

Contribution Title^{*}

First Author¹[0000–1111–2222–3333], Second Author^{2,3}[1111–2222–3333–4444], and
Third Author³[2222–3333–4444–5555]

¹ Princeton University, Princeton NJ 08544, USA

² Springer Heidelberg, Tiergartenstr. 17, 69121 Heidelberg, Germany
lncs@springer.com

<http://www.springer.com/gp/computer-science/lncs>

³ ABC Institute, Rupert-Karls-University Heidelberg, Heidelberg, Germany
{abc,lncs}@uni-heidelberg.de

Abstract The abstract should briefly summarize the contents of the paper in 15–250 words.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

1.1 Motivation

Many processes, often medical, economical, or administrative in nature, are governed by sequential events and their contextual environment. Many of these events and their order of appearance play a crucial part in the determination of every possible outcome?. With the rise of AI and the increased abundance of data in recent years, several techniques emerged that help to predict the outcomes of complex processes in the real world. A field that focuses on modelling processes is Process Mining (PM).

Research in the Process Mining discipline has shown that it is possible to predict the outcome of a particular process fairly well??.

For instance, in the medical domain, models have been shown to predict the outcome or trajectory of a patient’s condition?. In the private sector, process models can be used to detect faults or outliers. The research discipline Deep Learning has shown promising results within domains that have been considered difficult for decades. The Moravex Paradox?, which postulates that machines are capable of doing complex computations easily while failing in tasks that seem easy to humans such as object detection or language comprehension, does not hold anymore. Meaning that with enough data to learn, machines are capable of learning highly sophisticated tasks better than any human. The same holds for predictive tasks. However, while many prediction models can predict certain outcomes, it remains a difficult challenge to understand their reasoning.

This difficulty arises from models, like neural networks, that are so-called *blackbox models*. Meaning, that their inference is incomprehensible, due to the

^{*} Supported by organization x.

vast amount of parameters involved. This lack of comprehension is undesirable for many fields like IT or finance. Not knowing why a loan was given, makes it impossible to rule out possible biases. Knowing what will lead to a system failure will help us knowing how to avoid it. In critical domains like medicine, the reasoning behind decisions becomes crucial. For instance, if we know that a treatment process of a patient reduces the chances for survival, we want to know which treatment step is the critical factor we ought to avoid. To summarise, knowing the outcome of a process often leads us to questions on how to change it. Formally, we want to change the outcome of a process instance by making it maximally likely with as little interventions as possible?. Figure 1 is a visual representation of the desired goal.

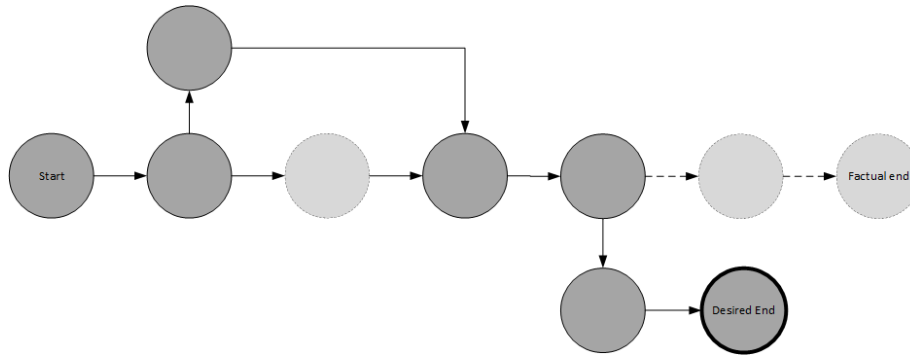


Figure 1: This figure illustrates a model, that predicts a certain trajectory of the process. However, we want to change the process steps in such a way, that it changes the outcome.

One way to better understand the Machine Learning (ML) models lies within the eXplainable AI (XAI) discipline. XAI focuses the developments of theories, methods, and techniques that help explaining blackbox models to humans. Most of the discipline's techniques produce explanations that guide our understanding. Explanations can come in various forms, such as IF-THEN rules(?, p.90) or feature importance(?, p.45), but some are more comprehensible for humans than others.

A prominent and human-friendly approach are *counterfactuals*(?, p. 221). Counterfactuals within the AI framework help us to answer hypothetical "what-if" questions. Basically, if we know *what* would happen *if* we changed the execution of a process instance, we could change it for the better. In this thesis, we raise the question how we can use counterfactuals to change the trajectory of a process models' prediction towards a desired outcome. Knowing the answers not only increases the understanding of blackbox models, but also help us avoid or enforce certain outcomes.

1.2 Problem Space

In this thesis, we approach the problem of generating counterfactuals for processes. The literature has provided a multitude of techniques to generate counterfactuals for AI models, that are derived from static data⁴. However, little research has focussed on counterfactuals for dynamic data⁵.

For process data, the literature often uses terms like structured and semi-structured, as they are related to the staticity and dynamicity. Both, structuredness and semi-structuredness, often relate to the data model, in which we structure the information at hand. As static data neither changes over time nor changes its structure, we can use structured data-formats such as tables to capture the information where each data point is an independent entity. We can take the MNIST dataset^{??} or Iris dataset^{??} as examples for structured and static data. In both datasets, all data points are independent and have the same amount of attributes. In contrast, semi-structured data does not have to follow these strict characteristics. Here, data points often belong to a group of data points which constitutes the full entity. Furthermore, the attributes of each data point may vary. The grouping mechanism could take the form of associative links, class associations or temporal cause-effect relationships. Examples of these are Part-of-Speech datasets like Penn Treebank set[?]. Here, we often associate each data point with a sentence. However, the temporal relationship between words is debatable and hence, whether the data is *dynamic*, as well. So, not all semi-structured datasets are dynamic and vice versa. However, structured data will almost always be static, with the exception of time-series. Lastly, there is also unstructured data, which does not incorporate any specific data model. Corpora like the Brown dataset[?], for instance, are collections of text heavy unstructured information. In Figure 2, we show various examples of data.

A major reason, why there has not been much research on counterfactuals for dynamic semi-structured data, emerges from a multitude of challenges, when dealing with counterfactuals and sequences. Three of these challenges are particularly important.

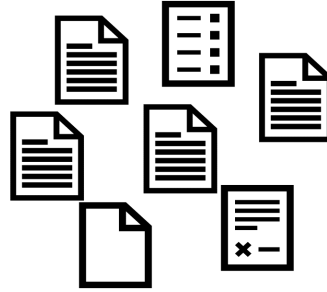
First, counterfactuals within AI attempt to explain outcomes which never occurred. *What-if* questions often refer to hypothetical scenarios. Therefore, there is no evidential data from which we can infer predictions. Subsequently, this lack of evidence further complicates the evaluation of generated counterfactuals. In other words, you cannot validate the correctness of a theoretical outcome that has never occurred.

Second, sequential data is highly variable in length, but process steps have complicated factors, too. The sequential nature of the data impedes the tractability of many problems due to the combinatorial explosion of possible sequences. Furthermore, the data generated is seldomly one-dimensional or discrete. Henceforth, each dimension's contribution can vary in dependance of its context, time and magnitude.

⁴ With static data, we refer to data that does not change over a time dimension.

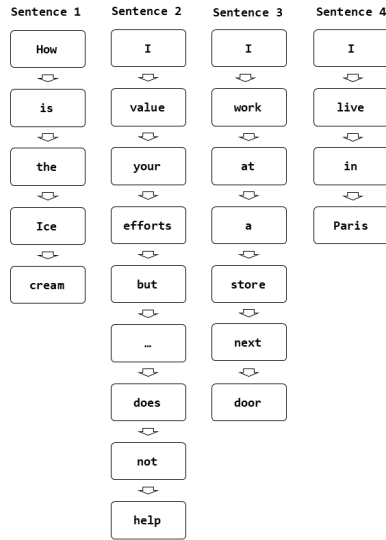
⁵ With dynamic data, we refer to data that has a temporal relationship as a major component, which is also inherently sequential

s.length	s.width	p.length	p.width	variety
6.5	2.8	4.6	1.5	Versicolor
5.8	2.7	4.1	1.0	Versicolor
6.7	3.3	5.7	2.5	Virginica
4.6	3.4	1.4	0.3	Setosa
6.4	3.2	5.3	2.3	Virginica
5.9	3.0	4.2	1.5	Versicolor
7.4	2.8	6.1	1.9	Virginica
5.5	2.4	3.8	1.1	Versicolor
5.6	2.5	3.9	1.1	Versicolor
5.0	3.4	1.5	0.2	Setosa
6.9	3.1	5.4	2.1	Virginica
5.5	2.5	4.0	1.3	Versicolor
5.7	2.6	3.5	1.0	Versicolor
5.8	2.7	3.9	1.2	Versicolor
7.6	3.0	6.6	2.1	Virginica
6.7	3.3	5.7	2.1	Virginica
5.0	3.5	1.6	0.6	Setosa
7.7	2.8	6.7	2.0	Virginica
6.4	2.7	5.3	1.9	Virginica
7.7	3.8	6.7	2.2	Virginica
5.2	3.5	1.5	0.2	Setosa
5.7	3.8	1.7	0.3	Setosa



(a) An excerpt of the MNIST dataset. This is a structured dataset.

(b) A number of heterogenous documents. A dataset like this is unstructured.



(c) Multiple sequences of words. Each word forms a sentence of different lengths. Therefore, this data is semi-structured.

Figure 2: Schematic examples of static structured, dynamic semi-structured data and unstructured data.

Third, process data often requires knowledge of the causal structures that produced the data in the first place. However, these structures are often hidden and it is a NP-hard problem to elicit them?

These challenges make the field, in which we can contribute, a vast endeavor.

1.3 Related Literature

Many researchers have worked on counterfactuals and PM. Here, we combine the important concepts and discuss the various contributions to this thesis.

1.4 Generating Counterfactuals

The topic of counterfactual generation as explanation method was introduced by ? in ???. The authors defined a loss function which incorporates the criteria to generate a counterfactual which maximizes the likelihood for a predefined outcome and minimizes the distance to the original instance. However, the solution of ? did not account for the minimisation of feature changes and does not penalize unrealistic features. Furthermore, their solution cannot incorporate categorical variables.

A newer approach by ? incorporates four main criteria for counterfactuals (see ??) by applying a genetic algorithm with a multi-objective fitness function?. This approach strongly differs from gradient-based methods, as it does not require a differentiable objective function. However, their solution was only tested on static data.

1.5 Generating Counterfactual Sequences

When it comes to sequential data most researchers work on ways to generate counterfactuals for natural language. This often entails generating univariate discrete counterfactuals with the use of Deep Learning techniques. ? and later ? are early examples of counterfactual NLP research??. Their approach strongly focuses on the manipulation of sentences to achieve the desired outcome. However, as ? puts it, their counterfactuals do not comply with *realisticness*?

Instead, ? showed that it is possible to generate realistic counterfactuals with a Generative Adversarial Model (GAN)?. They use the model to implicitly capture a latent state space and sample counterfactuals from it. Apart from implicitly modelling the latent space with GANs, it is possible to sample data from an explicit latent space. Examples of these approaches often use an encoder-decoder pattern in which the encoder encodes a data instance into a latent vector, which will be perturbed and then decoded into a similar instance??. By modelling the latent space, we can simply sample from a distribution conditioned on the original instance. ? provide an overview of the strengths and weaknesses of common generative models.

Even though, a single latent vector model can theoretically produce multivariate sequences, it may still be too restrictive to capture the combinatorial

space of multivariate sequences. Hence, most of the models within Natural Language Processing (NLP) were not used to produce a sequence of vectors, but a sequence of discrete symbols. For process instances, we can assume a causal relation between state vectors in a sequential latent space. We call models that capture a sequential latent state-space, which has causal relations, *dynamic*?. Early models of this type of dynamic latent state-space models are the well-known *Kalman-Filter* for continuous states and Hidden Markov Model (HMM) for discrete states. In recent literature, many techniques use Deep Learning to model complex state-spaces. The first models of this type were developed by ????. Their Deep Kalman Filter (DKF) and subsequent Deep Markov Model (DMM) approximate the dynamic latent state-space by modelling the latent space given the data sequence and all previous latent vectors in the sequence. There are many variations???? of ?'s model, but most use Evidence Lower-Bound (ELBO) of the posterior for the current Z_t given all previous $\{Z_{t-1}, \dots, Z_1\}$ and X_t ?

1.6 Generating Counterfactual Time-Series

Within the *multivariate time-series* literature two recent approaches yield ideas worth discussing.

First, ? introduce a case-based reasoning to generate counterfactuals?. Their method uses existing counterfactual instances, or *prototypes*, in the dataset. Therefore, it ensures, that the proposed counterfactuals are *realistic*. However, case-based approaches strongly depend on the *representativeness* of the prototypes(? , p. 192). In other words, if the model displays behaviour, which is not captured within the set of prototypical instances, most case-based techniques will fail to provide viable counterfactuals. The likelihood of such a break-down increases due to the combinatorial explosion of possible behaviours if the *true* process model has cycles or continuous event attributes. Cycles may cause infinite possible sequences and continuous attributes can take values on a domain within infinite negative and positive bounds. These issues have not been explored in the paper of ?, as it mainly deals with time series classification?. However, despite these shortcomings, case-based approaches may act as a valuable baseline against other sophisticated approaches.

The second paper within the multivariate time series field by ? also uses a case-based approach?. However, it contrasts from other approaches, as it does not specify a particular model but proposes a general framework instead. Hence, within this framework, individual components could be substituted by better performing components. Describing a framework, rather than specifying a particular model, allows to adapt the framework, due to the heterogeneous process dataset landscape. In this paper, we also introduce a framework that allows for flexibility depending on the dataset.

1.7 Generating Counterfactuals for Business Processes

So far, none of the techniques have been applied to process data.

Within PM, Causal Inference has long been used to analyse and model business processes. Mainly, due to the causal relationships underlying each process. However, early work has often attempted to incorporate domain-knowledge about the causality of processes in order to improve the process model itself. Among these, [1] approach is one of the first to include counterfactual reasoning for process optimization. [1] use counterfactuals to generate alternative solutions to treatments, which lead to a desired outcome. Again, the authors do not attempt to provide an explanation of the models outcome and therefore, disregard multiple viability criteria for counterfactuals in XAI. [1] published the most recent paper on the counterfactual generation of explanations. The authors use a known Structural Causal Model (SCM) to guide the generation of their counterfactuals. However, this approach requires a process model which is as close as possible to the *true* process model. For our approach, we assume that no knowledge about the dependencies are known.

Within the XAI context, [1] develop the first explanation method for process data. However, their work closely resembles the work of [2] and treat the task as Markov Decision Process (MDP). This extension of a regular Markov Process (MP) assumes that an actor influences the outcome of a process given the state. This formalisation allows the use of Reinforcement Learning (RL) methods like Q-learning or SARSA. However, this often requires additional assumptions such as a given reward function and an action-space. For counterfactual sequence generation, there is no obvious choice for the reward function or the action-space.

Nonetheless, both [1] and [2] contribute an important idea. The idea of incrementally generating the counterfactual instead of the full sequence. [2] has recently published an approach that builds on the same notion of incremental generation. Their approach has a very similar structure to our approach and appears to be the only one that we can compare our counterfactuals against.

For this reason, this thesis highlights some key differences and similarities. However, to understand the differences and similarities, we first have to establish some core concepts. In this section, we only discuss their approach, briefly.

The authors recognised that some processes have critical events which govern the overall outcome. Hence, by simply avoiding the undesired outcome from critical event to critical event, it is possible to limit the search space and compute viable counterfactuals. They use an extension of DiCE [3] to generate counterfactuals. However, their approach requires concrete knowledge about these critical points. We propose a Framework that avoids this constraint.

To our knowledge, the authors are also the first authors that try to optimize their counterfactual process generation based on criteria that ensure their viability. However, in our approach, we use different operationalisations to quantify the criteria.

1.8 Research Question

As we seek to make data-driven process models interpretable, we have to understand the exact purpose of this thesis. Hence, we establish the open challenges and how this thesis attempts to solve them.

1.9 Outline

2 First Section

2.1 A Subsection Sample

Please note that the first paragraph of a section or subsection is not indented. The first paragraph that follows a table, figure, equation etc. does not need an indent, either.

Subsequent paragraphs, however, are indented.

Sample Heading (Third Level) Only two levels of headings should be numbered. Lower level headings remain unnumbered; they are formatted as run-in headings.

Sample Heading (Fourth Level) The contribution should contain no more than four levels of headings. Table 1 gives a summary of all heading levels.

Table 1: Table captions should be placed above the tables.

Heading level	Example	Font size and style
Title (centered)	Lecture Notes	14 point, bold
1st-level heading	1 Introduction	12 point, bold
2nd-level heading	2.1 Printing Area	10 point, bold
3rd-level heading	Run-in Heading in Bold. Text follows	10 point, bold
4th-level heading	<i>Lowest Level Heading.</i> Text follows	10 point, italic

Displayed equations are centered and set on a separate line.

$$x + y = z$$

(1)

Theorem 1. *This is a sample theorem. The run-in heading is set in bold, while the following text appears in italics. Definitions, lemmas, propositions, and corollaries are styled the same way.*

Proof. Proofs, examples, and remarks have the initial word in italics, while the following text appears in normal font.

For citations of references, we prefer the use of square brackets and consecutive numbers. Citations using labels or the author/year convention are also acceptable. The following bibliography provides a sample reference list with entries for journal

Bibliography