Continuous-time learning of probability distributions via neural ODEs with applications in continuous glucose monitoring data

Antonio Álvarez-López

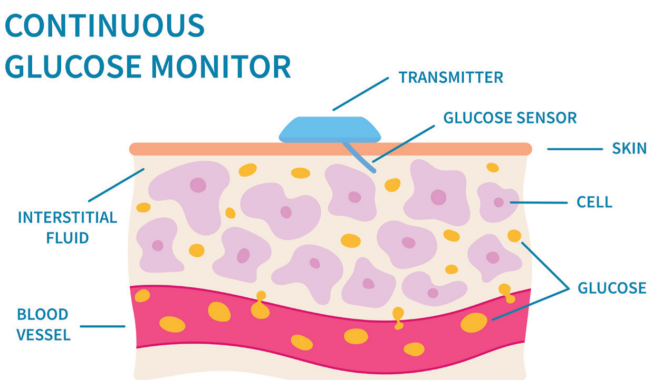
Departamento de Matemáticas,
Universidad Autónoma de Madrid

December 16, 2025

Joint work with...

Marcos Matabuena (Harvard University)
Incoming professor in MBUZAI

Motivation: Digital health



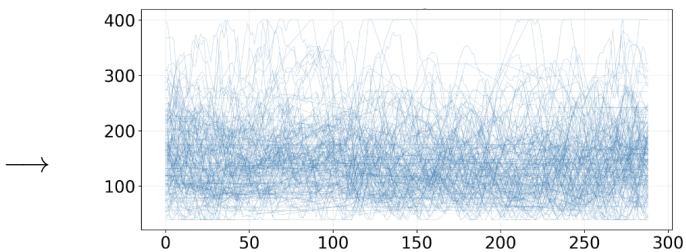
Source: (Messer et al. (2018)).

- Sends a measure every 5 minutes.
- **Advantages:**
  - Fewer finger-stick tests.
  - Alerts for hypo- and hyperglycaemia, or wide fluctuations in glucose levels
  - Treatment by closed-loop insulin pump.
- **Limitations:**
  - Periodic sensor replacement, higher cost.

Data

Continuous Glucose Monitoring produces (irregular) data streams.

	0	5	10	15	20	25	30	35	40	45	50	55	60	65
1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
2	216	222	226	246	250	254	256	250	248	246	242	238	234	230
3	114	114	112	112	110	108	108	108	108	106	104	104	102	100
4	178	180	182	184	186	188	188	190	190	190	192	192	192	186
5	178	174	170	168	168	168	168	168	166	166	164	162	164	166
6	196	196	190	186	190	194	196	196	192	190	186	186	188	190
7	70	74	76	76	76	74	72	74	74	76	78	78	78	78
8	186	186	186	188	188	190	190	192	194	196	198	200	202	204
9	222	220	216	214	212	210	210	210	206	202	202	204	208	210

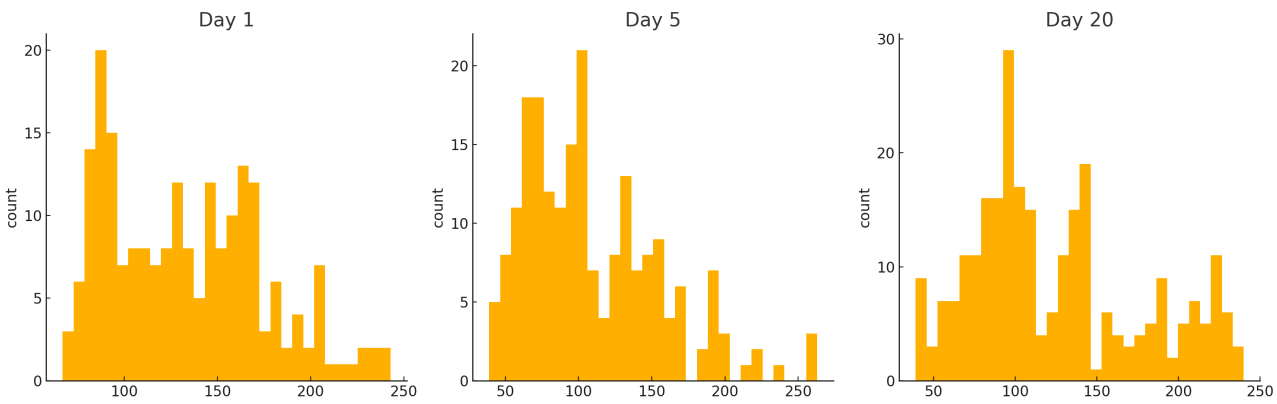


Glucose (mg/dL) vs. Measurement (total: 288/day)

Data source: PEDAP trial public dataset (Wadwa et al., NEJM 2023; Jaeb Center Public Study Websites).

DATA

We build **glucodensities** (Matabuena, Petersen, Vidal, Gude (2021)).



Idea: Treat CGM readings as samples and estimate density of % time spent at each glucose level.

Goals

- **Mathematical:** Model an evolving probability law from samples observed at discrete times.
- **Clinical:** Track glucose distribution in time, reflecting disease progression or treatment efficacy.
- **Data difficulties:**
  - Empirical / discrete distributions from irregular streams with sensor noise.
  - Multimodality: Mixing times of day $\Rightarrow$ different metabolic states.

Goals

- **Mathematical:** Model an evolving probability law from samples observed at discrete times.
- **Clinical:** Track glucose distribution in time, reflecting disease progression or treatment efficacy.
- **Data difficulties:**
  - Empirical / discrete distributions from irregular streams with sensor noise.
  - Multimodality: Mixing times of day $\Rightarrow$ different metabolic states.
- **Existing approaches:**
  - **Classical density/CDF estimation** (KDE, etc.), repeated over time (Chacón (2018))
  - **Deep generative density models** (e.g., normalizing flows) (Papamakarios et al. (2021))
  - **Semiparametric time-varying models** (GAMLSS) (Rigby & Stasinopoulos (2005))
  - **Functional-quantile / multilevel distributional frameworks** (Matabuena et al. (2025))
- **Main limitations:**
  - KDE: bandwidth sensitivity + curse of dimensionality.
  - Flow methods: often hard to interpret clinically.
  - GAMLSS: typically scalar responses and can impose rigid functional forms.
  - Functional-quantile dynamics: interpretable but often restricted to (near-)linear dynamics.

Model

Gaussian mixture with time-evolving weights

$$f_\theta(x,t) = \sum_{s=1}^{K} \alpha_s(t)\, \mathcal{N}(x \,|\, \mu_s, \Sigma_s), \qquad x \in \mathbb{R}^d$$

where $\mu_s \in \mathbb{R}^d$, $\Sigma_s \in \mathbb{S}_+^d$ are fixed; and

$$(\alpha_1, \ldots, \alpha_K) : [0,T] \longrightarrow \Delta_{K-1} \coloneqq \Big\{ \alpha \in \mathbb{R}^K \;\Big|\; \alpha_i \geq 0,\, \sum_i \alpha_i = 1 \Big\},$$

$$\dot{\alpha}(t) = \mathrm{NODE}_\phi\big(\alpha(t), t\big).$$

[1] Universal approximation of neural ODEs for dynamic behavior + Density of Gaussians (N. Wiener, Tauberian theorems, 1932).

Model

Gaussian mixture with time-evolving weights

$$f_\theta(x,t) = \sum_{s=1}^{K} \alpha_s(t)\, \mathcal{N}(x \mid \mu_s, \Sigma_s), \qquad x \in \mathbb{R}^d$$

where $\mu_s \in \mathbb{R}^d$, $\Sigma_s \in \mathbb{S}^d_+$ are fixed; and

$$(\alpha_1, \dots, \alpha_K) : [0,T] \longrightarrow \Delta_{K-1} \coloneqq \Big\{ \alpha \in \mathbb{R}^K \;\Big|\; \alpha_i \geq 0,\ \sum_i \alpha_i = 1 \Big\},$$

$$\dot{\alpha}(t) = \mathrm{NODE}_\phi\big(\alpha(t), t\big).$$

- **Interpretability:** Fixed Gaussian components $(m_s, \Sigma_s) \equiv$ Reference glycaemic states.
- By choosing K large enough, the parametric family of f_θ offers universal approximation[1] over

$$\Big\{ f : \mathbb{R}^d \times [0,T] \to \mathbb{R}_{\geq 0} \;\Big|\; \int_{\mathbb{R}^d} f = 1 \Big\}, \qquad \|f\| \coloneqq \sup_{t \in [0,T]} \|f(\cdot,t)\|_{L^1(\mathbb{R}^d)}.$$

[1] Universal approximation of neural ODEs for dynamic behavior + Density of Gaussians (N. Wiener, Tauberian theorems, 1932).

Two-stage algorithm

1: **Input:** Time series $\{(t_i, X_i)\}_{i=1}^n$, number of Gaussians K
2: Construct aggregated data:

$$X = \bigcup_{i=1}^{n} X_i$$

3: **Initialization:** Run KMeans on X to obtain initial parameters $\{\alpha_s, \mu_s, \sigma_s^2\}_{s=1}^K$
4: **for** $\ell = 1$ to n_{iter} **do**
5: **Global GMM Fitting (Gradient Descent):** On n_{grad} iterations, find

$$\{\mu_s, \sigma_s^2\}_{s=1}^K = \arg\min_{\mu,\sigma^2} \text{MMD}^2\Big(P_X, \sum_{s=1}^{K} \alpha_s \mathcal{N}(\mu_s, \sigma_s^2)\Big)$$

6: **Local Weight Estimation:** For each t_i, compute

$$(\alpha_1^{(i)}, \ldots, \alpha_K^{(i)}) = \arg\min_{\alpha} \text{MMD}^2\Big(P_{X_i}, \sum_{s=1}^{K} \alpha_s \mathcal{N}(\mu_s^*, \sigma_s^{2*})\Big)$$

7: **end for**
8: **Neural ODE Modeling:** Define the weight dynamics

$$\frac{d\alpha(t)}{dt} = f(\alpha(t), \psi), \qquad \alpha = (\alpha_1, \ldots, \alpha_K).$$

9: **Parameter Estimation:** Find

$$\psi^* = \arg\min_{\psi} \sum_{i=1}^{n} \left\| \alpha^{(i)} - \alpha(t_i; \psi) \right\|^2,$$

where $\alpha(t_i; \psi)$ is the solution of the Neural ODE at time t_i.

Why two stages? Joint problem is strongly non-convex ⇒ Single pass converges to poor local minima.
Alternating strategy yields stable updates

3.–7. Discrete-time fit:

- Minimize a discrepancy (MMD^2) for each t_i.
- Yields preliminary weights $\alpha^{(i)}$.

8.–9. Continuous-time fit:

- Fit neural ODE interpolating the weights $\alpha^{(i)}$.
- Enforce temporal smoothness.

Stage 1. Discrete-time fitting: Maximum Mean Discrepancy

Kernel $k\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ symmetric, positive-definite. Common choice is the Gaussian kernel

$$k(x, x') \;=\; \exp\!\Big(-\tfrac{\|x-x'\|^2}{2\sigma^2}\Big), \qquad\qquad \sigma \;=\; \text{median}\big\{\|x_i - x_j\|\big\}_{1\le i<j\le n} \qquad \text{(heuristic pick)}$$

Definition. Let μ, ν probability measures on $\mathbb{R}^d$. Define

$$\mathrm{MMD}^2(\mu,\nu) = \iint k(x,x')\,\mathrm{d}\mu(x)\,\mathrm{d}\mu(x') \;+\iint k(y,y')\,\mathrm{d}\nu(y)\,\mathrm{d}\nu(y') - 2\iint k(x,y)\,\mathrm{d}\mu(x)\,\mathrm{d}\nu(y).$$

MMD is a **distance** if and only if k is **characteristic**[2]; then $\quad \mathrm{MMD} = 0 \iff \mu = \nu.$

[2] $\mu \mapsto \mathbb{E}_\mu[k(x,\cdot)]$ injective. For instance, the Gaussian kernel.

Stage 1. Discrete-time fitting: Maximum Mean Discrepancy

Kernel $k\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ symmetric, positive-definite. Common choice is the Gaussian kernel

$$k(x, x') \;=\; \exp\!\Big(-\tfrac{\|x-x'\|^2}{2\sigma^2}\Big), \qquad\qquad \sigma \;=\; \text{median}\big\{\|x_i - x_j\|\big\}_{1\le i<j\le n} \qquad \text{(heuristic pick)}$$

Definition. Let μ, ν probability measures on $\mathbb{R}^d$. Define

$$\mathrm{MMD}^2(\mu,\nu) = \iint k(x,x')\,\mathrm{d}\mu(x)\,\mathrm{d}\mu(x') \;+\iint k(y,y')\,\mathrm{d}\nu(y)\,\mathrm{d}\nu(y') - 2\iint k(x,y)\,\mathrm{d}\mu(x)\,\mathrm{d}\nu(y).$$

MMD is a **distance** if and only if k is **characteristic**[2]; then $\quad \mathrm{MMD} = 0 \iff \mu = \nu$.

Advantages:

- *Efficient:* For μ mixture and ν discrete, we use a closed-form expression for squared MMD.
- *Robust* to the presence of outliers. Also to non-overlapping supports of μ and ν
- *Differentiable:* gradients back-propagate through k.

[2] $\mu \mapsto \mathbb{E}_\mu[k(x,\cdot)]$ injective. For instance, the Gaussian kernel.

Stage 2. Weight evolution: Neural ODEs

- **Continuous-time model.** Replace discrete network layers with an ODE:

$$\dot{\alpha}(t) = \mathrm{NODE}_\phi\big(\alpha(t), t\big), \qquad \alpha(0) = \alpha_0 \quad \rightarrow \quad \text{Output: } \alpha(t) = \mathsf{ODESolve}(\alpha_0, t_0, t, \mathrm{NODE}_\phi)$$

- **Projection to simplex**:

$$\alpha(t) \leftarrow \alpha(t)/1^\top \alpha(t) \qquad \forall t$$

- **Training loss.**

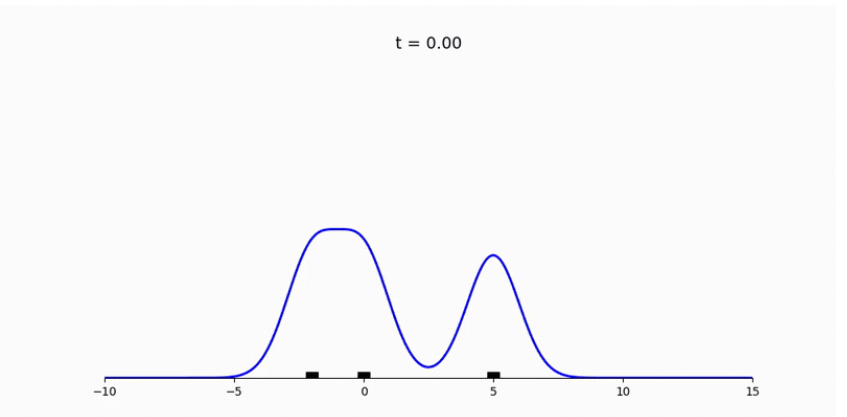
$$\mathcal{L}_{\mathrm{NODE}}(\phi) = \sum_i \big\|\alpha(t_i;\phi) - \alpha^{(i)}\big\|^2 + \nu\|\phi\|^2, \qquad \nu \geq 0 \text{ fixed}$$

  $\nabla_\phi \mathcal{L}_{\mathrm{NODE}}$ computed by adjoint method $\Rightarrow$ constant-memory back-propagation.

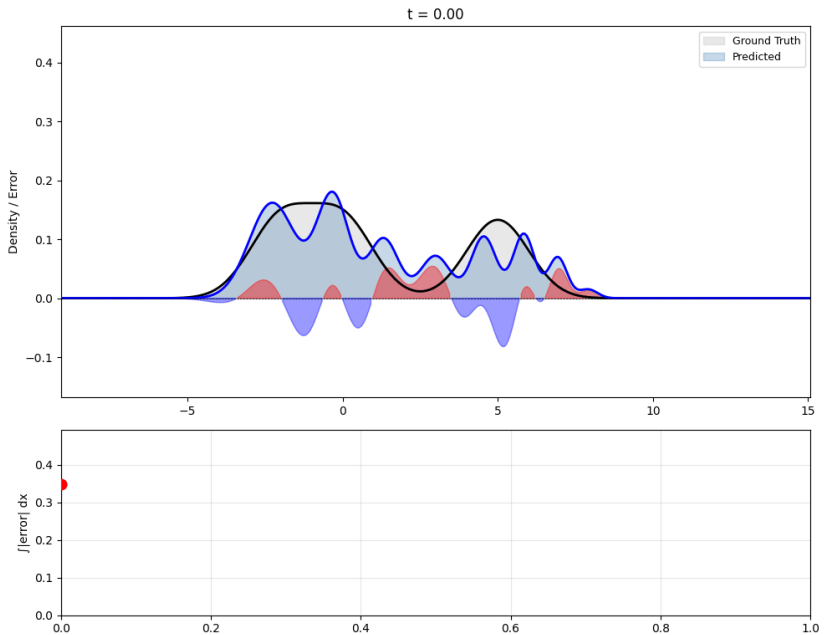
- **Advantages**

  1. Performance in high dimensions.
  2. Parameter-efficient (one vector field = arbitrary depth).
  3. Invertible flow (useful for generative models).
  4. Smooth latent trajectories.

Validation (Synthetic data)



Validation (Synthetic data)



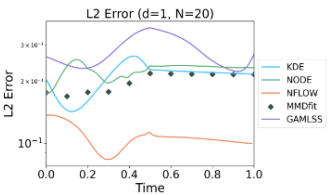
MMD Fitting Parameters

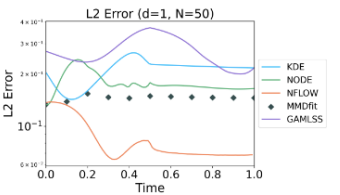
Parameter	Value	Parameter	Value
Max iterations	20	*Rel. tol. param.*	10^{-9}
Rel. tol. err.	10^{-7}	*Abs. tol.*	10^{-6}
Components (K)	10	*Learning rate*	0.01
Grad. steps	10	*Ridge reg.* (λ)	0.1

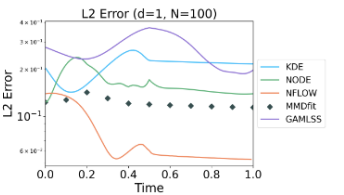
Neural ODE Training Parameters

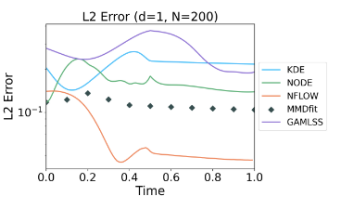
Parameter	Value	Parameter	Value
Seed	42	*Hidden dim*	200
Activation (σ)	ReLU	*Layers*	2
Optimizer	Adam	*Integration time* (T)	1.0
Step size (dt)	0.01	*Integrator*	RK4
Learning rate	10^{-3}	*L2 reg.* (λ)	0.001
Max epochs	2000	*MC samples* (n_{MC})	10000
Rel. tol.	10^{-6}	*Abs. tol.*	10^{-6}

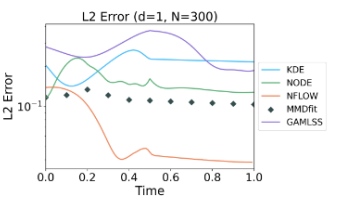
Comparison with other models: $d = 1$



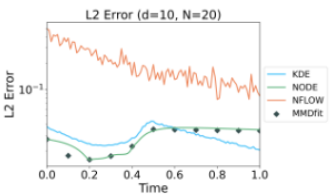


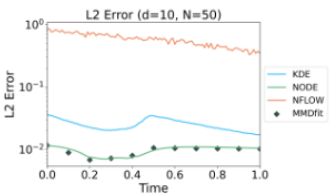


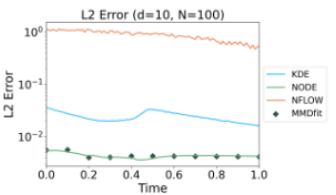


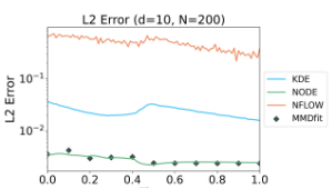


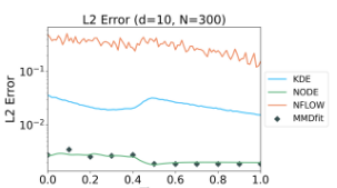
Comparison with other models: $d = 10$



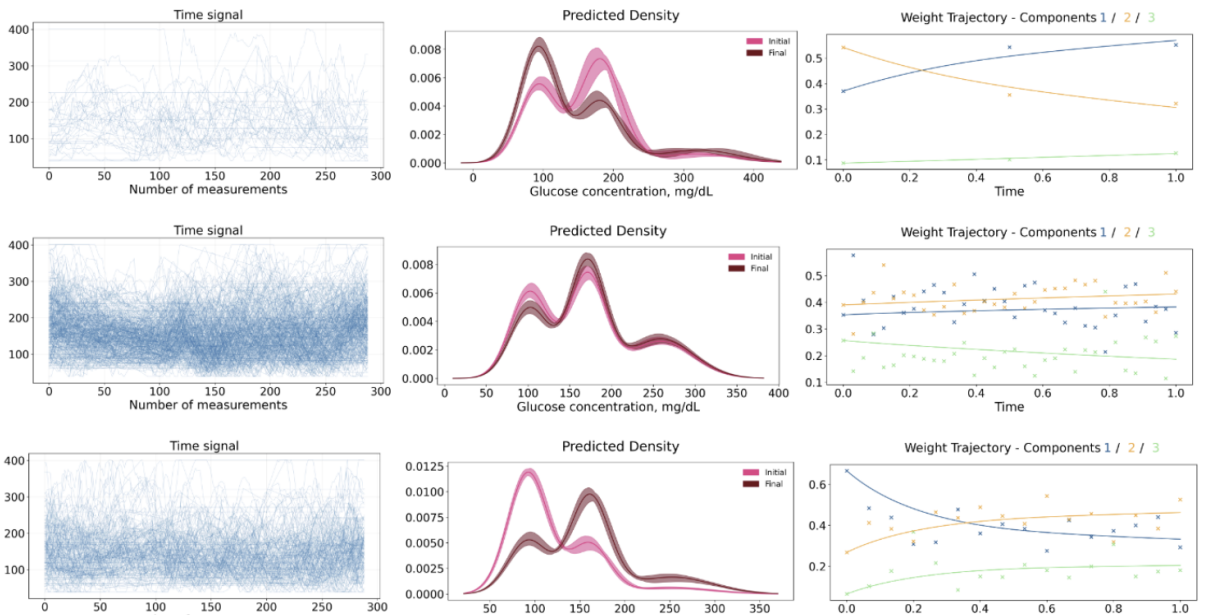




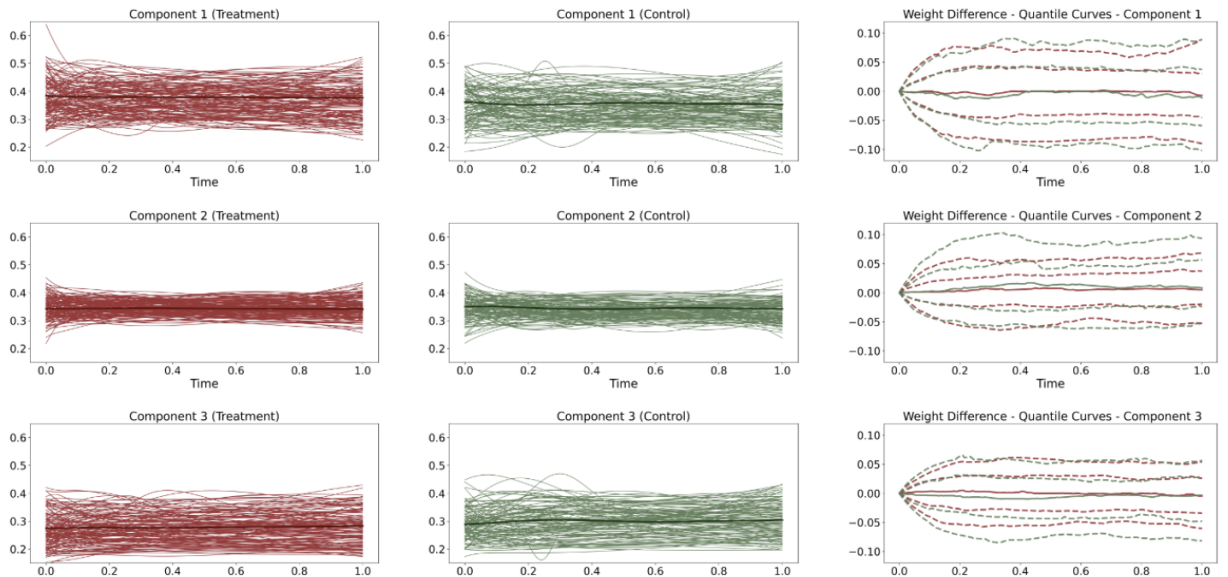




Clinical case study



Clinical case study



Columns 1 and 2: Component weight trajectories in the treatment group (left) and control group (right).
Column 3: $(0.1, 0.25, 0.5, 0.75, 0.9)$-quantiles of the process $Z_{is}(t) = \alpha_{is} - \alpha_{is}(0)$ $(s = 1, 2, 3)$ for each group.

FUTURE WORK

- Use the model to predict clinical features/outcomes by regression.
- Generalize to purely **functional data** (each sample is an element of a Hilbert space) $\Longrightarrow$ probability over a functional space. This is motivated by biomechanics.
- Add **control** term to model action of insulin pumps and allow for design of controllers.

FUTURE WORK

- Use the model to predict clinical features/outcomes by regression.
- Generalize to purely **functional data** (each sample is an element of a Hilbert space) $\Longrightarrow$ probability over a functional space. This is motivated by biomechanics.
- Add **control** term to model action of insulin pumps and allow for design of controllers.

Thanks for the attention!

EXTRA I: RKHS, KDE AND MMD

1. Reproducing Kernel Hilbert Space (RKHS): A Hilbert space of functions where the evaluation functional is linear and bounded.
This defines a positive-definite kernel with the reproducing property:

$$f(x) = \langle f,\, k(x,\cdot)\rangle_{\mathcal{H}_k}, \qquad \mathcal{H}_k := \overline{\text{span}}\{k(x,\cdot)\}_{x\in\mathcal{X}}.$$

Functions in $\mathcal{H}_k$ can thus be evaluated via dot products $\Rightarrow$ kernel trick.

2. Kernel Mean Embedding (KME): Image of a distribution P in the RKHS (it codifies all the means of functions in $\mathcal{H}_k$)

$$\mu_P = \mathbb{E}_{x\sim P}\big[k(x,\cdot)\big] \in \mathcal{H}_k, \qquad \hat{\mu}_P = \tfrac{1}{m}\sum_{i=1}^{m} k(x_i,\cdot).$$

If the kernel is characteristic, the map $P \mapsto \mu_P$ is injective.

3. Maximum Mean Discrepancy (MMD): Distance between two KMEs

$$\text{MMD}_k(P,Q) = \|\mu_P - \mu_Q\|_{\mathcal{H}_k}, \qquad \text{MMD}_k(P,Q) = 0 \iff P = Q \quad (\text{for characteristic } k).$$

$$\text{Kernel} \Rightarrow \text{RKHS} \Rightarrow \text{KME} \Rightarrow \text{MMD}$$

Extra II: Compute MMD

Closed-form expression for the MMD between a Gaussian mixture and an empirical distribution:

$$f_i(x) = \sum_{s=1}^{K} w_s \, \mathcal{N}(x \mid m_s, \Sigma_s).$$

At each $t_i \in \tau$, we use a Gaussian kernel k_i such as (2), with $\sigma_i^2 \approx (\underset{j \neq k}{\text{median}} \|X_{t_i,j} - X_{t_i,k}\|)^2$. Thus,

$$\text{MMD}^2(P_{t_i}, Q) = \sum_{s=1}^{K}\sum_{r=1}^{K} w_s w_r \, I_{i,s,r} - \frac{2}{n_i}\sum_{s=1}^{K}\sum_{j=1}^{n_i} w_s \, J_{i,s,j} + \frac{1}{n_i^2}\sum_{j=1}^{n_i}\sum_{\ell=1}^{n_i} k_i(X_{t_i,j}, X_{t_i,\ell}).$$

The two first terms admit closed-form expressions:

$$I_{i,s,r} = \frac{(\sigma_i^2)^{d/2}}{\sqrt{\det(\Sigma_s + \Sigma_r + \sigma_i^2 \mathsf{Id})}} \exp\Big(-\frac{1}{2}(m_s - m_r)^\top (\Sigma_s + \Sigma_r + \sigma_i^2 \mathsf{Id})^{-1}(m_s - m_r)\Big),$$

$$J_{i,s,j} = \frac{(\sigma_i^2)^{d/2}}{\sqrt{\det(\Sigma_s + \sigma_i^2 \mathsf{Id})}} \exp\Big(-\frac{1}{2}(X_{t_i,j} - m_s)^\top (\Sigma_s + \sigma_i^2 \mathsf{Id})^{-1}(X_{t_i,j} - m_s)\Big),$$