Glitches in the Matrix: Threats to the Validity of Data-driven Business Process Simulation

Robert Blümel[1,2*], Lukas Kirchdorfer[1,3], Jana-Rebecca Rehse[3], Timotheus Kampik[1,4], Han van der Aa[5]

[1*] SAP Signavio, Dietmar-Hopp-Allee 16, Walldorf, Germany.
[2] Doctoral School Computer Science, University of Vienna, Währinger Str. 29, Vienna, Austria.
[3] University of Mannheim, Schloss, Mannheim, Germany.
[4] Umeå University, MIT-huset, Umeå, Sweden.
[5] University of Vienna, Nordbergstr. 1, Vienna, Austria.

*Corresponding author(s). E-mail(s): robert.bluemel@sap.com;
Contributing authors: lukas.kirchdorfer@sap.com;
rehse@uni-mannheim.de; timotheus.kampik@sap.com;
han.van.der.aa@univie.ac.at;

Abstract

Business Process Simulation (BPS) is commonly used to conduct what-if analyses and assess the impact of changes to processes and their underlying information systems. In a typical data-driven BPS approach, a simulation model is derived from an event log and evaluated through a six-step method. However, following these steps can introduce systematic errors that potentially bias evaluation results and, thereby, lead to incorrect assessments of simulation model quality. To address this, we identify five systematic errors as threats to the internal validity of evaluation results, highlighting the problem and forming the central contribution of this paper. Through empirical examples, we demonstrate that these threats can distort evaluation outcomes, both individually and in combination, even when the same process and simulation model are applied. Building upon this analysis, we outline a conceptual framework that integrates these insights with established methods from related fields to guide mitigation of the identified threats. By recognizing the threats, we highlight how simulation accuracy

may be misjudged, and by proposing this framework, we enable more valid evaluations that better support decisions about redesigning business processes and information systems.

Keywords: Business process simulation, Process mining, Data-driven discovery, Evaluation, Validity concerns

1 Introduction

Business Process Simulation (BPS) is an important tool for the design and management of (process-aware) information systems (Dumas et al. 2018). By creating a digital twin of a process and simulating a large number of hypothetical cases, organizations are able to conduct what-if analyses and assess the impact of changes to the underlying information system before implementing them (Chapela-Campa et al. 2025). Ideally, this reduces the risk and potential cost of changes, facilitating continuous process improvement and innovation. However, obtaining a simulation model for a process is inherently complex (Banks 1998). In the early days of process simulation, experts had to manually model processes and their simulation parameters, which was time-consuming and complicated (Martin et al. 2016). To address this, researchers proposed data-driven simulation approaches, which derive process knowledge and create simulation models in a fully automated way (Camargo et al. 2020, 2021, 2023; Kirchdorfer et al. 2025; López-Pintado et al. 2024; López-Pintado and Dumas 2025; Meneghello et al. 2023). Such data-driven BPS approaches leverage historic process data from an event log to establish a model that can accurately simulate a given process.

To evaluate a data-driven BPS approach, researchers rely on a research method that links the algorithm to an external indicator of its quality. As illustrated in Figure 1, the main idea is to evaluate the approach indirectly through the BPS model it discovers. For that, researchers extract an event log from a source system and use it

as input for their approach. The BPS model that the approach discovers is then used to generate a simulated log. If that simulated log resembles the original event log, the BPS model is assumed to be of high quality, which is taken as evidence for the quality of the underlying data-driven BPS approach. In this way, the research method forms a chain of transformations in which the quality of the algorithm is inferred indirectly through the performance of the BPS model and, ultimately, through a comparison between the simulated and the original log.

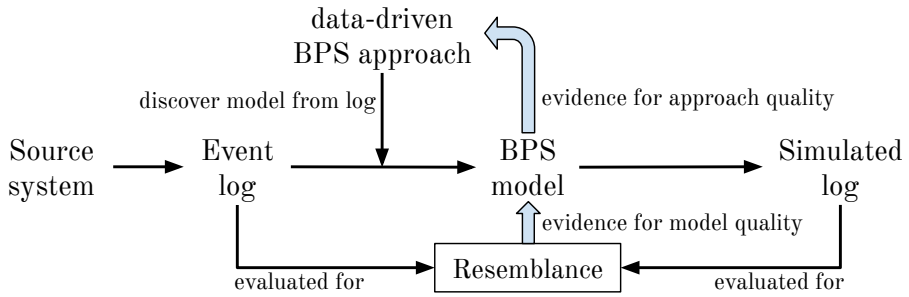


Figure 1: Evaluation logic of data-driven BPS approaches.

Although not formally standardized, the method is widely used (Camargo et al. 2019, 2020, 2021, 2022, 2023; Chapela-Campa and Dumas 2024; Estrada-Torres et al. 2021; Kirchdorfer et al. 2024, 2025; López-Pintado and Dumas 2022, 2023, 2025; López-Pintado et al. 2024; Meneghello et al. 2023, 2025) and hence apparently considered practical. However, in this work, we argue that, notwithstanding its practicality and lack of alternatives, this research method is subject to multiple validity concerns, such that the evaluation results may not be free from systematic error (Swanborn 1996). Specifically, we say that by mechanically following this method, researchers might threaten the internal validity of the evaluation results and hence misestimate the quality of their simulation model and, with that, also the quality of the BPS approach. This could imply that—when companies adopt BPS models based on these evaluation

results—information system changes that are expected to be favorable according to the performed simulations, could have undesired real-world consequences.

Understanding internal validity is key to addressing these concerns. A method "is internally valid when its manipulation is causally responsible for an observed effect" (Mendling et al. 2021). For an algorithm (in our context, a data-driven BPS approach), this means that its evaluation results should only depend on the algorithm's quality and not on other factors. Such factors include influences introduced by the research method itself, for instance, through decisions made during the processing of event logs. In this paper, we show that certain steps in the research method can significantly impact the evaluation of a BPS approach. For example, splitting an event log into a training and a test log to evaluate the resemblance between the simulated log and unseen test data results in a change to the underlying data used for evaluation. Consequently, differences in evaluation results may partially reflect these methodological decisions rather than the actual quality of the BPS approach, thereby threatening internal validity. For researchers, this means that when evaluating a data-driven BPS approach, they have to be aware of these potential impacts, so that they can mitigate them when following the research method.

The objective of this work is to ultimately increase the degree to which evaluations of data-driven BPS approaches accurately reflect the true quality of the underlying algorithms. To achieve this, we aim to enhance the validity of BPS evaluations by uncovering and mitigating threats to internal validity within the commonly used research method. To this end, the paper makes the following contributions:

(1) We explicate and assess both the structure and steps of the commonly used research method for data-driven BPS and identify five threats to its internal validity that originate from these steps.

(2) For each threat, we show how it can impact BPS evaluation results and lead to inaccurate conclusions about the quality of a BPS approach. We illustrate this with examples based on event logs typically used in BPS evaluations.

(3) We further quantify the joint effects of the threats, examine how structural characteristics of the log affect the severity of the trade-off between data splitting and leakage, and show that the resulting distortions can produce conflicting conclusions in what-if analyses.

(4) We outline a conceptual framework that embeds concrete refinements into the established research method, providing procedural guidance for addressing the identified threats, where possible.

With these contributions, we address a gap between existing methodical considerations and empirical practice in evaluating data-driven BPS. Although elements of the validity threats we identify have been explored in different research communities, such as general risks of simulation in organizations (van der Aalst 2015), validity issues in predictive process monitoring (PPM) (Weytjens and De Weerdt 2022), and classical simulation theory (Law 2015), a systematic identification and empirical demonstration of such threats in the context of data-driven BPS is missing. However—as we demonstrate in this paper—in the context of data-driven BPS, these insights are partially not incorporated into the research method, and key requirements for internal validity are frequently overlooked or violated in practice. Our work thus draws attention to these threats in a structured way and offers concrete guidelines for addressing them.

The remainder of the paper is structured as follows. Section 2 discusses the background in evaluations in simulation and process mining. Section 3 describes how we derive the five threats to validity. Section 4 presents the identified validity threats in detail and shows their effects on BPS evaluation, while Section 5 shows joint effects and implications experimentally. Section 6 proposes a conceptual framework to mitigate these threats, before Section 7 concludes the paper.

2 Background

Analyzing research methods for threats and proposing recommendations to mitigate them has a long history. Such analyses typically gain relevance after new research methods have been introduced or have matured, since it is then possible to identify which specific threats apply. Simulation research provides an example of this development: research methods were first introduced, followed by systematic analyses of their threats to validity. In this section, we trace the intertwined development of simulation research methods and the recognition of threats to their validity, serving as a reference for positioning our analysis of data-driven BPS. Section 2.1 summarizes the broader computer simulation literature, where structured methods and corresponding threats are well documented. Section 2.2 then turns to BPS, discussing an early research method and its associated threats, as well as the more recent shift to data-driven BPS approaches that derive models from event logs and evaluate them using methods inspired by PPM. While threats are well explored in simulation and increasingly in PPM, they have not yet been systematically analyzed for data-driven BPS, motivating the research gap derived in Section 2.3.

2.1 Simulation

Simulation research has a long history of developing research methods for conducting a simulation study, including manually building models and ensuring their validity. In the following, we first highlight these established methods before discussing literature that identifies threats to them.

Research methods. The general literature on simulation describes multiple research methods for conducting a simulation study (Balci 2012; Law 2015; Banks 1998). These methods guide the transformation from problem formulation to executable models and decision-relevant evidence, emphasizing real-world applicability. Common steps across these methods include problem definition, data collection and conceptual

model development, model implementation, experimentation and analysis, as well as documentation and presentation (Balci 2012; Law 2015; Banks 1998).

A further common characteristic of these methods is their iterative structure, with verification (building the model right) and validation (building the right model) embedded throughout. Other literature refines these activities by linking the system to be simulated, the conceptual model, and the computerized simulation model through iterative checks (Sargent 2001), formulating principles for their execution (Balci 1994), and providing guidance on selecting and embedding validation techniques into simulation model development (Troost et al. 2023; Roungas et al. 2018).

Threats to the research methods. Based on the above research methods, further work has identified threats that may compromise the validity of simulation studies. These threats, often referred to as pitfalls, are observed across all steps of a simulation study. Typical examples include: (i) unclear or missing objectives that yield models answering the wrong question (Law 2003; Sadowski 1989), (ii) inadequate or insufficient data that distort inputs (Law 2003; Banks and Chwif 2011), (iii) insufficient verification and validation that produces misleading conclusions (Law 2003; Pace 2004; Banks and Chwif 2011), and (iv) imbalanced complexity or poor communication, which, even for technically correct models, make experiments infeasible or results untrusted by stakeholders (Barth et al. 2012; Sadowski 1989; Pace 2004).

Additional work targets domain- or audience-specific threats, refining general simulation threats by focusing on managerial and researcher-oriented perspectives. These include project management deficiencies, insufficient skills, missing scientific questions, tunnel vision, and threats to internal validity such as inappropriate experimental designs, biased simplifying assumptions, and inconsistent datasets (Williams and Ülgen 2012; Koivisto 2017; Uhrmacher 2012; de França and Travassos 2015). In parallel to the identification of these threats to the research methods proposed for simulation, BPS has developed further into a new direction, as we outline in the following.

2.2 Process Mining

Process mining comprises methods that enable the data-driven analysis of organizational processes by analyzing event data recorded by information systems (Van der Aalst 2016). The emergence of process mining as a data science discipline (Van der Aalst 2016) requires more empirical research methods, which introduce greater threats to validity than, e.g., mathematical proofs (Rehse et al. 2024). Researchers have focused on identifying and addressing these threats in multiple process mining sub-disciplines, particularly process discovery and PPM (predictive process monitoring) (Rehse et al. 2024; Van der Werf et al. 2023; Mendling et al. 2021; Weytjens and De Weerdt 2022). In the following, we first emphasize similar advancements associated with data-driven BPS approaches and then highlight their methodological alignment with PPM literature.

Business process simulation (BPS). BPS is a tool for organizations to conduct what-if analyses on hypothetical changes to their processes or information systems, allowing them to assess the impact of changes before implementing them and thereby reducing risk and cost. BPS research has evolved through two main phases: (i) an early research method with associated threats, and (ii) modern data-driven approaches that derive models from historical data and focus on quantitative evaluation.

Early BPS research method and threats. Early work on BPS-specific research methods and threats has been closely aligned with general simulation research methods. For instance, van der Aalst (2015) proposed an end-to-end research method covering problem definition, model conceptualization, validation, and the use of simulation results to address the problem. The threats identified for this method either align with those known from general simulation research (Section 2.1), such as unclear problem definitions, hidden assumptions, and inappropriate levels of detail, or are specific to BPS, where real-world human behavior may be misrepresented, for example, when models ignore multitasking, variable work pace, or batch execution (van der Aalst 2015).

Data-driven BPS. Since the proposal of this first research method and the associated threats, researchers have introduced various data-driven BPS approaches, which automatically derive a simulation model from historic data using process mining (Park et al. 2024). This development has been adopted for different simulation paradigms such as discrete event simulation (Camargo et al. 2020, 2021, 2023; Meneghello et al. 2023; López-Pintado et al. 2024; López-Pintado and Dumas 2025), system dynamics (Pourbafrani and van Der Aalst 2022), and agent-based modeling and simulation (Bemthuis and Lazarova-Molnar 2025; Kirchdorfer et al. 2024), as well as across various business process domains (see, e.g., (Oldenburg et al. 2025; Halawa et al. 2021)). This shift has also changed how research on data-driven BPS is conducted, with quantitative evaluations of BPS models receiving increased attention. Those evaluations focus on how well the discovered simulation model reflects the original data (Rozinat et al. 2009).

Two developments have shaped the evaluation of data-driven approaches. First, splitting is essential to ensure that model discovery and evaluation are performed on disjoint data, thereby preventing overly optimistic results. This has evolved from no splitting (Rozinat et al. 2009) to temporal splitting to capture temporal dependencies (Camargo et al. 2019). Second, evaluation metrics complement data splitting by providing systematic ways to quantify how faithfully simulated data reproduces observed process behavior. Initially, these metrics covered control-flow, data, and resource perspectives without automation (Rozinat et al. 2009), before automated evaluation emerged through trace-level comparisons on control-flow and time using edit distance and mean absolute error (Camargo et al. 2019). A significant advancement was the adoption of earth mover's distance-based metrics to compare activity durations (Camargo et al. 2020) and the approach's extension to eight metrics across control-flow, time, congestion, and resource perspectives (Chapela-Campa et al. 2023, 2025). Today, this approach is widely adopted in data-driven BPS research (Camargo

et al. 2021, 2023; Meneghello et al. 2023; López-Pintado et al. 2024; López-Pintado and Dumas 2025; Kirchdorfer et al. 2025; Chapela-Campa and Dumas 2024; Kirchdorfer et al. 2025). A recent evaluation framework shifts from comparing logs to assessing downstream utility, evaluating whether PPM models trained on simulated logs achieve performance comparable to those trained on real logs (Özdemir et al. 2025). However, it remains to be seen whether this framework will be widely adopted in practice or replace the research method we address in this paper.

Predictive process monitoring (PPM). PPM aims to predict the future of ongoing process cases based on partial execution traces (prefixes) and historical event data. Typical examples include predicting the remaining time until a case completes, the next activity in the process, or the final outcome (e.g., whether an order will be delivered) (Di Francescomarino and Ghidini 2022).

In terms of evaluation strategy, PPM is similar to BPS (Pfeiffer et al. 2025): both split historical data into training and test logs, apply models to predict or simulate process behavior, and compare predictions or simulations with observed outcomes. Despite these similarities, BPS and PPM differ in the objectives for their evaluation: PPM predicts the next steps or outcomes of individual process cases (Di Francescomarino and Ghidini 2022), requiring evaluation at the event level, whereas BPS simulates behavior at the system level, focusing on reproducing overall process behavior (Chapela-Campa et al. 2025).

These objectives lead to three main differences in evaluation. First, event log splitting differs: PPM splits on trace prefixes (Weytjens and De Weerdt 2022), whereas BPS splits at the trace level, resulting in different procedures for executing the split. Second, the approaches themselves vary: PPM relies heavily on machine learning, whereas BPS only partly uses machine learning and instead models overall process behavior, which can introduce distinct threats. Third, evaluation metrics differ: PPM

metrics assess the accuracy of individual event predictions, whereas BPS metrics measure how well the simulated process reproduces overall behavior. With that, these differences affect several steps in the evaluation of both PPM and BPS approaches.

Efforts to enhance PPM evaluation validity have addressed three key areas. First, event log splitting has evolved from random (Senderovich et al. 2017) to temporal (Weytjens and De Weerdt 2022) to preserve intra-log dependencies between the training and test logs. Second, research has analyzed event logs for trends, such as changes in process behavior over time, which inform evaluations by revealing underlying patterns (Wuyts et al. 2023). Third, benchmark datasets enhance comparability across evaluations, with researchers proposing measures to address threats to evaluation validity and comparability, such as the use of different event logs across studies, improper splitting, and biases in case numbers and durations at the boundaries of event logs (Weytjens and De Weerdt 2022).

2.3 Research Gap

Comparing the research methods of general simulation and early non-data-driven BPS with that of newer data-driven BPS reveals three key distinctions.

(1) **Scope differences:** Whereas general simulation literature and early BPS work propose research methods covering steps from problem definition to the communication of results, research on data-driven BPS focuses primarily on model discovery and quantitative evaluation, with less emphasis on the end-to-end research method required to obtain a data-driven BPS model. Consequently, several steps and associated threats emphasized in general simulation are less relevant or may apply differently in data-driven BPS.

(2) **Validation focus:** Whereas general simulation research methods are designed to facilitate various validation techniques, with face validity playing a predominant role in practice (Sargent 2020), data-driven BPS relies primarily on historic data

validation. This shift is important because the validation technique dictates the procedural steps and introduces different potential threats.

(3) **Domain specificity:** Whereas classical simulation research methods are designed for general applicability across domains, BPS is inherently tied to organizational processes. Together with the focus on historic data validation, this makes the development and evaluation of data-driven BPS models highly dependent on domain-specific data structures. As highlighted in the related PPM literature, this can introduce threats that are unique to this domain.

Taken together, these differences imply that threats from general simulation are not all directly transferable to data-driven BPS, and, conversely, that not all threats to data-driven BPS validity can be derived from general simulation literature. From these observations, we identify three concrete research gaps:

(1) There is no comprehensive description of the research method currently used in data-driven BPS. This is required as a basis for analyzing its threats.

(2) It remains unclear how these threats affect evaluation outcomes in data-driven BPS. Concrete examples could showcase how these threats affect the research method and impact results.

(3) Without an explicated research method for data-driven BPS and its threats, there is no domain-specific framework linking common choices in the research method to the validity risks they introduce and to practicable mitigations.

In the following, we address these gaps by (i) explicating the data-driven BPS research method and identifying its threats to internal validity (Section 3), (ii) demonstrating how these threats affect evaluation results (Section 4), and (iii) proposing a conceptual framework to address them (Section 6).

3 Identification of Threats

In this section, we show how we identified threats to internal validity in the evaluation of data-driven BPS. For that purpose, we conducted a two-step investigation of the research method that is commonly used to develop a data-driven BPS approach. In the first step (Section 3.1), we explicate this research method by describing the key entities of the underlying domain and by deriving its process steps from 15 relevant BPS papers. Building on this shared understanding, we identify threats to internal validity in the second step (Section 3.2) by systematically analyzing the decisions made within each step of the research method.

3.1 Research Method of Data-driven BPS

Data-driven BPS commonly follows a specific research method to determine the quality of a discovered simulation model (Camargo et al. 2019, 2020; López-Pintado and Dumas 2022; Estrada-Torres et al. 2021; Camargo et al. 2021, 2022, 2023; Meneghello et al. 2023, 2025; López-Pintado et al. 2024; López-Pintado and Dumas 2023, 2025; Kirchdorfer et al. 2024, 2025; Chapela-Campa and Dumas 2024). To explicate this research method, we follow Frank's (2006) definition of a research method that states that a research method consists of a *terminology* and a corresponding *process*. The terminology structures the underlying domain, whereas the process outlines the steps taken to achieve the objectives of the research method. In the following, we explicate both of these components.

Terminology. Figure 2 depicts the key entities and relationships in the terminology of the data-driven BPS research method as a UML-based conceptual model. We define the entities and their relationships as follows:

- The **Process** is the real-world entity being simulated. It is a series of linked events, activities, and decisions involving actors and objects that together achieve a valuable outcome (Dumas et al. 2018).

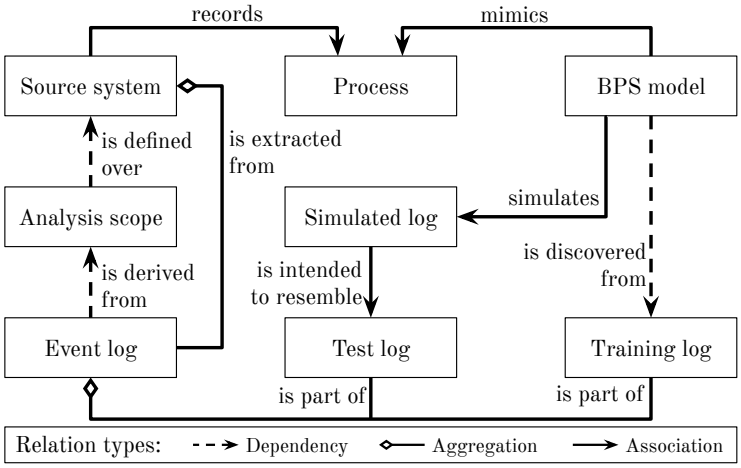


Figure 2: Terminology of the data-driven BPS research method.

- The **Source system** records and stores event data of the process. This system can be a workflow management system, an enterprise resource planning system, or any other information system holding event data (Van der Aalst 2016).
- The **Analysis scope** is a specification of parameters for guiding the extraction of event data from the source system, including timeframe, granularity, and relevant data attributes (Van Eck et al. 2015).
- The **Event log** is a set of event data extracted from the source system in accordance with the analysis scope. It is structured into cases (i.e., process instances), each represented as a trace of events ordered over time (Van der Aalst 2016).
- The **Training log** and the **Test log** are typically disjoint subsets of the event log, representing data from different timeframes of the process.
- The **BPS model** is a representation of the process that aims to mimic the behavior of the original process and is discovered from the training log.
- The **Simulated log** is the output produced as the BPS model simulates the process. This log is intended to resemble the test log.

Process steps. Based on this terminology, we derive the process steps of the data-driven BPS research method from the literature. Table 1 indicates which process

Table 1: Process steps taken to evaluate BPS models in the analyzed papers.

Paper	Process steps			
	Split	**Discovery**	**Simulation**	**Evaluation**
Camargo et al. (2019)	No split	Train log	Full size	Sim vs Orig
Camargo et al. (2020)	No split	Full log	Full size	Sim vs Orig
López-Pintado and Dumas (2022)	No split	Full log	Full size	Sim vs Orig
Estrada-Torres et al. (2021)	T/T split (n.d.)	Train log	Test size	Sim vs Test
Camargo et al. (2021)	Temp 80/20	Train (disc+val)	Test size	Sim vs Test
Camargo et al. (2022, 2023)	Temp 80/20 (no overlap)	80% disc / 20% val (train)	Test size	Sim vs Test
Meneghello et al. (2023, 2025)	Temp 60/20/20	Train (disc), Val	Test size	Sim vs Test
López-Pintado et al. (2024)	Temp 50/50	Train log	Test size	Sim vs Test
Chapela-Campa and Dumas (2024)	Temp 50/50 (no overlap)	Train log	Test size	Sim vs Test
López-Pintado and Dumas (2023, 2025)	Temp 50/50	Train log	Test size	Sim vs Test
Kirchdorfer et al. (2024, 2025)	Temp 80/20 (no overlap)	80% disc / 20% val (train)	Test size	Sim vs Test

Legend: T/T = train/test; Temp = temporal split; n.d. = not described; disc = discovery; val = validation; Train = training log; Val = validation log; Test = test log; Full = full event log; Full size = same size as full log; Test size = same size as test log; Sim = simulated logs; Orig = original log; Sim vs Orig/Test = comparison between simulated and original/test logs.

steps are executed to evaluate a data-driven BPS approach in 15 papers that were published between 2019 and 2025 and which constitute a representative set of recent publications in the field (Camargo et al. 2019, 2020; López-Pintado and Dumas 2022; Estrada-Torres et al. 2021; Camargo et al. 2021, 2022, 2023; Meneghello et al. 2023, 2025; López-Pintado et al. 2024; López-Pintado and Dumas 2023, 2025; Kirchdorfer et al. 2024, 2025; Chapela-Campa and Dumas 2024). From these papers, we derive a process model for the data-driven BPS research method, shown in Figure 3, which outlines the six steps taken to achieve the research method's objective of establishing a BPS model of high quality. In the following, we discuss how these steps are executed in the literature and abstract them into a unified process.

All papers rely on pre-existing event logs, so the first explicitly reported step in most papers is the splitting of the event log. However, event logs necessarily originate from a source system, and the two steps that obtain an event log from a source system are fundamental for process mining and commonly described in literature (Jans et al.

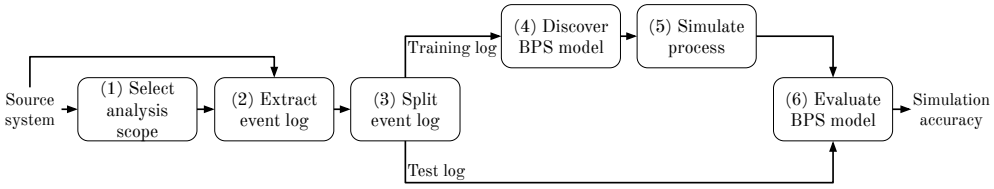


Figure 3: Process of the data-driven BPS research method.

2019), even if they are not explicitly reported in every paper. We therefore include them to ensure a complete representation of the research method and to capture all potential threats.

(1) **Select analysis scope:** The process begins by selecting an analysis scope over the source system. This involves determining the specific timeframe for extracting data from the source system and deciding on the construction of the event log, particularly the granularity—such as the object level that will define the process instance (Jans et al. 2019)—and identifying which data attributes are necessary for the analysis (Van Eck et al. 2015).

(2) **Extract event log:** Using the selected analysis scope, event data is extracted from the source system to create an event log. The chosen granularity and attributes guide the construction of events into cases, and the selection of the timeframe requires handling traces that only partially fit within it.

Starting from the extracted event log, the ensuing steps have evolved slightly over time. As Table 1 shows, three earlier papers do not include a train–test split and use the full event log for both model discovery and evaluation, whereas all later papers adopt temporal splitting to separate training and evaluation data. The reason for that may be that later researchers were more concerned with the quality of their approaches, thus needing more sophisticated evaluation on unseen data. On all other steps, the three early papers already followed the same method as later work, and all later work conducts the train–test split. We therefore do not regard the omission of

the split as a separate option within the research method, but as an early version that has evolved since then.

(3) **Split event log:** The extracted event log is then split into two separate logs: a training log and a test log. To preserve temporal dependencies within the data that are critical in simulation, logs are typically split temporally (Chapela-Campa et al. 2025). This splitting requires selecting a splitting point (e.g., after 70% of traces have started) and then determining what to do with traces that intersect with it to create the fully disjoint logs.

Although the exact split proportions differ (e.g., 50/50, 80/20) according to Table 1, the structure of this step is consistent. Some papers additionally set aside a validation partition within the training log, but this step is not applied consistently and is therefore not treated as essential in our formalization. With training and test logs defined, the next step is to discover a BPS model from the training log.

(4) **Discover BPS model:** The training log is used to discover a BPS model that can mimic the process. The specific actions in this step vary based on the discovery approach being evaluated.

Following model discovery, all papers use the BPS model to simulate new logs. Although the size of the simulated logs varies to match the log used for comparison, Table 1 shows that this step is consistently applied across all papers. This indicates that all papers rely on historical data for validation rather than expert judgment to assess model quality.

(5) **Simulate process:** Using the discovered BPS model, the process is simulated, producing a simulated log. To facilitate later comparisons, the simulated log is typically simulated with a fixed number of cases, matching the size of the test log.

Finally, the simulated log is used as input for evaluation. The three older papers compare the simulated logs against the entire event log, but most papers compare the simulated log to the test log, as documented in Table 1.

(6) **Evaluate BPS model:** The BPS model's accuracy is evaluated by comparing the simulated log to the test log to determine their resemblance. The accuracy is interpreted as the model's quality, supporting the research method's objective of establishing a high-quality BPS model.

Additional steps (such as data preprocessing) may be introduced in practice but are not considered in our analysis, since the wide range of possible data manipulations they entail makes it infeasible to systematically identify their potential threats to validity. We therefore restrict our examination to the six minimally required steps outlined above, with which we establish a shared understanding of the research method. This provides the necessary foundation for the systematic identification of threats to internal validity in the subsequent analysis.

3.2 Identification of Threats to Validity

To identify potential threats to internal validity, we analyze each step of the process of the data-driven BPS research method. Each step entails decisions and actions, such as selecting the analysis scope or determining how to split the event log. Crucially, while these steps are necessary—and there is no single "correct" way to perform them—the specific decisions made can significantly alter the underlying data, with consequences that propagate through later stages of the research method. As a result, even when using the same simulation model, evaluation outcomes may vary depending on the earlier steps in the process. This threatens internal validity, as the results may reflect these prior decisions rather than the true quality of the model itself.

To make these effects visible, we systematically examine each step of the data-driven BPS research method. For each step, we identify the decisions and actions that researchers must make and analyze how these can influence the execution of subsequent steps and the resulting evaluation outcomes. This analysis considers what

decisions are made and how they may affect the results. With that, we derive a comprehensive set of concrete threats to internal validity inherent in the method itself.

(1) **Select analysis scope:** The decisions involved in selecting the analysis scope can affect the validity of evaluation results in different ways. While decisions regarding granularity or specific attributes affect event log construction, posing issues of a different validity type (Rehse et al. 2024), selecting the timeframe can threaten internal validity. The threat arises from the assumption that the simulation model is evaluated on the same process from which it was derived, even while maintaining a separation between training and test data to prevent overfitting and ensure an objective evaluation on unseen data. This assumption allows any errors in quality assessment to be attributed directly to the model. However, when the process evolves and different versions emerge over time, also referred to as concept drift (Sato et al. 2021), this assumption does not hold: the model may end up simulating an outdated process version, with its evaluation influenced by different process versions in the training and test data rather than the model's true quality. As a result, the internal validity of the quality assessment is compromised by Threat T1: Concept Drift.

(2) **Extract event log:** After setting the timeframe of the analysis scope, each trace either falls entirely within this timeframe or only partially, i.e., by starting too early or finishing too late. Managing these partially covered traces involves strategies such as keeping, deleting, or cutting them to match the timeframe. Regardless of the strategy used, these decisions create discrepancies between the log and the actual characteristics of the process. Specifically, the event log may exhibit periods of time at the beginning and end of the log, referred to as fading-in and fading-out phases, where the log contains fewer active cases than the actual process. This causes evaluation results to vary depending on the chosen strategy, even for the

same BPS model. Consequently, Threat T2: Fading-in and out Phases in Extracted Logs emerges, compromising the internal validity of the evaluation.

(3) **Split event log:** Once the event log is extracted, deciding how to split it into training and test logs requires determining what to do with traces that intersect with the splitting point. If intersecting traces are, e.g., assigned to the training log, the log will contain a fading-out phase after the splitting point, during which fewer cases are active in the log than in the actual process. The test log starts with a fading-in phase with the same issue. Consequently, the evaluation of the BPS model can vary, influenced by the transitions created by these handling decisions rather than the inherent quality of the model itself. Thus, the presence of Threat T3: Fading-out and Fading-in Phases around the Splitting Point emerges, compromising the internal validity of the evaluation.

(4) **Discover BPS model:** When discovering the BPS model from the training log, the discovery can be artificially enhanced if the training and test logs are not properly separated. Even if the traces are disjoint, the timeframes of the logs may overlap, allowing the training log to convey information about the process state during the test period, such as overall case load or resource utilization. This issue, referred to as data leakage (Swanborn 1996), can lead to overly optimistic evaluations. Hence, by obscuring the model's true capabilities in mimicking the process, Threat T4: Data Leakage becomes a critical concern for maintaining internal validity in data-driven BPS.

(5) **Simulate process:** Since the simulated log is used to evaluate the simulation model, it must faithfully represent the model. Dynamic parameters, such as load conditions that are not specifically modeled, are often overlooked during the simulation. Consequently, the inadequate handling of load conditions can lead to a warm-up phase with no active cases and a cool-down phase where cases end without new ones starting. These low load levels affect other parameters, such as

waiting and lead times, so that the log fails to represent the model under normal conditions. It then becomes unclear whether the assessment truly measures the BPS model's ability to mimic the process or reflects the effects of the varying load conditions from the simulation step. Therefore, the distortion caused by overlooked load conditions constitutes Threat T5: Warm-up and Cool-down Phases of Simulated Logs, significantly risking the internal validity of the evaluation.

(6) **Evaluate BPS model:** To evaluate the BPS model, the simulated and the test log are compared for their resemblance. In contrast to the preceding steps, the decisions made here neither alter the data on which the model is evaluated nor affect the model's discovery. The concerns that arise in this step, therefore, do not threaten internal validity but relate to the measures used for the comparison. These measures have been shown to capture the intended differences between two logs (Chapela-Campa et al. 2023, 2025). Their use is nevertheless constrained by (i) their reliance on event data that may only partially reflect the underlying process, (ii) the fact that they cannot be normalized for comparison across different logs without losing information, and (iii) outliers that can negatively impact the resulting distances (Özdemir et al. 2025). Such concerns, alongside the question of how often a simulation should be replicated to support reliable conclusions, fall outside the scope of this analysis.

As this detailed walkthrough of the research method for data-driven BPS demonstrates, we can identify potential threats to validity stemming from the actions and choices made throughout the process. In the following section, we will explore each identified threat and its impact in greater detail.

4 Presence and Impact of Threats to Validity in BPS Evaluation

In this section, we examine the presence and impact of the five threats to the internal validity of the data-driven BPS research method. We first analyze how these threats are discussed in the papers from which the method was derived. Subsequently, in Section 4.1–Section 4.5, we illustrate each threat's impact on evaluation results through experiments, using established BPS evaluation metrics (Chapela-Campa et al. 2025). All data and experiment details are available in our repository[1].

Table 2: Identified threats to internal validity across the analyzed BPS papers.

Paper	Threats				
	T1	**T2**	**T3**	**T4**	**T5**
Camargo et al. (2019)	ND	ND	–	NA	–
Camargo et al. (2020)	ND	ND	–	NA	ND
López-Pintado and Dumas (2022)	ND	ND	–	NA	ND
Estrada-Torres et al. (2021)	ND	ND	ND	PA	ND
Camargo et al. (2021)	ND	ND	ND	PA	–
Camargo et al. (2022, 2023)	ND	ND	ND	A	–
Meneghello et al. (2023, 2025)	ND	ND	ND	PA	–
López-Pintado et al. (2024)	ND	ND	ND	PA	ND
Chapela-Campa and Dumas (2024)	ND	ND	ND	A	ND
López-Pintado and Dumas (2023, 2025)	ND	ND	ND	PA	ND
Kirchdorfer et al. (2024, 2025)	ND	ND	ND	A	ND

Legend: ND = not discussed; NA = not addressed; PA = partly addressed; A = addressed; – = not applicable.

Discussion of the threats' presence in the analyzed papers. Table 2 shows that, across the 15 analyzed papers from which we derived the research method, the threats to validity are only partially discussed or addressed. In fact, most threats are not mentioned at all. T1 (concept drift) is never explicitly assessed, likely because drift is assumed absent from the evaluation logs. T2 (fading-in and fading-out phases in extracted logs) is likewise not discussed, presumably because the logs were already available before evaluation. T3 (fading-in and fading-out phases around the split) is

[1] https://github.com/robert-l-b/bps_validity_threats

also omitted, although splitting is applied in all but the first three papers, where T3 is not applicable. T5 (warm-up and cool-down phases in simulated logs) is likewise not recognized, though in some papers, the threat does not apply due to the predictive nature of their simulation models, which do not need to warm up. By contrast, T4 (data leakage) is partially or fully addressed in most papers. The first three papers do not separate training and test data and therefore do not address T4, while later work uses temporal splitting and, in five cases, additional measures to fully prevent leakage. This shows that almost all threats are neither discussed nor counteracted, and motivates a detailed discussion of each threat and its impact on evaluation results in the following.

4.1 T1: Concept Drift

Concept drift refers to changes in processes over time, caused by the dynamic environments they are embedded in (Sato et al. 2021). This can involve changes to a process's control flow, but may also relate to aspects such as arrival rates, execution times, or resource availability. Although ignoring concept drift has previously been identified as a risk when applying simulation (van der Aalst 2015), its potential as a threat to validity has become more pronounced with the rise of data-driven evaluation approaches (cf. Section 3) and the widespread use of the framework proposed by Chapela-Campa et al. (2025). Since evaluations in data-driven BPS use data from one time period to establish a model (the training log) and from a different period for assessment (the test log), any changes in the underlying process between or during these periods pose a considerable threat to the validity of the evaluation results. Specifically, even a simulation model that perfectly mimics the training log may yield misleading evaluation results when the test log's characteristics differ, while a less accurate model may appear superior and be selected for further analysis or redesign.

Experiment setup. To illustrate the impact of concept drift, we conduct an experiment using the event log of a help desk process[2], which experiences a significant decline in incoming cases at one point in time (see Figure 4). To show the effect of this drift on evaluation results, we temporally split the log into pre- and post-drift sub-logs and create two simulation scenarios based on this split:

(i) Pre-drift only: We split the pre-drift log into a training (80% of cases) and test log (20%), removing cases that span across the split, so that we train and evaluate the model only on pre-drift data.

(ii) Across drift: We evaluate the simulation model from scenario *(i)* against a test log from the post-drift sub-log (of the same size as the test log in *(i)*).

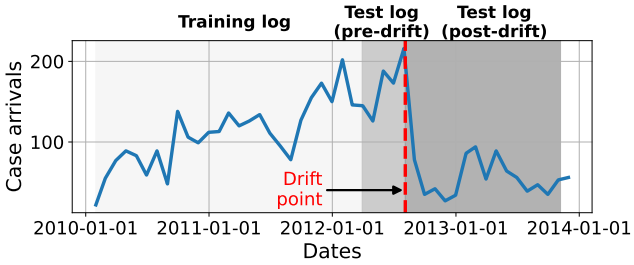


Figure 4: T1—Concept drift in arriving cases in help desk log.

We conduct simulations for each scenario by using the AgentSimulator approach (Kirchdorfer et al. 2025) to discover a BPS model and to simulate ten runs. To evaluate the results, we use an established set of BPS evaluation metrics (Chapela-Campa et al. 2025), which use the Earth Mover's distance to quantify differences in key process characteristics between a test and a simulated log, including control-flow, time, resource, and congestion. For this experiment, we specifically consider:

(1) Case-arrival distribution distance (CADD): This metric measures differences in the manner in which new cases arrive, i.e., the case-arrival distributions.

[2] https://doi.org/10.4121/uuid:0c60edf1-6f83-4e75-9367-4c63b3e9d5bb

(2) Absolute event distribution distance (AEDD): This metric measures differences in how events are distributed over the timeframe of the process, i.e., how many events are performed for specific time windows (e.g., per hour).

(3) Relative event distribution distance (REDD): This metric measures differences in execution patterns per trace, i.e., how cases progress over time, relative to their arrival time.

Collectively, these distances assess how closely the training log resembles the test log with respect to different aspects of the real process. Accordingly, they serve as direct measures of the model's overall quality (Chapela-Campa et al. 2025).

Results. As shown in Table 3, evaluating the same simulation model on test logs from both before and after the drift can lead to substantial differences in results. As expected, CADD shows a significant difference in simulated case arrivals for scenario *(ii)*, with a nearly ninefold increase in the distance measure compared to the pre-drift scenario. This shows that the simulation model can accurately mimic the manner in which new cases arrive to the process, but only when compared to a suitable test log.

Table 3: T1—Evaluation results for two simulation scenarios on the help desk log, evaluated using case-arrival distribution distance (CADD), absolute event distribution distance (AEDD), and relative event distribution distance (REDD).

Simulation scenario	CADD		AEDD		REDD	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**
(i) pre-drift only	428.0	0.4	457.9	4.8	96.6	1.5
(ii) across drift	3878.2	47.5	4140.1	42.7	230.0	1.0

While CADD results demonstrate the direct impact of concept drift on case-arrival rates, the drift also affects downstream metrics. For AEDD, the large increase in distance reflects a mismatch in case arrivals: because the BPS model simulates a higher arrival rate than found in the post-drift test log, all simulated events fall within

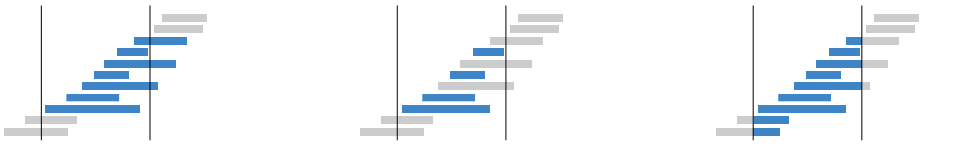
a relatively short timeframe of around 8 to 9 months, whereas the post-drift test log spreads the same number of cases over a 15-month period (compared to 6 months for the pre-drift log). Thus, the AEDD metric spikes for scenario *(ii)*, reflecting an inaccurate workload in terms of performed events over time. A similar trend can be observed in REDD, which more than doubles when comparing pre-drift to across-drift. In the pre-drift setting, most events are executed shortly after the beginning of a case, whereas in the post-drift setting, events are spread more evenly. Since the simulation model mimics the short execution patterns from before the drift, it replicates scenario *(i)* more closely than scenario *(ii)*.

In summary, the experiment highlights that concept drift can significantly affect BPS evaluations. The threat, however, extends beyond the specific conditions illustrated here. First, the reason for the drift can reflect both the evolution of the process and temporary changes in the environmental context. Second, similar issues arise even if the drift occurs within the timeframe of the training or test log (rather than exactly in between), since the drift may still cause the characteristics of the logs to differ considerably. Third, while this experiment focuses on a drift in case-arrival rates, any characteristic derived from the training log may exhibit a drift, and a drift in one such characteristic can propagate and distort multiple simulation dimensions, making its impact difficult to isolate. Therefore, concept drift poses a general threat to validity in data-driven BPS evaluations.

4.2 T2: Fading-in and out Phases in Extracted Logs

The extraction of event logs from source systems can result in *fading-in* and *fading-out* phases at the beginning and end of logs, during which process characteristics captured in the data differ from reality. These phases result from the need to handle traces that only partially fit within an event log's timeframe, by either keeping, deleting, or cutting them (see Figure 5). Depending on the chosen extraction strategy, resulting

event logs can have periods in which their workload (number of active cases) is lower than in the real process. Case-specific characteristics, such as case length and lead time, may also be affected (Weytjens and De Weerdt 2022). Since extracted logs, therefore, do not fully reflect the underlying process, fading-in and fading-out phases compromise the accuracy of simulation insights and pose a threat to validity.



(a) Extracting traces starting in timeframe.

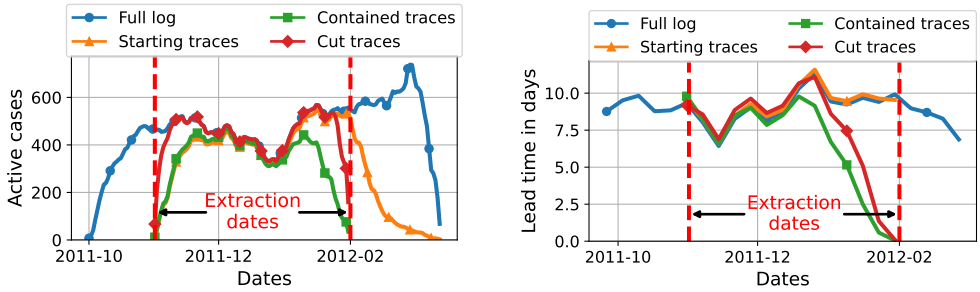
(b) Extracting traces contained in timeframe.

(c) Extracting traces cut to timeframe.

Figure 5: T2—Three event log extraction strategies that determine whether traces are included (blue) or excluded (light grey) in the event log.

Experiment setup. To show the impact of different extraction strategies on evaluation results, we consider three simulation scenarios. Since no publicly available database for event log extraction exists, we mimic the extraction step on a subset of the *BPI Challenge 2012* event log[3], filtered for human-executed tasks (the "W"-subset), with extraction boundaries at a distance from the beginning and end of the log to avoid its own fading-in and fading-out phases (see Figure 6a). Based on this, we define three simulation scenarios by extracting traces that are either *(i) starting*, *(ii) contained*, or *(iii) cut* within the timeframe (see Figure 5). For each scenario, we subsequently apply the same splitting strategy for which we remove traces spanning across both splits, discover a BPS model, and conduct ten simulation runs using AgentSimulator (Kirchdorfer et al. 2025). Besides the metrics used for threat T1, we use the lead time distribution distance (LTDD) to evaluate the simulation scenarios, which quantifies differences in the lead times of cases (Chapela-Campa et al. 2025).

[3]https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f



(a) Active cases of event logs created from three different extraction strategies.

(b) Lead time of event logs created from three different extraction strategies.

Figure 6: T2—Comparison of active cases and lead time for event logs created from three different extraction strategies using the BPI 2012 log.

Results. Our experiment confirms that different extraction strategies yield distinct event log characteristics, leading to inconsistent evaluation results. Figure 6a shows that each strategy produces a different representation of active cases at the start and end of the extracted logs, forming different fading-in and fading-out phases. Cutting the traces shortens both phases, whereas extracting cases that start within the timeframe better reflects the original process but shifts the fading-out phase beyond the timeframe. The same pattern appears in the lead times of starting cases over time (Figure 6b): extracting traces starting in the timeframe resembles the lead times of the original process most closely, while restricting the extraction to contained traces produces shorter traces at the end, and cutting traces produces a comparable but shorter fading-out phase.

Table 4 shows that these differences also affect evaluation results, with two key findings. First, no single strategy yields higher accuracy results across all metrics. Second, results differ significantly across strategies: AEDD and LTDD vary by a factor of two to three between *(i) starting* and *(ii) contained* (AEDD: 43 vs. 106, LTDD: 105 vs. 33). While the simulation based on *(i) starting* better resembles the test log's event distribution over two months (AEDD: 43), it simulates too short cases, leading to a higher difference in LTDD (105). In contrast, *(ii) contained* extracts shorter cases

Table 4: T2—Evaluation results for three simulation scenarios on the extracted versions of the BPI 2012 log using AEDD, REDD, and LTDD.

Extraction strategy	AEDD		REDD		LTDD	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**
(i) starting	42.7	6.7	73.2	7.4	104.8	4.0
(ii) contained	105.8	10.4	60.7	7.7	33.0	5.7
(iii) cut	118.5	12.2	83.2	10.1	41.2	3.6

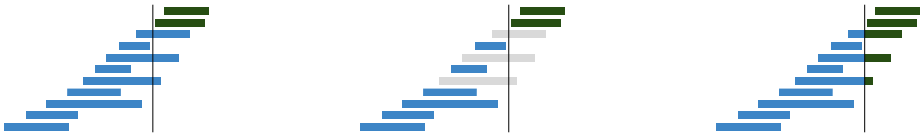
toward the end of the timeframe, resulting in a test log of less than a month that the simulation exceeds, and thus a high AEDD of 106. The shorter cases, however, align more with the simulated lead times, leading to a better LTDD of 33. The REDD results follow a similar pattern but with less variation, with differences of only about 20 across strategies, indicating smaller but still relevant differences in execution patterns.

In conclusion, the extraction step produces fading-in and fading-out phases that can affect process characteristics and threaten the internal validity of BPS evaluation. This highlights the need to carefully consider the chosen strategy and the interpretation of the resulting simulation results.

4.3 T3: Fading-out and in Phases around the Splitting Point

Splitting an event log into training and test logs can also create fading-out and fading-in phases around the splitting point, which may cause variations in process characteristics and evaluation results. Although such splitting is critical for objective evaluations, allowing for assessing the model against unseen data and reducing the risk of overfitting, with *temporal* splitting crucial for accurate process representation, the split itself can still alter process characteristics near the splitting point, similar to the effects during log extraction (see Section 4.2). For instance, it can change the process workload and affect case-specific characteristics, such as the evolution of case length over time (Weytjens and De Weerdt 2022), with the extent of the resulting

fading-out and fading-in phases varying across splitting strategies. Figure 7 illustrates three common strategies, *regular*, *intermediate*, and *strict*, which respectively keep (in the training log), delete, or cut off traces crossing the splitting point. Each affects the fading-out and fading-in phases differently and thus the evaluation results. Thereby, they threaten the internal validity of experiments and potentially lead to skewed conclusions.



(a) Regular temporal split.

(b) Intermediate temporal split.

(c) Strict temporal split.

Figure 7: T3—Three event log splitting strategies that assign traces to the training log (blue), test log (dark green), or exclude them (light grey).

Experiment setup. To illustrate the impact of data splitting strategies on BPS evaluation results, we define an experiment with three simulation scenarios. Building on our T2 example (Section 4.2), we use the log extracted from traces starting in the timeframe and establish three scenarios corresponding to the temporal splitting strategies in Figure 7: *(i) regular*, *(ii) intermediate*, and *(iii) strict*. For each scenario, we determine the splitting point by assigning the last 20% of traces to the test log and subsequently use AgentSimulator (Kirchdorfer et al. 2025) to discover a BPS model and execute ten simulation runs.

Results. The experiment shows that different splitting strategies create distinct fading-out and fading-in phases in the training and test logs, as shown by the number of active cases for each strategy (Figure 8). The strict split results in shorter phases than the regular and intermediate splits, which yield the same test logs but differ in the fading-out phase of the training log: this phase ends at the splitting point for the intermediate split, whereas it only begins there for the regular split, resulting in

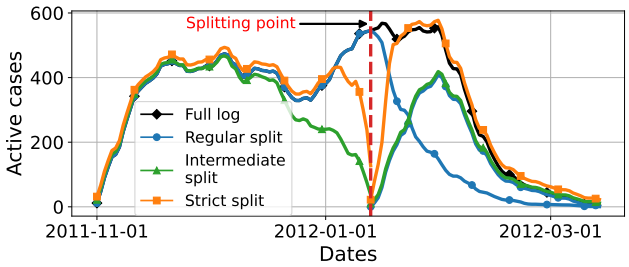


Figure 8: T3—Active cases of training and test logs created from three different splitting strategies for BPI 2012 log.

a longer full-load training period. Additionally, the split affects the training and test logs differently for these two strategies, as the test logs do not reach the event log's actual load.

Table 5: T3—Evaluation results for three simulation scenarios on the splitting strategies of the BPI 2012 log using AEDD, REDD, and LTDD.

Splitting strategy	**AEDD**		**REDD**		**LTDD**	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**
(i) regular	35.4	6.0	42.2	5.2	71.2	7.8
(ii) intermediate	42.7	6.7	73.2	7.4	104.8	4.0
(iii) strict	97.5	6.4	71.0	5.9	103.7	5.4

Table 5 shows that these differences also affect evaluation results. The regular split consistently yields the best results across all evaluation metrics, with approximately 40% better REDD and LTDD, while the intermediate and strict splits lead to comparable results, except for AEDD. The AEDD difference stems from the simulation model's inability to simulate incomplete traces: the simulated log exhibits a fading-in phase during which the load of active cases builds up, but this phase is much shorter in the strict-split test log, resulting in a comparatively high AEDD of 98. The differences between the regular and intermediate splits stem from the different characteristics of their training logs' fading-out phases: simulated logs based on the intermediate split

generally contain shorter cases than those from the regular split. This indicates that removing cases spanning the splitting point has influenced the simulation model's discovery, leading to greater distances than with the regular split.

Table 6: T3: List of real and synthetic logs, including their number of cases, the average case length, and the percentage of cases intersecting with a splitting point after 80% of cases have started.

Log	Syn./Real	Cases	Avg. Dur. (d)	Overlap (%)
EX (T3)	Real	5,010	8.9	10.9
BPI 12	Real	8,616	8.9	6.6
BPI 17	Real	30,276	12.7	3.7
ACR	Real	954	14.9	13.3
MP	Real	225	20.6	25.3
P2P	Synth.	608	21.5	9.0
CVS	Synth.	10,000	7.6	15.6
CFS	Synth.	1,000	0.9	0.6
CFM	Synth.	2,000	0.8	0.5

The key distinction between the splits is that the intermediate split omits the cases overlapping the splitting point: in our example, nearly 11% of the 5,010 cases do so, which both creates the fading-in and fading-out phases and contributes to systematic differences between the strategies. Table 6 shows that this is not an exception. Across a range of real and synthetic event logs commonly used in BPS evaluation (Camargo et al. 2020, 2021, 2023; Meneghello et al. 2023; Kirchdorfer et al. 2025), half show overlaps of around 10% or more, driven by longer case durations and lower case counts: the *purchase-to-pay* (P2P)[4], *academic credentials recognition* (ACR), and *manufacturing production* (MP)[5] logs, with fewer than 1,000 cases and average case durations of about 15 days or longer, reach overlaps of up to 25% (MP), and the *pharmacy retail process* (CVS)[1] reaches 15.6% via its medium case duration despite its larger case count. In contrast, logs with very short durations or many cases show much smaller overlaps: the confidential synthetic process versions CFS[1] and CFM[1] (less

[1] https://zenodo.org/records/4264885
[4] https://fluxicon.com/academic/material
[5] https://doi.org/10.4121/uuid:68726926-5ac5-4fab-b873-ee76ea412399

than a day) at 0.5% and 0.6%, and the *BPI Challenge 2017* log (filtered to the "W" subset)[6] at 3.7%, with 30,276 cases. These small overlaps yield minimal differences between regular and intermediate splits, indicating relatively short fading phases and less pronounced effects of T3 overall.

In summary, the experiment demonstrates that fading-out and fading-in phases introduced by splitting strategies can significantly influence BPS evaluations. Particularly, the comparison of the results from the regular and intermediate split is relevant in the context of common practices. While the regular split yields more accurate results in our experiment, the intermediate split is commonly used to avoid the following threat: data leakage (Kirchdorfer et al. 2025; Camargo et al. 2023).

4.4 T4: Data Leakage

The way event logs are split can also introduce a significant threat to the discovery step: data leakage. In data mining, data leakage occurs when information about the target variable inadvertently leaks into the training data, causing overestimated model performance (Kaufman et al. 2012). In BPS, leakage occurs when the training log contains events from the test log's timeframe—a risk that is also explicitly highlighted in data-driven BPS approaches (Camargo et al. 2023; López-Pintado et al. 2024; López-Pintado and Dumas 2025; Kirchdorfer et al. 2025). These events convey information about the process during the timeframe to be simulated, such as the load of cases, the utilization of specific resources, changes in patterns, or black swan events. Simpler discovery approaches that discover distributions over the entire training log are less affected, whereas more complex models, such as those based on deep learning, capture complex relationships and are therefore more likely to be impacted. As a result, the model may appear to perform better than it actually does, because it has

[6] https://doi.org/10.4121/uuid:5f3067df-f10b-45da-b98b-86ae4c7a310b

indirectly "seen" much of the test data. The regular temporal split (Section 4.3) exemplifies this issue: it is the most intuitive splitting strategy, but allows the described overlap of events and creates data leakage.

Experiment setup. The direct impact of data leakage is difficult to isolate and depends on whether an approach (aims to) exploit it. We therefore quantify its extent in BPS evaluation settings, since any leakage can in principle be exploited even if current approaches do not explicitly do so. For this, we measure the overlap of training-log events with the test log's timeframe for our example from Section 4.3 (Ex (T3)) and five additional event logs commonly used in BPS evaluation (Camargo et al. 2020, 2021, 2023; Meneghello et al. 2023; Kirchdorfer et al. 2025). For each log, we apply the regular temporal split and count the unique training-log events that fall within the test log's timeframe. We define *training log overlap* as this share relative to the training log events, and *test log overlap* as the same share relative to the test log events.

Table 7: T4—Percentage of training log events that overlap with the test log's timeframe and their share relative to the training and test logs.

Measures	Ex (T3)	BPI 12	BPI 17	P2P	ACR	CVS
Training log overlap (%)	15.4	8.8	2.8	10.5	6.7	11.5
Test log overlap (%)	59.9	37.5	12.3	64.3	33.4	46.0

Results. Data leakage is significant, as shown by the overlap between training and test log events. Table 7 quantifies this overlap across various event logs, highlighting the extent of the issue. In the example log from Section 4.3 (Ex (T3)), 15% of training log events overlap with the test log's timeframe, corresponding to nearly 60% of the test log's events. Across most event logs, events from the test log's timeframe constitute about 10% of the training log's events and more than one-third—once even 64%—of the respective test log's events. The BPI 2017 log is an exception, where these shares are lower because it contains twice as many events as the second-largest log (CVS).

In conclusion, the experiment highlights that data leakage poses a significant threat to BPS evaluations. Incorporating events from the test log's timeframe into the training data can lead to overly optimistic results. Our findings show leakage rates of up to 64% across different event logs, emphasizing the importance of selecting appropriate splitting strategies to ensure valid evaluation outcomes.

4.5 T5: Warm-up and Cool-down Phases of Simulated Logs

A final significant threat to the internal validity of BPS evaluations arises from the *warm-up* and *cool-down* phases in simulation. These phases emerge at the start and end of a simulation but are not explicitly incorporated into the BPS model. Since a typical BPS evaluation simulates a set number of cases (usually matching the test log) and starts no new cases once this number is reached, two distinct phases arise. The warm-up phase begins with an empty system, resulting in artificially low waiting times and workloads in the initial period. This *initialization bias* (Law 2015) differs from the fading-in and fading-out phases discussed earlier: the latter result from partially observed data under normal system load, whereas the warm-up phase emerges from the simulation starting with an empty system. Although the bias has been discussed in general simulation literature (Law 2015; Schruben 1982; Banks and Chwif 2011) and within the BPS domain (Peters et al. 2022; van der Aalst 2015), it is not addressed in the research method used to evaluate data-driven BPS approaches (Camargo et al. 2020; Kirchdorfer et al. 2025; Estrada-Torres et al. 2021; López-Pintado et al. 2024; López-Pintado and Dumas 2025). The cool-down phase, conversely, initiates no new cases and processes the remaining ones with diminishing load, producing similarly distorted waiting times and execution patterns. Both phases, therefore, cause the BPS model to be inaccurately represented by the simulated log, and, with that, distort evaluation results.

Experiment setup. To test the impact of warm-up and cool-down phases on BPS evaluation results, we define simulation scenarios for the P2P process that either replicate or mitigate the threat. To do so, we split the P2P log into training and test logs using an 80/20 intermediate temporal split, and outline four simulation scenarios:

(i) *Regular simulation*: The simulation starts with an empty system and ends without initiating new cases.

(ii) *Pre-warm-up*: The system warms up before logging events.

(iii) *Post-cool-down*: The system cools down after the logging is stopped.

(iv) *Pre-warm-up & post-cool-down*: Scenarios *(ii)* and *(iii)* are combined.

For each scenario, we discover a BPS model and simulate ten runs using AgentSimulator (Kirchdorfer et al. 2025).

Table 8: T5—Evaluation results for four simulation scenarios on the P2P process using AEDD, REDD, and LTDD.

Simulation scenario	AEDD		REDD		LTDD	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**
(i) regular simulation	1275.2	(7.6)	766.6	(7.0)	550.0	(13.6)
(ii) pre-warm-up	1208.4	(14.0)	729.4	(14.0)	487.9	(31.9)
(iii) post-cool-down	1168.0	(25.5)	698.5	(22.9)	430.3	(46.9)
(iv) pre-warm-up & post-cool-down	1116.9	(24.8)	635.7	(23.7)	347.1	(39.0)

Results. The experiment demonstrates that incorporating warm-up and cool-down phases into BPS evaluation improves accuracy. As shown in Table 8, scenario *(i)* exhibits the worst performance across all evaluation dimensions, while scenarios *(ii)* and *(iii)* show initial improvements over scenario *(i)*. As expected, the most significant improvements appear in the combined scenario *(iv)*, which warms up and cools down the system before and after generating the simulated log: compared to scenario *(i)*, it improves AEDD by 12%, REDD by 17%, and LTDD by 37%. These results can be explained by differences in waiting times. Figure 9 shows that the average weekly waiting times of scenarios *(ii)* to *(iv)*, which counteract the warm-up and cool-down phases,

are higher than those of scenario *(i)*. More importantly, it shows that those scenarios also lead to longer simulated logs in general, which results in better performance across all three metrics.

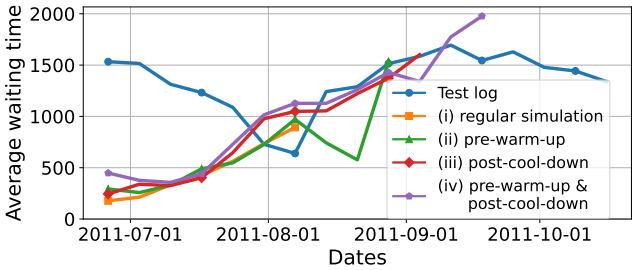


Figure 9: T5—Average waiting times per week for the test log and four experiment scenarios for the P2P process.

In conclusion, these findings highlight the importance of counteracting warm-up and cool-down phases to ensure accurate BPS evaluations. Without doing so, simulations may produce inaccurate representations of waiting times and execution patterns, leading to skewed conclusions about process performance and threatening the validity of the evaluation.

5 Joint Effects and Consequences of Threats to Validity

The experiments in Section 4 demonstrate that each threat to validity in BPS individually produces measurable distortions in evaluation results. In this section, we examine the joint effects of these threats and the consequences they carry for BPS evaluations. We first quantify how evaluation results vary across combinations of threats to validity and how much of this variation aligns with each threat individually or can be attributed to joint effects (Section 5.1). We then examine a particular joint effect

between two threats, namely the trade-off between splitting the log and the data leakage this may induce, and identify the conditions under which this trade-off becomes severe or makes evaluation infeasible (Section 5.2). Finally, we show how the distortions caused by the threats can lead to conflicting operational recommendations in what-if analyses (Section 5.3).

5.1 Joint Effects Among Threats

To examine the joint effects of the threats to validity, we design an experiment built on the research method used to develop a data-driven BPS approach. We carry out four of its steps in alternative ways and measure how evaluation results change across the resulting scenarios. Each of these four steps is associated with one of the threats: T1 (concept drift), T2 (fading-in and fading-out phases in extracted logs), T3 (fading-out and fading-in phases around the splitting point), and T5 (warm-up and cool-down phases of simulated logs). T4 (data leakage) is not included separately because it follows directly from the data-splitting strategy varied under T3. This design allows us to quantify both the total variation in evaluation results across varying scenarios and the share of that variation attributable to each individual threat.

Experiment setup.

- **T1:** To isolate the effect of concept drift, we generate two simulated versions of the underlying log with AgentSimulator: the *no-drift* alternative preserves a stable case-arrival rate across the timeframe, and the *across-drift* alternative introduces a change in the case-arrival rate midway through it.
- **T2:** From each of the two simulated event logs, we extract an event log under one of two extraction strategies that lead to T2. The *contained* strategy keeps only traces that fall entirely within the timeframe. The *ending* strategy keeps only traces that end within the timeframe, mirroring the *starting* strategy from

Section 4.2. Choosing these two strategies limits the difference between the resulting event logs to the beginning of the timeframe. This ensures that the end of the log, which is split off as the test log, is the same across all 16 scenarios.

- **T3:** We split each extracted event log into a training and a test log under one of two data-splitting strategies that lead to T3. The *regular* split assigns traces that span the splitting point to the training log, whereas the *intermediate* split discards these traces.
- **T5:** From each training log, we discover a BPS model and conduct ten simulation runs using AgentSimulator (Kirchdorfer et al. 2025) under one of two simulation conditions that lead to T5. The *regular simulation* starts with an empty system and ends without initiating new cases. The *pre-warm-up & post-cool-down* simulation includes a pre-warm-up phase before the test log's timeframe and a post-cool-down phase after it (see Section 4.5).

Metrics. To capture both the total variation across the 16 scenarios and the contribution of each individual threat, we use the four distance metrics introduced in Section 4 together with two aggregating measures. For each scenario, we compute CADD, AEDD, REDD, and LTDD, which capture differences in case-arrival, absolute event, relative event, and lead-time distributions, respectively (Chapela-Campa et al. 2025). To aggregate these values, we use two measures:

(1) *Coefficient of variation (CoV):* CoV measures how much a distance metric varies across the 16 scenarios. Specifically, with the standard deviation σ and the mean μ of the metric across the 16 scenarios, we compute $\text{CoV} = \sigma/\mu$.

(2) *Eta-squared (η^2):* η^2 quantifies how much of the total variation in a metric across the 16 scenarios is associated with a given threat. In statistical analyses on sampled data, η^2 is an effect-size measure typically paired with a hypothesis test to assess whether an observed effect generalizes beyond the sample (Cohen 1988). We instead use η^2 descriptively to characterize the extent to which variation in

the evaluation results aligns with each threat to validity individually. Specifically, let $x_1, \ldots, x_{16}$ be the metric values across the 16 scenarios, $\bar{x}$ their mean, and $\bar{x}_a, \bar{x}_b$ the means of the eight scenarios where the threat takes each of its two alternatives. We compute

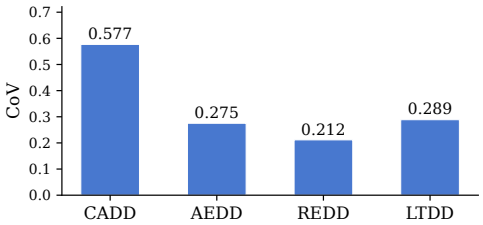
$$\eta^2 = \frac{8(\bar{x}_a - \bar{x})^2 + 8(\bar{x}_b - \bar{x})^2}{\sum_{i=1}^{16}(x_i - \bar{x})^2}.$$

A value close to one indicates that the threat accounts for most of the variation, while a value close to zero indicates that its two scenarios yield similar evaluation results on average. Because η^2 is computed for each threat to validity separately, the four values for a metric need not sum to one. Since the 16 scenarios differ only in the four threats varied in the experiment, the remaining share reflects joint effects among these threats.

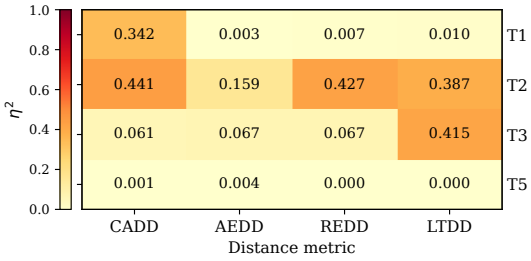
Results. Across the 16 scenarios, the evaluation results vary widely on all four distance metrics. We first quantify this variation across the scenarios and then discuss the attribution of the variance to individual threats and to their joint effects.

Variation across scenarios. The overall variation across all 16 scenarios and the four threats is captured in Figure 10a. The figure shows CoV per metric as a bar chart, with higher bars indicating greater relative variation across scenarios. CADD—the distribution distance of the case arrivals—shows the highest variation by a wide margin (CoV = 0.58), with mean CADD values ranging from 15 to 117 across the 16 scenarios. This large range can be explained by the concept drift, which we introduced directly to the case arrivals in half of the scenarios. The other three metrics fall within a tighter range (CoV between 0.21 and 0.29), but with absolute values still varying by a factor of 2 to 3: AEDD from 43 to 123, REDD from 40 to 95, and LTDD from 27 to 87. This shows that none of the four metrics remains stable across the 16 scenarios,

and there is also variation in the process dimensions that are not as directly affected by the creation of our scenarios.



(a) CoV per distance metric.



(b) η^2 per threat and distance metric.

Figure 10: Coefficient of variation (CoV) and eta-squared (η^2) across 16 scenarios (T1×T2×T3×T5) for four distance metrics.

Attribution to threats and joint effects. Figure 10b shows that the variation of two of the four distance metrics is largely explained by one or two threats to validity. CADD's variation, for instance, can be associated in large parts with T1 ($\eta^2 = 0.34$) and T2 ($\eta^2 = 0.44$). Concept drift (T1) directly drives much of the variation in case-arrival distribution distances, while the stronger association with T2 shows that the data extraction strategies also affect arrival rates: traces at the beginning of the training logs generated by one extraction strategy yield different discovered and simulated arrival rates than those generated by the others. Correspondingly, the mean CADD increases from 36 without drift to 71 with drift and from 33 under one extraction strategy to 73 under the other. A similar pattern holds for LTDD, whose variation is explained

by T2 ($\eta^2 = 0.39$) and T3 ($\eta^2 = 0.42$): the extraction and data-splitting strategies underlying these threats together determine which traces, especially the longer ones, are included in the training and test logs. In both cases, the four η^2 values sum to approximately 0.85–0.90, leaving only a small share to the joint effects of the threats.

For the other two metrics, REDD and AEDD, no single threat explains a comparable share of the variation. Unlike the case-arrival rate captured by CADD, the way events are distributed over the timeframe (AEDD) and within each case (REDD) is shaped by many factors that propagate through the simulation and interact with one another, so most of the variation arises from joint effects among the threats. REDD is partly associated with the extraction strategy underlying T2 ($\eta^2 = 0.43$), but no other threat reaches an η^2 above 0.07, leaving about half of the variation in how events are distributed within cases to the joint effects. AEDD shows no single dominant threat: its largest single attribution is T2 at $\eta^2 = 0.16$, even though its CoV (0.28) is comparable to LTDD's (0.29). About three quarters of AEDD's variation, therefore, arises from joint effects of the threats rather than from any one of them individually, a pattern that cannot be anticipated from the per-threat analysis of Section 4. Hence, for these two metrics, the conclusions a researcher draws about the simulation model reflect the combination of decisions made along the research method to build the BPS model rather than any individual decision. T5 (warm-up and cool-down phases), in contrast, contributes little to any of the four metrics, with $\eta^2 < 0.01$ throughout.

In summary, the threats contribute unevenly to evaluation variation in our experiment. T2, driven by the choice of data extraction strategy, is the most influential threat overall, contributing to variation across all four distance metrics. Concept drift (T1) and the choice of data-splitting strategy (T3) have more localized effects, consistent with the mechanisms identified in Section 4. However, the relative (REDD) and absolute (AEDD) event distribution show that larger parts of the variation arise from

joint effects among the threats and cannot be attributed to any single one. Taken together, these findings show that the conclusions researchers draw about how well a simulation model captures the real process depend on the decisions made along the research method that we investigate. The size of the identified effect may differ in other settings, as these findings are based on a single event log.

5.2 Trade-off Between Data Splitting and Leakage

In this section, we investigate the trade-off between T3 and T4 that arises from the choice of the data-splitting strategy: the regular split retains more training data but introduces data leakage (T4), whereas the intermediate split avoids leakage at the cost of discarding cases that span the splitting point and creating fading-out and fading-in phases (T3). As shown in Section 4.3, the size of the log and the average case duration determine how many cases span the splitting point and therefore drive the severity of this trade-off, so we design an experiment that varies these two factors.

Experiment setup. To examine how these two factors shape the trade-off, we apply both the regular and the intermediate data-splitting strategies to two logs that differ in average case duration. The first is the running example log from Section 4.2 and Section 4.3, derived from the *BPI Challenge 2012 W log* (5,010 cases, average case duration of 8.9 days). As a second log with longer cases, we use the ACR log (954 cases, average case duration of 14.9 days). To vary the size of the logs, we extract smaller versions of the log chronologically using 20, 40, 60, 80, and 100% of the cases from the full log. For each of these created logs, we then apply both splits and compare the resulting evaluation results. We report absolute distance values for AEDD, REDD, LTDD, and CADD under each data-splitting strategy and measure the data leakage overlap of the regular split, as defined in Section 4.4.

Results. The experiment shows that the relevance of the trade-off between T3 and T4 depends on the structural characteristics of the log. This becomes visible in Figure 11

and Figure 12, where both the distance values per strategy and the leakage overlap change differently with log size for the two logs.

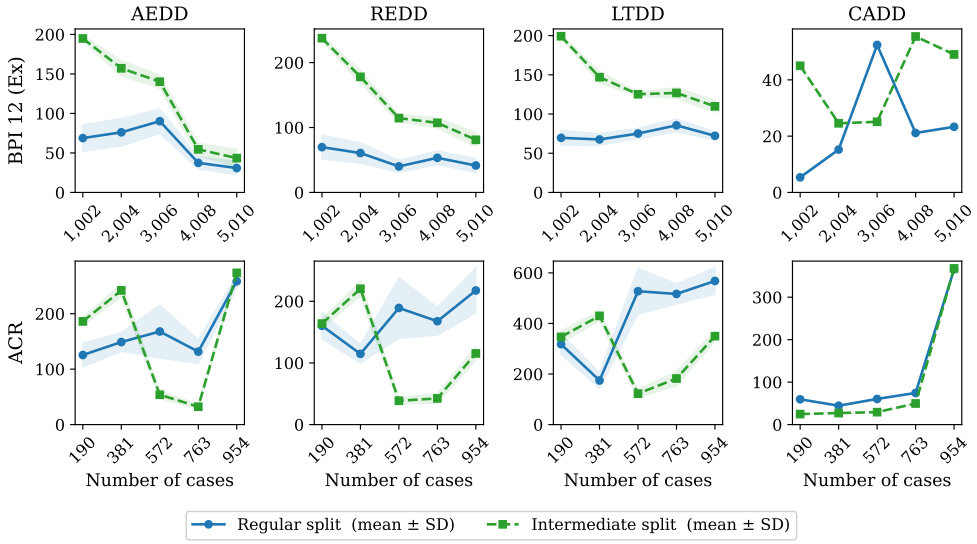


Figure 11: AEDD, REDD, LTDD, and CADD distances under the regular and the intermediate split across log sizes for the BPI 12 example log and the ACR log.

The BPI 12 example log shows that, given a sufficiently large log, the intermediate split becomes a viable alternative to the regular split, allowing the leakage from the regular split to be avoided without a meaningful loss in evaluation accuracy. As shown in the first row of Figure 11, the regular split's distances for AEDD, REDD, and LTDD stay relatively stable across log sizes, while the intermediate split's distances drop with log size: AEDD from 195 at 1,002 cases to 43 at 5,010 cases, with the same pattern for REDD and LTDD. Only CADD departs from this trend, probably due to changes in arrival rates towards the end of the full log. This convergence between the two splits is reflected in the share of training events that fall within the test log's timeframe under the regular split (Figure 12), which drops from around 190% at 1,002 cases to 60% at 5,010 cases. At large log sizes, the gap between the two splits becomes small, so the intermediate split can be chosen to avoid the remaining leakage

of around 60% under the regular split. This pattern is likely tied to the characteristics of the BPI 12 example log: a large case count, comparatively short and less variable case durations, and only six distinct activities.

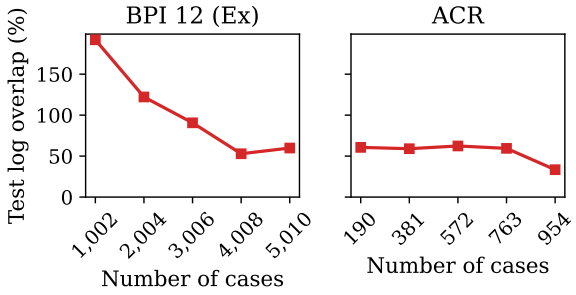


Figure 12: Share of training events that fall within the test log's timeframe under the regular split, across log sizes for the BPI 12 example log and the ACR log.

For the ACR log, evaluation results vary too widely across log sizes to attribute differences to the choice of split. Under the regular split, AEDD ranges from 126 to 258, and under the intermediate split from 32 to 273 (Figure 11). REDD and LTDD show the same pattern. As standard deviations remain mostly below 25, this variation cannot be explained by simulation randomness. Instead, it is likely caused by the structure of the ACR log. Its small case count, longer and more variable case durations, and 18 activities mean that each sampled log size captures the process differently, causing varying divergence between the discovered model and test log. The same effect appears at the full size of 954 cases, where both splits yield their highest AEDD and CADD values, consistent with a fading-out phase at the end of the original log (T2). In addition, the regular split's leakage on ACR is lower than on the BPI 12 example log: the share of training events that fall within the test log's timeframe (Figure 12) stays around 60% across most sizes, compared to around 190% observed for the BPI 12 example log at small sizes. For logs like ACR, the trade-off between T3 and T4 is

not the main concern, since other distortions tied to the log's structure have a larger effect on evaluation results than the choice of data-splitting strategy.

In summary, the relevance of the T3–T4 trade-off depends on the structural characteristics of the log. For long, regular logs, the trade-off resolves at a sufficiently large log size, and the intermediate split can be chosen to avoid leakage from the regular split. For logs with smaller case counts and longer, more variable case durations, the trade-off cannot be cleanly isolated because further distortions tied to the log's structure also shape evaluation results.

5.3 Impact on Downstream Decision Quality

The experiments in Section 5.1 and Section 5.2 show that the decisions made along the research method can distort the evaluation results of a BPS model. We now show that these distortions also impact downstream decision quality.

Experiment setup. To assess the impact of the threats on downstream decision quality, we conduct a what-if analysis for each of the 16 scenarios from Section 5.1. This requires defining a process change and a way to compare the conclusions drawn from the resulting simulation outputs. We base the change on the loan application process captured in the *BPI Challenge 2012 W log*. In this process, applications undergo a manual review before a final decision is made. We inject a change that halves the processing time of this review activity, representing the introduction of digital support for the assessment step. To derive and compare conclusions, we apply this change to the discovered BPS model of each scenario from Section 5.1 and conduct ten simulation runs using AgentSimulator (Kirchdorfer et al. 2025). We then compare the simulated logs from Section 5.1 with those generated from the changed BPS model to estimate the effect of the change on the lead time of cases, a key performance indicator (KPI) commonly used in operational decision support and typically weighed against the cost of the change. For each scenario, we compute the difference in lead time between the

simulations with and without the change and use the resulting estimates to compare the conclusions drawn across scenarios.

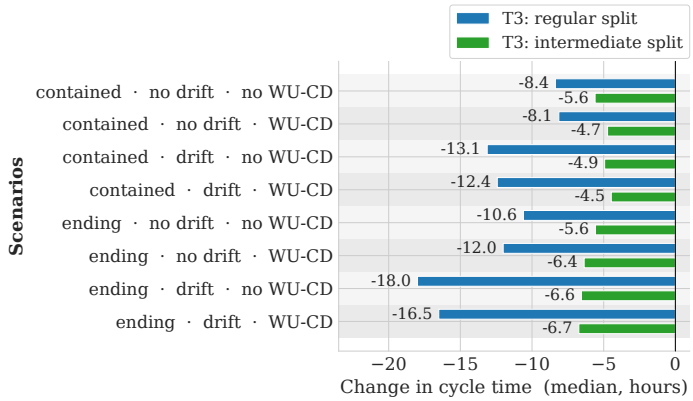


Figure 13: Lead-time reductions per scenario. Each row identifies one configuration by data extraction strategy (contained / ending), concept drift, and warm-up / cool-down (WU-CD) effect.

Results. The experiment shows that the estimated reduction in lead time depends on the decisions made along the research method, and that this dependence can flip the conclusion drawn from the analysis. Whether the digital support is worth introducing depends on weighing its cost against the operational gain from the reduced lead time, so a smaller estimated reduction makes the change less likely to be considered worthwhile, while a larger one makes it more likely. Figure 13 shows that this estimated reduction varies across the 16 scenarios from 4.5 to 18.0 hours per case. The eight regular-split scenarios cluster around a mean reduction of 12.4 hours per case, while the eight intermediate-split scenarios cluster around 5.6 hours per case. Hence, the data-splitting strategy underlying T3 alone separates the estimates by more than a factor of two. The remaining threats mainly affect the extent of variation within each group: under the intermediate split, estimates stay close to the 5.6-hour mean, whereas under the regular split they show greater variation around the 12.4-hour mean. This wider variation under the regular split is consistent with the inclusion of

cases that span the splitting point, which tend to be longer and exhibit more variable lead times and are omitted under the intermediate split. Depending on the scenario, the same proposed change is therefore read either as a reduction of nearly one working day per case, which is likely to justify the cost of the digital support, or as a gain of only a few hours, which is more likely to be judged too small to justify it. Thus, the same simulation model, the same proposed change, and the same input data can lead to conflicting investment decisions.

In summary, the threats to validity affect not only evaluation results but also the conclusions derived from them. In our experiment, T3 has the strongest influence on the estimated lead-time reduction, but the size and direction of the effect may differ for other event logs, what-if changes, or KPIs. This motivates the conceptual framework proposed in Section 6, which aims to guide researchers in reducing the distortions that give rise to these conflicting conclusions.

6 Conceptual Framework to Address Threats to Validity

This section introduces a conceptual framework to help researchers limit the impact of the identified threats to the internal validity of the data-driven BPS research method. Grounded in the related literature (cf. Section 2), the methodological analysis of the research method (cf. Section 3.2), the experimental results on the threats individually (cf. Section 4), and their joint effects and downstream consequences (cf. Section 5), it translates these insights into concrete refinements that can be embedded into the established research method. The framework provides procedural guidance that is aligned with the specific characteristics of the data and the BPS approach at hand. It clarifies (i) which steps are to be carried out, (ii) which fine-grained decisions must be made within those steps in light of the identified threats, and (iii) which established techniques from the literature can be used when such decisions require methodological

support that may be unfamiliar to the researcher. In the remainder of this section, we first present the framework and its refined operationalization of the different steps, and then discuss its applicability, limitations, and trade-offs.

6.1 Overview of the Conceptual Framework

The conceptual framework focuses on four steps of the established research method, where the identified threats to internal validity can be addressed through concrete interventions. Table 9 summarizes each step's inputs, outputs, addressed threats, and corresponding conceptual refinements.

Table 9: Core conceptual framework: inputs, outputs, threats, and conceptual refinements per step of the research method addressed.

Step	Input	Output	Threat(s)	Conceptual refinement	Figure
(2) Extract event log	Source system, analysis scope	Event log	T2	Extract traces—if supported by simulation approach—cut to timeframe to minimize misrepresentation of process	15
(Additional step:) Detect concept drift	Event log	Event log	T1	Apply concept drift detection; then—according to the analysis goal—re-scope or document drift	16
(3) Split event log	Event log	Training and test log	T3, T4	Split based on the capability of the simulation approach and the length of the log to avoid bias and leakage	17
(5) Simulate process	BPS model	Simulated log	T5	Based on the simulation approach and analysis goal, include warm-up and cool-down before and after logging data	18

The framework's pipeline, depicted in Figure 14, remains grounded in the original research method while explicitly embedding the proposed refinements. The refinements are threefold:

1. The framework provides detailed substeps for the three steps—Steps (2), (3), and (5)—in which the identified threats can be addressed, indicated by a process symbol in Figure 14.
2. We added a step, *Detect concept drift*, before splitting the event log. This step enables the use of established process-mining methods to address concept drift and is highlighted in grey in the figure.
3. The framework explicitly incorporates the role of the simulation approach at hand, since different approaches (e.g., queue-based versus timestamp-prediction) determine which refinements are applicable and how validity threats should be mitigated.

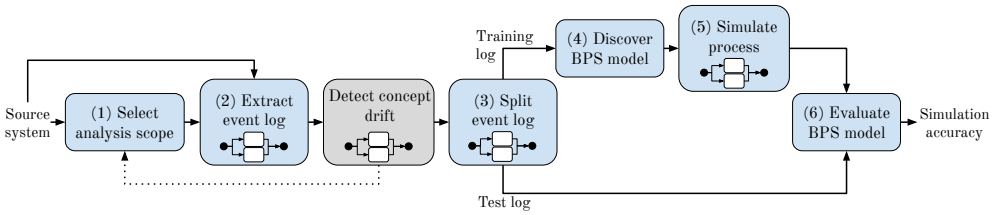


Figure 14: Overview of how the conceptual framework relates to the identified data-driven BPS research method.

Taken together, these refinements extend the six-step research method into a form that operationalizes threat mitigation while preserving the method's original structure and objective. Practically, researchers are encouraged to apply the refinements that are compatible with their simulation approach and data, and explicitly document any choices and deviations. Such documentation matters because decisions at different steps are not independent: evaluation results reflect the combined effect of these decisions. Section 5.1 shows that this combined effect can in part be attributed to individual threats and in part arises from joint effects among them. Given this, the framework refines each step where a threat can be addressed, while the joint

effects across steps remain reflected in the evaluation results and should be made explicit through the documentation of all methodological choices. In the following, we go through the four adapted steps to detail the substeps of the framework.

6.2 Operationalization of Step 2: Extract Event Log

The framework operationalizes Step 2 of the research method to address T2—the distortion caused by fading-in and fading-out phases when extracting event logs. The goal is to obtain an event log from the source system that reflects the underlying process as accurately as possible across the chosen timeframe. To reduce T2's impact, the extraction should avoid strategies that bias the workload or case characteristics at the start and end of the log.

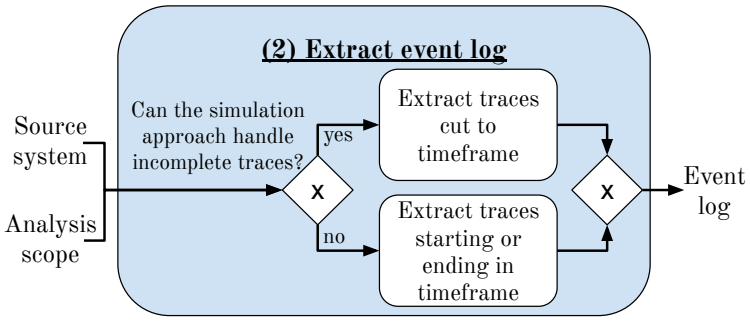


Figure 15: Subprocess of Step 2: Extract event log.

Figure 15 shows two alternatives to achieve this:

- If the simulation approach can handle incomplete traces, all events within the analysis timeframe should be extracted by cutting all traces that intersect the timeframe boundaries. This reduces the effect of the fading-in and fading-out phases on the workload in the event log.
- If the simulation approach cannot handle incomplete traces, all traces that either start or end within the chosen analysis timeframe (i.e., traces that intersect

it) should be extracted. Including intersecting traces prevents a bias toward systematically shorter cases at the timeframe boundaries that would result from retaining only fully contained traces instead, as shown in Section 4.2.

While these strategies cannot fully mitigate T2, they reduce its impact on evaluation results.

6.3 Operationalization of the Additional Step: Detect Concept Drift

In this step, the framework explicitly addresses T1—concept drift, i.e., changes in the process over time that can distort evaluation results if left unchecked. The goal is to detect whether concept drift is present and, if so, adjust the timeframe or subsequent steps of the research method to ensure valid evaluation results.

As shown in Figure 16, the step has two parts. First, concept drift detection needs to be applied to identify whether it exists in the data. This can be done by using established process mining approaches, including control-flow-based approaches (Elkhawaga et al. 2020), visual techniques (Wuyts et al. 2023), and quantitative methods (Kraus and Van der Aa 2025) extending beyond the control-flow perspective. Since these approaches require an event log as input, this additional step in the framework is required to address T1 after the event log extraction.

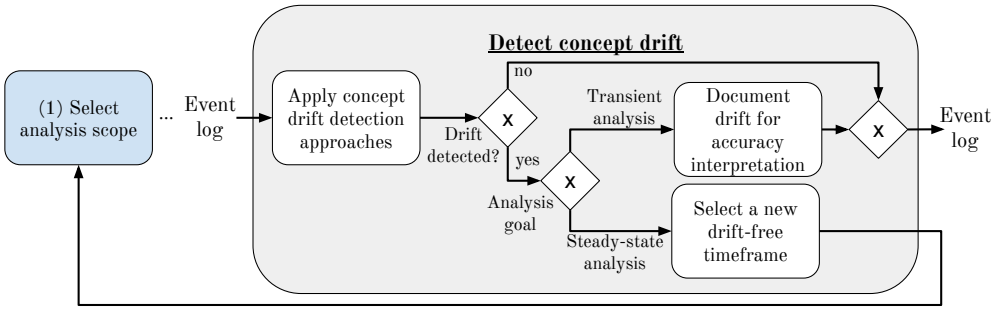


Figure 16: Subprocess of the additional step: Detect concept drift

Second, if no drift is detected, the framework continues with Step 3. If, however, concept drift is detected, the next substeps depend on the simulation's analysis goal:

- *Steady-state analysis* aims to analyze the long-run stationary behavior of a process, which requires the process to be in such a steady state for a longer period of time (van der Aalst et al. 2010). As discussed in Section 4.1, concept drift opposes this requirement and threatens the validity of such an analysis. Hence, the detection results should be used to reselect the timeframe of the analysis scope in Step 1 and extract a drift-free event log in Step 2 before continuing.
- *Transient analysis* focuses on the process behavior starting from a recent state and over a limited horizon (van der Aalst et al. 2010). Because drift is expected to continue in such a setting, documenting detected drift becomes essential for correctly interpreting evaluation results. The measured distance between the simulated and the test log reflects not only the BPS model's capacity to capture the training log but also its ability to recognize and continue the drift. As shown in Section 4.1, this interpretation should extend beyond the process perspective directly affected by the drift to other perspectives to which the drift may have propagated.

Through this handling, the framework proceeds with a log suitable for the chosen analysis goal.

6.4 Operationalization of Step 3: Split Event Log

The goal of the framework's conceptual refinement of Step 3 is to produce strictly separated training and test logs. More specifically, the goal is to (i) mitigate the risks of fading-out and fading-in phases around the splitting point (T3) and data leakage (T4) and (ii) preserve sufficient data for model discovery and evaluation. To achieve this, the framework recommends a splitting strategy that fits the simulation approach and characteristics of the event log at hand. The substeps shown in Figure 17 operationalize these goals as follows.

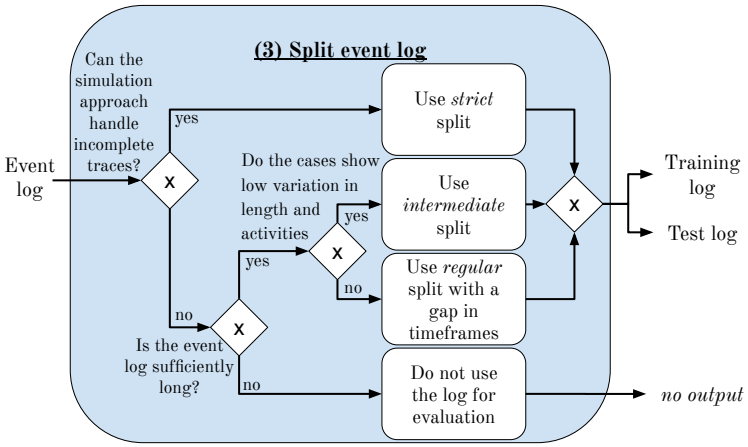


Figure 17: Subprocess of Step 3: Split event log.

- For simulation approaches capable of handling incomplete traces, applying a strict split helps reduce the effects of T3 while preserving the accuracy of BPS model discovery.
- If this is not the case, the choice depends on the structural characteristics of the event log (cf. Section 5.2). For a sufficiently long event log, there are two options:
  - If the cases are short with low variation in length and number of activities, the intermediate split should be applied to strictly separate the training and test log. On such logs, it avoids the data leakage of the regular split without meaningfully distorting evaluation results (cf. Section 5.2).
  - Else, if the cases are longer with higher variation, a regular split may be applied, incorporating an additional gap between the training and test timeframes to allow the training cases to fade out before the test cases start. This is analogous to practices in PPM (Weytjens and De Weerdt 2022) but differs in that full cases are preserved.
- If the event log is not sufficiently long, the trade-off between T3 and T4 cannot be resolved through the choice of data-splitting strategy (cf. Section 5.2). In this case, the log should not be used for evaluation.

Thus, the output of all three options in this step is a strictly separated training log and test log.

6.5 Operationalization of Step 5: Simulate Process

The framework's refinement of Step 5 generates a simulated log under realistic workload conditions by ensuring that warm-up and cool-down phases occur before and after the simulation of the log, thereby avoiding the distortions caused by T5.

Whether managing these phases requires additional steps depends on the chosen simulation approach. Simulation approaches that do not explicitly manage a process's internal states, including resources and their queues, require no further handling. For example, timestamp-prediction methods (e.g., deep learning approaches (Camargo et al. 2019)) inherently avoid warm-up and cool-down distortions and may proceed by simulating the process directly. In contrast, if simulation approaches internally manage resources and their queues, the substeps illustrated in Figure 18 should be applied.

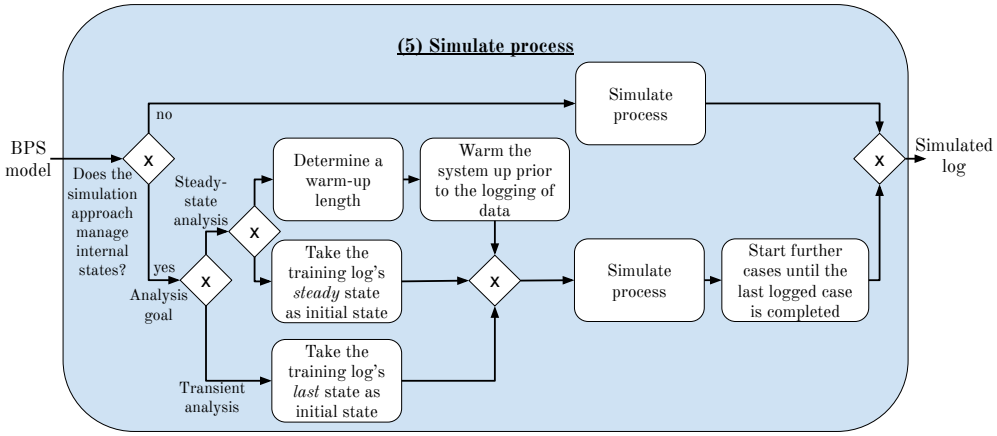


Figure 18: Subprocess of Step 5: Simulate process.

Queue-based systems require two actions for realistic workload conditions:

- *Warm-up:* How to avoid initialization bias depends on the analysis goal:
  - For *steady-state analysis*, which targets long-term performance evaluation, the system should reach a stable state before evaluation begins (Law 2015). This

can be achieved either by warming up the simulation before data logging starts—with the length of the warm-up period determined using techniques based on resource utilization and stability detection (Mahajan and Ingalls 2004)—or by initializing the simulation from a steady state reconstructed from the training log (Pourbafrani et al. 2023).

  - For *transient analysis*, the goal is to fast-forward from the most recent process state, which should be derived from the end of the training log (Rozinat et al. 2009). Recently, an approach to automatically discover this state has been proposed (Avramenko et al. 2025), allowing for a seamless transition from the training log to transient analysis.

- *Cool down:* Toward the end of the simulation, further cases should continue to start until all ongoing cases in the simulated log are completed. These additional cases are not included in the simulated log but allow ongoing cases to finish naturally, mitigating premature workload reductions.

By integrating these substeps, the simulated log more accurately represents the true quality of the simulation model, free from distortions due to artificially low load conditions during simulation.

6.6 Limitations and Discussion of the Conceptual Framework

This section reflects on the limitations of the conceptual framework and discusses its implications for applying the data-driven BPS research method. Specifically, it examines (i) its dependence on event data characteristics, (ii) its reliance on external analytical techniques, and (iii) its grounding in the empirical results reported in Section 4 and Section 5.

A first limitation concerns the framework's dependence on the characteristics of the available event data. For instance, addressing concept drift (T1) requires a sufficiently long, drift-free timeframe (which may be difficult to obtain in fast-evolving

environments), and the substeps for data extraction and splitting (T2, T3, and T4) similarly assume enough historical data. Despite this limitation, the framework helps identify when data is insufficient, and the log should not be used for evaluation.

A second limitation is that the framework cannot provide full procedural guidance for all potentially required external techniques. Instead, it references established approaches (e.g., for concept drift detection, steady-state identification, or warm-up determination), specifies when they should be applied, and indicates why they are critical depending on the context. This is particularly important because several of the approaches originate from distinct research communities and are not yet widely adopted in data-driven BPS.

A third limitation is that the framework has not been fully validated. Nonetheless, each refinement is grounded in the empirical results reported in Section 4 and Section 5, which demonstrate how threats T1–T5 affect evaluation outcomes. For instance, Section 4.1 shows that concept drift impacts evaluation results when compared to a drift-free setting. The framework, therefore, describes how researchers can detect concept drift and select a drift-free timeframe. In this sense, the experiments already show how applying each refinement improves internal validity relative to ignoring the threat. Thus, while the framework has not been evaluated as a whole, its individual refinements are directly supported by empirical evidence.

Taken together, these limitations emphasize that not all identified threats can always be eliminated by following the framework. Rather, it helps mitigate distortions where possible and makes the potential impact of remaining threats explicit. We therefore advocate for explicit reporting of all choices (such as timeframe boundaries, data-splitting strategies, and warm-up settings) and sensitivity analyses showing how evaluation results vary across configurations. Such transparency improves the validity and replicability of individual studies (Rehse et al. 2024) and lays the groundwork for systematic comparisons and future refinements in BPS evaluation.

7 Conclusion

In this paper, we examined the internal validity of BPS evaluations, a critical factor for ensuring that simulation-based assessments of process changes are accurate and reliable. Specifically, we studied the research method that is typically used in data-driven BPS to achieve a high-quality BPS model. In doing so, we identified five key threats to internal validity that arise from different steps in the method and demonstrated how these can distort evaluation results. Based on our analysis, we derived a conceptual framework to help mitigate these threats where possible.

In light of long-standing efforts to achieve valid evaluations in other fields of process mining, we aim to raise awareness of these issues in BPS. We do not claim that our list of threats is complete; in fact, any design decision in an evaluation can introduce a new potential threat to validity (Rehse et al. 2024). The impact of these threats is also specific to the event log in question: depending on factors like the length of the fading-in phase, some threats may be more or less prominent.

Future research should focus on identifying additional threats to the validity of BPS evaluations and exploring ways to mitigate them. While this work focuses on internal validity, other aspects, such as experimental design or generalizability, may also threaten validity and should be considered in future studies. Looking forward, the development of concrete strategies to address and minimize these threats will be essential to improve the reliability of evaluation results. Although these threats cannot be entirely avoided, careful decision-making, as shown in our examples and framework, can reduce or at least vary some of their impacts.

We believe that as the field of BPS advances, its evaluation practices should evolve in response to these insights, and that our work contributes to helping researchers and practitioners better understand where simulation approaches excel and where further improvements are needed.

Statements and Declarations

Ethics approval and consent to participate

Not applicable.

Funding

Timotheus Kampik was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

Competing Interests

The authors have no relevant financial or non-financial interests to disclose.

Availability of data and materials

The implementation, datasets, experimental details, and obtained raw results are available through the repository linked in the Evaluation section.

Authors' contributions

All authors contributed to the work's conception; R.B. suggested the main idea and conducted the experiments; R.B, L.K., J.R., T.K., and H.A. wrote and revised the manuscript throughout all its versions.

References

Avramenko, M., D. Chapela-Campa, M. Dumas, and F. Milani 2025. What's coming next? short-term simulation of business processes from current state. In *ICPM*, pp. 1–8. IEEE.

Balci, O. 1994. *Principles of simulation model validation, verification, and testing.* Virginia Polytechnic Institute & State University.

Balci, O. 2012. A life cycle for modeling and simulation. *Simulation 88*(7): 870–883 .

Banks, J. 1998. *Handbook of simulation: principles, methodology, advances, applications, and practice.* John Wiley & Sons.

Banks, J. and L. Chwif. 2011. Warnings about simulation. *Journal of simulation 5*(4): 279–291 .

Barth, R., M. Meyer, and J. Spitzner. 2012. Typical pitfalls of simulation modeling: lessons learned from armed forces and business. *The journal of artificial societies and social simulation 15*(2): 5 .

Bemthuis, R.H. and S. Lazarova-Molnar. 2025. Towards integrating process mining with agent-based modeling and simulation: State of the art and outlook. *Expert Systems with Applications*: 127571 .

Camargo, M., D. Báron, M. Dumas, and O. González-Rojas. 2023. Learning business process simulation models: a hybrid process mining and deep learning approach. *Information Systems* 117: 102248 .

Camargo, M., M. Dumas, and O. González-Rojas 2019. Learning accurate LSTM models of business processes. In *BPM*, pp. 286–302. Springer.

Camargo, M., M. Dumas, and O. González-Rojas. 2020. Automated discovery of business process simulation models from event logs. *Decision Support Systems* 134: 113284 .

Camargo, M., M. Dumas, and O. González-Rojas. 2021. Discovering generative models from event logs: data-driven simulation vs deep learning. *PeerJ Computer Science* 7:

e577 .

Camargo, M., M. Dumas, and O. González-Rojas 2022. Learning accurate business process simulation models from event logs via automated process discovery and deep learning. In X. Franch, G. Poels, F. Gailly, and M. Snoeck (Eds.), *CAiSE*, pp. 55–71. Springer International Publishing.

Chapela-Campa, D., I. Benchekroun, O. Baron, M. Dumas, D. Krass, and A. Senderovich 2023. Can i trust my simulation model? measuring the quality of business process simulation models. In *BPM*, pp. 20–37. Springer.

Chapela-Campa, D., I. Benchekroun, O. Baron, M. Dumas, D. Krass, and A. Senderovich. 2025. A framework for measuring the quality of business process simulation models. *Information Systems* 127: 102447 .

Chapela-Campa, D. and M. Dumas. 2024. Enhancing business process simulation models with extraneous activity delays. *Information Systems* 122: 102346 .

Cohen, J. 1988. *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum Associates.

de França, B. and G. Travassos. 2015. Simulation based studies in software engineering: A matter of validity. *CLEI electronic journal 18*(1): 4–1 .

Di Francescomarino, C. and C. Ghidini. 2022. Predictive process monitoring, *Process mining handbook*, 320–346. Springer International Publishing Cham.

Dumas, M., M. Rosa, J. Mendling, and H. Reijers. 2018. *Fundamentals of business process management*. Springer.

Elkhawaga, G., M. Abuelkheir, S. Barakat, A. Riad, and M. Reichert. 2020. Conda-pm—a systematic review and framework for concept drift analysis in process

mining. *Algorithms 13*(7): 161 .

Estrada-Torres, B., M. Camargo, M. Dumas, L. García-Bañuelos, I. Mahdy, and M. Yerokhin. 2021. Discovering business process simulation models in the presence of multitasking and availability constraints. *Data & Knowledge Engineering* 134: 101897 .

Frank, U. 2006. Towards a pluralistic conception of research methods in information systems research. ICB-Research Report 7, Universität Duisburg-Essen.

Halawa, F., S.C. Madathil, and M.T. Khasawneh. 2021. Integrated framework of process mining and simulation–optimization for pod structured clinical layout design. *Expert Systems with Applications* 185: 115696 .

Jans, M., P. Soffer, and T. Jouck. 2019. Building a valuable event log for process mining: an experimental exploration of a guided process. *Enterprise Information Systems 13*(5): 601–630 .

Kaufman, S., S. Rosset, C. Perlich, and O. Stitelman. 2012. Leakage in data mining: Formulation, detection, and avoidance. *TKDD 6*(4): 1–21 .

Kirchdorfer, L., R. Blümel, T. Kampik, H. Van der Aa, and H. Stuckenschmidt 2024. Agentsimulator: An agent-based approach for data-driven business process simulation. In *ICPM*, pp. 97–104. IEEE.

Kirchdorfer, L., R. Blümel, T. Kampik, H. van der Aa, and H. Stuckenschmidt. 2025. Discovering multi-agent systems for resource-centric business process simulation. *Process Science 2*(1): 4 .

Kirchdorfer, L., K. Özdemir, S. Kusenic, H. van der Aa, and H. Stuckenschmidt 2025. A divide-and-conquer approach for modeling arrival times in business process

simulation. In *BPM*, pp. 325–342. Springer.

Koivisto, M. 2017. Pitfalls in modeling and simulation. *Procedia computer science* 119: 8–15 .

Kraus, A. and H. Van der Aa 2025. On the use of steady-state detection for process mining: Achieving more accurate insights. In *CAiSE*, pp. 204–220. Springer.

Law, A. 2015. *Simulation Modeling and Analysis* (5 ed.). New York: McGraw-Hill Education.

Law, A.M. 2003. Designing a simulation study: how to conduct a successful simulation study. In S. E. Chick, P. J. Sanchez, D. M. Ferrin, and D. J. Morrice (Eds.), *WSC*, pp. 66–70. IEEE Computer Society.

López-Pintado, O. and M. Dumas 2022. Business process simulation with differentiated resources: Does it make a difference? In *BPM*, pp. 361–378. Springer International Publishing.

López-Pintado, O. and M. Dumas 2023. Discovery and simulation of business processes with probabilistic resource availability calendars. In *ICPM*, pp. 1–8.

López-Pintado, O. and M. Dumas. 2025. Business process simulation: Probabilistic modeling of intermittent resource availability and multitasking behavior. *Information Systems* 127: 102471 .

López-Pintado, O., S. Murashko, and M. Dumas 2024. Discovery and simulation of data-aware business processes. In *ICPM*, pp. 105–112. IEEE.

Mahajan, P. and R. Ingalls 2004. Evaluation of methods used to detect warm-up period in steady state simulation. In *WSC*, Volume 1, pp. 663–671. IEEE.

Martin, N., B. Depaire, and A. Caris. 2016. The use of process mining in business process simulation model construction: structuring the field. *Business & Information Systems Engineering* 58: 73–87 .

Mendling, J., B. Depaire, and H. Leopold. 2021. Theory and practice of algorithm engineering. *arXiv preprint arXiv:2107.10675* .

Meneghello, F., C. Di Francescomarino, and C. Ghidini 2023. Runtime integration of machine learning and simulation for business processes. In *ICPM*, pp. 9–16. IEEE.

Meneghello, F., C.D. Francescomarino, C. Ghidini, and M. Ronzani. 2025. Runtime integration of machine learning and simulation for business processes: Time and decision mining predictions. *Information Systems* 128: 102472 .

Oldenburg, F., K. Hoberg, and H. Leopold. 2025. Process mining in supply chain management: state-of-the-art, use cases and research outlook. *International Journal of Production Research 63*(8): 2889–2904 .

Özdemir, K., L. Kirchdorfer, K.A. Elyasi, H. van der Aa, and H. Stuckenschmidt 2025. Rethinking business process simulation: A utility-based evaluation framework. In *BPM Forum*, pp. 128–144. Springer.

Pace, D.K. 2004. Modeling and simulation verification and validation challenges. *Johns Hopkins APL technical digest 25*(2): 163–172 .

Park, G., M. Cho, and J. Lee. 2024. Leveraging machine learning for automatic topic discovery and forecasting of process mining research: A literature review. *Expert Systems with Applications* 239: 122435 .

Peters, S., Y. Kerner, R. Dijkman, I. Adan, and P. Grefen. 2022. Fast and accurate quantitative business process analysis using feature complete queueing models.

Information systems 104: 101892 .

Pfeiffer, P., L. Abb, P. Fettke, and J.R. Rehse. 2025. Learning from the data to predict the process. *Business & Information Systems Engineering*: 1–27 .

Pourbafrani, M., N. Lücking, M. Lucke, and W. Van der Aalst 2023. Steady state estimation for business process simulations. In *BPM*, pp. 178–195. Springer.

Pourbafrani, M. and W.M. van Der Aalst. 2022. Discovering system dynamics simulation models using process mining. *IEEE Access* 10: 78527–78547 .

Rehse, J.R., S. Leemans, P. Fettke, and J. Van der Werf. 2024. On process discovery experimentation. *ACM Transactions on Software Engineering and Methodology* .

Roungas, B., S.A. Meijer, and A. Verbraeck. 2018. A framework for optimizing simulation model validation & verification. *International Journal on Advances in Systems and Measurements 11*(1): 137–152 .

Rozinat, A., R. Mans, M. Song, and W. Van der Aalst. 2009. Discovering simulation models. *Information systems 34*(3): 305–327 .

Rozinat, A., M.T. Wynn, W.M. van der Aalst, A.H. ter Hofstede, and C.J. Fidge. 2009. Workflow simulation for operational decision support. *Data & Knowledge Engineering 68*(9): 834–850 .

Sadowski, R. 1989. The simulation process: avoiding the problems and pitfalls. In *WSC*, New York, NY, USA, pp. 72–79. Association for Computing Machinery.

Sargent, R.G. 2001. Some approaches and paradigms for verifying and validating simulation models. In *WSC*, Volume 1, pp. 106–114. IEEE.

Sargent, R.G. 2020. Verification and validation of simulation models: an advanced tutorial. In *WSC*, pp. 16–29. IEEE.

Sato, D., S. De Freitas, J. Barddal, and E. Scalabrin. 2021. A survey on concept drift in process mining. *CSUR 54*(9): 1–38 .

Schruben, L.W. 1982. Detecting initialization bias in simulation output. *Operations Research 30*(3): 569–590 .

Senderovich, A., C. Di Francescomarino, C. Ghidini, K. Jorbina, and F. Maggi 2017. Intra and inter-case features in predictive process monitoring: A tale of two dimensions. In *BPM*, pp. 306–323. Springer.

Swanborn, P. 1996. A common base for quality control criteria in quantitative and qualitative research. *Quality and quantity 30*(1): 19–35 .

Troost, C., R. Huber, A.R. Bell, H. Van Delden, T. Filatova, Q.B. Le, M. Lippe, L. Niamir, J.G. Polhill, Z. Sun, et al. 2023. How to keep it adequate: A protocol for ensuring validity in agent-based simulation. *Environmental Modelling & Software* 159: 105559 .

Uhrmacher, A.M. 2012. Seven pitfalls in modeling and simulation research. In *WSC*, pp. 1–12. IEEE.

van der Aalst, W. 2015. Business process simulation survival guide, In *Handbook on Business Process Management 1, Introduction, Methods, and Information Systems, 2nd Ed*, eds. vom Brocke, J. and M. Rosemann, International Handbooks on Information Systems, 337–370. Springer.

Van der Aalst, W. 2016. *Process Mining: Data Science in Action*. Berlin Heidelberg: Springer.

van der Aalst, W., J. Nakatumba, A. Rozinat, and N. Russell. 2010. Business process simulation, In *Handbook on Business Process Management 1: Introduction, Methods, and Information Systems*, eds. Brocke, J.v. and M. Rosemann, International Handbooks on Information Systems, 313–338. Springer.

Van der Werf, J., A. Polyvyanyy, B. Van Wensveen, M. Brinkhuis, and H. Reijers. 2023. All that glitters is not gold: Four maturity stages of process discovery algorithms. *Information Systems* 114: 102155 .

Van Eck, M., X. Lu, S. Leemans, and W. Van der Aalst 2015. Pm^2: a process mining project methodology. In *CAiSE*, pp. 297–313. Springer.

Weytjens, H. and J. De Weerdt 2022. Creating unbiased public benchmark datasets with data leakage prevention for predictive process monitoring. In *BPM Workshops*, pp. 18–29. Springer.

Williams, E.J. and O.M. Ülgen 2012. Pitfalls in managing a simulation project. In *WSC*, pp. 1–8.

Wuyts, B., H. Weytjens, S. Vanden Broucke, and J. De Weerdt 2023. DyLoPro: Profiling the dynamics of event logs. In *BPM*, pp. 146–162. Springer.