From Black Box to Glass Box: Evaluating Faithfulness of Process Predictions with GCNNs

Myriam Schaschek, Fabian Gwinner, Benedikt Hein, and Axel Winkelmann

Julius-Maximilians-University Wuerzburg

Abstract. The volatile digital economy forces enterprises to tap into the potential of data-driven decision-making. Accordingly, proactive management of business processes is increasingly gaining momentum in information system research. In addition to the superior model performance of predictive models, the explainability of deep learning models becomes a crucial requirement for real-world applications. Although recent works on explainable predictive business process monitoring propose various explainability approaches, preliminary research has been conducted on evaluating explanations regarding their faithfulness. Since human-created ground truth for evaluating algorithms is often unavailable or subjective, objective metrics are needed to assess the faithfulness of explanations. We contribute to this research gap by quantitatively and qualitatively investigating the capabilities of different explainability methods for Graph Convolutional Neural Networks in the context of outcome prediction.

Keywords: Explainable AI · Evaluation · Graph Neural Networks · Predictive Business Process Monitoring.

1 Introduction

Artificial intelligence (AI) has permeated various disciplines, driving its application in information systems research. This is particularly evident in the number of studies conducted on the impact of AI on businesses. There is a significant shift in the perception of deep learning (DL) model performance for predicting process-related measures, which is no longer focused solely on achieving high performance, but on providing explanations to users. This raises vital questions about how we can evaluate the performance of explanation techniques. Accordingly, we investigate the capabilities of three explainability methods for state-of-the-art predictive business process monitoring (PBPM) methods [37], namely Graph Convolutional Neural Networks (GCNNs).

Inspired by the graph nature of processes, some authors transformed the eventlogs of process executions into graph structure for predicting their process measures of interest [19]. Graph Neural Networks (GNNs) can process the topology information of graph-structured data to reach superior performance in prediction tasks. First works in PBPM research have proven GNNs applicable for outcome prediction [31] or next activity prediction [39]. Due to the superior model performance, GNNs have become increasingly popular for various PBPM tasks.

In general, it is necessary to provide human-intelligible explanations for users to ensure their acceptance of graph-based DL models in the business environment [38]. However, due to their sophisticated and complex internal representations, DL models are referred to as opaque "black-box" models that lack interpretability to humans [25]. A novel research stream in PBPM explores explainability algorithms to suitably explain the PBPM predictions of deep models [30]. While recent research has begun investigating post-hoc explainability using model-agnostic or model-specific techniques, we think it is critical to understand how to evaluate the explanation results, as this evaluation influences users' adoption decisions [28].

Against this background, we discovered a research gap in the PBPM literature: the absence of work evaluating explainability, particularly model-specific explanations of GCNNs. Appropriate explanation results should faithfully explain the behavior of the applied DL model [4] and should be helpful for the user [18]. One challenge in evaluating GNN explanation techniques is that it often relies on human-made ground truth, highly dependent on subjective understanding [5]. Therefore, using quantitative metrics to assess the quality of GNN explanations is a representative sample to evaluate the results from the model's perspective [45]. However, methods for assessing explanation faithfulness for graph-structured data from eventlogs remain unexplored.

To address this research gap, we investigate the quality of three post-hoc GCNN-based explanation techniques for process outcome predictions and discuss their general suitability for PBPM. Thereon, we methodologically contribute to the machine learning research in information systems and address the perennial need for deep model explanations in the business process environment.

2 Preliminaries

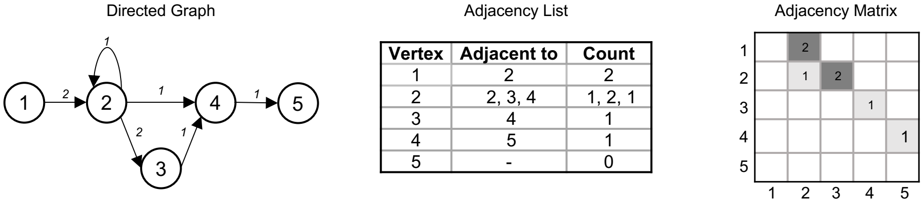
Graph-based PBPM techniques process adjacency matrices of process graphs derived from eventlogs to predict the desired measures of interest. Eventlogs consist of process instances that represent the sequential execution of process events[1].

Definition 1 (Event, Trace, Process Instance, Eventlog). *An event e is the smallest instance and denotes a tuple $(a, c, t, (d_1, v_1), ..., (d_m, v_m))$ where a is the activity name, c is the case id, t is the timestamp, and $(d_1, v_1), ..., (d_m, v_m)$ are event attributes and their respective values. Traces consist of non-empty sequences of events $\sigma = \langle e_1, ..., e_n \rangle$, and each event can be represented as a vector $x^i \in \mathbb{R}^n$ containing information associated to each event. The time order of events within a trace is denoted in superscripts $\sigma = \langle x^{(1)}, ..., x^{(n)} \rangle$. A trace contains all events up to the current time instant, whereas a process instance (or case) contains all past and future events. A set of traces is referred to as an eventlog $L = \{\sigma_1, ..., \sigma_n\}$.*

[1] Definitions are inspired by the work of [40,32].

Definition 2 (Graph, Adjacency Matrix). *Intuitively, the sequences of events, represented by a set of direct edges E connecting two nodes (events) V, can be represented as a direct graph. Let $G = (V, E, \boldsymbol{X})$ denote a directed graph, where V is a set of nodes, E is a set of edges, and $\boldsymbol{X} = \{x_1, ..., x_n\}$ a set of node feature vectors corresponding to the nodes in V. The node feature vectors $\boldsymbol{X}$ represent event attributes and their corresponding values. Equivalently, G can be represented as adjacency matrix $A \in \mathbb{R}^{|V|\times|V|}$. The discrete values in row i and column j denote existing edges between the associated nodes v_i and v_j. Adjacency matrices express the topology (control flow) information of directed graphs as illustrated in 1.*

Definition 3 (Label). *Supervised PBPM approaches assume a labelled event-log for training. A label defines a certain outcome of a process, given a trace $\sigma = \langle e_1, ..., e_n \rangle$ and indicates a certain class that the classifier has to learn. For outcome prediction using GCNNs, the task is to classify a trace σ to its corresponding label $y_i = \{1, .., n\}$.*



Vertex	Adjacent to	Count
1	2	2
2	2, 3, 4	1, 2, 1
3	4	1
4	5	1
5	-	0

1		2			
2		1	2		
3				1	
4					1
5					
	1	2	3	4	5

Fig. 1: Sequences of events represented by a directed graph and its transformation to an weighted adjacency matrix.

3 Related Work

In the following, we summarize recent developments in explainable PBPM and in evaluating explainability to guide our proposed approach that draws inspiration from both the field of PBPM and recent work on DL operating on graphs.

3.1 Explainability and predictive business process monitoring

In PBPM research, a variety of DL models have been proposed to represent the properties of eventlogs in neural network architecture, as they show superior performance for various process prediction tasks [37]. The representation of processes as graphs inspires the use of GNN architectures to predict various process measures, such as the next event [37,39], the outcome [8,31], or the prediction of the remaining time [21].

Generally speaking, GNNs are neural network architectures that enable the processing of graph-structured input data. They compute non-linear transformation functions to map graph-structured input data into compact vector embeddings [14]. A plethora of GNN architectures has been developed for specific application purposes [43]. Convolutional GNNs, often referred to as GCNNs, are a special form of GNN which generalizes the characteristics of Convolutional Neural Networks (CNN) for the processing of non-euclidean data [13]. They learn higher abstractions with stacked multiple convolutional layers at the cost of complexity and lack of interpretability. In this context, [45] investigate GCNN as a basis for post-hoc explainers and demonstrate their performance for various graph prediction tasks.

Examples of post-hoc GCNN explainers are Grad-CAM, GNNExplainer, or PGExplainer. Grad-CAM is a gradient-based explainability approach initially designed for CNNs. [22] extend the method to GCNNs and prove its superior performance compared to two other methods initially designed for CNNs. Grad-CAM uses the gradient of a GNN to produce heat maps with back-propagation. Explainability methods developed explicitly for GNN architectures are GNNExplainer and PGExplainer. Both are perturbation-based methods and derive subgraph structures to explain a set of nodes of a given class. This approach is that simple surrogate or gradient-based methods fall short in learning the semantically essential structures and graphs' topology information. GNNExplainer [44] is the first developed GNN-based explainable method that optimizes masks to identify crucial edges and features. Meanwhile, PGExplainer [17] learns a parameterized model to predict essential edges.

While several classes of GNN methods have been proposed in recent PBPM literature, more studies are needed to explain their predictions [40].

Explainable predictive business process monitoring Several works on explainable PBPM focus on *model-agnostic* explanations, e.g., [29,26,23]. However, little research focuses on *model-specific* explanations [30]. Recent works on model-specific explanation methods in PBPM are [41,29] that aim at building interpretable DL-based models using attention mechanisms or [40] that use layer-wise relevance propagation. Due to the unique nature of graphs, the emerging research field on explainable GNNs is developing model-specific algorithms specialized in interpreting topological structures. These post-hoc explainers show a promising path to explainable PBPM, which is of pivotal interest to the various business domains and thus to the information system community.

3.2 Evaluating the Performance of Explainability Methods

With explainability as a crucial design criterion for future decision support systems [38], it becomes essential that explanation results *faithfully* explain the behavior of the predictive model at hand [4]. Comparing explainability techniques for determining performance in terms of faithfulness involves two different thrusts. First, the interpretability and second, the usefulness of the explanations through visualizations of the results for the user is a decisive evaluation criterion [18]. Literature on explanation methods divides corresponding evaluation approaches into application-based, human-based, and functional-based [3].

Recent work on explainable PBPM mainly focuses on evaluating their predictive quality by visualizing the explanations for predictions [30]. [24] are the first to address the user perspective and investigate how to assist users (e.g., process managers) in decision-making. Even though explanation visualization allows users to evaluate the explanation as reasonable, the evaluation is highly dependent on subjective understanding and cannot serve as an objective evaluation criterion. Further, the ground truth needed to evaluate performance is not always available, and evaluating explanations by hand can be tedious and, therefore, easily prone to error [5]. Accordingly, evaluation metrics that quantify the studied explanation results should be used to measure whether the explanations are faithful to the model [11,42].

While recent research makes initial attempts to evaluate post-hoc explanations for PBPM applications [34,36,35], GNN explainability techniques have not yet been considered as powerful tools for an adequate representation of graph-structured data, such as eventlogs. In general, the research area of GNN explainability is still in its infancy. Thus, only a few works empirically evaluate the applied explainability methods and metrics [1,45,5,27]. [1] provide theoretical guarantees and empirical evidence on faithfulness, stability, and fairness preservation for nine diverse state-of-the-art GNN explanation methods. A unified approach to evaluating GNN explanations present [45] that taxonomize explanation methods and propose using fidelity, sparsity, and stability as key evaluation metrics for GNN explanations. In contrast, [27] quantitatively evaluate graph input attribution methods with several metrics and [5] shows the drawback of GNN explanations and why they might not detect the ground truth.

4 Evaluating Explainable Graph Convolutional Neural Networks for Outcome Prediction

Since explanations are essential to evaluate predictive quality, we evaluate explanations regarding fidelity and sparsity. Built on the recent success of GNNs, we rely on a GCNN architecture for predicting business process outcomes due to the architectural advantage favoring the use of graph-structured data, such as in chemical molecules or business processes [1].

Our approach to assessing the quality of outcome prediction explanations involves three steps (Figure 2). First, we pre-process eventlog data for graph-based classification, explain predictions, and evaluate the explanations. The left part of Figure 2 shows our proposed approach for explainable outcome prediction on a high-level level. We train a GCNN-based neural network architecture to learn a binary graph classification task. Subsequently, we employ an explanation method specified for GCNNs and evaluate explanations. Quantitative analysis of explanation evaluation includes computed evaluation metrics fidelity and sparsity. To qualitatively assess the performance and faithfulness of explanations, we

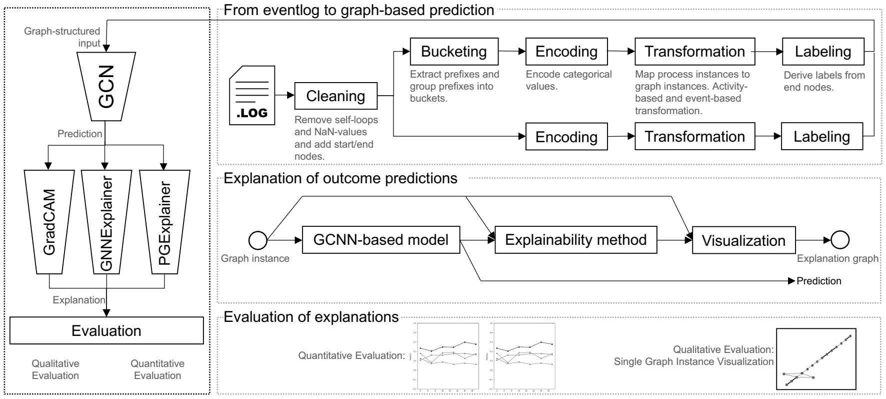

Graph-structured input
GCN
Prediction
GradCAM
GNNExplainer
PGExplainer
Explanation
Evaluation
Qualitative Evaluation
Quantitative Evaluation
From eventlog to graph-based prediction
.LOG
Cleaning
Remove self-loops and NaN-values and add start/end nodes.
Bucketing
Extract prefixes and group prefixes into buckets.
Encoding
Encode categorical values.
Transformation
Map process instances to graph instances. Activity-based and event-based transformation.
Labeling
Derive labels from end nodes.
Encoding
Transformation
Labeling
Explanation of outcome predictions
Graph instance
GCNN-based model
Explainability method
Visualization
Explanation graph
Prediction
Evaluation of explanations
Quantitative Evaluation:
Qualitative Evaluation:
Single Graph Instance Visualization

Fig. 2: A three-step process for explainable outcome predictions.

visualize single graph instances of processes. The right part of Figure 2 provides detailed insights into the processes in each step. In the following, we present insights into our method.

Data Preparation Eventlogs require pre-processing before being fed to a GCNN-based predictor to predict the process outcomes. Thus, we pre-process raw eventlogs following two distinct approaches and convert them from tabular to graph data structures. Pre-processing includes data cleaning, bucketing, encoding, transforming, and labelling. The first step includes modifying headers, removing irrelevant columns, replacing unknown values, and adding start/end nodes. After cleaning, we further pre-process the dataset in different ways according to the application objective. Either whole process instances are processed or extracted prefixes are collected in buckets [33]. Following [31,33], we subsequently encode categorical variables with one-hot-encoding. The conversion of process instances from the eventlog to a graph structure is accomplished using activity-based [31] and event-based [39]. Subsequently, we label the process instances for the binary classification of process outcomes and delete the events that derive the labels.

GCNN-based Prediction The GCNN model contains three components [43]: multiple generalized CNN layers [13] stacked one after another to learn node representations, a read-out layer to summarize the learned node-based features and generate a graph level prediction, and finally a linear classifier to determine probabilities for class membership. Our proposed model consists of three GCNN layers following a ReLu activation function and a global pooling layer as a read-out layer to compute a graph's average overall node features. Then, a linear layer returns the class membership of the processed instance. We train the model in batches, allowing the model to process multiple graphs in parallel and reducing training time. In addition, the model contains a dropout layer to prevent overfitting. Figure 3 illustrates the proposed GCNN architecture.

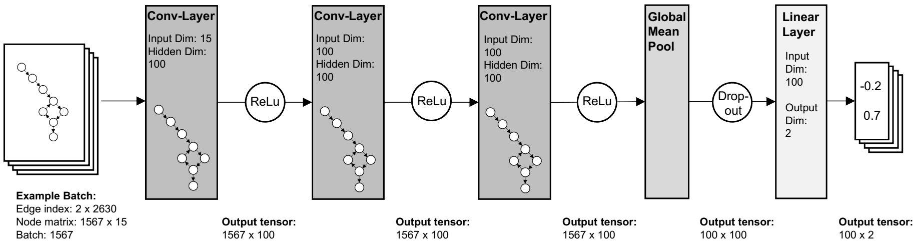


Fig. 3: Visualization of the proposed GCNN model architecture.

Explanation After predicting process outcomes with GCNNs, we use three explanation methods (Grad-CAM, GNNExplainer, PGExplainer) to identify essential components of the instance graphs for the prediction. We select explanation methods based on the desirable property generalizability since all methods can be applied to different graph prediction tasks and GNN architectures. The returned explanation consists of edge masks combined with the original graph's components to form an explanation subgraph. This allows the visualization of the critical subgraphs as part of the original graph, where edges included in the critical subgraph's edge list are highlighted in bold. The steps necessary for explainable predictions using post-hoc methods depict the central component of Figure 2. For explaining process outcome predictions depicting a graph classification, instance-level explanation techniques are a good starting point. Instance-level explanations identify essential graph components instead of model-level explanations that derive instance-independent explanations. For PBPM, instance-level explanations provide more detailed insights into the parts of processes that lead to the prediction. Thus, we deploy and subsequently compare instance-level explanation techniques to reveal essential process characteristics.

Evaluation of Explanation GCNN explanations should highlight the node or edge features the predictive model leverages to make the classification prediction. In the neural architecture of GNN, the message-passing scheme creates local views of the graph created by embeddings along the edges in the node's local neighborhood. Thus, a faithful explanation corresponds to assessing whether the explainer identifies critical components for the prediction [45]. There exists a plethora of quantitative performance measures for explainability methods. However, some metrics correlate with each other. For example, making an explanation more straightforward by increasing sparsity leads to a deterioration of other metrics, such as the fidelity metric.

Following recent research on explainability in GNNs, we use two commonly applied metrics to evaluate the faithfulness of GNN explanation results [45,10], namely fidelity+ and fidelity- [22]. Fidelity+ (1) is defined as the difference in prediction probabilities of the model given the original graph and a constructed new graph as input [22]. The constructed new graph contains masked components decisive for the prediction. Vice versa fidelity- (2) masks decision-irrelevant components. Explanations with "low fidelity-" and "high fidelity+" show a faithful representation, and sparse explanations are considered more comprehensible. The sparsity metric (3) measures the sparsity of an explanation as to the proportion of the subgraph identified as decision-relevant in the original total graph [22]. Following [15,45], we evaluate explanations with fidelity metrics for different sparsity levels.

$$Fidelity{+}^{prob} = 1\frac{1}{N}\sum_{i=1}^{N}\big(f(G_i)_{y_i} - f(G_i^{1-m_i})_{y_i}\big) \tag{1}$$

$$Fidelity{-}^{prob} = 1\frac{1}{N}\sum_{i=1}^{N}\big(f(G_i)_{y_i} - f(G_i^{m_i})_{y_i}\big) \tag{2}$$

$$Sparsity = 1\frac{1}{N}\sum_{i=1}^{N}\big(1 - \frac{|m_i|}{|M_i|}\big) \tag{3}$$

Where $f()$ denotes a trained GNN model based on the input graph G_i and m_i specifies the mask generated by the explainable method. Thus, $G_i^{m_i}$ indicates the generated subgraph containing components essential for the prediction. Vice versa, $G_i^{1-m_i}$ comprises the remaining irrelevant components. $|m_i|$ represents the number of essential graph components, and $|M_i|$ is the number of original graph components.

5 Experimental Evaluation

Our experimental analysis of GCNN explanation techniques involves a quantitative and qualitative analysis of the explainable outcome prediction results. To this end, we compare the three different GNN explanation methods GradCAM, GNNExplainer, and PGExplainer, using fidelity [22] under similar sparsity levels. The experimental comparison aims at evaluating the performance of explanation techniques on business process data represented as graphs.

5.1 Experiment Design

We designed experiments to reflect the practical use of GCNN explanation methods for outcome predictions in PBPM. We experiment with two application domains, namely finance and editorial decision-making. As mentioned above, we limit the analyzed explanatory methods to those that are particularly applicable to GNN model architectures in a model-specific way. The good performances on diverse graph-structured datasets shown in [45,1] by GCNN-based explanation techniques made them natural candidates for our proposed approach[2].

Prediction. We first train a single 3-layer GCNN and then use Grad-CAM, GNNExplainer, and PGExplainer to explain the predictions made by the GCNN. We partition the datasets into 70:20:10 train, test, and validation sets for training and evaluation. We used a GCN architecture with the following configurations. Models are trained using the ADAM optimizer with grid-searched learning rate (0.05, 0.00005) [12].

Explanation. For an intuitive interpretation of experimental results, we establish a performance baseline for explanation techniques by randomly selecting a set of graph nodes of the original graph according to the required sparsity. Then, the evaluation metrics are computed on the randomly generated explanation graphs to verify that the results obtained are better than a random explanation. Finally, we grid-search hyperparameters of explanation techniques for a fair comparison and fine-tune them, if necessary.

Evaluation. After explaining the prediction, we quantitatively and qualitatively evaluate the different explanation methods. For an objective assessment of explanation faithfulness, we compare different methods with fidelity scores under similar sparsity levels. Five evaluation runs are performed for ascending sparsity values. We randomly select 300 instance graphs from each dataset to calculate the evaluation metrics not taken for training the explainer. In addition, we visually compare the explanation results.

5.2 Datasets

We consider real-world and synthetic eventlogs suitable for the task of graph classification, or in other words, for outcome prediction of processes. The synthetic dataset comprises an intuitively comprehensible process model that enables non-experts to understand which events influence process outcomes. Consequently, the eventlog is suitable for a qualitative analysis via visualization. The opposite applies to the real-world log, which is more complex.

The **loan eventlog**[3] is a commonly used eventlog in process mining research. The eventlog covers loan applications and related loan application processing events in an online system. The structure contains three categories of events:

[2] The experimental evaluation is implemented in Python 3.8.5. We use PyTorch Geometrics [6] and PyTorch [20] back-ends for an efficient GPU-based implementation. For the model implementation, we take advantage of the open-source library DIG [16], which provides a module dedicated to explainability in GNNs. For the visualization of the explanation results and processes, we employ NetworkX [7], a Python package for graph data structures in particular, as well as PM4Py, a Python package for process mining [2]. To this end, we can implement and evaluate the chosen techniques in a unified environment. In the spirit of promoting open research and fostering transparency in model training and evaluation, we provide access to the source code at `https://github.com/myrmsch/From-Black-Box-to-Glass-Box-Evaluating-Faithfulness-of-Process-Predictions-with-GCNNs.git` [9].

[3] BPI Challenge 2017 - Loan eventlog

application status changes, offer status changes, and workflow events. Following [31], we reduce the eventlog to workflow events to gain clarity of process flows. Workflow events represent the actions of the employees of the credit-granting institution with the information system. Our objective is to train a classifier that predicts whether the loan will be granted or not.

The **review eventlog**[4] is a published synthetic eventlog depicting the paper approval process. First, the decision maker invites three reviewers to review the article, leading to an accept, reject, or no reviewer's response. Subsequently, the editor collects the reviews and makes a final approval decision. In doubt, the reviewer can request further reviews. This can continue in a loop until a final decision is reached. Thus, the result of the process is to accept or reject the paper based on previous rounds of reviews conducted. Predicting whether an article will ultimately be accepted or rejected is the goal of classifying the process outcome.

We retrieve four datasets to train on during data pre-processing since we transform the datasets twofold, namely activity-based (ab) and event-based (eb). For a better understanding of the data, we compute the basic statistics of the pre-processed datasets (eventlogs) and provide advanced graph-based characteristics in Table 1.

	Instances	**Instances class 0**	**Instances class 1**	**Mean number of nodes**	**Max. number of nodes**	**Min. number of nodes**	**Number of features**
Review-Log (ab)	10.000	4.932	5.068	15	16	11	19
Review-Log (eb)	10.000	4.932	5.068	22	84	9	29
Loan-Log (ab)	31.411	14.183	17.228	16	26	5	15
Loan-Log (eb)	31.411	14.183	17.228	24	151	3	183

Table 1: Overview of pre-processed eventlogs.

5.3 Results and Analysis

The results of the comparative evaluation of GCNN predictions on synthetic and real-world datasets are discussed in the following. The evaluation metric accuracy for the trained model on train, validation, and test set for whole graphs as input reports Table 2. The review dataset achieves high accuracy values for both activity-based and event-based datasets. It is worth highlighting that the activity-based approach results in almost four percentage points better accuracy for the loan event log. One possible explanation is the transformation into graph structures, where event-based coding leads to more nodes.

[4] Synthetic eventlog - Review example

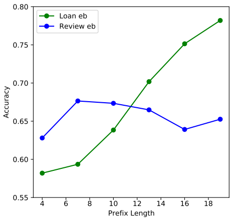


Fig. 4: Accuracy depending on the prefix length of the underlying buckets based on the validation set.

Model	Epochs	Accuracy Train	 Val	 Test
GCN review log ab	50	99.9%	99.8%	99.8%
GCN review log eb	200	99.9%	100.0%	100.0%
GCN loan log ab	300	89.4%	88.9%	88.0%
GCN loan log eb	300	92.0%	86.6%	85.7%

Table 2: Overview of training results for GCNN models trained on whole instance graphs.

To study the accuracy concerning prefix length, we illustrate the achieved validation accuracy values for different prefix lengths for both datasets (Figure 4). The accuracy of the predictions for the review dataset initially increases with increasing prefix length but decreases with the prefix length of ten. Even with a more extended prefix, the prediction quality is significantly lower than with whole graphs as input. This shows the importance of the last nodes for the prediction of the model.

Quantitative Analysis of Explainabililty The results of the quantitative comparative evaluation of explanation methods summarize Figures 5 and 6. The following observations emerge from the results.

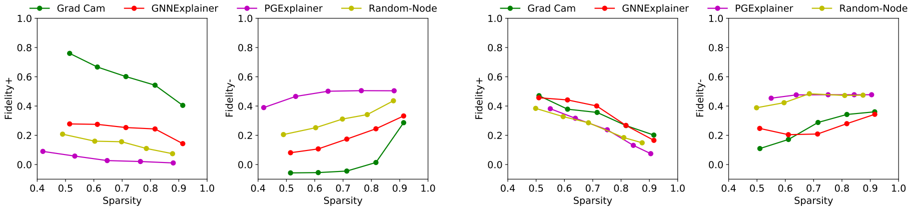


Fig. 5: Results of quantitative analysis of explainable techniques applied to the loan dataset (eb results on the left; ab results on the right).

All evaluation methods outperform the baseline (consisting of randomly selected graph nodes from the original graph) for the activity-based encoded review dataset. Best results achieve Grad-CAM for explanations with a sparsity of 0.5. We interpret the fidelity+ of almost 0.8 and fidelity- of almost 0 as meaningful explanations as they can identify distinctive graph components. The PGExplainer approaches Grad-CAM fidelity performance with increasing sparsity, and GNNExplainer achieves better values than the baseline but remains just below it. When applied to the event-based review dataset, PGExplainer and Grad-CAM score worse than the activity-based approach. GNNEXplainer has comparable fidelity values concerning the baseline and barely outperforms them. Grad-CAM shows the best fidelity+ values (0.58) at a sparsity of 0.5. However, the fidelity values are significantly lower than for the activity-based approach. As sparsity increases, fidelity values increase, leading to less meaningful explanations. Notably, at a sparsity of 0.6, the PGExplainer obtains a fidelity+ score of 0.1, almost 0.3 lower than the baseline. Accordingly, it identifies insignificant graph components as it fails to learn appropriate structures. Moreover, all techniques fail to determine unique discriminative components, as was previously the case with the activity-based approach.

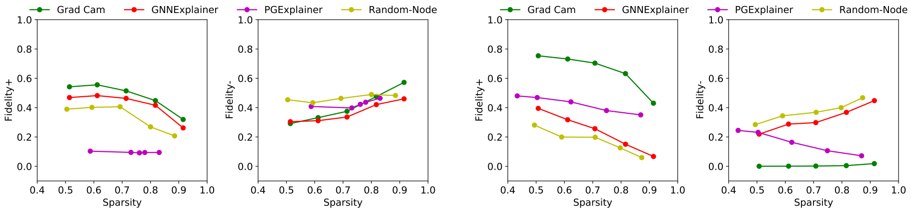


Fig. 6: Results of quantitative analysis of explainable techniques applied to the review dataset (eb results on the left; ab results on the right).

The evaluation of the activity-based loan eventlog shows that compared to the baseline method, only Grad-CAM and GNNExplainer achieve both a higher fidelity+ and fidelity- for all sparsity values. Note that fidelity- is generally more distinctive from the baseline, meaning on average the excluded graph components lead to a smaller change in the prediction. This indicates that only a part of decisive graph components are identified. In contrast to the activity-based loan dataset, Grad-CAM can achieve high fidelity values on the event-based loan dataset. At a sparsity of 0.5, it achieves a fidelity+ of 0.8 and a fidelity- below 0. The reached scores compare to the results for the activity-based review dataset. With increasing sparsity, fidelity+ decreases while fidelity- decreases. GNNExplainer also outperforms the baseline but does not generate meaningful explanations as Grad-CAM. The PGExplainer performs worse than the baseline for all sparsity values considered. Parameter tuning has further shown that the prediction scores become worse as the output loss of the PGExplainer decreases. We suggest that the algorithm learns sub-optimal structures.

We demonstrate the meaningfulness of explanations using model-specific explainable techniques for GNNs. Grad-CAM obtains high fidelity+ and low fidelity- values for the activity-based review and the event-based loan dataset. However, we notice that the performance of Grad-CAM varies across all datasets due to model training. The learning rate significantly affects the performance of Grad-GAM, while the prediction accuracy remains the same. The GNNExplainer consistently outperforms the baseline on all datasets but generates less meaningful explanations than Grad-CAM. In contrast, the PGExplainer can only exceed the baseline for the activity-based review dataset. The technique underperforms the baseline for event-based datasets, suggesting that the underlying graph structure cannot be processed.

Qualitative Analysis of Explainability After examining the performance of the explainability methods, we visualize the explanation graphs to illustrate their interpretability (see Figure 7).

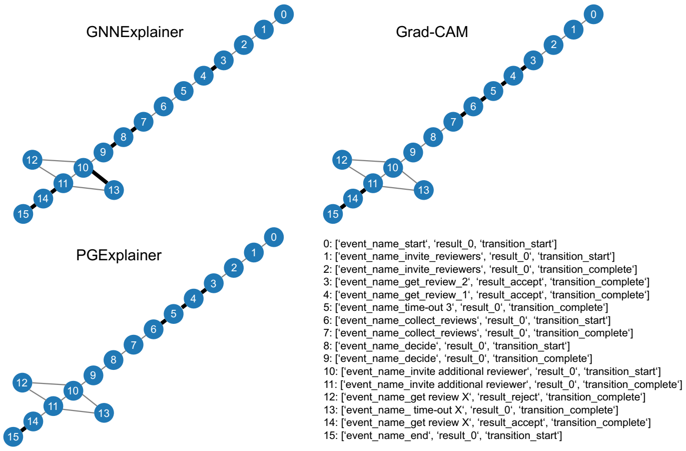

GNNExplainer
Grad-CAM
PGExplainer
0: [‘event_name_start‘, ‘result_0, ‘transition_start‘]
1: [‘event_name_invite_reviewers‘, ‘result_0‘, ‘transition_start‘]
2: [‘event_name_invite_reviewers‘, ‘result_0‘, ‘transition_complete‘]
3: [‘event_name_get_review_2‘, ‘result_accept‘, ‘transition_complete‘]
4: [‘event_name_get_review_1‘, ‘result_accept‘, ‘transition_complete‘]
5: [‘event_name_time-out 3‘, ‘result_0‘, ‘transition_complete‘]
6: [‘event_name_collect_reviews‘, ‘result_0‘, ‘transition_start‘]
7: [‘event_name_collect_reviews‘, ‘result_0‘, ‘transition_complete‘]
8: [‘event_name_decide‘, ‘result_0‘, ‘transition_start‘]
9: [‘event_name_decide‘, ‘result_0‘, ‘transition_complete‘]
10: [‘event_name_invite additional reviewer‘, ‘result_0‘, ‘transition_start‘]
11: [‘event_name_invite additional reviewer‘, ‘result_0‘, ‘transition_complete‘]
12: [‘event_name_get review X‘, ‘result_reject‘, ‘transition_complete‘]
13: [‘event_name_ time-out X‘, ‘result_0‘, ‘transition_complete‘]
14: [‘event_name_get review X‘, ‘result_accept‘, ‘transition_complete‘]
15: [‘event_name_end‘, ‘result_0‘, ‘transition_start‘]

Fig. 7: Visualization of the generated explanation graphs for an instance of class 1 in the review dataset (ab).

We use masks of the explanation graph to represent graph-based structures that are significant, starting from a single graph instance. The explanation technique ranks bold black edges as relevant. To showcase the analysis, we use the activity-based review dataset due to its easy-to-understand process model. The review process involves preparing multiple reviews that we deem relevant for the final decision on rejection or acceptance. If only rejecting reviews are available, the overall process is likely to result in rejection. Furthermore, we anticipate that reviews that led to a final acceptance will be more critical to the model.

Grad-CAM marks both the edges between nodes three to six, which contain the results of the first three reviews and the edges with the included expert opinion. However, GNNExplainer marks the front reviews but not the last review, and PGExplainer does not mark edges to the crucial nodes, so its explanation is weak. Our results indicate that the graph-based representation form of the explanations can intuitively present the results to people. The explanations of Grad-Cam, agree with the superior fidelity values. Similarly, the visualizations show that the other techniques only partially recognize decisive areas of the original graph, highlighting the worse quantitative evaluation values.

6 Discussion and Future Research

With the increase in the use of explainable methods in PBPM, we ask the key question if *explanations are faithfully* describing the underlying model. We contribute to this knowledge base by exploring the capabilities of a novel, explainable GCNN-based approach to predicting process outcomes using quantitative and qualitative evaluation. An essential novelty of our approach is the consideration of objective performance measures for model-specific post-hoc GCNN explanations to evaluate their faithfulness.

Limitations concerning our approach include that our evaluation uses three explainability methods and two standard evaluation metrics. While we employ state-of-the-art algorithms and metrics, it is still possible that other explainable algorithms provide superior performance. In the same way, our evaluation is bound to two datasets, which might not yield generalizable results. However, our initial findings on the limits of explainable GNN provide a starting point for exploring the circumstances under which one post-hoc explainer should be preferred over another. The results of our research will enable us to shed light on the application of PBPM in practice. Various stakeholders can use our findings to develop explainable PBPM systems. For example, explainable PBPM enables process owners to reverse engineer processes by extracting information about the knowledge learned during the training phase. In this context, explanation methods would enable an understanding of the main drivers or influences on process performance.

7 Conclusion

Facing the practical need for explainable PBPM, we empirically evaluate explainable outcome-oriented PBPM methods. Our work compares different explainability techniques specialized for GCNNs in PBPM. We present promising results for GCNN explainability methods. Through an experimental evaluation, we demonstrate that exploiting the underlying graph structure of process data enables the generation of intuitive explanations for humans in the form of graphs and offers promising quantitative results. Our comprehensive experimental analysis shows that the examined GNN-based explanation methods can be implemented successfully to explain predictions for whole instance graphs and prefixes. Grad-CAM has achieved the best results in both use cases in this context. We see it as beneficial to compare other explainability techniques and create a broader spectrum of possible applications in PBPM for practice to foster the acceptance of algorithms in user interaction.

References

1. Agarwal, C., Zitnik, M., Lakkaraju, H.: Probing gnn explainers: A rigorous theoretical and empirical analysis of gnn explanation methods. In: arXiv preprint arXiv:2106.09078v2. p. 11 (2022)

2. Berti, A., Van Zelst, S.J., van der Aalst, W.: Process mining for python (pm4py): bridging the gap between process-and data science. In: arXiv preprint arXiv:1905.06169. pp. 1–4 (2019)
3. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning. In: arXiv preprint arXiv:1702.08608. pp. 1–13 (2017)
4. Du, M., Liu, N., Hu, X.: Techniques for interpretable machine learning. Communications of the ACM **63**(1), 68–77 (2019)
5. Faber, L., K. Moghaddam, A., Wattenhofer, R.: When comparing to ground truth is wrong: On evaluating gnn explanation methods. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. pp. 332–341 (2021)
6. Fey, M., Lenssen, J.E.: Fast graph representation learning with pytorch geometric. In: arXiv preprint arXiv:1903.02428. pp. 1–9 (2019)
7. Hagberg, A., Swart, P., S Chult, D.: Exploring network structure, dynamics, and function using networkx. Tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States) (2008)
8. Harl, M., Weinzierl, S., Stierle, M., Matzner, M.: Explainable predictive business process monitoring using gated graph neural networks. Journal of Decision Systems **29**(sup1), 312–327 (2020)
9. Hein, B., Schaschek, M.: GitHub - myrmsch/From-Black-Box-to-Glass-Box-Evaluating-Faithfulness-of-Process-Predictions-with-GCNNs — github.com. `https://github.com/myrmsch/From-Black-Box-to-Glass-Box-Evaluating-Faithfulness-of-Process-Predictions-with-GCNNs/tree/main`, [Accessed 31-07-2023]
10. Hooker, S., Erhan, D., Kindermans, P.J., Kim, B.: A benchmark for interpretability methods in deep neural networks. Advances in neural information processing systems **32** (2019)
11. Jacovi, A., Goldberg, Y.: Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? In: arXiv preprint arXiv:2004.03685. pp. 1–15 (2020)
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: arXiv preprint arXiv:1412.6980. pp. 1–15 (2014)
13. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks pp. 1–14 (2016)
14. Li, M.M., Huang, K., Zitnik, M.: Representation learning for networks in biology and medicine: Advancements, challenges, and opportunities. In: arXiv e-prints: 2104.04883. pp. 1–18 (2021)
15. Li, P., Yang, Y., Pagnucco, M., Song, Y.: Explainability in graph neural networks: An experimental survey. In: arXiv preprint arXiv:2203.09258. pp. 1–8 (2022)
16. Liu, M., Luo, Y., Wang, L., Xie, Y., Yuan, H., Gui, S., Yu, H., Xu, Z., Zhang, J., Liu, Y., et al.: Dig: a turnkey library for diving into graph deep learning research. Journal of Machine Learning Research **22**(240), 1–9 (2021)
17. Luo, D., Cheng, W., Xu, D., Yu, W., Zong, B., Chen, H., Zhang, X.: Parameterized explainer for graph neural network. Advances in neural information processing systems **33**, 19620–19631 (2020)
18. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. Artificial intelligence **267**, 1–38 (2019)
19. Oberdorf, F., Schaschek, M., Stein, N., Flath, C.M.: Neural process mining: Multi-headed predictive process analytics in practice. In: Proceedings of the 29th European Conference on Information Systems (ECIS) (2021)

20. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
21. Philipp, P., Georgi, R.X.M., Beyerer, J., Robert, S.: Analysis of control flow graphs using graph convolutional neural networks. In: 2019 6th International Conference on Soft Computing & Machine Intelligence (ISCMI). pp. 73–77. IEEE (2019)
22. Pope, P.E., Kolouri, S., Rostami, M., Martin, C.E., Hoffmann, H.: Explainability methods for graph convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10772–10781 (2019)
23. Ribeiro, M.T., Singh, S., Guestrin, C.: " why should i trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1135–1144 (2016)
24. Rizzi, W., Comuzzi, M., Di Francescomarino, C., Ghidini, C., Lee, S., Maggi, F.M., Nolte, A.: Explainable predictive process monitoring: A user evaluation. In: arXiv preprint arXiv:2202.07760. p. 51 (2022)
25. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence **1**(5), 206–215 (2019)
26. Sachan, S., Yang, J.B., Xu, D.L., Benavides, D.E., Li, Y.: An explainable ai decision-support-system to automate loan underwriting. Expert Systems with Applications **144**, 113100 (2020)
27. Sanchez-Lengeling, B., Wei, J., Lee, B., Reif, E., Wang, P., Qian, W., McCloskey, K., Colwell, L., Wiltschko, A.: Evaluating attribution for graph neural networks. Advances in neural information processing systems **33**, 5898–5910 (2020)
28. Shin, D.: The effects of explainability and causability on perception, trust, and acceptance: Implications for explainable ai. International Journal of Human-Computer Studies **146**, 102551 (2021)
29. Sindhgatta, R., Ouyang, C., Moreira, C.: Exploring interpretability for predictive process analytics. In: International Conference on Service-Oriented Computing. pp. 439–447. Springer (2020)
30. Stierle, M., Brunk, J., Weinzierl, S., Zilker, S., Matzner, M., Becker, J.: Bringing light into the darkness-a systematic literature review on explainable predictive business process monitoring techniques. In: ECIS 2021 Research-in-Progress Papers. p. 8 (2021)
31. Stierle, M., Weinzierl, S., Harl, M., Matzner, M.: A technique for determining relevance scores of process activities using graph-based neural networks. Decision Support Systems **144**, 113511 (2021)
32. Taymouri, F., Rosa, M.L., Erfani, S., Bozorgi, Z.D., Verenich, I.: Predictive business process monitoring via generative adversarial nets: the case of next event prediction. In: International Conference on Business Process Management. pp. 237–256. Springer (2020)
33. Teinemaa, I., Dumas, M., Rosa, M.L., Maggi, F.M.: Outcome-oriented predictive process monitoring: Review and benchmark. ACM Transactions on Knowledge Discovery from Data (TKDD) **13**(2), 1–57 (2019)
34. Velmurugan, M., Ouyang, C., Moreira, C., Sindhgatta, R.: Evaluating explainable methods for predictive process analytics: A functionally-grounded approach. In: arXiv preprint arXiv:2012.04218. p. 15 (2020)

35. Velmurugan, M., Ouyang, C., Moreira, C., Sindhgatta, R.: Developing a fidelity evaluation approach for interpretable machine learning. In: arXiv preprint arXiv:2106.08492. pp. 1–28 (2021)
36. Velmurugan, M., Ouyang, C., Moreira, C., Sindhgatta, R.: Evaluating fidelity of explainable methods for predictive process analytics. In: International Conference on Advanced Information Systems Engineering. pp. 64–72. Springer (2021)
37. Venugopal, I., Töllich, J., Fairbank, M., Scherp, A.: A comparison of deep-learning methods for analysing and predicting business processes. In: 2021 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2021)
38. Wanner, J., Heinrich, K., Janiesch, C., Zschech, P.: How much ai do you require? decision factors for adopting ai technology. In: Proceedings of the 31st International Conference on Information Systems (ICIS). p. 10 (2020)
39. Weinzierl, S.: Exploring gated graph sequence neural networks for predicting next process activities. In: International Conference on Business Process Management. pp. 30–42. Springer (2021)
40. Weinzierl, S., Zilker, S., Brunk, J., Revoredo, K., Matzner, M., Becker, J.: Xnap: Making lstm-based next activity predictions explainable by using lrp. In: International Conference on Business Process Management. pp. 129–141. Springer (2020)
41. Wickramanayake, B., He, Z., Ouyang, C., Moreira, C., Xu, Y., Sindhgatta, R.: Building interpretable models for business process prediction using shared and specialised attention mechanisms. In: arXiv preprint arXiv:2109.01419. p. 40 (2021)
42. Wiegreffe, S., Pinter, Y.: Attention is not not explanation. In: arXiv preprint arXiv:1908.04626 (2019)
43. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y.: A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems **32**(1), 4–24 (2021)
44. Ying, Z., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: Gnnexplainer: Generating explanations for graph neural networks. Advances in neural information processing systems **32** (2019)
45. Yuan, H., Yu, H., Gui, S., Ji, S.: Explainability in graph neural networks: A taxonomic survey. In: arXiv preprint arXiv:2012.15445. p. 14 (2020)