

Functional Graph Convolutional Networks: A Unified Multi-task and Multi-modal Learning Framework to Facilitate Health and Social-Care Insights

Tobia Boschi¹, Francesca Bonin¹, Rodrigo Ordonez-Hurtado¹,
Cécile Rousseau², Alessandra Pascale¹, and John Dinsmore³

¹IBM Research Europe, Dublin

²École Centrale de Lille

³Trinity Centre for Practice and Healthcare Innovation, Trinity College of Dublin

tobia.boschi@ibm.com, FBonin@ie.ibm.com, Rodrigo.Ordonez.Hurtado@ibm.com,
cecile.rousseau@centrale.centralelille.fr, apascale@ie.ibm.com, dinsmorj@tcd.ie

Abstract

This paper introduces a novel Functional Graph Convolutional Network (funGCN) framework that combines Functional Data Analysis and Graph Convolutional Networks to address the complexities of multi-task and multi-modal learning in digital health and longitudinal studies. With the growing importance of health solutions to improve health care and social support, ensure healthy lives, and promote well-being at all ages, funGCN offers a unified approach to handle multivariate longitudinal data for multiple entities and ensures interpretability even with small sample sizes. Key innovations include task-specific embedding components that manage different data types, the ability to perform classification, regression, and forecasting, and the creation of a knowledge graph for insightful data interpretation. The efficacy of funGCN is validated through simulation experiments and a real-data application. funGCN source code is publicly available at <https://github.com/IBM/funGCN>.

1 Introduction

Ensure healthy lives and promote well-being for all at all ages and leave no one behind – this idea is embedded in the core vision of the UN’s 2030 Agenda for Sustainable Development¹ [Resolution and others, 2015]. In line with this, in the contemporary healthcare landscape, digital health solutions and longitudinal studies have become increasingly significant to monitor health care and wellbeing of populations around the world. These approaches enable a comprehensive understanding of health outcomes over time, providing valuable insights into disease progression, treatment efficacy, and overall patients’ well-being [Garssen *et al.*, 2021][Cuff, 2023]. Digital health solution are therefore critical tools for achieving the UN sustainable development goals.

With the advent of new technologies, data are collected from multiple sources, including surveys, phone interviews, wearable devices, and mobile health applications, to cite a few.

These tools allow patients to capture a wide array of health-related signals from the comfort of their homes. They facilitate the entry of multiple measurements over time [Vijayan *et al.*, 2021] and enhance medical professionals’ and policymakers’ understanding of a population’s health, the impact of interventions, or, more in general, the well-being of subjects. However, they also add to the complexity of the analysis.

Artificial intelligence (AI) plays a key role in effectively dealing with such diverse information. Investigating these studies necessitates a variety of analytical tasks, including *classification*, *longitudinal regression*, and *forecasting*. Classification is crucial for separating patient subgroups or “arms”, assessing treatments’ effectiveness, or identifying the presence of diseases. Longitudinal regression can estimate critical variables that evolve over the entire study’s duration, like patient well-being or healthcare service utilization, and can help determine the impact of a trial or a solution on individual patients. Forecasting, on the other hand, predicts future trends, informing on a trial’s potential success and interventions’ forthcoming effects [Hu *et al.*, 2015]. On top of the different required analytical tasks, the analysis of the collected data presents substantial challenges. First, the variables are of different modalities. Second, the longitudinal signals can be recorded at various times and frequencies. Third, the difficulties of recruiting participants may result in small sample sizes, even smaller than the number of variables [Ildstad *et al.*, 2001]. Last, the interpretability of the results is crucial: understanding the relationships between variables and how they influence specific outcomes can enhance future trial designs, improve patient care, and optimize resource allocation [Hakkoum *et al.*, 2022][Farah *et al.*, 2023]. All the above challenges constitute a complex methodological barrier, even for the most advanced AI and machine learning (ML) approaches. Hence, there is a need for new informative methodologies able to i) deal with multi-modal data and multivariate longitudinal signals for multiple entities, ii) perform different inference tasks, and iii) be effective even when the sample size is small. In response to these demands, we introduce a novel approach called Functional Graph Convolutional Network (funGCN), which synergizes Functional Data Analysis (FDA) [Ramsay and Silverman, 2005][Kokoszka and Reimherr, 2017] and Graph Convolutional Networks (GCNs) [Zhang *et al.*, 2019].

¹SDG, goal 3, <https://sdgs.un.org/goals/goal3>

FDA a dynamic area of statistical research that allows working with multivariate longitudinal data by estimating smooth curves across a continuous domain. GCNs are renowned for their capacity to discern relations between variables. By integrating FDA and GCNs, we create a unified framework for multi-task learning with multi-modal longitudinal data. The innovative aspects of `funGCN` include:

- The introduction of two task-specific embeddings designed for multi-modality. Specifically, they can handle longitudinal, categorical, and constant scalar variables (the latter are numerical values that remain constant over time) and facilitate the comparison of diverse variables and statistical entities throughout the temporal domain.
- The capability to execute three analytics tasks: classification, (longitudinal) regression, and forecast. Moreover, given the embeddings’ flexibility, `funGCN` can concurrently perform regression and classification of multiple target variables within a single training session.
- The generation of a knowledge graph that quantifies the connections between all variables, facilitating dimensional reduction and providing interpretable insights into the outcomes, even in scenarios with small sample sizes.

The effectiveness of `funGCN` is demonstrated through simulation experiments and a real-data application, showcasing its ability to construct an informative knowledge graph and offering a new approach for analyzing complex longitudinal datasets.

This work is part of the SEURO Project² that aims to develop and evaluate a digital social health platform to improve and advance home-based self-management and integrated care for older adults (over 65 years) with multiple chronic conditions. The project focuses on deploying new technologies for improving health care in elderly population and on evaluating innovative, flexible, and individual-centered healthcare models through data from diverse longitudinal studies. The project brings together academic/research institutions, Small to Medium Enterprises (SMEs), health service providers and NGOs and benefits from the a multidisciplinary collaboration between computer scientists/statistician, health-care professionals, stakeholders in the health/social care field.

Related work. Multi-modal GCN, Temporal GCN, and time-series analysis and forecasting are all active research fields. However, aside from `funGCN`, there seems to be a lack of a comprehensive framework that combines all their capabilities. For a clearer understanding of the terminology associated with the applicability of the various models, see Table 1.

Recently work in [Langbridge *et al.*, 2023] introduced a *Causal Temporal GCN* for multivariate time series, while novel approaches in [Jiang *et al.*, 2022], [Vijay *et al.*, 2023], and [Zhou *et al.*, 2024] extended *transformers* and *large language models* to time series. Yet, these approaches do not consider multi-modality or multiple statistical entities. [D’Souza *et al.*, 2022] presents a multi-modal GCN for classification but, differently from `funGCN`, does not tackle longitudinal

variables. *Long short-term memory (LSTM) networks* [Hochreiter and Schmidhuber, 1997] and *gated recurrent unit (GRU) networks* [Chung *et al.*, 2014] allow to perform multiple-task for longitudinal variable and multiple-entities, however they cannot handle multi-modal data. Similarly, classical ML algorithms such *supporting vector machines (SVM)* [Müller *et al.*, 1997] and *random forest* [Breiman, 2001], can manage multi-modal data only after flattening the longitudinal features, losing the time dependency information and interpretability while, at the same time, increasing the total number of variables. On the other hand, FDA has shown promising results in dealing with multi-modal variables across numerous longitudinal applications [Ullah and Finch, 2013][Cremona *et al.*, 2019][Boschi *et al.*, 2021] and, recently, has been effectively extended to deep learning methodologies [Rao and Reimherr, 2023]. `funGCN` is designed to combine the strengths of the FDA in handling multi-modal longitudinal data with the GCN’s capabilities at detecting their interrelationships and producing interpretable outcomes.

In the remainder of this paper, we first present our methodology (Section 2), then we evaluate the performance of our approach against various competing algorithms through simulations (Section 3), and investigate a real-world application using the longitudinal SHARE dataset (Section 4). Finally, we draw our conclusions in Section 5. The Appendix material can be found at <https://arxiv.org/pdf/2403.10158>.

2 Methodology

The architecture of `funGCN` is illustrated in Figure 1. The input data are transformed through two distinct *embedding* modules, each serving a preparatory role — *KG embedding*, estimating the knowledge graph through a node-wise feature selection approach, and *GCN embedding*, preparing the input for the GCN component. The embedding modules’ outputs are tensors of dimensions (n, p, k_{graph}) and (n, p, k_{gcn}) , respectively. The GCN takes as input both the knowledge graph and its specifically embedded data, and performs three analytics tasks: regression, classification, and forecasting. While regression estimates the feature across the entire time domain, forecasting predicts the *horizon* (future) trajectory of a longitudinal feature given its *history* (past) values.

The problem dimensions are defined by n , representing the number of statistical entities, and $p = p_l + p_c + p_s$, which denotes the total number of variables (or features), where p_l, p_c, p_s indicate the numerosity of each input data modality. Longitudinal variables (p_l) evolve over time, categorical variables (p_c) can have an arbitrary number of levels, and scalar features (p_s) are numerical values that remain constant.

Assumptions on longitudinal features. For each of the n statistical entities, we observe p_f distinct longitudinal variables, denoted as $\mathcal{X}_{i,j}$ for $i = 1, \dots, n$ and $j = 1, \dots, p_f$. Typically, in real-world scenarios, these variables are sampled at discrete time points, which might differ across variables and entities [Ramsay and Silverman, 2005][Ullah and Finch, 2013]. However, FDA provides various established preprocessing tools to construct smooth and continuous curves from discrete measurements, such as *basis expansions*, *moving average smoothing*, and *sparse conditional estimation* [Yao *et*

²<https://seuro2020.eu/>

Term	Definition
Multivariate time series	Multiple longitudinal variables for one entity/subject.
Multi-modal model	Model dealing with variables of different modalities (e.g. longitudinal, categorical,...).
Multiple entities model	Model dealing with more statistical entities (e.g. several subjects).
Multi-task	Model performing more than one analytical tasks (i.e. regression, classification, forecast).

Table 1: *Definitions.* Terminology associated with a model applicability.

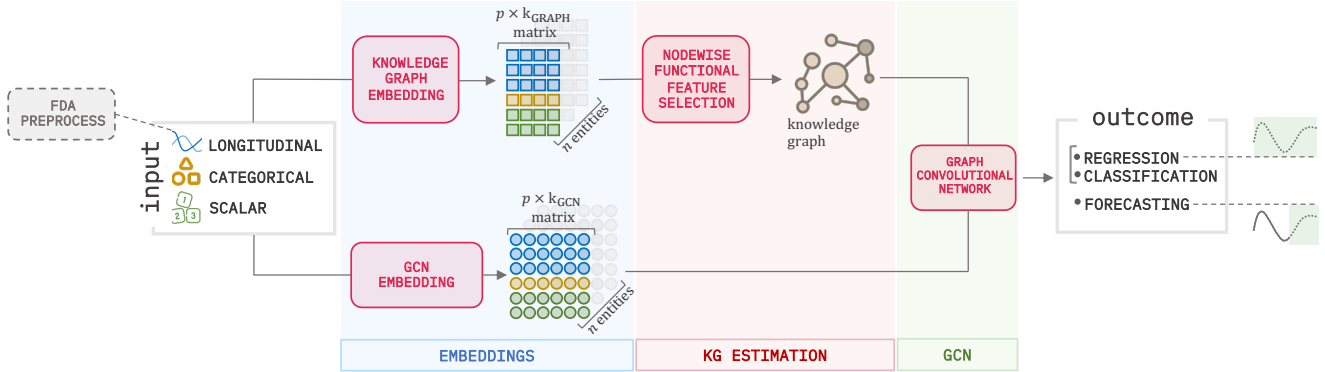


Figure 1: **funGCN architecture.** Embedding modules are discussed in Subsection 2.1, the knowledge graph estimation in Subsection 2.2, and the GCN module in Subsection 2.3.

et al., 2005]. Throughout this paper, we assume that such pre-processing has been applied where necessary and consider $\mathcal{X}_{ij} \in \mathbb{L}^2(\mathcal{T})$, the space of square-integrable functions over the closed and bounded interval \mathcal{T} . This implies that if f is a function in $\mathbb{L}^2(\mathcal{T})$, then $\|f\|_{\mathbb{L}^2}^2 = \langle f, f \rangle_{\mathbb{L}^2} = \int_{\mathcal{T}} f^2(t) dt < \infty$. Hence, \mathcal{X}_{ij} are infinite-dimensional mathematical objects.

2.1 Embedding Modules

The two embedding modules differ on the representation of longitudinal features, while the same transformation is applied to categorical and scalar variables.

Longitudinal features. To transform the longitudinal variable into finite-dimensional objects, we employ a tensor representation, expressing each variable as linear combinations of basis functions [Ramsay and Silverman, 2005]. Let (b_1^j, \dots, b_k^j) denote the k functional basis for the feature j . Then, the embedded longitudinal features are a tensor $X^l \in \mathbb{R}^{n \times p_l \times k}$ defined as:

$$X_{ijs}^l = \langle \mathcal{X}_{ji}, b_s^j \rangle_{\mathbb{L}^2} = \int_{\mathcal{T}} X_{ji}(t) b_s^j(t) dt.$$

Hence, a functional feature for a specific entity is converted to a vector in \mathbb{R}^k , with the coefficients derived by computing the \mathbb{L}^2 -inner product between the curve and each basis function. The selection of the basis system is crucial, as it influences the characteristics of the resulting embedding. To effectively exploit their properties, we propose two different sets of basis functions for the knowledge graph and the GCN embedding modules.

KG embedding. The longitudinal features are represented using their first k_{graph} *functional principal components (FPCs)*, derived through an eigen-decomposition of the functional covariance operator [Kokoszka and Reimherr, 2017]. FPCs capture the directions of larger variability of the curves and are

a well-established technique in FDA, generally used to solve a wide range of problems, including regression [Reiss and Ogden, 2007] and classification [Wang *et al.*, 2016]. Given their proven effectiveness in detecting significant relationships among variables [Boschi *et al.*, 2023], FPCs are an optimal basis system to determine the structure of the node-wise feature selection module input.

GCN embedding. The longitudinal features are embedded using k_{gcN} *equispaced cubic B-splines*. B-splines, known for their piece-wise polynomial structure, offer a flexible yet stable method for curves representation [Eilers and Marx, 1996]. Differently from FPCs, which are derived from the data and vary across features, B-spline bases are pre-determined and consistent across different variables. Moreover, while FPCs can influence the entire time domain, cubic B-splines are non-zero over a limited range, defined by a three-knot interval. The localized influence is crucial for effective forecasting while preserving the original embedded form. It enables a clear distinction between the coefficients associated with past values and those associated with the future, which ensures continuity between the history and the horizon prediction and makes B-splines an ideal tool to define the GCN input.

Categorical features. Each of the p_c categorical variables, which can have an arbitrary number of levels, is mapped to a k -dimensional space using a standard embedding layer in PyTorch [Paszke *et al.*, 2019]. This process assigns a unique k -dimensional vector to every level of the categorical variable, with $k = k_{\text{graph}}, k_{\text{gcN}}$ depending on the embedding module. Unlike the typical method where vector representations are learned and optimized within the GCN module, funGCN pre-determines these vectors before the network's initialization. This approach ensures a static representation that, while not updated during GCN training, allows the integration of multi-

Algorithm 1 Node-wise functional feature selection

GOAL: Estimate adjacency matrix $A \in \mathbb{R}^{p \times p}$

1. SET $A = 0$. FOR $j = 1, \dots, p$:
 - Perform *functional feature selection*
 - SET target = j and penalty $\lambda = c_\lambda \lambda_{\max}$
 - SET $c_\lambda = 1$ and active set $\mathcal{S} = \emptyset$
 - WHILE $|\mathcal{S}| < p_{\max}$:
 - (a) IF feature t selected and $t \notin \mathcal{S}$:
 - add t to \mathcal{S} and set $A_{jt} = c_\lambda$
 - (b) decrease c_λ
2. Make A symmetric: $A = \frac{1}{2}AA^T$
3. Prune A : if $A_{ij} < \theta \Rightarrow A_{ij} = 0$ and normalize it.

modal data by maintaining consistent embeddings.

Scalar features. Scalar features are treated as constant functions over time. We follow the same module-specific basis representation used for the longitudinal variables.

The embedding modules transform each feature modality into k_{graph} and k_{gcn} -dimensional vectors, resulting in output tensors $X_{\text{graph}} \in \mathbb{R}^{n \times p \times k_{\text{graph}}}$ and $X_{\text{gcn}} \in \mathbb{R}^{n \times p \times k_{\text{gcn}}}$, respectively. These tensors are then standardized to ensure that, if the third dimension is fixed, feature values across all entities have a mean of 0 and a standard deviation of 1. This process guarantees uniform representation of features. For each entity, the features are encapsulated in a $p \times k$ matrix. This structure aids the comparison of different entities and enables the fusion of different modalities, while preserving the time-dependency information of the longitudinal features.

Selection of k_{graph} and k_{gcn} . The choice of k for longitudinal features determines how closely the basis representation approximates the original curves. A lower k_{graph} , such as fewer than 5, offers computational efficiency and effectiveness. Indeed, FPCs provide a parsimonious yet comprehensive basis system: few components capture most of the curves' variability and identify key feature relationships. On the other hand, B-splines require a larger number of basis functions to reconstruct the original signal. For capturing smooth curves accurately, it is recommended to set k_{gcn} around 10, with adjustments based on signal complexity and task.

2.2 Knowledge Graph Estimation

The aim of the *node-wise functional feature selection module* is the creation of a knowledge graph with nodes representing the p features and edges indicating their association strength. Hence, the graph information can be encapsulated in a $p \times p$ adjacency matrix A . To estimate A , a node-wise regression method is employed, conducting separate feature selection for each feature as a target variable [Lee and Hastie, 2015]. This strategy enhances the accuracy and specificity of the knowledge graph by allowing for individualized analysis of feature relationships.

We adopt the functional feature selection methodology proposed in [Boschi *et al.*, 2023]. This approach was designed to work with longitudinal features and basis representation. Still, given the flexible structure of X_{graph} , we can enhance its applicability to multi-modal data. The node-wise strategy is outlined in Algorithm 1. For each target variable j , $j = 1, \dots, p$, we explore different penalty parameters $\lambda = c_\lambda \lambda_{\max}$. Initially, c_λ is set to 1, corresponding to no active features. As c_λ gradually reduces, the number of selected features in the model increases. We stop the procedure when this number reaches an upper bound of p_{\max} . In the adjacency matrix A , for each row corresponding to a target variable j , we mark the columns linked to selected features with their respective c_λ values, indicating the order of selection within the model. The diagonal elements are set to 1 for self-association, and all other non-selected feature columns are zeroed out. The earlier a feature is chosen, the higher its c_λ value. Thus, the association strength reflects the selection sequence of features.

Once the node-wise selection routine is completed, first, we compute the symmetric matrix $\tilde{A} = \frac{1}{2}(A + A^T)$, and we then prune the matrix by setting values below a certain threshold θ to zero. Finally, we normalize it as follows:

$$\bar{A} = D\tilde{A}D, \text{ with } D = \text{diag}\left(\left(\sum_{j=1}^p \tilde{A}_{jj}\right)^{-\frac{1}{2}}\right).$$

This pruning and normalization process further reduces the problem dimension and sharpens the GCN focus on the most significant connections, enhancing the efficiency of training.

Note that, despite the single feature selection procedure being highly computationally efficient, the graph construction cost grows with the number of features p . However, the graph is independent of the specific tasks or targets, and, therefore, it can be used across different GCN training scenarios without being recomputed.

2.3 Graph Convolutional Network

The GCN module has a simple structure consisting of two *convolutional* layers, each followed by a *ReLU* activation function, and a final *linear* layer.

Input. The GCN receives as input the preprocessed tensor $X_{\text{gcn}} \in \mathbb{R}^{n \times p \times k_{\text{gcn}}}$ and the adjacency matrix \bar{A} . At this stage, it is necessary to define one or more target variables, denoted as p_{target} . In forecasting, X_{gcn} is divided into 2 parts based on its third dimension. We define $k_2 = r_f * k_{\text{gcn}}$ and $k_1 = k_{\text{gcn}} - k_2$. Here, r_f represents the ratio of the time domain that will be forecasted, and k_1 and k_2 denote the number of coefficients associated with history and horizon, respectively. We can simply consider $k_1 = k_2 = k_{\text{gcn}}$ for the regression and classification tasks.

Training. We employ the *Adam* optimizer [Kingma and Ba, 2014] and the *means-square error loss* to deal with finite vectors. The n entities are divided into training and validation sets of dimension n_{train} and n_{val} , respectively. Throughout each epoch, the training process iterates over the n_{train} entities feeding the GCN a $p \times k_1$ matrix and receiving a $p_{\text{target}} \times k_2$ output. The optimization procedure occurs after each epoch.

Monitoring validation loss enables early stopping if no improvement is observed for v_{stop} consecutive epochs or upon reaching the maximum number of epochs.

Mapping back to original features. For longitudinal and scalar variables, their original form can be easily recovered once the training is completed due to the adaptable nature of their embeddings. Given the estimated coefficients \hat{c}_{ijs} and the basis system (b_1^j, \dots, b_k^j) , the curves are computed as follows [Kokoszka and Reimherr, 2017]:

$$\hat{\chi}_{ij} = \sum_{s=1}^{k_{\text{gen}}} \hat{c}_{ijs} b_s^j.$$

However, for categorical variables, the embedding cannot be directly inverted to the original space. Instead, a nearest *neighbors algorithm* [Cover and Hart, 1967] is employed to match each estimated vector with the nearest original category, ensuring a meaningful approximation of the categorical data.

The GCN learns the most important connections between different types of features from the adjacency matrix. Despite its straightforward architecture, it handles multi-modal data and executes different analytics tasks. Notably, p_{target} can be greater than one, meaning the GCN can predict multiple targets after one training. Furthermore, it is feasible to train the GCN for both classification and regression simultaneously, given that the problem dimensionality is the same for both tasks.

3 Simulation Experiments

Methods comparison. This section assesses the `funGCN` model’s effectiveness with synthetic data, comparing it to `lstm`, `gru`, `svm`, and functional regression (`fReg`). The implementation and hyper-parameters used for all the methods are detailed in Section A of the Appendix. To the best of our knowledge, these are the only methodologies able to handle multivariate longitudinal data and multiple entities.

Table 2 compares the selected methods based on their task execution capabilities, applicable scenarios, and interpretability. Notably, `funGCN` is the only framework supporting diverse tasks for multi-modal data and offering comprehensive insights into feature relationships. `lstm` and `gru` lack interpretability and support for categorical features, necessitating that we remove such variables before executing these methods. `svm` is limited to classification and requires one-hot encoding of categorical features and flattening of longitudinal variables, resulting in the loss of the time-dependency information and interpretability. Finally, `fReg` cannot handle classification and scenarios with more features than entities.

Experiments settings. We set the number of entities $n = 300$ and examine four distinct scenarios with varying numbers of features: $p = 20, 50, 100, 500$. In each scenario, the proportion of longitudinal, categorical, and scalar variables is 0.6, 0.2, and 0.2, respectively, and the number of interconnected features is $p_0 = 10$. We split $p_0 = 10$ into four longitudinal, 4 categorical (2 with two levels, 1 with three levels, and 1 with four levels), and 2 scalar features. The synthetic data generation process is described in Section B of the Appendix.

In each scenario, the target variables are the 4 longitudinal and the 4 categorical from the p_0 interconnected features. For forecasting, we set the forecast ratio at 0.3, utilizing 70% of the domain for predictions. For each target, we conduct 20 replications using 225 entities for training and 75 for testing across all methods, totaling 80 replications per task. For `fReg`, we limit to 10 replications per target in the scenario with 50 features due to its computational complexity (see Table 3) and do not consider scenarios with 100 and 500 features due to its inability to deal with more parameters than entities.

Regression and forecasting tasks are assessed using the *standardized root mean-squared error*, which for the longitudinal target \mathcal{Y} is computed as:

$$\text{std-RMSE} = \left(\frac{1}{n} \sum_{i=1}^n \frac{1}{\text{sd}(\mathcal{Y}_i)} \int_{\mathcal{T}} (\mathcal{Y}_i - \hat{\mathcal{Y}}_i)^2 dt \right)^{\frac{1}{2}}.$$

Classification performances are evaluated in terms of *accuracy*, reflecting the proportion of correctly predicted instances.

Results. The results of the simulation experiments are presented in Figure 2. For each task across varying p , the dots illustrate the average performance derived from 80 replications (20 for each target), with the width of the intervals indicating the standard deviation. In regression tasks, the performance of all algorithms is comparable, though `funGCN` exhibits a marginal advantage. In forecasting and classification, `funGCN` outperforms its competitors, which achieve similar results among themselves. These findings underline the versatility and effectiveness of `funGCN` across different tasks and numbers of features, even in challenging scenarios where the sample size is small relative to the number of variables.

Table 3 shows the average CPU time required for completing all replications across various tasks performed by each algorithm. The computation time for `funGCN` is divided into two parts: the creation of the knowledge graph and the training of the GCN module. Although the graph was recomputed for each simulation, it is important to note that this computation is required only once in real-world applications, as the graph does not need to be recomputed at inference time. `svm` is the most time-efficient algorithm, whereas `fReg` is significantly more computationally expensive. Compared to other deep learning methods, `funGCN` exhibits greater efficiency, particularly as the number of variables increases. Notably, for scenarios with $p = 500$, `funGCN`’s inference phase even surpasses `svm` in speed, highlighting its efficiency in handling a large number of variables.

4 SHARE Analysis

The Survey of Health, Ageing and Retirement in Europe (SHARE) is a pivotal research project that examines the impact of health, social, economic, and environmental policies on the life course of European citizens [Börsch-Supan *et al.*, 2013][Bergmann *et al.*, 2017][Börsch-Supan, 2020]. SHARE stands out for its longitudinal design, tracking the same individuals across eight survey waves from 2004 to 2020. We restrict our analysis to the 1,518 participants who were part of at least seven waves, ensuring sufficient measurements for reliable curve estimation. We investigate a subset of 35 variables

	tasks			capability			feature interpretability		
	regress	classify	forecast	multi-modal	multi-entities	$p > n$	single effect	importance	global relation
funGCN	✓*	✓*	✓	✓	✓	✓	✗	✓	✓
lstm	✓	✓	✓	✗	✓	✓	✗	✗	✗
gru	✓	✓	✓	✗	✓	✓	✗	✗	✗
fReg	✓	✗	✓	✓	✓	✗	✓	✓	✗
svm	✗	✓	✗	▲	✓	✓	✗	✗	✗

Table 2: *Algorithms comparison.* The table displays tasks, capability, and interpretability properties of the proposed funGCN and other (existing) evaluated algorithms. The *task* columns refer to the possibility of performing either regression, classification, or forecast. The *capability* columns refer to the ability to handle i) multi-modal data, ii) multiple entities, or iii) the number of variables being larger than the sample size. The *interpretability* columns refer to the ability to estimate i) effect of single feature on the target, ii) importance of the feature for the specific task, or iii) the global relationship between all the variables. [Notation: * indicates that funGCN is the only algorithm able to concurrently perform regression and classification of multiple target variables within just one training process; ▲ denotes that svm can handle multi-modal data only after flattening the longitudinal variables, losing the time information.]

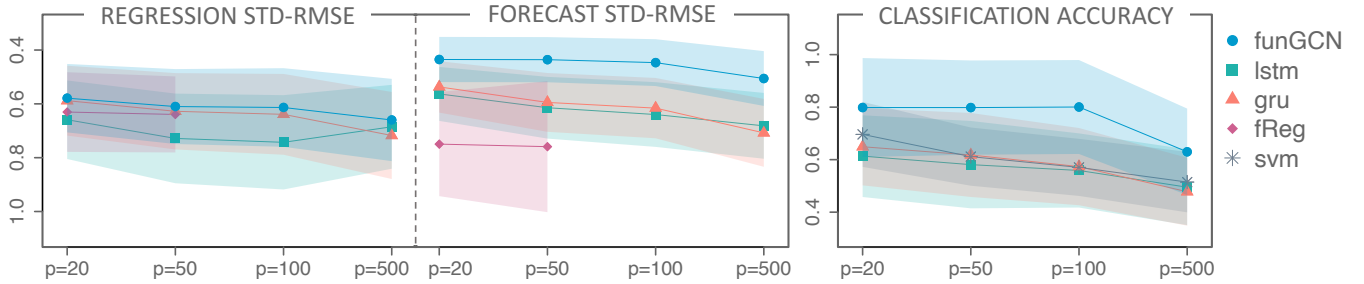


Figure 2: **Simulation results.** For each investigated scenario ($p = 20, 50, 100, 500$), dots represent the *average* over 80 replications (20 for each target), and the width of the interval indicates the *standard deviation*. The *y-axis* for *regression* and *forecast* are inverted: higher values correspond to better performance in all the panels.

p	funGCN _{kg}	funGCN _{inf}	lstm	gru	fReg	svm
20	0.8	2.5	4.3	10.8	472.5	0.1
50	1.8	2.2	9.7	18.6	5,006.5	0.2
100	3.5	2.4	16.8	25.1	—	0.4
500	16.1	3.1	80.7	80.8	—	4.7

Table 3: *Simulation CPU time.* Average CPU processing time, in seconds, computed over 80 replications (20 for each target), and across all possible tasks an algorithm performs in each instance. For funGCN we report the time required to create the knowledge graph (kg) and the time needed to complete the inference task (inf). All the computations were executed on a MacBookPro 2021 with an M1 Max processor and 32GB of RAM.

from the EasySHARE dataset [Gruber *et al.*, 2014], a preprocessed version of the SHARE data. These variables, which can be longitudinal, categorical, or scalar, span physical and mental health, socio-demographics, and healthcare, offering a rich dataset for comprehensive analysis — the complete list of the examined variables is reported in Appendix Table C1.

Following the strategy in [Boschi *et al.*, 2024], we employ cubic B-splines for smoothing longitudinal variables, with knots at survey dates and a roughness penalty on the curve’s second derivative. The optimal smoothing parameter is determined by minimizing the generalized cross-validation criterion [Craven and Wahba, 1978]. Despite variations in survey dates and data points across participants, this approach offers a consistent functional representation, allowing for a natural imputation of missing values and facilitating comparisons across the entire timeline.

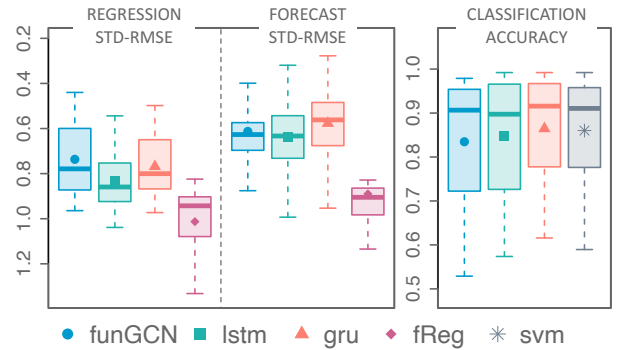


Figure 3: **SHARE results.** Boxplots generated from the distribution obtained from a total of 260 replications for regression and forecasting tasks, and 80 replications for classification – encompassing 20 replications for each target. The dots and the lines indicate the means and medians of the distributions, respectively. The *y-axis* for *regression* and *forecast* are inverted: higher values correspond to better performance in all the panels.

Across task results. To evaluate the algorithms’ performance across the 3 different tasks, we select each of the 13 longitudinal and 4 categorical variables from the chosen 35-feature subset as target variables (detailed in Appendix Table C1). We conduct 20 replications for each target, randomly splitting the 1,518 subjects into a training set of 1,139 and a test set of 379. The aggregate average results are depicted in Figure 3, while the individual targets performance are avail-

Acknowledgments

The authors received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement No. 945449. The authors also thank their partners at IMEC for helping them access the SHARE data, and Abigail Langbridge for the useful discussions on Convolutional Graphical Networks.

References

- [Bergmann *et al.*, 2017] Michael Bergmann, Thorsten Kneip, Giuseppe De Luca, and Annette Scherpenzeel. Survey participation in the survey of health, ageing and retirement in europe (share), wave 1-6. *Munich: Munich Center for the Economics of Aging*, 2017.
- [Börsch-Supan *et al.*, 2013] Axel Börsch-Supan, Martina Brandt, Christian Hunkler, Thorsten Kneip, Julie Korbmacher, Frederic Malter, Barbara Schaan, Stephanie Stuck, and Sabrina Zuber. Data resource profile: the survey of health, ageing and retirement in europe (share). *International journal of epidemiology*, 42(4):992–1001, 2013.
- [Börsch-Supan, 2020] Axel Börsch-Supan. Survey of health, ageing and retirement in europe (share) wave 5. *Release version*, 7(0), 2020.
- [Boschi *et al.*, 2021] Tobia Boschi, Jacopo Di Iorio, Lorenzo Testa, Marzia A Cremona, and Francesca Chiaromonte. Functional data analysis characterizes the shapes of the first covid-19 epidemic wave in italy. *Scientific reports*, 11(1):1–15, 2021.
- [Boschi *et al.*, 2023] Tobia Boschi, Lorenzo Testa, Francesca Chiaromonte, and Matthew Reimherr. Fasten: an efficient adaptive method for feature selection and estimation in high-dimensional functional regressions. *arXiv preprint arXiv:2303.14801*, 2023.
- [Boschi *et al.*, 2024] Tobia Boschi, Francesca Bonin, Rodrigo Ordonez-Hurtado, Alessandra Pascale, and Jonathan Epperlein. A new computationally efficient algorithm to solve feature selection for functional data classification in high-dimensional spaces. *arXiv preprint arXiv:2401.05765*, 2024.
- [Breiman, 2001] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.
- [Chung *et al.*, 2014] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- [Cover and Hart, 1967] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.
- [Craven and Wahba, 1978] Peter Craven and Grace Wahba. Smoothing noisy data with spline functions. *Numerische mathematik*, 31(4):377–403, 1978.
- [Cremona *et al.*, 2019] Marzia A Cremona, Hongyan Xu, Kateryna D Makova, Matthew Reimherr, Francesca Chiaromonte, and Pedro Madrigal. Functional data analysis for computational biology. *Bioinformatics (Oxford, England)*, 35(17):3211, 2019.
- [Cuff, 2023] Alison Cuff. The evolution of digital health and its continuing challenges, 2023.
- [D’Souza *et al.*, 2022] Niharika S D’Souza, Hongzhi Wang, Andrea Giovannini, Antonio Foncubierto-Rodriguez, Kristen L Beck, Orest Boyko, and Tanveer Syeda-Mahmood. Fusing modalities by multiplexed graph neural networks for outcome prediction in tuberculosis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 287–297. Springer, 2022.
- [Eilers and Marx, 1996] Paul HC Eilers and Brian D Marx. Flexible smoothing with b-splines and penalties. *Statistical science*, 11(2):89–121, 1996.
- [Farah *et al.*, 2023] Line Farah, Juliette M Murriss, Isabelle Borget, Agathe Guilloux, Nicolas M Martelli, and Sandrine IM Katsahian. Assessment of performance, interpretability, and explainability in artificial intelligence-based health technologies: What healthcare stakeholders need to know. *Mayo Clinic Proceedings: Digital Health*, 1(2):120–138, 2023.
- [Garssen *et al.*, 2021] Bert Garssen, Anja Visser, and Grieteke Pool. Does spirituality or religion positively affect mental health? meta-analysis of longitudinal studies. *The International Journal for the Psychology of Religion*, 31(1):4–20, 2021.
- [Gruber *et al.*, 2014] Stefan Gruber, Christian Hunkler, and Stephanie Stuck. Generating easyshare: guidelines, structure, content and programming. Technical report, SHARE Working Paper Series 17-2014. Munich, 2014.
- [Hakkoum *et al.*, 2022] Hajar Hakkoum, Ibtissam Abnane, and Ali Idri. Interpretability in the medical field: A systematic mapping and review study. *Applied Soft Computing*, 117:108391, 2022.
- [Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [Hu *et al.*, 2015] Zhongkai Hu, Shiyong Hao, Bo Jin, Andrew Young Shin, Chunqing Zhu, Min Huang, Yue Wang, Le Zheng, Dorothy Dai, Devore S Culver, et al. Online prediction of health care utilization in the next six months based on electronic health record information: a cohort and validation study. *Journal of medical Internet research*, 17(9):e4976, 2015.
- [Ildstad *et al.*, 2001] Suzanne T Ildstad, Charles H Evans Jr, et al. Small clinical trials: Issues and challenges. 2001.
- [Jiang *et al.*, 2022] Hao Jiang, Lianguang Liu, and Cheng Lian. Multi-modal fusion transformer for multivariate time series classification. In *2022 14th International Conference on Advanced Computational Intelligence (ICACI)*, pages 284–288. IEEE, 2022.
- [Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

- [Kokoszka and Reimherr, 2017] Piotr Kokoszka and Matthew Reimherr. *Introduction to functional data analysis*. CRC Press, 2017.
- [Langbridge *et al.*, 2023] Abigail Langbridge, Fearghal O’Donncha, Amadou Ba, Fabio Lorenzi, Christopher Lohse, and Joern Ploennigs. Causal temporal graph convolutional neural networks (ctgcn). *arXiv preprint arXiv:2303.09634*, 2023.
- [Lee and Hastie, 2015] Jason D Lee and Trevor J Hastie. Learning the structure of mixed graphical models. *Journal of Computational and Graphical Statistics*, 24(1):230–253, 2015.
- [Müller *et al.*, 1997] Klaus-Robert Müller, Alexander J. Smola, Gunnar Rätsch, Bernhard Schölkopf, Jens Kohlmorgen, and Vladimir Vapnik. Predicting time series with support vector machines. In *International conference on artificial neural networks*, pages 999–1004. Springer, 1997.
- [Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
- [Petrie *et al.*, 2018] John R Petrie, Tomasz J Guzik, and Rhian M Touyz. Diabetes, hypertension, and cardiovascular disease: clinical insights and vascular mechanisms. *Canadian Journal of Cardiology*, 34(5):575–584, 2018.
- [Powell-Wiley *et al.*, 2021] Tiffany M Powell-Wiley, Paul Poirier, Lora E Burke, Jean-Pierre Després, Penny Gordon-Larsen, Carl J Lavie, Scott A Lear, Chiadi E Ndumele, Ian J Neeland, Prashanthan Sanders, et al. Obesity and cardiovascular disease: a scientific statement from the american heart association. *Circulation*, 143(21):e984–e1010, 2021.
- [Ramsay and Silverman, 2005] James O. Ramsay and Bernard W. Silverman. *Functional data analysis*. Springer, 2 edition, 2005.
- [Rao and Reimherr, 2023] Aniruddha Rajendra Rao and Matthew Reimherr. Nonlinear functional modeling using neural networks. *Journal of Computational and Graphical Statistics*, pages 1–10, 2023.
- [Reiss and Ogden, 2007] Philip T Reiss and R Todd Ogden. Functional principal component regression and functional partial least squares. *Journal of the American Statistical Association*, 102(479):984–996, 2007.
- [Resolution and others, 2015] General Assembly Resolution et al. Transforming our world: the 2030 agenda for sustainable development. *UN Doc. A/RES/70/1 (September 25, 2015)*, 2015.
- [Sivertsen *et al.*, 2015] Heidi Sivertsen, Guro Hanevold Bjørkløf, Knut Engedal, Geir Selbæk, and Anne-Sofie Helvik. Depression and quality of life in older persons: a review. *Dementia and geriatric cognitive disorders*, 40(5-6):311–339, 2015.
- [Tan *et al.*, 2012] Dawn Tan, Mary Danoudis, Jennifer McGinley, and Meg E Morris. Relationships between motor aspects of gait impairments and activity limitations in people with parkinson’s disease: a systematic review. *Parkinsonism & related disorders*, 18(2):117–124, 2012.
- [Ullah and Finch, 2013] Shahid Ullah and Caroline F Finch. Applications of functional data analysis: A systematic review. *BMC medical research methodology*, 13(1):1–12, 2013.
- [Vijay *et al.*, 2023] Ekambaram Vijay, Arindam Jati, Nam Nguyen, Gift Sinthong, and Jayant Kalagnanam. Tsmixer: lightweight mlp-mixer model for multivariate time series forecasting. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2023.
- [Vijayan *et al.*, 2021] Vini Vijayan, James P Connolly, Joan Condell, Nigel McKelvey, and Philip Gardiner. Review of wearable devices and data collection considerations for connected health. *Sensors*, 21(16):5589, 2021.
- [Walker *et al.*, 2014] Robin L Walker, Guanmin Chen, Finlay A McAlister, Norm RC Campbell, Brenda R Hemmelgarn, Elijah Dixon, William Ghali, Doreen Rabi, Karen Tu, Nathalie Jette, et al. Relationship between primary care physician visits and hospital/emergency use for uncomplicated hypertension, an ambulatory care-sensitive condition. *Canadian Journal of Cardiology*, 30(12):1640–1648, 2014.
- [Wang *et al.*, 2016] Jane-Ling Wang, Jeng-Min Chiou, and Hans-Georg Müller. Functional data analysis. *Annual Review of Statistics and its application*, 3:257–295, 2016.
- [Yao *et al.*, 2005] Fang Yao, Hans-Georg Müller, and Jane-Ling Wang. Functional data analysis for sparse longitudinal data. *Journal of the American statistical association*, 100(470):577–590, 2005.
- [Zanin, 2017] Luca Zanin. Education and life satisfaction in relation to the probability of social trust: a conceptual framework and empirical analysis. *Social Indicators Research*, 132(2):925–947, 2017.
- [Zhang *et al.*, 2019] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1):1–23, 2019.
- [Zhou *et al.*, 2024] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36, 2024.