the vectors  $\boldsymbol{\xi}_k$ , for  $k \in [K - 1]$ , and is given by

$$g(\boldsymbol{\xi}; \boldsymbol{\Psi}^{(s)}) = \frac{\partial Q(\boldsymbol{\xi}; \boldsymbol{\Psi}^{(s)})}{\partial \boldsymbol{\xi}} = [g_1(\boldsymbol{\xi}^{(t)}), \dots, g_{K-1}(\boldsymbol{\xi}^{(t)})]^\top,$$

where, for  $k \in [K - 1]$ ,  $g_k(\boldsymbol{\xi}^{(t)}) = \frac{\partial Q(\boldsymbol{\xi}; \boldsymbol{\Psi}^{(s)})}{\partial \boldsymbol{\xi}_k} = \sum_{i=1}^n [\tau_{ik}^{(s)} - \pi_k(\boldsymbol{\xi}^{(t)}; \mathbf{r}_i)] \mathbf{r}_i^\top$ .

**Updating the experts network parameters** Updating the experts network's parameters  $\boldsymbol{\theta}_k$  consists of maximizing the function  $Q(\boldsymbol{\theta}_k; \boldsymbol{\Psi}^{(s)})$  given by

$$\begin{aligned} Q(\boldsymbol{\theta}_k; \boldsymbol{\Psi}^{(s)}) &= \sum_{i=1}^n \tau_{ik}^{(s)} \log \phi(y_i; \beta_{k,0} + \boldsymbol{\eta}_k^\top \mathbf{x}_i, \sigma_k^2) \\ &= -\frac{1}{2\sigma_k^2} \sum_{i=1}^n \tau_{ik}^{(s)} [y_i - (\beta_{k,0} + \boldsymbol{\eta}_k^\top \mathbf{x}_i)]^2 - \frac{n}{2} \log(2\pi\sigma_k^2). \end{aligned}$$

Thus, updating  $\boldsymbol{\theta}_k = (\beta_{k,0}, \boldsymbol{\eta}_k^\top, \sigma_k^2)^\top$ , consists of a weighted Gaussian regression problem where the weights are the conditional memberships  $\tau_{ik}^{(s)}$ , and the updates are given by (17).

## B EM-Lasso for $\ell_1$ -regularized MLE of the FME model

The EM-Lasso algorithm for the maximization of (18) firstly requires the construction of the penalized complete-data log-likelihood

$$\mathcal{L}_c(\boldsymbol{\Psi}) = \log L_c(\boldsymbol{\Psi}) - \text{Pen}_{\lambda, \chi}(\boldsymbol{\Psi}) \quad (45)$$

where  $\log L_c(\boldsymbol{\Psi})$  is the non-regularized complete-data log-likelihood defined by (42). Thus, the EM algorithm for the FME model is implemented as follows. After starting with an initial solution  $\boldsymbol{\Psi}^{(0)}$ , it alternates between the two following steps, until convergence (when there is no longer a significant change in the values of the penalized log-likelihood (18)).

**E-step.** This step computes the expectation of the complete-data log-likelihood (45), given the observed data  $\mathcal{D}$ , using the current parameter vector  $\boldsymbol{\Psi}^{(s)}$ :

$$\mathcal{Q}_{\lambda, \chi}(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(s)}) = \mathbb{E}[\mathcal{L}_c(\boldsymbol{\Psi}) | \mathcal{D}; \boldsymbol{\Psi}^{(s)}] = Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(s)}) - \text{Pen}_{\lambda, \chi}(\boldsymbol{\Psi}), \quad (46)$$

which only requires the computation of the posterior probabilities of component membership  $\tau_{ik}^{(s)}$  ( $i \in [n]$ ), for each of the  $K$  experts as defined by (15).

**M-step.** This step updates the value of the parameter vector  $\Psi$  by maximizing the  $Q$ -function (46) with respect to  $\Psi$ , that is, by computing the parameter vector update

$$\Psi^{(s+1)} = \arg \max_{\Psi} Q_{\lambda, \chi}(\Psi; \Psi^{(s)}). \quad (47)$$

The maximization is performed by separate maximizations w.r.t. the gating network parameters and the experts network parameters.

## B.1 Updating the gating network parameters

Updating the gating network parameters at the  $s$ th EM iteration consists of maximizing the following function

$$Q_{\chi}(\xi; \Psi^{(s)}) = Q(\xi; \Psi^{(s)}) - \chi \sum_{k=1}^{K-1} \|\zeta_k\|_1,$$

with

$$Q(\xi; \Psi^{(s)}) = \sum_{i=1}^n \left[ \sum_{k=1}^{K-1} \tau_{ik}^{(s)} (\alpha_{k,0} + \zeta_k^{\top} \mathbf{r}_i) - \log \left( 1 + \sum_{k'=1}^{K-1} \exp\{\alpha_{k',0} + \zeta_{k'}^{\top} \mathbf{r}_i\} \right) \right]$$

where  $\xi = (\alpha_{1,0}, \zeta_1^{\top}, \dots, \alpha_{K-1,0}, \zeta_{K-1}^{\top})^{\top} \in \mathbb{R}^{(q+1)(K-1)}$  is the gating network parameter vector and  $\Psi^{(s)}$  is the current estimation of model's parameters. One can see this is equivalent to solving a weighted regularized multinomial logistic problem for which  $Q_{\chi}(\xi; \Psi^{(s)})$  is its penalized log-likelihood. There is no closed-form solution for this kind of problem. We then use an iterative optimization algorithm to seek for a maximizer of  $Q_{\chi}(\xi; \Psi^{(s)})$ , i.e., an update for the parameters of the gating network. The idea is to update only a single gate at a time, while fixing other gate's parameters to their previous estimates. Again, to update that single gate, we only update one component at a time, while fixing the other components to their previous values. This procedure for updating the gating network parameters is supported by the methodology of coordinate ascent algorithm: if the objective function consists of a concave, differentiable function and a sum of concave functions then the maximizer can be achieved by iteratively maximizing with respect to each coordinate direction at a time.

**Coordinate ascent for updating the gating network.** Suppose at the  $s$ th EM iteration, we wish to update the gates one by one such that it maximizes  $Q_{\chi}(\xi; \Psi^{(s)})$ . To do that, we create an outer loop, indexed by  $t$ , which cycles over the gates. For each single gate, say gate  $k$ , we partially approximate the smooth part of  $Q_{\chi}(\xi; \Psi^{(s)})$  with respect to  $(\alpha_{k,0}, \zeta_k)$  at  $\xi^{(t)}$ , then optimize the obtained objective function (with respect to  $(\alpha_{k,0}, \zeta_k)$ ) by solving a penalized weighted least square problem using coordinate ascent algorithm. Note that  $\xi^{(t)}$  denotes the current value of  $\xi$  at the iteration  $t$ th of the outer loop, while  $\xi^{(s)}$  is the value of  $\xi$  before entering the outer loop.

In particular, using Taylor expansion, one has a quadratic approximation for smooth part of  $Q_{\chi}(\xi; \Psi^{(s)})$  with respect to  $(\alpha_{k,0}, \zeta_k)$  at  $\xi^{(t)}$  given by

$$l_k(\alpha_{k,0}, \zeta_k) = -\frac{1}{2} \sum_{i=1}^n w_{ik} (c_{ik} - \alpha_{k,0} - \mathbf{r}_i^{\top} \zeta_k)^2 + C(\xi^{(t)}),$$

where

$$\begin{aligned} w_{ik} &= \pi_k(\boldsymbol{\xi}^{(t)}; \mathbf{r}_i) [1 - \pi_k(\boldsymbol{\xi}^{(t)}; \mathbf{r}_i)], \quad (\text{weights}) \\ c_{ik} &= \alpha_{k,0}^{(t)} + \mathbf{r}_i^\top \boldsymbol{\zeta}_k^{(t)} + \frac{\tau_{ik}^{(s)} - \pi_k(\boldsymbol{\xi}^{(t)}; \mathbf{r}_i)}{w_{ik}}, \quad (\text{working response}) \end{aligned}$$

and  $C(\boldsymbol{\xi}^{(t)})$  is a function of  $\boldsymbol{\xi}^{(t)}$ . After calculating the partial quadratic approximation  $l_k(\alpha_{k,0}, \boldsymbol{\zeta}_k)$  about the current estimator  $\boldsymbol{\xi}^{(t)}$ , we then solve the following penalized weighted least square problem

$$\max_{(\alpha_{k,0}, \boldsymbol{\zeta}_k)} l_k(\alpha_{k,0}, \boldsymbol{\zeta}_k) - \chi \|\boldsymbol{\zeta}_k\|_1, \quad \chi > 0, \quad (48)$$

to obtain an update for the parameters of gate  $k$ .

As mentioned above, this problem could be solved by coordinate ascent algorithm. This means we will create an inner loop, indexed by  $m$ , cycles over the components of  $(\alpha_{k,0}, \boldsymbol{\zeta}_k)$  and update them one by one until the objective function of (48) does not gain any significant increase. For each  $j \in [q]$ , using the soft-thresholding operator (see [Hastie et al. \(2015\)](#), sec. 5.4), one can obtain the closed form update for  $\zeta_{kj}$  as follows

$$\zeta_{kj}^{(m+1)} = \frac{\mathcal{S}_\chi\left(\sum_{i=1}^n w_{ik} r_{ij} (c_{ik} - \tilde{c}_{ikj}^{(m)})\right)}{\sum_{i=1}^n w_{ik} r_{ij}^2},$$

in which  $\tilde{c}_{ikj}^{(m)} = \alpha_{k,0}^{(m)} + \mathbf{r}_i^\top \boldsymbol{\zeta}_k^{(m)} - \zeta_{kj}^{(m)} r_{ij}$  is the fitted value excluding the contribution from  $r_{ij}$ ,  $\mathcal{S}_\chi(\cdot)$  is a soft-thresholding operator defined by  $\mathcal{S}_\chi(u) = \text{sign}(u)(|u| - \chi)_+$  and  $(v)_+$  a shorthand for  $\max\{v, 0\}$ . Note that at each iteration of the inner loop, only one component is updated while the others are kept to their previous values, that means  $\zeta_{kh}^{(m+1)} = \zeta_{kh}^{(m)}$  for all  $h \neq j$ . For  $\alpha_{k,0}$ , the closed-form update is given by

$$\alpha_{k,0}^{(m+1)} = \frac{\sum_{i=1}^n w_{ik} (c_{ik} - \mathbf{r}_i^\top \boldsymbol{\zeta}_k^{(m+1)})}{\sum_{i=1}^n w_{ik}}.$$

Once the inner loop converges, the new values of  $(\alpha_{k,0}, \boldsymbol{\zeta}_k)$  are used for the updating procedure of the next gate. When all the gates have their new values, i.e., after  $K - 1$  inner loops, we perform a backtracking line search before actually updating the gating network's parameters for the next  $t$ -indexed iteration. More precisely, the update is  $\boldsymbol{\xi}^{(t+1)} = (1 - \nu)\boldsymbol{\xi}^{(t)} + \nu\bar{\boldsymbol{\xi}}^{(t)}$  where  $\bar{\boldsymbol{\xi}}^{(t)}$  is the output after  $K - 1$  inner loops and  $\nu$  is backtrackingly determined to ensure  $\mathcal{Q}_\chi(\boldsymbol{\xi}^{(t+1)}; \boldsymbol{\Psi}^{(s)}) > \mathcal{Q}_\chi(\boldsymbol{\xi}^{(t)}; \boldsymbol{\Psi}^{(s)})$ .

We keep running the  $t$ -indexed loop until convergence, i.e., when there is no significant relative variation in  $\mathcal{Q}_\chi(\boldsymbol{\xi}; \boldsymbol{\Psi}^{(s)})$ . Once  $\alpha_{k,0}$  and  $\zeta_{kj}$  reach their optimal values  $\tilde{\alpha}_{k,0}$  and  $\tilde{\zeta}_{kj}$  for all  $k \in [K - 1], j \in [q]$ , the update for gating network's parameters is then  $\boldsymbol{\xi}^{(s+1)} = (\tilde{\alpha}_{1,0}, \tilde{\boldsymbol{\zeta}}_1^\top, \dots, \tilde{\alpha}_{K-1,0}, \tilde{\boldsymbol{\zeta}}_{K-1}^\top)^\top$ .

## B.2 Updating the experts network parameters

The maximization step for updating the expert parameters  $\boldsymbol{\theta}_k$  consists of maximizing the function  $\mathcal{Q}_\lambda(\boldsymbol{\theta}_k; \boldsymbol{\Psi}^{(s)})$  given by

$$\mathcal{Q}_\lambda(\boldsymbol{\theta}_k; \boldsymbol{\Psi}^{(s)}) = Q(\boldsymbol{\theta}_k; \boldsymbol{\Psi}^{(s)}) - \lambda \|\boldsymbol{\eta}_k\|_1,$$

with

$$Q(\boldsymbol{\theta}_k; \boldsymbol{\Psi}^{(s)}) = -\frac{1}{2\sigma_k^2} \sum_{i=1}^n \tau_{ik}^{(s)} [y_i - (\beta_{k,0} + \boldsymbol{\eta}_k^\top \mathbf{x}_i)]^2 - \frac{n}{2} \log(2\pi\sigma_k^2),$$

where  $\boldsymbol{\theta}_k = (\beta_{k,0}, \boldsymbol{\eta}_k^\top, \sigma_k^2)^\top \in \mathbb{R}^{p+2}$  is the unknown vector and  $\boldsymbol{\Psi}^{(s)}$  is the current estimation of model's parameters. There is no closed-form solution for this optimization problem, we then solve it by an iterative optimization algorithm similarly to updating the gating network parameters. We first perform the update for  $(\beta_{k,0}, \boldsymbol{\eta}_k)$  while fixing  $\sigma_k^2$ . This corresponds to solving a weighted LASSO problem where the weights are the posterior experts memberships  $\tau_{ik}^{(s)}$ . Once  $(\beta_{k,0}, \boldsymbol{\eta}_k)$  has new value, the variance  $\sigma_k^2$  is updated straightforwardly by the standard update of a weighted Gaussian regression.

More specifically, when  $\sigma_k^2$ , the variance of expert  $k$ , is fixed to  $\sigma_k^{2(s)}$ , updating  $(\beta_{k,0}, \boldsymbol{\eta}_k)$  consists of solving the following weighted LASSO problem:

$$\max_{(\beta_{k,0}, \boldsymbol{\eta}_k)} -\frac{1}{2\sigma_k^{2(s)}} \sum_{i=1}^n \tau_{ik}^{(s)} [y_i - (\beta_{k,0} + \boldsymbol{\eta}_k^\top \mathbf{x}_i)]^2 - \frac{n}{2} \log(2\pi\sigma_k^{2(s)}) - \lambda \sum_{j=1}^q |\boldsymbol{\eta}_{kj}|, \quad \lambda > 0, \quad (49)$$

which can be solved by coordinate ascent algorithm. For each  $j \in [p]$ , the closed-form update for  $\boldsymbol{\eta}_{kj}$  is given by

$$\boldsymbol{\eta}_{kj}^{(m+1)} = \frac{\mathcal{S}_{\lambda\sigma_k^{2(s)}}\left(\sum_{i=1}^n \tau_{ik}^{(s)} \mathbf{x}_{ij}(y_i - \tilde{y}_{ij}^{(m)})\right)}{\sum_{i=1}^n \tau_{ik}^{(s)} \mathbf{x}_{ij}^2},$$

in which  $\tilde{y}_{ij}^{(m)} = \beta_{k,0}^{(m)} + \mathbf{x}_i^\top \boldsymbol{\eta}_k^{(m)} - \boldsymbol{\eta}_{kj}^{(m)} \mathbf{x}_{ij}$  is the fitted value excluding the contribution from  $\mathbf{x}_{ij}$  and  $\mathcal{S}_\lambda(\cdot)$  is the soft-thresholding operator (see [Hastie et al. \(2015\)](#), sec. 5.4). Here  $m$  denotes the  $m$ th iteration of the coordinate ascent algorithm. The update for  $\beta_{k,0}$  is

$$\beta_{k,0}^{(m+1)} = \frac{\sum_{i=1}^n \tau_{ik}^{(s)} (y_i - \mathbf{x}_i^\top \boldsymbol{\eta}_k^{(m+1)})}{\sum_{i=1}^n \tau_{ik}^{(s)}}.$$

We keep updating the components of  $(\beta_{k,0}, \boldsymbol{\eta}_k)$  cyclically until the change in objective function of (49) is small enough. So, the update for  $(\beta_{k,0}, \boldsymbol{\eta}_k)$  in this EM iteration is then  $(\beta_{k,0}^{(s+1)}, \boldsymbol{\eta}_k^{(s+1)}) = (\beta_{k,0}^*, \boldsymbol{\eta}_k^*)$  where the latter is the optimal solution of (49). Finally, the update for  $\sigma_k^2$  is given by

$$\sigma_k^{2(s+1)} = \frac{\sum_{i=1}^n \tau_{ik}^{(s)} (y_i - \beta_{k,0}^{(s+1)} - \mathbf{x}_i^\top \boldsymbol{\eta}_k^{(s+1)})^2}{\sum_{i=1}^n \tau_{ik}^{(s)}}.$$

Hence, the update for the value of parameters vector  $\boldsymbol{\Psi}$  at M-step, i.e., the solution to problem (47), is  $\boldsymbol{\Psi}^{(s+1)} = (\boldsymbol{\xi}^{(s+1)}, \boldsymbol{\theta}_1^{(s+1)}, \dots, \boldsymbol{\theta}_K^{(s+1)})$  where  $\boldsymbol{\xi}^{(s+1)}$  and  $\boldsymbol{\theta}_k^{(s+1)}, k \in [K]$ , are solved by maximizing  $\mathcal{Q}_\chi(\boldsymbol{\xi}; \boldsymbol{\Psi}^{(s)})$  and  $\mathcal{Q}_\lambda(\boldsymbol{\theta}_k; \boldsymbol{\Psi}^{(s)})$ , respectively, using the algorithms described above. The EM algorithm monotonically increases (18). Furthermore, the sequence of parameter estimates generated by the EM algorithm converges toward a local maximum of the log-likelihood function ([Dempster et al., 1977](#); [McLachlan and Krishnan, 2008](#); [Wu, 1983](#)).

## C EM-iFME for updating iFME model parameters

### C.1 Updating the gating network parameters

This section presents the using of Dantzig selector to solve problem (37). Let us simplify the subscript  $k$  in the notations and rewrite the problem under matrix form as follows

$$\begin{aligned} \max_{\tilde{\omega} \in \mathbb{R}^{2q+1}} \quad & -\frac{1}{2} \|\mathbf{c}_w - \mathbf{X}_w \tilde{\omega}\|_2^2 - \chi \|\Omega \tilde{\omega}\|_1 \\ \text{subject to} \quad & \mathbf{A} \tilde{\omega} = \mathbf{0}_q, \end{aligned} \quad (50)$$

where  $\tilde{\omega} = (\alpha_0, \boldsymbol{\omega}^{[d_1]^\top}, \boldsymbol{\omega}^{[d_2]^\top})^\top$  is the unknown coefficients vector,  $\mathbf{c}_w = (\sqrt{w_1}c_1, \dots, \sqrt{w_n}c_n)^\top$  is the weighted working response vector,  $\mathbf{X}_w = [\sqrt{w}|\mathbf{S}_w|\mathbf{0}_{n \times q}] \in \mathbb{R}^{2q+1}$  is the weighted design matrix,  $\Omega = \text{diag}(0, \mathbf{1}_q^\top, \varrho \mathbf{1}_q^\top)$  is the diagonal weighting matrix and  $\mathbf{A} = [\mathbf{0}_q | \mathbf{A}_q^{[d_2]} \mathbf{A}_q^{[d_1]^{-1}} | -\mathbf{I}_q]$  is the constraints matrix. Here,  $\sqrt{w} = (\sqrt{w_1}, \dots, \sqrt{w_n})^\top$ ,  $\mathbf{S}_w = [\sqrt{w_1}\mathbf{s}_1, \dots, \sqrt{w_n}\mathbf{s}_n]^\top$ , with  $\mathbf{s}_i$  are the design vectors (see (28)),  $\mathbf{0}_{n \times q} \in \mathbb{R}^{n \times q}$  contains 0's,  $\mathbf{0}_q \in \mathbb{R}^q$  contains 0's,  $\mathbf{1}_q \in \mathbb{R}^q$  contains 1's and  $\mathbf{I}_q$  is the identity matrix in  $\mathbb{R}^{q \times q}$ . This problem can be viewed as the problem of finding a sparse solution via Lasso estimate for a linear regression model with constraints. Therefore, we can solve it alternatively by Dantzig selector estimate as the solution to the following problem

$$\begin{aligned} \max_{\tilde{\omega} \in \mathbb{R}^{2q+1}} \quad & -\|\Omega \tilde{\omega}\|_1 \\ \text{subject to} \quad & \begin{cases} |\mathbf{X}_w^\top (\mathbf{c}_w - \mathbf{X}_w \tilde{\omega})| \leq \chi \mathbf{1}_{2q+1}, \\ \mathbf{A} \tilde{\omega} = \mathbf{0}_q, \end{cases} \end{aligned}$$

where the absolute value operator is understood componentwise. By decomposing  $\tilde{\omega}$  into its positive and negative parts,  $\tilde{\omega} = \tilde{\omega}_+ - \tilde{\omega}_-$ , the above problem becomes

$$\begin{aligned} \max_{(\tilde{\omega}_+, \tilde{\omega}_-) \in \mathbb{R}^{4q+2}} \quad & -[0, \mathbf{1}_q^\top, \varrho \mathbf{1}_q^\top, 0, \mathbf{1}_q^\top, \varrho \mathbf{1}_q^\top] \begin{bmatrix} \tilde{\omega}_+ \\ \tilde{\omega}_- \end{bmatrix} \\ \text{subject to} \quad & \begin{cases} \begin{bmatrix} \mathbf{X}_w^\top \mathbf{X}_w & -\mathbf{X}_w^\top \mathbf{X}_w \\ -\mathbf{X}_w^\top \mathbf{X}_w & \mathbf{X}_w^\top \mathbf{X}_w \end{bmatrix} \begin{bmatrix} \tilde{\omega}_+ \\ \tilde{\omega}_- \end{bmatrix} \leq \begin{bmatrix} \chi + \mathbf{X}_w^\top \mathbf{c}_w \\ \chi - \mathbf{X}_w^\top \mathbf{c}_w \end{bmatrix}, \\ \begin{bmatrix} \mathbf{A} & -\mathbf{A} \end{bmatrix} \begin{bmatrix} \tilde{\omega}_+ \\ \tilde{\omega}_- \end{bmatrix} = \mathbf{0}_{4q+2}, \\ \tilde{\omega}_+ \geq \mathbf{0}_{2q+1}, \quad \tilde{\omega}_- \geq \mathbf{0}_{2q+1}, \end{cases} \end{aligned} \quad (51)$$

which is a standard linear program with  $4q + 2$  variables, therefore can be easily solved by available toolboxes for linear programming, e.g., Matlab's `linprog` function. Finally, the solution  $\tilde{\omega}^* = (\alpha_0^*, \boldsymbol{\omega}^{[d_1]*^\top}, \boldsymbol{\omega}^{[d_2]*^\top})^\top$  to the original problem could be retrieved by the relation  $\tilde{\omega}^* = \tilde{\omega}_+^* - \tilde{\omega}_-^*$  where  $(\tilde{\omega}_+^{*\top}, \tilde{\omega}_-^{*\top})^\top$  is the solution of (51).

### C.2 Updating the expert network parameters

This section presents how to solve problem (38). Firstly, we fix  $\sigma_k^2$  to its previous estimate and perform an update for  $(\beta_{k,0}, \boldsymbol{\gamma}_k^{[d_1]})$ , it corresponds to solving a penalized weighted least square

problem with constraints. From now on, let us simplify the subscript  $k$  in the notations and rewrite the problem under matrix form as follows

$$\begin{aligned} \max_{\tilde{\gamma} \in \mathbb{R}^{2p+1}} \quad & -\frac{1}{2} \|\mathbf{y}_{\sigma\tau} - \mathbf{X}_{\sigma\tau} \tilde{\gamma}\|_2^2 - \lambda \|\Lambda \tilde{\gamma}\|_1 - \frac{n_k}{2} \log(2\pi\sigma^2) \\ \text{subject to} \quad & \mathbf{A} \tilde{\gamma} = \mathbf{0}_p, \end{aligned}$$

where  $\tilde{\gamma} = (\beta_0, \gamma^{[d_1]^\top}, \gamma^{[d_2]^\top})^\top$  is the unknown coefficients vector,  $\mathbf{y}_{\sigma\tau} = (\sigma\sqrt{\tau_1}y_1, \dots, \sigma\sqrt{\tau_n}y_n)^\top \in \mathbb{R}^n$  is the weighted response vector,  $\mathbf{X}_{\sigma\tau} = \sigma \odot [\sqrt{\tau}|\mathbf{V}_\tau| \mathbf{0}_{n \times p}] \in \mathbb{R}^{n \times (2p+1)}$  is the weighted design matrix,  $\Lambda = \text{diag}(0, \mathbf{1}_p^\top, \rho \mathbf{1}_p^\top)$  is the diagonal weighting matrix,  $n_k = \sum_{i=1}^n \tau_{ik}$  and  $\mathbf{A} = [\mathbf{0}_p | \mathbf{A}_p^{[d_2]} \mathbf{A}_p^{[d_1]^{-1}} | -\mathbf{I}_p] \in \mathbb{R}^{p \times (2p+1)}$  is the constraints matrix. Here,  $\sqrt{\tau} = (\sqrt{\tau_1}, \dots, \sqrt{\tau_n})^\top \in \mathbb{R}^n$ ,  $\mathbf{V}_\tau = [\sqrt{\tau_1} \mathbf{v}_1, \dots, \sqrt{\tau_n} \mathbf{v}_n]^\top \in \mathbb{R}^{n \times p}$ , with  $\mathbf{v}_i$  are the design vectors (see (28)),  $\mathbf{0}_{n \times p} \in \mathbb{R}^{n \times p}$  contains 0's,  $\mathbf{0}_p \in \mathbb{R}^p$  contains 0's,  $\mathbf{1}_p \in \mathbb{R}^p$  contains 1's and  $\mathbf{I}_p$  is the identity matrix in  $\mathbb{R}^{p \times p}$ . As the last term in the objective function is independent of  $\tilde{\gamma}$ , this problem is similar to the problem (50) and then can be solved by Dantzig selector estimate as the solution to the following problem

$$\begin{aligned} \max_{\tilde{\gamma} \in \mathbb{R}^{2p+1}} \quad & -\|\Lambda \tilde{\gamma}\|_1 \\ \text{subject to} \quad & \begin{cases} |\mathbf{X}_{\sigma\tau}^\top (\mathbf{y}_{\sigma\tau} - \mathbf{X}_{\sigma\tau} \tilde{\gamma})| \leq \lambda \mathbf{1}_{2p+1}, \\ \mathbf{A} \tilde{\gamma} = \mathbf{0}_p. \end{cases} \end{aligned}$$

Similarly to the problem in the gating network update, by decomposing  $\tilde{\gamma}$  into its positive and negative parts,  $\tilde{\gamma} = \tilde{\gamma}_+ - \tilde{\gamma}_-$ , the above problem becomes

$$\begin{aligned} \max_{(\tilde{\gamma}_+, \tilde{\gamma}_-) \in \mathbb{R}^{4p+2}} \quad & -[0, \mathbf{1}_p^\top, \rho \mathbf{1}_p^\top, 0, \mathbf{1}_p^\top, \rho \mathbf{1}_p^\top] \begin{bmatrix} \tilde{\gamma}_+ \\ \tilde{\gamma}_- \end{bmatrix} \\ \text{subject to} \quad & \begin{cases} \begin{bmatrix} \mathbf{X}_{\sigma\tau}^\top \mathbf{X}_{\sigma\tau} & -\mathbf{X}_{\sigma\tau}^\top \mathbf{X}_{\sigma\tau} \\ -\mathbf{X}_{\sigma\tau}^\top \mathbf{X}_{\sigma\tau} & \mathbf{X}_{\sigma\tau}^\top \mathbf{X}_{\sigma\tau} \end{bmatrix} \begin{bmatrix} \tilde{\gamma}_+ \\ \tilde{\gamma}_- \end{bmatrix} \leq \begin{bmatrix} \lambda + \mathbf{X}_{\sigma\tau}^\top \mathbf{y}_{\sigma\tau} \\ \lambda - \mathbf{X}_{\sigma\tau}^\top \mathbf{y}_{\sigma\tau} \end{bmatrix}, \\ \begin{bmatrix} \mathbf{A} & -\mathbf{A} \end{bmatrix} \begin{bmatrix} \tilde{\gamma}_+ \\ \tilde{\gamma}_- \end{bmatrix} = \mathbf{0}_{4p+2}, \\ \tilde{\gamma}_+ \geq \mathbf{0}_{2p+1}, \quad \tilde{\gamma}_- \geq \mathbf{0}_{2p+1}, \end{cases} \end{aligned} \quad (52)$$

which is a standard linear program with  $4q + 2$  variables and can be solved similarly to the gating network case. The solution  $\tilde{\gamma}^* = (\beta_0^*, \gamma^{[d_1]*^\top}, \gamma^{[d_2]*^\top})^\top$  to the original problem is retrieved by the relation  $\tilde{\gamma}^* = \tilde{\gamma}_+^* - \tilde{\gamma}_-^*$  where  $(\tilde{\gamma}_+^{*\top}, \tilde{\gamma}_-^{*\top})^\top$  is the solution of (52).

Finally, the update for  $\sigma_k^2$  is

$$\sigma_k^{2(s+1)} = \frac{\sum_{i=1}^n \tau_{ik}^{(s)} (y_i - \beta_{k,0}^{(s+1)} - \mathbf{v}_i^\top \gamma_k^{[d_1]^{(s+1)}})^2}{\sum_{i=1}^n \tau_{ik}^{(s)}},$$

in which  $\beta_{k,0}^{(s+1)}, \gamma_k^{[d_1]^{(s+1)}}$  are the new updates for  $\beta_{k,0}$  and  $\gamma_k^{[d_1]}$ .

## D Detailed Data generating protocol, simulation parameters and experimental protocol

### D.1 Data generating protocol

In the simulated data, the data generation protocol is as follows. We consider a  $K$ -component functional mixture of Gaussian experts model that relates a scalar response  $y \in \mathbb{R}$  to a univariate functional predictor  $X(t), t \in \mathcal{T}$  defined on a domain  $\mathcal{T} \subset \mathbb{R}$ . Given the model parameters (defined in the next paragraph)  $\boldsymbol{\beta} = \{\beta_{k,0}, \beta_k(t), \sigma_k^2\}_{k=1}^K$  and  $\boldsymbol{\alpha} = \{\alpha_{k,0}, \alpha_k(t)\}_{k=1}^K$ ,  $t \in \mathcal{T}$ , we first construct the functional predictors  $X_i(\cdot)$  for  $i \in [n]$  using the representation defined in (5), with detailed parameterization in (54). Then, for each  $i \in [n]$ , conditional on the functional predictor  $X_i(\cdot)$ , a hidden categorical random variable  $Z_i \in [K]$  is generated following the multinomial distribution  $\mathcal{M}\left(1, (\pi_1(X_i(t); \boldsymbol{\alpha}), \dots, \pi_K(X_i(t); \boldsymbol{\alpha}))\right)$ , where  $\pi_k(X_i(t); \boldsymbol{\alpha})$  for  $k \in [K]$  is given by (3). Finally, conditional on  $Z_i = z_i$  and  $X_i(\cdot)$ , the scalar response  $Y_i$  is obtained by sampling from the Gaussian distribution with mean  $\beta_{z_i,0} + \int_{\mathcal{T}} X_i(t) \beta_{z_i}(t) dt$  and variance  $\sigma_{z_i}^2$ . The value  $z_i$  is then the true cluster label of the predictor  $X_i(\cdot)$ . This hierarchical generative process can be summarized as follows:

$$\begin{aligned} Y_i | Z_i = z_i, X_i(t) &\sim \mathcal{N}\left(\beta_{z_i,0} + \int_{\mathcal{T}} X_i(t) \beta_{z_i}(t) dt; \sigma_{z_i}^2\right), \\ Z_i | X_i(t) &\sim \mathcal{M}\left(1, (\pi_1(X_i(t); \boldsymbol{\alpha}), \dots, \pi_K(X_i(t); \boldsymbol{\alpha}))\right). \end{aligned} \quad (53)$$

The true generated data, i.e.,  $(X_i(\cdot), Y_i, Z_i)_{i=1}^n$ , are used for evaluating the prediction and clustering performance.

Finally, to mimic real data, in our simulations, the generated predictors  $X_i(\cdot)$  are contaminated with measurement errors, that means we will not use  $X_i(\cdot)$  for analysis, but

$$U_i(t) = X_i(t) + \delta_i(t),$$

with  $\delta_i(t)$  is an independent Gaussian noise with mean zero and constant variance  $\sigma_{\delta}^2$  for all  $t \in \mathcal{T}$ . We considered different noise levels  $\sigma_{\delta}^2$  (see Table 9).

### D.2 Simulation parameters and experimental protocol

The parameters that were used in the data generating process (53) are as follows. We consider  $K = 3$  components and a time domain  $\mathcal{T} = [0, 1]$ .

**Functional experts and gating parameters:** The functional experts parameters are given by

$$\begin{aligned}\beta_1(t) &= \begin{cases} -50(t-0.5)^2 + 4 & \text{if } 0 \leq t < 0.3, \\ 0 & \text{if } 0.3 \leq t < 0.7, \\ 50(t-0.5)^2 - 4 & \text{if } 0.7 \leq t \leq 1, \end{cases} \\ \beta_2(t) &= -\beta_1(t), \\ \beta_3(t) &= 100(t-0.5)^2 - 10, \quad 0 \leq t \leq 1, \\ (\beta_{1,0}, \beta_{2,0}, \beta_{3,0})^\top &= (-5, 0, 5)^\top, \\ (\sigma_1^2, \sigma_2^2, \sigma_3^2)^\top &= (5, 5, 5)^\top,\end{aligned}$$

and the functional gating network parameters are given by

$$\begin{aligned}\alpha_1(t) &= 80(t-0.5)^2 - 8, \\ \alpha_2(t) &= -\alpha_1(t), \quad \alpha_3(t) = \mathbf{0}, \quad 0 \leq t \leq 1, \\ (\alpha_{1,0}, \alpha_{2,0}, \alpha_{3,0})^\top &= (-10, -10, 0)^\top.\end{aligned}$$

Note that to satisfy the identifiability condition (see [Jiang and Tanner \(1999\)](#)), the experts are ordered, for instance  $(\beta_{1,0}, \beta_1, \sigma_1^2) \prec \dots \prec (\beta_{K,0}, \beta_K, \sigma_K^2)$ , and the last gating network parameters,  $\alpha_{K,0}, \alpha_K(t)$ , are initialized, i.e. fixed to zeros. Here, “ $\prec$ ” is the lexicographical order on  $\mathbb{R}^{p+2}$ .

As it can be seen in Figure 11, the expert parameter functions  $\beta_1(t)$  and  $\beta_2(t)$  have a flat region in the interval  $0.3 \leq t < 0.7$ , outside of which they are quadratic, while  $\beta_3(t)$  and the gating parameter functions  $\alpha_1(t)$ ,  $\alpha_2(t)$  are all quadratic on the whole domain. By choosing these functional experts and gating parameters, we can later compare the sparsity in the zeroth and third derivatives of the solutions obtained by the proposed models.

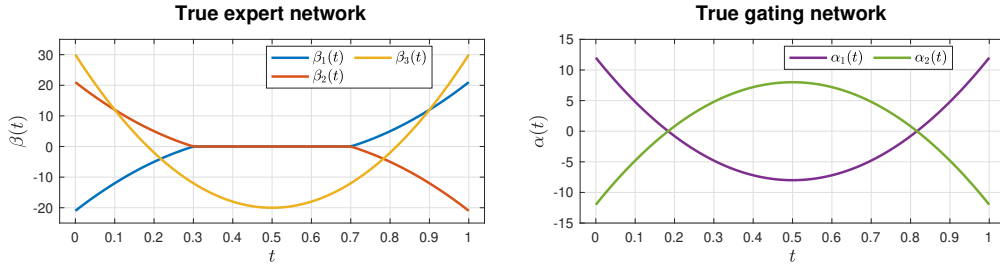


Figure 11: The true expert and gating functions used in simulations.

**Functional predictors parameters:** In this simulation, the functional predictors  $X_i(\cdot)$  are constructed using the following formula,

$$X_i(t) = \mathbf{x}_i^\top \mathbf{b}(t), \quad t \in \mathcal{T}, \quad (54)$$

in which  $\mathbf{b}(t) = [b_1(t), \dots, b_{10}(t)]^\top$  is a 10-dimensional B-spline basis  $\mathbf{x}_i \in \mathbb{R}^{10}$  is a coefficient vector defined as  $\mathbf{x}_i = \mathbf{W} \mathbf{v}_i$ , where  $\mathbf{W} \in \mathbb{R}^{10 \times 10}$  is a matrix of i.i.d. random values from the uniform distribution  $\mathcal{U}(0, 1)$  and  $\mathbf{v}_i \in \mathbb{R}^{10}$  is a vector of i.i.d. random values from the normal distribution  $\mathcal{N}(0, 10)$ . Here, the matrix  $\mathbf{W}$  acts as a factor to facilitate the fluctuation of the generated  $X_i(\cdot)$ .

Scenario	$S1$	$S2$	$S3$	$S4$
$\sigma_\delta^2$	1	1	4	4
$m$	300	100	300	100

Table 9: Simulation scenarios with different noise level  $\sigma_\delta^2$  and curve sampling level  $m$ .

**Noisy functional predictors:** Since in real practical situations we do not usually directly observe the true functional predictors  $X(\cdot)$ 's, but only a noisy and discretized version of them, we thus consider several scenarios with different noise and sampling levels. To this end, in the data generating protocol we consider noisy versions  $U_i(t_j) = X_i(t_j) + \delta_i(t_j)$  of the functional predictors values  $X_i(t_j)$ , where  $\delta_i(t_j) \sim \mathcal{N}(0, \sigma_\delta^2)$  is a centred Gaussian noise with variance  $\sigma_\delta^2$ , for all  $i \in [n]$ ,  $j \in [m]$ . We investigate simulated scenarios  $S1, \dots, S4$  with curve length  $m$  and the noise level  $\sigma_\delta^2$  of the functional predictors, including two levels of sampling  $m \in \{300, 100\}$ , and two levels of noise  $\sigma_\delta^2 \in \{1, 4\}$ . The resulting four considered scenarios of simulated data are presented in Table 9. For each considered scenario, we generate  $N = 100$  datasets, each dataset contains  $n$  pairs of  $(U_i(t), Y_i)$ , with  $n \in \{200, 400, 600, 800, 1000\}$ .