Knowledge-Driven Hallucination in Large Language Models: An Empirical Study on Process Modeling

Humam Kourani[1,2][0000−0003−2375−2152], Anton Antonov[1,2][0009−0004−1044−4884], Alessandro Berti[2][0000−0002−3279−4795], and Wil M.P. van der Aalst[2,1][0000−0002−0955−6940]

[1] Fraunhofer Institute for Applied Information Technology FIT, Schloss Birlinghoven, 53757 Sankt Augustin, Germany
{humam.kourani,anton.antonov}@fit.fraunhofer.de
[2] RWTH Aachen University, Ahornstraße 55, 52074 Aachen, Germany
{a.berti,wvdaalst}@pads.rwth-aachen.de

Abstract. The utility of Large Language Models (LLMs) in analytical tasks is rooted in their vast pre-trained knowledge, which allows them to interpret ambiguous inputs and infer missing information. However, this same capability introduces a critical risk of what we term *knowledge-driven hallucination*: a phenomenon where the model's output contradicts explicit source evidence because it is overridden by the model's generalized internal knowledge. This paper investigates this phenomenon by evaluating LLMs on the task of *automated process modeling*, where the goal is to generate a formal business process model from a given source artifact. The domain of Business Process Management (BPM) provides an ideal context for this study, as many core business processes follow standardized patterns, making it likely that LLMs possess strong pre-trained schemas for them. We conduct a controlled experiment designed to create scenarios with deliberate conflict between provided evidence and the LLM's background knowledge. We use inputs describing both standard and deliberately atypical process structures to measure the LLM's fidelity to the provided evidence. Our work provides a methodology for assessing this critical reliability issue and raises awareness of the need for rigorous validation of AI-generated artifacts in any evidence-based domain.

Keywords: Large Language Models · Hallucination · Generative AI · Trustworthy AI · Process Modeling.

1 Introduction

The integration of Large Language Models (LLMs) and other foundation models into analytical and data-driven workflows promises to automate complex tasks and democratize access to specialized domains. A key capability driving this transformation is the models' ability to leverage vast, pre-trained knowledge to

interpret ambiguous inputs, infer missing details, and generate coherent, structured outputs. This capacity for *intelligent inference* is fundamental to their performance across a range of applications, from code generation to data analysis.

Reliance of an LLM on its internal knowledge base introduces a critical challenge that we term *knowledge-driven hallucination*. This conflict arises when the explicit evidence provided in a user's prompt (e.g., a source document, a dataset, or a simple text) is inconsistent with the generalized patterns and "common sense" knowledge the model has acquired during its training. In such situations, the model faces a dilemma: should it remain faithful to the provided evidence, even if it appears anomalous or counter-intuitive, or should it "correct" the output based on its pre-trained understanding of what is typical or plausible?

The outcome of this conflict poses implications for the trustworthiness and reliability of AI-driven systems. An LLM that prioritizes its internal knowledge over explicit evidence may generate outputs that are dangerously misleading. The generated artifact might appear well-formed, logical, and plausible, yet fail to accurately represent the specific reality of the input data. This risk is particularly acute in specialized domains where processes, rules, or data may be intentionally unconventional and deviate from established norms.

This paper investigates the knowledge-driven hallucination of LLMs through a systematic, empirical study within the domain of *Business Process Management (BPM)*. The task of *process modeling* (i.e., generating a formal process model from a source artifact) is particularly well-suited for our investigation due to the nature of business processes themselves. Many core business operations, such as purchase-to-pay, order-to-cash, or incident management, follow well-established, standardized patterns across different organizations. Consequently, it is highly probable that an LLM has been exposed to extensive documentation and descriptions of these standard processes during its training, endowing it with a strong pre-trained "schema" of how such processes "should" operate. This creates a powerful and realistic dilemma for our study: what occurs when the evidence provided for a specific organization's process directly contradicts the generalized, common-sense model of that process residing within the LLM?

To isolate and quantify this conflict, we conduct a controlled experiment using a set of standard process models (M^+) that represent conventional process flows. For each standard model, we create two deliberately conflicting variants: a reversed model (M^-), where the sequence of activities is causally inverted, and a shuffled model (M^*), where the original activity labels are randomly permuted across the model's structure. We then task an LLM to generate process models from textual descriptions and event logs derived from these variants. By comparing the generated models against both the source evidence and the conventional standard model, we measure the degree to which the LLM adheres to the provided evidence versus reverting to its internal knowledge.

The structure of this paper is as follows. Section 2 discusses related work on process modeling and LLM hallucinations. Section 3 details our experimental

methodology, while Section 4 presents our findings. Finally, Section 5 concludes the paper and discusses the broader implications of our work.

2 Related Work

This section reviews related research, focusing on LLM hallucination and process modeling techniques.

2.1 LLM Hallucination

The phenomenon of *hallucination* in LLMs—where models generate outputs that appear plausible yet are factually incorrect—has been extensively studied in the literature [30]. The underlying causes of such behavior can broadly be categorized into three groups: hallucinations arising from data, from training, and from inference [20].

LLMs are trained on two main types of data: pre-training data, which imparts general and factual knowledge [9], and alignment data, which instructs models to follow human preferences and respond to user intent [29]. However, the pre-training corpus is inherently limited and often biased toward general knowledge [21], restricting the model's ability to generalize to domain-specific queries [7].

The training process itself imposes further constraints. Pre-training limits the effective context length a model can utilize during inference [25,22], leading to incomplete conditioning on the user's input. Moreover, fine-tuning with human feedback may encourage the model to favor responses that align with user preferences, even when they are not strictly truthful [14,24].

At inference time, LLMs generate output in an autoregressive manner, predicting the next token based on previously generated tokens and their internalized knowledge [5]. As responses become longer, models are increasingly prone to forgetting earlier parts of the prompt [3], which can degrade the coherence and factuality of their outputs. Apart from that, flawed reasoning may also introduce hallucinations. For instance, Berglund et al. [23] identify the "Reversal Curse", where a model that correctly answers "A is B" may fail when asked to infer "B is A".

2.2 Process Modeling

Business process modeling is the structured representation of the tasks, decisions, and flow within a business process [10]. Organizations often rely on process models to document their workflows, and the creation of these models typically involves collaboration between business analysts and domain experts to ensure clarity and accuracy [11].

Traditionally, creating business process models required substantial manual effort and expertise in complex modeling languages. To automate this, early approaches primarily relied on traditional Natural Language Processing (NLP) and rule-based techniques [28]. These methods exploited dependency parsing, part-of-speech tagging, and semantic role labeling to identify process elements from unstructured text [27,8]. For instance, researchers combined NLP with computational linguistics to generate BPMN models [12], used text mining to derive

Table 1: Characteristics of the selected processes.

Process	Ground Truth Petri Net				
	#Activities	#Nodes	Decisions	Cycles	Concurrency
Sales Order (p1)	8	26	×		×
Booking System (p7)	13	49	×	×	×
Complaint Handling (p13)	9	21	×		
Internal Audit (p16)	24	63	×	×	×

models directly from text [1], and applied NLP to extract structured process representations [26]. However, these traditional methods were often hindered by the inherent ambiguity and variability of natural language, necessitating significant human intervention and preventing full automation [2].

The advent of LLMs has marked a paradigm shift in this domain. A significant line of research investigates the use of LLMs for generating process models directly from various inputs. Studies have demonstrated the ability of LLMs to generate process models from unstructured text [17,6] and to translate textual descriptions into both procedural and declarative process model constraints [13]. Beyond direct generation, other methods explore more interactive approaches, such as creating models through dialogue-based systems and chatbots [15].

3 Evaluation Methodology

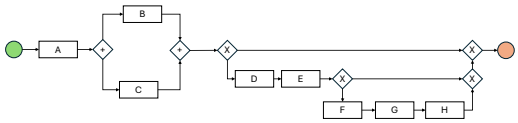
Our methodology is designed to create a controlled environment where we can systematically evaluate an LLM's tendency for knowledge-driven hallucination. The experiment consists of three main stages: (1) the generation of standard and conflicting process artifacts, (2) the procedure for generating process models using LLMs, and (3) the protocol for evaluating the generated models.

3.1 Artifact Generation

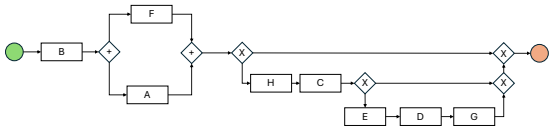
We base our experiments on four diverse business processes selected from the benchmark presented in [16]. For each of these processes, a set of ground truth artifacts already exists, which we designate as the *standard* or *expected* versions: a standard model (M^+), a corresponding natural language description (D^+), and a simulated event log (L^+). Table 1 summarizes key dimensions of the selected processes. For each process, we report the number of activities and the number of nodes (transitions + places) in its ground truth Petri net. Furthermore, we indicate whether the process contains key control-flow constructs: decision points, cycles, and concurrency.

From these standard artifacts, we systematically generate two sets of conflicting evidence:

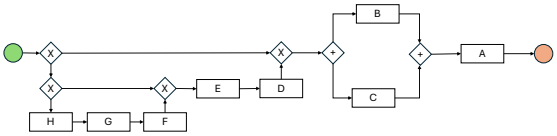
- **Reversed Artifacts (M^-, L^-, D^-):** The reversed model (M^-) was created by manually reversing all sequential dependencies in M^+. The reversed log (L^-) was generated by reversing the event order in each trace of L^+. Finally, the reversed description (D^-) was created by manually adjusting D^+ to match the new process flow of M^-.



(a) Original process model (M^+).



(b) Shuffled process model (M^*).



(c) Reversed process model (M^-).

Abbreviation	Original Activity Label
A	Receive customer inquiry
B	Collect customer information
C	Address customer concerns or questions
D	Guide customer in selecting product/service
E	Provide quote
F	Place order
G	Record order in system
H	Send order confirmation to customer

(d) Mapping of abbreviations used in the figures to the original activity labels.

Fig. 1: Ground truth process models for the sales order process.

Listing 1: Reversed textual description (D^-) for the sales order process.

```
First a confirmation of the order may be sent to the customer. If the customer
 receives a confirmation, then the order is recorded in the system and the
 order is placed. After that, the sales representative provides a quota and
 the customer is guided in selecting product or services. However, all
 previous steps can be skipped. Then, the sales staff or customer support
 addresses any concerns or questions and collects relevant information, at
 the same time. Finally, the department receives a potential customer inquiry
 about a product or service.
```

– **Shuffled Artifacts (M^*, L^*, D^*):** The shuffled model (M^*) was created by applying a random, bijective mapping of activity labels to the standard model M^+, preserving its control-flow structure. The shuffled log (L^*) was created by applying the same mapping to the activity names in L^+. The corresponding description (D^*) was then derived manually.

This setup provides us with six distinct input scenarios for the LLM: three based on text descriptions (D^+, D^-, D^*) and three based on event logs (L^+, L^-, L^*). To compactly represent event logs as textual input for LLMs, we generate a textual abstraction for each log using the process mining library PM4Py [4].

To illustrate the conflicting artifacts, Figure 1d shows the standard, shuffled, and reversed models for the sales order process. The corresponding reversed description (D^-) and the textual abstraction of the reversed event log (L^-) are provided in Listing 1 and Listing 2, respectively.

Listing 2: The textual abstraction generated with PM4Py [4] for the reversed event log (L^-) for the sales order process.

```
Send order confirmation to customer -> Record order in system -> ...
Send order confirmation to customer -> Record order in system -> ...
Provide quote -> Guide customer in selecting product/service -> ...
Provide quote -> Guide customer in selecting product/service -> ...
Collect customer information -> Address customer concerns or questions -> ...
Address customer concerns or questions -> Collect customer information -> ...
```

Table 2: Characteristics of the evaluated LLMs, including open-source status, reasoning capabilities, parameter estimates, announcement dates, and LiveBench 2025-07-30 `https://livebench.ai/` leaderboard scores.

Model	Open-Source	Reasoning	Parameters	Announcement	LB Score
command-r	×		35B	2024-08-30	27.15
gemini-2.5-flash		×	est. 400B, 20B act.	2025-06-17	64.42
gemini-2.5-pro		×	est. 1500B, 40B act.	2025-06-17	69.39
gpt-4.1-nano			est. 18B, 2B act.	2025-04-14	40.51
grok-3-fast			est. 2700B, 50B act.	2025-04-09	56.05
grok-3-mini-fast		×	est. 250B, 35B act.	2025-04-09	62.36
kimi-k2	×		≈ 1000B, 32B act.	2025-07-11	62.70
o3		×	est. 200B	2025-04-16	71.98
o4-mini		×	est. 60B, 8B act.	2025-04-16	66.87
qwen3-235b-a22b	×		235B, 22B act.	2025-07-25	64.72

3.2 LLM-based Model Generation Procedure

The characteristics of the large language models evaluated in this study are summarized in Table 2. We define two primary tasks for the LLMs:

- **Text-to-Model Generation:** The LLM is prompted to generate a process model from each of the textual descriptions (D^+, D^-, D^*). We utilize the ProMoAI framework from [18], which generates models in the POWL language [19] and subsequently converts them into Petri nets or BPMN diagrams for analysis.
- **Log-to-Model Generation:** The LLM is provided with the textual abstractions of the event logs (L^+, L^-, L^*) and is prompted to discover a process model that explains the behavior in each log. This experiment was also executed using the ProMoAI framework, with the event log abstraction serving as the input process description.

To investigate the LLM's sensitivity to prompting, we conducted all experiments under two distinct conditions:

- **Standard Prompt:** The original, optimized prompt from ProMoAI.
- **Strict Adherence Prompt:** The standard prompt is adjusted with an explicit instruction for the LLM to disregard its background knowledge and fully rely on the provided input.

To ensure that a quantitative comparison is meaningful and can be fully automated, we standardize the activity labels across all experiments. For each

process, the prompt provided to the LLM is extended to include the complete list of valid activity labels. This experimental design focuses the evaluation purely on the discovered control-flow structure, thereby removing any ambiguity that could arise from the LLM generating synonymous or differently phrased activity names.

3.3 Evaluation Protocol

Our evaluation protocol aims to quantify the tension between evidence adherence and knowledge reversion. To measure the relationship between a generated model and the ground truth variants, we compute their *semantic similarity*. This is quantified using the behavioral-footprint similarity implemented in the PM4Py library [4]. We acknowledge that more robust evaluation techniques, such as formal conformance checking as proposed in [16], exist for assessing model quality. However, the primary objective of our study is not to ascertain the absolute correctness of the generated models, but rather to perform a relative comparison. Our goal is to determine which of the three ground truth variants (M^+, M^-, or M^*) a generated model most closely resembles. The footprint-based similarity metric is well-suited for this purpose, as our analysis focuses on identifying the highest similarity score in each comparison rather than on the absolute values of the scores themselves.

For each LLM-generated process model, we compute its semantic similarity against all three ground truth models: M^+, M^-, and M^*. Ideally, a model generated from conflicting evidence (e.g., from D^- or L^-) should exhibit high similarity to its corresponding ground truth (M^-). Our central hypothesis is that knowledge-driven hallucination will cause the generated model to show significant similarity to the standard model (M^+) instead.

4 Results and Discussion

The results of our experiments are summarized[3] in Table 3 and Table 4. For each process model generated by an LLM, we report three semantic similarity scores, comparing it against the standard (M^+), reversed (M^-), and shuffled (M^*) ground truth models. To facilitate analysis, we highlight the highest similarity score for each generated model. The cell is colored green if the highest score corresponds to the correct ground truth artifact (e.g., a model from D^- is most similar to M^-), indicating successful adherence to the provided evidence. Conversely, the cell is colored red if the generated model is most similar to the standard process model (M^+) despite being generated from conflicting evidence (D^-, D^*, L^-, or L^*), signaling a clear instance of knowledge-driven hallucination. We omit this highlighting in cases where all similarity scores for a given model are below 0.1, as a comparison at such a low level of quality is no longer insightful.

To summarize overall performance, we report average scores in the columns $\overline{diag}$ in Table 3 and Table 4. The first subcolumn in $\overline{diag}$ shows average scores

[3] All artifacts and results are available at `https://github.com/antonov1/process-hallucinations`.

Table 3: Semantic similarity scores for models generated from textual descriptions using standard and strict adherence prompts.

LLM		Standard Prompt														Strict Adherence Prompt													
		Sales Order			Booking			Complaint			Audit			*diag*		Sales Order			Booking			Complaint			Audit			*diag*	
		M^+	M^-	M^*	M^+	M^-	M^*	M^+	M^-	M^*	M^+	M^-	M^*		All	M^+	M^-	M^*	M^+	M^-	M^*	M^+	M^-	M^*	M^+	M^-	M^*		All
command-r	D^+	0.20	0.05	0.00	0.37	0.06	0.06	0.04	0.00	0.01	0.17	0.09	0.05	0.20	0.18	0.10	0.00	0.00	0.10	0.06	0.05	0.19	0.00	0.00	0.12	0.07	0.02	0.13	0.17
	D^-	0.03	0.00	0.03	0.12	0.02	0.02	0.00	0.90	0.06	0.12	0.23	0.04	0.29		0.10	0.16	0.05	0.12	0.02	0.02	0.00	0.58	0.00	0.12	0.23	0.04	0.25	
	D^*	0.24	0.17	0.05	0.04	0.00	0.06	0.01	0.00	0.04	0.04	0.03	0.05	0.05		0.19	0.06	0.12	0.04	0.00	0.06	0.05	0.05	0.33	0.02	0.02	0.02	0.13	
gemini-2.5-flash	D^+	0.40	0.08	0.04	0.48	0.11	0.03	1.00	0.00	0.06	0.53	0.26	0.02	0.60	0.45	0.40	0.08	0.04	0.48	0.11	0.03	0.62	0.00	0.11	0.65	0.22	0.03	0.54	0.46
	D^-	0.09	0.60	0.04	0.59	0.23	0.05	0.00	1.00	0.06	0.07	0.17	0.01	0.50		0.09	0.60	0.04	0.10	0.08	0.06	0.00	1.00	0.06	0.14	0.29	0.01	0.49	
	D^*	0.05	0.05	0.33	0.05	0.06	0.29	0.58	0.06	0.12	0.07	0.04	0.21	0.24		0.08	0.04	0.40	0.03	0.06	0.35	0.14	0.00	0.39	0.05	0.03	0.23	0.34	
gemini-2.5-pro	D^+	1.00	0.11	0.00	0.52	0.13	0.04	1.00	0.00	0.06	0.68	0.22	0.02	0.80	0.41	1.00	0.11	0.00	0.41	0.12	0.06	0.90	0.00	0.06	0.77	0.23	0.02	0.77	0.47
	D^-	0.44	0.08	0.08	0.38	0.13	0.05	0.00	1.00	0.06	0.19	0.12	0.01	0.33		0.08	0.40	0.04	0.15	0.10	0.08	0.00	1.00	0.06	0.14	0.17	0.01	0.42	
	D^*	0.35	0.08	0.08	0.27	0.09	0.06	0.80	0.00	0.12	0.08	0.06	0.17	0.11		0.07	0.04	0.32	0.09	0.09	0.16	0.58	0.06	0.12	0.04	0.04	0.30	0.22	
gpt-4.1-nano	D^+	0.46	0.00	0.00	0.01	0.00	0.01	0.69	0.00	0.05	0.01	0.00	0.01	0.29	0.23	0.91	0.11	0.00	0.23	0.07	0.04	0.08	0.00	0.00	0.06	0.04	0.02	0.32	0.22
	D^-	0.14	0.00	0.14	0.34	0.07	0.00	0.00	0.73	0.06	0.04	0.04	0.01	0.21		0.27	0.00	0.15	0.38	0.10	0.02	0.00	0.50	0.06	0.19	0.37	0.02	0.24	
	D^*	0.33	0.07	0.07	0.05	0.05	0.25	0.19	0.00	0.36	0.08	0.02	0.06	0.18		0.33	0.00	0.07	0.02	0.06	0.12	0.36	0.00	0.25	0.00	0.00	0.00	0.11	
grok-3-fast	D^+	1.00	0.11	0.00	0.21	0.07	0.04	1.00	0.00	0.06	0.42	0.21	0.00	0.66	0.50	0.55	0.00	0.00	0.20	0.06	0.06	1.00	0.00	0.06	0.45	0.22	0.01	0.55	0.55
	D^-	0.11	0.67	0.11	0.66	0.15	0.02	0.00	1.00	0.06	0.17	0.39	0.03	0.55		0.09	0.60	0.04	0.38	0.08	0.00	0.00	1.00	0.06	0.15	0.29	0.03	0.49	
	D^*	0.00	0.12	0.50	0.05	0.09	0.14	0.54	0.05	0.11	0.05	0.03	0.45	0.30		0.00	0.11	1.00	0.04	0.09	0.23	0.06	0.06	0.80	0.06	0.03	0.38	0.60	
grok-3-mini-fast	D^+	0.55	0.00	0.00	0.29	0.08	0.04	1.00	0.00	0.06	0.08	0.05	0.04	0.48	0.41	0.55	0.00	0.00	0.38	0.08	0.00	1.00	0.00	0.06	0.49	0.17	0.02	0.60	0.58
	D^-	0.11	1.00	0.11	0.24	0.14	0.08	0.00	1.00	0.06	0.16	0.33	0.01	0.62		0.11	1.00	0.11	0.17	0.00	0.04	0.00	1.00	0.06	0.16	0.33	0.01	0.58	
	D^*	0.55	0.00	0.00	0.08	0.07	0.15	0.55	0.00	0.13	0.02	0.03	0.30	0.14		0.05	0.10	0.83	0.05	0.06	0.24	0.00	0.06	0.70	0.03	0.03	0.42	0.55	
kimi-k2	D^+	0.44	0.08	0.00	0.54	0.11	0.00	0.47	0.00	0.09	0.38	0.15	0.02	0.46	0.40	1.00	0.11	0.00	0.54	0.11	0.00	1.00	0.00	0.06	0.47	0.23	0.02	0.75	0.44
	D^-	0.24	0.62	0.11	0.38	0.08	0.00	0.00	0.90	0.06	0.12	0.54	0.03	0.53		0.24	0.62	0.11	0.22	0.09	0.04	0.00	0.14	0.05	0.11	0.44	0.03	0.32	
	D^*	0.04	0.00	0.19	0.05	0.05	0.21	0.27	0.03	0.06	0.04	0.03	0.31	0.19		0.05	0.05	0.33	0.05	0.07	0.15	0.33	0.00	0.23	0.04	0.01	0.26	0.24	
o3	D^+	0.50	0.00	0.06	0.34	0.09	0.03	0.90	0.00	0.06	0.66	0.22	0.01	0.60	0.42	1.00	0.11	0.00	0.26	0.08	0.03	0.90	0.00	0.06	0.75	0.24	0.01	0.73	0.61
	D^-	0.25	0.25	0.25	0.45	0.19	0.03	0.00	1.00	0.06	0.27	0.24	0.02	0.42		0.11	1.00	0.11	0.13	0.14	0.06	0.00	1.00	0.06	0.09	0.18	0.05	0.58	
	D^*	0.06	0.06	0.50	0.07	0.05	0.14	0.58	0.06	0.12	0.04	0.02	0.24	0.25		0.05	0.10	0.83	0.05	0.05	0.19	0.00	0.06	0.64	0.03	0.02	0.44	0.53	
o4-mini	D^+	0.55	0.00	0.00	0.31	0.09	0.03	0.90	0.00	0.06	0.35	0.19	0.03	0.53	0.55	1.00	0.11	0.00	0.41	0.12	0.04	0.90	0.00	0.06	0.54	0.26	0.02	0.71	0.63
	D^-	0.11	1.00	0.11	0.66	0.15	0.03	0.00	1.00	0.06	0.30	0.40	0.02	0.64		0.11	1.00	0.11	0.17	0.19	0.05	0.00	1.00	0.06	0.15	0.40	0.02	0.65	
	D^*	0.05	0.10	0.64	0.05	0.06	0.26	0.06	0.06	0.80	0.03	0.03	0.28	0.50		0.05	0.10	0.64	0.05	0.06	0.26	0.06	0.06	0.80	0.03	0.04	0.44	0.54	
qwen3-235b-a22b	D^+	1.00	0.11	0.00	0.30	0.09	0.03	1.00	0.00	0.06	0.64	0.24	0.01	0.73	0.45	0.50	0.00	0.06	0.52	0.13	0.04	0.57	0.00	0.10	0.25	0.16	0.03	0.46	0.52
	D^-	0.24	0.62	0.11	0.66	0.15	0.02	0.00	1.00	0.06	0.26	0.23	0.03	0.50		0.11	1.00	0.11	0.34	0.11	0.04	0.00	1.00	0.06	0.16	0.51	0.04	0.66	
	D^*	0.10	0.00	0.10	0.13	0.04	0.03	0.73	0.06	0.06	0.04	0.01	0.28	0.11		0.00	0.06	0.46	0.03	0.06	0.35	0.20	0.00	0.64	0.05	0.03	0.29	0.43	

Table 4: Semantic similarity scores for models generated from event logs using standard and strict adherence prompts.

LLM		Standard Prompt														Strict Adherence Prompt													
		Sales Order			Booking			Complaint			Audit			$\overline{diag}$		Sales Order			Booking			Complaint			Audit			$\overline{diag}$	
		M^+	M^-	M^*	M^+	M^-	M^*	M^+	M^-	M^*	M^+	M^-	M^*		All	M^+	M^-	M^*	M^+	M^-	M^*	M^+	M^-	M^*	M^+	M^-	M^*		All
command-r	L^+	0.60	0.09	0.04	0.12	0.03	0.03	0.80	0.00	0.06	0.03	0.03	0.03	0.39	0.26	0.60	0.09	0.04	0.12	0.03	0.03	0.80	0.00	0.06	0.03	0.03	0.03	0.39	0.30
	L^-	0.08	0.42	0.08	0.00	0.19	0.02	0.00	0.00	0.00	0.04	0.27	0.01	0.22		0.06	0.36	0.12	0.00	0.19	0.02	0.00	0.62	0.05	0.04	0.27	0.01	0.36	
	L^*	0.00	0.06	0.42	0.04	0.04	0.11	0.11	0.00	0.18	0.00	0.00	0.00	0.18		0.05	0.05	0.33	0.04	0.03	0.11	0.00	0.02	0.15	0.00	0.00	0.00	0.15	
gemini-2.5-flash	L^+	1.00	0.11	0.00	0.75	0.19	0.03	1.00	0.00	0.06	0.68	0.18	0.05	0.86	0.57	1.00	0.11	0.00	0.62	0.15	0.02	1.00	0.00	0.06	0.67	0.18	0.04	0.82	0.70
	L^-	1.00	0.11	0.00	0.91	0.22	0.05	0.20	0.29	0.06	0.47	0.18	0.01	0.20		0.11	1.00	0.11	0.91	0.22	0.05	0.00	1.00	0.06	0.62	0.20	0.02	0.61	
	L^*	0.00	0.10	0.69	0.03	0.05	0.62	0.06	0.06	1.00	0.06	0.03	0.32	0.66		0.00	0.11	1.00	0.09	0.11	0.25	0.06	0.06	1.00	0.02	0.01	0.34	0.67	
gemini-2.5-pro	L^+	1.00	0.11	0.00	0.75	0.19	0.03	1.00	0.00	0.06	0.68	0.18	0.05	0.86	0.59	1.00	0.11	0.00	0.75	0.19	0.03	1.00	0.00	0.06	0.68	0.18	0.05	0.86	0.59
	L^-	1.00	0.11	0.00	0.91	0.22	0.05	1.00	0.00	0.06	0.68	0.15	0.02	0.12		1.00	0.11	0.00	0.91	0.22	0.05	0.00	1.00	0.06	0.68	0.15	0.03	0.37	
	L^*	0.00	0.11	1.00	0.03	0.03	0.75	0.06	0.06	1.00	0.03	0.03	0.47	0.80		0.04	0.00	0.19	0.02	0.03	0.60	0.06	0.06	1.00	0.06	0.07	0.41	0.55	
gpt-4.1-nano	L^+	0.83	0.10	0.00	0.03	0.00	0.00	0.62	0.00	0.05	0.21	0.01	0.05	0.42	0.42	1.00	0.11	0.00	0.06	0.06	0.04	0.82	0.00	0.05	0.04	0.00	0.01	0.48	0.31
	L^-	0.11	1.00	0.11	0.09	0.07	0.05	0.00	1.00	0.06	0.00	0.19	0.04	0.57		0.00	0.55	0.06	0.08	0.16	0.04	0.00	0.00	0.00	0.01	0.24	0.04	0.23	
	L^*	0.00	0.11	1.00	0.06	0.05	0.08	0.00	0.00	0.00	0.05	0.04	0.05	0.28		0.00	0.10	0.77	0.03	0.03	0.09	0.00	0.00	0.04	0.00	0.00	0.01	0.23	
grok-3-fast	L^+	1.00	0.11	0.00	0.75	0.19	0.03	1.00	0.00	0.06	0.68	0.18	0.05	0.86	0.53	1.00	0.11	0.00	0.75	0.19	0.03	1.00	0.00	0.06	0.68	0.18	0.05	0.86	0.49
	L^-	1.00	0.11	0.00	0.50	0.12	0.06	0.20	0.29	0.06	0.75	0.26	0.01	0.19		0.83	0.10	0.05	0.50	0.12	0.06	0.20	0.29	0.06	0.15	0.07	0.03	0.14	
	L^*	0.00	0.11	1.00	0.04	0.09	0.40	0.04	0.00	0.41	0.05	0.03	0.40	0.55		0.05	0.05	0.64	0.05	0.10	0.42	0.00	0.04	0.41	0.05	0.03	0.42	0.47	
grok-3-mini-fast	L^+	1.00	0.11	0.00	0.75	0.19	0.03	1.00	0.00	0.06	0.68	0.18	0.05	0.86	0.70	0.47	0.09	0.04	0.75	0.19	0.03	1.00	0.00	0.06	0.68	0.18	0.05	0.72	0.71
	L^-	0.11	1.00	0.11	0.75	0.19	0.05	0.00	1.00	0.06	0.15	0.51	0.01	0.67		0.06	0.42	0.10	0.19	0.75	0.05	0.00	1.00	0.06	0.15	0.51	0.01	0.67	
	L^*	0.04	0.14	0.67	0.08	0.08	0.45	0.06	0.06	1.00	0.11	0.07	0.20	0.58		0.00	0.11	1.00	0.03	0.03	0.63	0.06	0.06	1.00	0.06	0.04	0.31	0.74	
kimi-k2	L^+	1.00	0.11	0.00	0.62	0.15	0.02	0.90	0.00	0.06	0.65	0.18	0.05	0.79	0.52	1.00	0.11	0.00	0.36	0.08	0.02	1.00	0.00	0.06	0.13	0.05	0.04	0.62	0.46
	L^-	1.00	0.11	0.00	0.62	0.15	0.02	0.00	0.00	0.00	0.48	0.15	0.04	0.10		0.03	0.03	0.03	0.20	0.07	0.02	0.20	0.29	0.06	0.33	0.17	0.01	0.14	
	L^*	0.00	0.11	1.00	0.10	0.10	0.36	0.06	0.06	1.00	0.06	0.04	0.28	0.66		0.00	0.11	1.00	0.11	0.07	0.43	0.06	0.06	1.00	0.04	0.03	0.09	0.63	
o3	L^+	1.00	0.11	0.00	0.75	0.19	0.03	1.00	0.00	0.06	0.68	0.18	0.05	0.86	0.65	1.00	0.11	0.00	0.83	0.18	0.03	1.00	0.00	0.06	0.68	0.18	0.05	0.88	0.82
	L^-	1.00	0.11	0.00	0.91	0.22	0.05	0.00	1.00	0.06	0.52	0.13	0.02	0.37		0.10	0.83	0.10	0.22	0.91	0.05	0.00	1.00	0.06	0.10	0.41	0.01	0.79	
	L^*	0.00	0.11	1.00	0.04	0.06	0.37	0.06	0.06	1.00	0.03	0.01	0.49	0.71		0.00	0.11	1.00	0.03	0.05	0.59	0.06	0.06	1.00	0.02	0.01	0.52	0.78	
o4-mini	L^+	1.00	0.11	0.00	0.75	0.19	0.03	0.70	0.00	0.06	0.68	0.18	0.05	0.78	0.69	1.00	0.11	0.00	0.75	0.19	0.03	1.00	0.00	0.06	0.68	0.18	0.05	0.86	0.78
	L^-	0.11	1.00	0.11	0.50	0.12	0.06	0.00	1.00	0.06	0.07	0.38	0.01	0.63		0.10	0.83	0.10	0.15	0.62	0.05	0.00	1.00	0.06	0.15	0.51	0.01	0.74	
	L^*	0.00	0.11	1.00	0.00	0.05	0.42	0.06	0.06	1.00	0.04	0.03	0.28	0.68		0.00	0.11	1.00	0.03	0.03	0.71	0.06	0.06	1.00	0.08	0.06	0.31	0.75	
qwen3-235b-a22b	L^+	1.00	0.11	0.00	0.75	0.19	0.03	1.00	0.00	0.06	0.65	0.18	0.05	0.85	0.50	1.00	0.11	0.00	0.71	0.18	0.03	1.00	0.00	0.06	0.68	0.18	0.05	0.85	0.72
	L^-	0.04	0.16	0.04	0.58	0.16	0.02	1.00	0.00	0.06	0.23	0.26	0.01	0.15		0.10	0.83	0.10	0.19	0.60	0.09	0.00	1.00	0.06	0.15	0.51	0.01	0.73	
	L^*	0.00	0.07	0.45	0.09	0.11	0.14	0.06	0.06	1.00	0.06	0.02	0.40	0.50		0.00	0.07	0.45	0.04	0.07	0.46	0.06	0.06	1.00	0.03	0.02	0.36	0.57	

for matching pairs: (M^+ vs. D^+/L^+), (M^- vs. D^-/L^-), and (M^* vs. D^*/L^*). The second subcolumn in $\overline{diag}$ reports the mean of these three scores, offering a single, aggregated measure of adherence quality for each LLM. In Table 5, we additionally report the score differences between the best performing LLM and each LLM's average diagonal score.

Our findings strongly support our central hypothesis: LLMs exhibit a significant tendency for knowledge-driven hallucination when faced with atypical process structures. This is evident from two key observations. First, we identified numerous instances where models generated from reversed or shuffled artifacts were substantially more similar to the standard ground truth model (M^+) than to their own source evidence. As the red cells in the tables show, this phenomenon occurred across all tested LLMs, with no model achieving full adherence to atypical evidence. Second, even in cases where the LLM correctly

Table 5: Gap to best average diagonal score ($\overline{diag}$) for each configuration. Values show the difference between the best performing LLM and each LLM's average diagonal score.

LLM	Textual Description		Log Abstraction	
	Standard Prompt	Strict Prompt	Standard Prompt	Strict Prompt
command-r	0.37	0.46	0.44	0.52
gemini-2.5-flash	0.11	0.18	0.13	0.12
gemini-2.5-pro	0.14	0.16	0.11	0.22
gpt-4.1-nano	0.32	0.41	0.28	0.50
grok-3-fast	0.05	0.08	0.17	0.32
grok-3-mini-fast	0.14	0.05	0.00	0.11
kimi-k2	0.16	0.19	0.19	0.35
o3	0.13	0.02	0.06	0.00
o4-mini	0.00	0.00	0.01	0.03
qwen3-235b-a22b	0.10	0.12	0.21	0.10

followed the atypical process structure (green cells for D^-, D^*, L^-, L^*), the quality of the generated model, as measured by the similarity score, was generally lower than that achieved for the standard process (D^+, L^+). For example, while gpt-4.1-nano with the strict prompt achieved a perfect score (1.00) for the sales order process from L^+, the discovered models for the atypical artifacts (L^- and L^*) received lower scores (0.55 and 0.77, respectively). This suggests that even when LLMs do not fully hallucinate, their performance is degraded when the input contradicts their internal knowledge, as they struggle to reconcile the evidence with their pre-trained schemas.

4.1 The Influence of Experimental Inputs and Prompts

Effect of Strict Prompting: Our experiment shows that explicitly instructing the LLM to adhere strictly to the provided input can mitigate, but not eliminate, this issue. While all models were susceptible, their responsiveness to the strict prompt varied. For instance, *o3* showed a marked improvement with the strict prompt, correctly modeling several atypical processes it had previously failed on. In contrast, other models like *grok-3-fast* continued to hallucinate frequently even under strict instructions. With standard prompts, we observed 27 clear cases of hallucination from textual descriptions and 20 from event logs. The strict adherence prompt reduced these numbers to 13 and 10, respectively. While this improvement confirms that prompt engineering is a helpful mitigation strategy, its inability to fully resolve the problem underscores how deeply ingrained the model's background knowledge is.

Effect of Artifact Type: The type of input artifact also appears to play a role. We observed fewer hallucinations when models were generated from event logs compared to textual descriptions (a total of 30 instances for logs vs. 40 for text across both prompt types). This is logical, as the structured and unambiguous format of an event log may serve as stronger evidence for the LLM compared to the inherent ambiguity of natural language. However, the persistence of the issue in log-based generation confirms that even structured data is not immune to being overridden by the model's internal schemas.

4.2 Analysis of LLM-Specific Characteristics

Impact of Model Properties: Our analysis reveals that knowledge-driven hallucination is a general weakness across different LLMs, though its severity varies. Interestingly, we found no direct relationship between an LLM's size (parameter count) and its ability to adhere to atypical evidence. For instance, the massive 2.7T-parameter *grok-3-fast* and the compact 18B-parameter *gpt-4.1-nano* showed comparable weaknesses, while the mid-sized *o4-mini* was a top performer. Similarly, while reasoning capabilities are often associated with better performance, they do not guarantee immunity to this type of hallucination. Both *gemini-2.5-pro* and *o4-mini* are considered reasoning models, yet *o4-mini* demonstrated significantly better adherence to atypical evidence. This suggests that neither raw scale nor general reasoning ability alone predicts a model's fidelity to source evidence when it conflicts with pre-trained knowledge.

Correlation Between General Capability and Knowledge Hallucination: A particularly revealing finding is that high performance on standard tasks does not guarantee robustness against knowledge hallucination. A prime example is *gemini-2.5-pro*, which shows strong performance on a wide range of tasks (as measured by the LiveBench benchmark). It was also the top-performing model in our experiment when using the standard prompt and the textual descriptions (D^+), achieving the highest average score (0.80). However, its performance collapsed when faced with conflicting artifacts, dropping to 0.33 for the reversed descriptions (D^-) and just 0.11 for the shuffled ones (D^*). This dramatic drop suggests that the model's well-formed internal schema for the standard process is so dominant that it consistently overrides conflicting source evidence, making it highly prone to knowledge-driven hallucination.

5 Conclusion

This paper introduced and empirically investigated the phenomenon of *knowledge-driven hallucination* in Large Language Models (LLMs), where a model's pre-trained knowledge overrides explicit source evidence, leading to factually incorrect but plausible-looking outputs. Through a controlled experiment in the domain of automated process modeling, we systematically evaluated the fidelity of ten state-of-the-art LLMs when tasked with generating process models from standard and deliberately atypical process evidence.

Our findings demonstrate that LLMs exhibit a tendency to prioritize their generalized internal schemas over contradictory evidence provided in the prompt. This was evident as models frequently reverted to generating a standard process flow even when the input described a reversed or structurally shuffled version. We observed this behavior across all tested LLMs, regardless of their size or specialization, and with both unstructured text and structured event log inputs. Even in cases where the models did not fully hallucinate, their performance in correctly modeling atypical processes was significantly degraded compared to standard ones, highlighting the disruptive effect of the conflict between evidence and internal knowledge.

The implications of our findings extend far beyond process modeling and raise critical concerns about the reliability of LLMs in any evidence-based domain. The danger of knowledge-driven hallucination lies in its deceptive nature; the generated artifacts are often coherent, logical, and well-formed, masking the fact that they do not accurately represent the source data. This "plausibility trap" poses a significant risk in fields such as legal analysis, financial reporting, and scientific research, where strict adherence to source evidence is essential. Our work underscores that simple prompt engineering, such as instructing the model to be faithful to the input, can mitigate but not eliminate this deep-seated behavior.

Future work should focus on two key areas. First, there is a clear need to develop more robust mitigation techniques beyond prompting that allow for better control over the influence of pre-trained knowledge. Second, this experimental methodology could be adapted to investigate knowledge-driven hallucination in other structured generation tasks, such as code generation from legacy specifications or data schema creation from business requirements. Ultimately, our study serves as a critical reminder that as we delegate more complex analytical tasks to AI, we must also develop rigorous methods to validate its outputs and ensure that its powerful inferential capabilities do not come at the cost of factual integrity.

Acknowledgment The project on which this work is based upon was funded by the German Federal Ministry of Research, Technology and Space Travel (grant 01IS23065). The responsibility for the content of this publication lies with the authors.

References

1. de A. R. Gonçalves, J.C., Santoro, F.M., Baião, F.A.: Let me tell you a story - on how to build process models. J. Univers. Comput. Sci. **17**(2), 276–295 (2011)
2. van der Aa, H., Carmona, J., Leopold, H., Mendling, J., Padró, L.: Challenges and opportunities of applying natural language processing in business process management. In: COLING 2018. pp. 2791–2801 (2018)
3. Alberto Blanco-Justicia et al.: Digital forgetting in large language models: a survey of unlearning methods. Artif. Intell. Rev. **58**(3), 90 (2025)
4. Berti, A., van Zelst, S.J., Schuster, D.: PM4Py: A process mining library for python. Softw. Impacts **17**, 100556 (2023)
5. Brown, T.B., Mann, B., Ryder, N., et al., M.S.: Language models are few-shot learners. In: NeurIPS 2020 (2020)
6. Busch, K., Leopold, H.: Towards a benchmark for large language models for business process management tasks. CoRR **abs/2410.03255** (2024)
7. Chen Ling et al.: Beyond one-model-fits-all: A survey of domain specialization for large language models. CoRR **abs/2305.18703** (2023)
8. Chen Qian et al.: An approach for process model extraction by multi-grained text classification. In: CAiSE 2020. LNCS, vol. 12127, pp. 268–282. Springer (2020)
9. Chunting Zhou et al.: LIMA: less is more for alignment. In: NeurIPS 2023 (2023)
10. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management, Second Edition. Springer (2018)
11. Forster, S., Pinggera, J., Weber, B.: Toward an understanding of the collaborative process of process modeling. In: CAiSE'13 Forum. pp. 98–105 (2013)

12. Friedrich, F., Mendling, J., Puhlmann, F.: Process model generation from natural language text. In: CAiSE 2011. pp. 482–496 (2011)
13. Grohs, M., Abb, L., Elsayed, N., Rehse, J.: Large language models can accomplish business process management tasks. In: BPM 2023 Workshops. LNBIP, vol. 492, pp. 453–465. Springer (2023)
14. Hosking, T., Blunsom, P., Bartolo, M.: Human feedback is not gold standard. In: ICLR 2024. OpenReview.net (2024)
15. Klievtsova, N., Benzin, J., Kampik, T., Mangler, J., Rinderle-Ma, S.: Conversational process modelling: State of the art, applications, and implications in practice. In: BPM 2023 Forum. pp. 319–336 (2023)
16. Kourani, H., Berti, A., Schuster, D., van der Aalst, W.M.P.: Evaluating large language models on business process modeling: Framework, benchmark, and self-improvement analysis. CoRR **abs/2412.00023** (2024)
17. Kourani, H., Berti, A., Schuster, D., van der Aalst, W.M.P.: Process modeling with large language models. In: Enterprise, Business-Process and Information Systems Modeling - BPMDS 2024 and EMMSAD 2024, Limassol, Cyprus, June 3-4, 2024, Proceedings. pp. 229–244 (2024)
18. Kourani, H., Berti, A., Schuster, D., van der Aalst, W.M.P.: ProMoAI: Process modeling with generative AI. In: IJCAI 2024, Jeju, South Korea, August 3-9, 2024. pp. 8708–8712 (2024)
19. Kourani, H., van Zelst, S.J.: POWL: partially ordered workflow language. In: BPM 2023. pp. 92–108 (2023)
20. Lei Huang et al.: A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. ACM Trans. Inf. Syst. **43**(2), 42:1–42:55 (2025)
21. Leo Gao et al.: The pile: An 800gb dataset of diverse text for language modeling. CoRR **abs/2101.00027** (2021)
22. Longze Chen et al.: Long context is not long at all: A prospector of long-dependency data for large language models. In: ACL 2024. pp. 8222–8234. Association for Computational Linguistics (2024)
23. Lukas Berglund et al.: The reversal curse: Llms trained on "a is b" fail to learn "b is a". In: ICLR 2024. OpenReview.net (2024)
24. Mrinank Sharma et al.: Towards understanding sycophancy in language models. In: ICLR 2024. OpenReview.net (2024)
25. Roberts, M., Anderson, J., Delgado, W., Johnson, R., Spencer, L.: Extending contextual length and world knowledge generalization in large language models (2024)
26. Sholiq, S., Sarno, R., Astuti, E.S.: Generating BPMN diagram from textual requirements. J. King Saud Univ. Comput. Inf. Sci. **34**(10 Part B), 10079–10093 (2022)
27. Sintoris, K., Vergidis, K.: Extracting business process models using natural language processing (NLP) techniques. In: CBI 2017. pp. 135–139 (2017)
28. Woensel, W.V., Motie, S.: NLP4PBM: a systematic review on process extraction using natural language processing with rule-based, machine and deep learning methods. Enterp. Inf. Syst. **18**(11) (2024)
29. Yufei Wang et al.: Aligning large language models with human: A survey. CoRR **abs/2307.12966** (2023)
30. Ziwei Ji et al.: Survey of hallucination in natural language generation. ACM Comput. Surv. **55**(12), 248:1–248:38 (2023)