Diagnosing LLM Hallucinations in Process Mining Tasks: a Taxonomy and a Benchmark

Alessandro Berti[1][0000−0002−3279−4795] and Humam Kourani[2,1][0000−0003−2375−2152]

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

Abstract. Large language models (LLMs) are increasingly used to support process mining tasks, yet their answers may contain *hallucinations*, content that is unfaithful to the prompt, unsupported by evidence, or invalid with respect to PM formalisms. The PM-LLM-Benchmark evaluates model answers with an expert LLM-as-a-judge that assigns a numeric score and a short explanation. This paper introduces a complementary *hallucination audit* that reads the judge's explanation and converts it into structured annotations about the answering model. We map issues mentioned by the judge onto a PM-tailored taxonomy (four families, twelve sub-types) with severities, aggregate them per model and task family, and analyze their relationships with benchmark performance and model characteristics. Across models, higher PM-LLM-Benchmark scores align with fewer hallucinations; reasoning-oriented and newer models show more favorable profiles; and hallucination families tend to co-occur rather than trade off. We discuss safeguards (reinforce context fidelity, disciplined reasoning, and validate structured outputs), that directly target the dominant failure modes.

Keywords: Process Mining · Large Language Models · Hallucination Detection · Benchmarking

1 Introduction

Process mining significantly benefits from Large Language Models (LLMs) capable of interpreting, explaining, and synthesizing domain-specific artifacts such as event logs, process models, and diagrams [6]. To systematically assess these capabilities, the PM-LLM-Benchmark provides a structured evaluation framework where candidate models respond to process mining prompts [5]. Each response is then evaluated by an expert judge, who assigns a numerical score along with a brief explanatory rationale. While these scores effectively summarize the overall performance of the models, practitioners additionally require detailed insights into the specific ways models fail [7]. Understanding these failure modes is crucial for implementing effective safeguards and making informed decisions about model selection and deployment.

To bridge this gap, we introduce a structured *hallucination audit* grounded in the judges' feedback narratives. Assuming the accuracy and faithfulness of the judge's qualitative assessments, we systematically analyze these narratives, categorizing identified issues according to a clearly defined hallucination taxonomy tailored explicitly for process mining applications. This taxonomy includes four primary categories (*input misalignment*, *factual errors*, *logical errors*, and *technical errors*) further subdivided into twelve specific sub-types. By classifying hallucination occurrences and assigning severity ratings, we transform the judges' qualitative comments into structured diagnostics.

The contribution of this paper is a structured auditing framework that systematically translates PM-LLM-Benchmark rationales into detailed hallucination annotations for model responses, publicly available at `https://github.com/fit-alessandro-berti/pm-llm-benchmark/tree/main/hallucinations`. We propose a comprehensive process-mining-specific hallucination taxonomy, complete with illustrative examples. Furthermore, we provide an aggregated analysis linking hallucination patterns to benchmark scores and model characteristics, uncovering robust trends that can guide model selection and deployment strategies. Finally, we offer recommendations for practitioners in the form of practical guidelines, alongside openly accessible resources to ensure reproducibility and facilitate community-driven enhancements.

The paper is organized as follows. Section 2 introduces the base PM-LLM-Benchmark. Section 3 details the auditing benchmark and hallucination taxonomy. Sections 4 and 5 present and interpret the analysis results and their implications. We discuss related work in Section 6 and conclude the paper in Section 7.

2 Preliminaries: The PM-LLM-Benchmark

The PM-LLM-Benchmark [5] `https://github.com/fit-alessandro-berti/pm-llm-benchmark` assesses the capability of LLMs in addressing various process mining tasks. Each evaluated task comprises a prompt to which the candidate LLM provides a response. Subsequently, an expert judge evaluates the answer, assigning a numerical score between 1.0 and 10.0 along with a concise explanatory justification. These individual scores are aggregated across all prompts, generating an overall performance score for each model, where a higher score indicates superior performance. The benchmark includes eight distinct categories of process mining tasks: (**C1**) context understanding, (**C2**) conformance and anomaly detection, (**C3**) model generation and modification, (**C4**) process querying, (**C5**) hypothesis and question generation, (**C6**) fairness analysis, (**C7**) visual diagram interpretation, and (**C8**) optimization.

Table 1 presents the top-performing LLMs in the date `2025-08-15` along with their average scores across the eight task categories of the PM-LLM-Benchmark. Additionally, the leaderboard clearly indicates whether each listed model is open source and whether it explicitly emphasizes reasoning capabilities. This detailed breakdown enables informed comparisons of model performance across different

Table 1: Top models on the PM-LLM-Benchmark leaderboard, available at `https://github.com/fit-alessandro-berti/pm-llm-benchmark`, with category breakdown (higher is better).

	Model	Score	OS	Reasoning	C1 PCo	C2 CC	C3 PMo	C4 PQ	C5 HG	C6 FA	C7 VI	C8 OPT
1	Qwen3-235B-A22B-Thinking-2507	45.4	Yes	Yes	6.9	8.0	6.9	6.3	6.2	6.4	0.0	4.8
2	gpt-5-2025-08-07-HIGH	43.3	No	Yes	6.8	8.3	4.6	6.0	6.5	6.2	4.9	4.8
3	gpt-5-2025-08-07	43.0	No	Yes	7.1	8.4	5.2	5.8	6.0	6.0	4.0	4.6
4	gemini-2.5-pro-thinkhigh	42.4	No	Yes	6.4	8.3	5.2	5.6	6.2	6.0	4.7	4.6
5	gpt-5-mini-2025-08-07	42.3	No	Yes	6.3	8.6	4.9	5.9	6.1	5.8	3.8	4.8
6	gemini-2.5-pro-thinklow	42.2	No	Yes	5.7	8.4	5.3	5.5	6.1	6.5	5.4	4.7
7	gemini-2.5-flash-thinkhigh	41.0	No	Yes	6.1	8.3	4.8	5.4	6.2	5.5	5.1	4.7
8	o3-pro-2025-06-10-search	40.1	No	Yes	6.9	8.4	4.2	4.5	6.3	5.3	5.0	4.7
9	grok-4-0709	40.1	No	Yes	6.0	8.7	5.4	4.3	5.6	5.4	3.1	4.6
10	o3-pro-2025-06-10	39.9	No	Yes	6.2	7.7	5.3	5.2	5.3	5.9	5.0	4.3
11	Qwen3-30B-A3B-2507-Thinking	39.7	Yes	Yes	6.2	7.7	4.8	4.8	5.7	6.0	0.0	4.6

PM tasks and provides insights into the influence of model characteristics such as openness and reasoning orientation on overall effectiveness.

3 Hallucinations Benchmark

To bridge the limitations of numeric scores in revealing actionable insights into model failures, we develop a hallucination auditing benchmark that parses the judge's qualitative rationales into a taxonomy of error types, severities, and task associations.

3.1 Scope and Availability

This study enhances the *PM-LLM-Benchmark* by translating the qualitative feedback provided by the judge LLM into structured, process-mining-specific hallucination annotations. The key contribution is the introduction of an auditing layer utilizing a lightweight classifier LLM (*gpt-4.1-mini*) to systematically interpret the judge's narrative, mapping identified issues to a defined process-mining-oriented hallucination taxonomy. An overview of the end-to-end evaluation pipeline is shown in Figure 1.

To facilitate transparency, reproducibility, and practical usage, all artifacts associated with the hallucinations benchmark for process mining have been made openly accessible. The main repository of the hallucinations benchmark, including comprehensive documentation, data, and analysis resources, can be found at `https://github.com/fit-alessandro-berti/pm-llm-benchmark/tree/main/hallucinations`.

Within this repository, several core files are particularly important to the analyses presented in this paper:

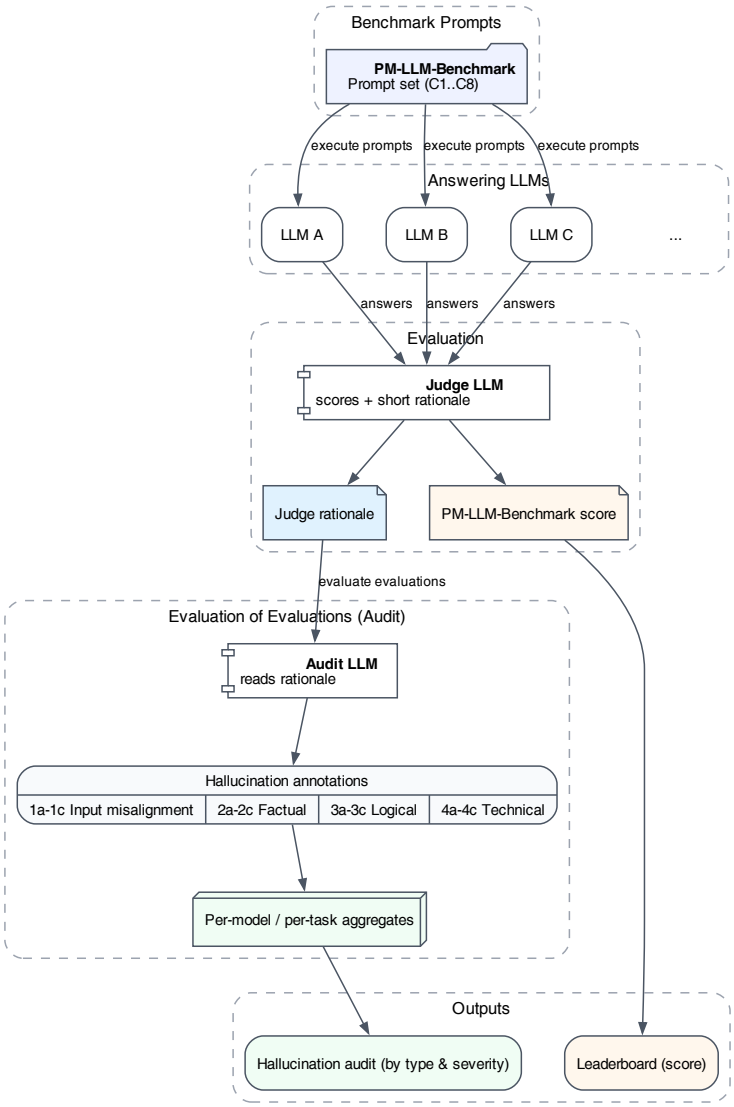

Fig. 1: Execution flow of the PM-LLM-Benchmark and the evaluation-of-evaluations audit. For each answering LLM, the shared PM prompt set (C1–C8) is run; an expert Judge LLM assigns a numeric score with a short rationale; a lightweight Audit LLM then reads the judge's rationale to produce structured hallucination annotations (1a–4c) that are aggregated per model and task, yielding a hallucination audit alongside the standard leaderboard score.

- `hallucination_report.md`: per-model hallucination breakdown by taxonomy (counts and brief notes).
- `pm_llm_category_table.md`: cross-tab mapping sub-types to PM task families (C1–C8).
- `CORRELATION.md`: summary of correlations between hallucination rates, benchmark scores, and model attributes.

3.2 Hallucination Taxonomy

We employ a comprehensive hallucination taxonomy adapted from broader hallucination categories established in existing literature. The taxonomy is structured into four main categories, each containing specific subcategories explicitly designed for process mining tasks. The taxonomy is exemplified in Listing 1. Below we describe each category and its respective subcategories:

Category 1: Input Misalignment. This category addresses issues where the generated response does not correctly adhere to the provided prompt instructions and context. It comprises:

- *1a. Instruction Override:* Occurs when the model ignores explicit constraints specified by the prompt. Example: Prompt states, "List only activity names, no explanation", but the answer returns a narrative with recommendations and metrics.
- *1b. Context Omission:* Refers to situations where the model silently omits necessary contextual information from the prompt, leading to incomplete or irrelevant responses. Example: The prompt provides a rework loop between *Review* and *Fix*; the answer ignores it and discusses unrelated steps.
- *1c. Prompt Contradiction:* Happens when the model explicitly provides responses opposite to the requirements stated in the prompt. Example: Prompt instructs, "Assume conformance and explain why", but the answer claims compliance violations.

Category 2: Factual Errors. This category identifies inaccuracies related to factual correctness and grounding within the responses. It includes:

- *2a. Concept Fabrication:* Involves the invention or fabrication of process mining concepts, methods, or KPI definitions that are not established within the PM literature. Example: The answer cites a non-existent algorithm like "Quantum BPM Miner" or invents a KPI definition not used in PM literature.
- *2b. Spurious Numeric:* Refers to numeric claims or quantitative assertions presented without support from the prompt or provided data. Example: The answer states, "throughput improved by 20%", although no numbers were provided or derivable from the prompt.
- *2c. False Citation:* Occurs when the model incorrectly attributes statements to non-existent or incorrect sources. Example: Attributes a claim to a paper or standard that does not exist or misquotes a real source.

Category 3: Logical Errors. This category covers errors related to flawed or inconsistent reasoning within generated responses. It consists of:

- *3c. Unsupported Leaps:* Pertains to conclusions made without sufficient logical or factual backing from preceding statements. Example: Infers that "de-

Listing 1: Judge evaluation annotated with hallucination tags (about the *answering model*). Tags follow the PM-specific taxonomy: 1a–1c (Input Misalignment), 2a–2c (Factual Errors), 3a–3c (Logical Errors), 4a–4c (Technical Errors).

```
Prompt (excerpt): "Assume the process conforms to the provided model. List ONLY the
     activity names in order. Do NOT explain or recommend."
Context (excerpt): Event log summary states 18% rework from "Review" back to "Fix"; no
     resource data provided. The BPMN shows Start -> Register -> Review -> Fix (optional
     rework) -> Approve -> End; no decision gateways.

--- Judge Evaluation (score + narrative) ---

Score: 4.3 / 10. The answer does not meet the prompt constraints and includes several
     issues.
(1) Format and scope. The prompt requires listing activity names only; the answer adds
     analysis and recommendations.
    [1a Instruction override]
(2) Use of provided context. The BPMN and context include an optional "Fix" step and 18%
     rework; the answer omits "Fix" and claims there is no rework.
    [1b Context omission]
(3) Consistency with prompt intent. Despite "assume conformance," the answer asserts non-
     conformance without basis.
    [1c Prompt contradiction][3a Unsupported leap]
(4) Quantitative claims. The answer states throughput improved by "20%" and reports "92%
     precision / 95% recall" without sources or derivation.
    [2b Spurious numeric]
(5) References. The answer cites a non-existent 2021 journal article ("X Algorithm") as
     evidence.
    [2c False citation]
(6) Internal coherence. The answer first claims "no bottlenecks" then calls "Approval" the
     main bottleneck; the justification is tautological.
    [3b Self-contradiction][3c Circular reasoning]
(7) Causal attribution. The answer attributes delays to a "back-office resource shortage"
     although no resource data are provided.
    [3a Unsupported leap]
(8) Domain concepts. The answer introduces a fictitious metric ("case convergence ratio")
     and tool ("Quantum BPM Miner 3000").
    [2a Concept fabrication]
(9) Structured output. The requested JSON is malformed (missing bracket/quote), so it
     cannot be parsed.
    [4a Syntax error]
(10) Modeling and visuals. The proposed BPMN "two Start Events merging at an AND before
     Register" violates notation semantics, and the described decision gateway after "
     Review" does not appear in the provided diagram.
    [4b Modeling-semantics breach][4c Visual-description mismatch]

Overall rationale: the score reflects multiple high-impact issues across alignment, logic,
     and structure. A corrected answer should (only) list activities in order, cite the
     rework loop, avoid unsupported numerics/claims, and provide valid JSON if requested.
```

Legend. *1a* Instruction override; *1b* Context omission; *1c* Prompt contradiction; *2a* Concept fabrication; *2b* Spurious numeric; *2c* False citation; *3a* Unsupported leap; *3b* Self-contradiction; *3c* Circular reasoning; *4a* Syntax error; *4b* Modeling-semantics breach; *4c* Visual-description mismatch. Tags refer to hallucinations *in the answering model*, extracted from the judge's rationale.

lays are caused by resource shortage" without any resource information in the provided data.

- *3b. Self-Contradiction:* Describes instances where the model generates conflicting statements within the same response, undermining internal consistency. Example: Initially claims, "no bottlenecks are present", but later asserts, "*Approval* is the main bottleneck".
- *3c. Circular Reasoning:* Refers to cases where the model uses a claim as its own justification, resulting in logically circular arguments. Example: "Activity X is a bottleneck because it slows the process, which makes it a bottleneck".

Category 4: Technical Errors. This category addresses issues arising from generating structured outputs, adherence to modeling conventions, or accuracy in interpreting visual materials. Subcategories include:

- *4a. Syntax Error:* Occurs when the model produces invalid structured outputs such as JSON, code, or other structured formats that cannot be parsed or executed. Example: Emits malformed JSON, code, or query that cannot be parsed or executed as requested.
- *4b. Model Semantics Breach:* Involves generating outputs that violate established modeling notation rules or semantic conventions in process mining. Example: Describes a BPMN fragment violating semantics, like two start events merging directly at a gateway.
- *4c. Visual Description Mismatch:* Refers to descriptions of diagrams or images that reference components or aspects not present in the provided visuals. Example: Describes a gateway and extra path not shown in the provided process diagram.

3.3 Evaluation Metrics

We introduce an approach to measuring hallucinations based explicitly on the judge's textual feedback. The following aspects are evaluated:

- *Incidence by Type:* For each evaluated answer, we record the presence or absence of specific hallucination sub-types, as identified by the judge. These sub-types fall within a comprehensive taxonomy of 12 categories grouped into four broader families.
- *Severity Levels:* Each identified hallucination sub-type is assigned a severity rating (*low*, *medium*, *high*, or *critical*), indicating its potential impact on practical PM tasks.
- *Aggregated Summaries:* We generate aggregated summaries at both per-prompt and per-model levels. These summaries include total counts, frequency distributions, and severity distributions, allowing comparative analysis across the eight task categories of the PM-LLM-Benchmark: context understanding, conformance/anomalies, model generation/modification, querying, hypothesis generation, fairness, diagram interpretation, and optimization.

These metrics, derived solely from the judge's explanations, provide interpretable signals about the reliability of each answering LLM.

3.4 Measurement Procedure

The evaluation process involves several sequential steps:

1. *Ingestion:* We begin by collecting the narrative feedback from the judge.
2. *Normalization:* The judge's narrative feedback is passed to the classifier LLM, instructed explicitly to produce structured annotations based on the judge's descriptions.
3. *Taxonomy Tagging:* The classifier identifies and tags each hallucination subtype present, assigns corresponding severity levels, and extracts brief textual examples (trigger spans) from the judge's narrative as supporting evidence.
4. *Consolidation and Aggregation:* Identified tags are de-duplicated for each response, and aggregated across models and PM task categories to generate summarized metrics.

The structured annotations derived from the judge's feedback facilitate an understanding of each LLM's error patterns and associated risks. Specifically, the annotations clarify *where* (error types), *how often* (frequency), and *how severely* (impact) each model fails.

4 Results

In this section, we summarize the key outcomes of the hallucination audit and their relationship to benchmark scores, model characteristics, and process mining task families.

Overall Alignment Between Scores and Hallucinations. Across models, higher PM-LLM-Benchmark scores are strongly associated with fewer hallucinations. The total hallucination incidence exhibits a pronounced negative correlation with the benchmark score ($r = -0.866$), confirming that the numeric evaluation offered by the judge LLM is a reliable summary of answer fidelity. The effect is especially marked for *logical errors* ($r = -0.800$) and *input misalignment* ($r = -0.767$), while *factual* and *technical* errors also decrease, albeit more moderately ($r = -0.543$ and $r = -0.533$, respectively), as shown in Table 2.

Where the Reductions Come From. Within families, the largest improvements for better-scoring LLMs concentrate on high-impact subtypes. *Unsupported leaps* in reasoning drive the logical family and show the strongest single correlation with score ($r = -0.811$). On the alignment side, *context omission* is the most prevalent subtype and declines with score ($r = -0.666$), followed by *instruction override* ($r = -0.629$). Among factual errors, *spurious numeric claims* decrease with performance ($r = -0.522$), whereas fabricated citations are comparatively rare and weakly linked to score. Technical reliability improves mainly through fewer *model-semantics breaches* ($r = -0.546$) and some reduction of *syntax errors*. By contrast, *circular reasoning* is infrequent and shows no meaningful relationship with score. These patterns, summarized in Table 2, indicate that gains in disciplined inference and faithful use of context account for most of the aggregate reduction.

Table 2: Pearson correlations between model features and hallucination categories/types. *Coloring:* blue = negative correlation (feature increases as hallucinations decrease), red = positive; darker shade marks $|r| \geq 0.70$, lighter shade marks $0.50 \leq |r| < 0.70$. Bold = strongest absolute correlation in each row.

Category / Type	PM-LLM-Benchmark Score	Is Open Source	Is Reasoning Model	Model Size	Days Since 2024-01-01
Overall (Total Hallucinations)	−0.866	0.225	−0.601	−0.174	−0.334
Category 1: Input Misalignment	−0.767	0.196	−0.559	−0.168	−0.582
Category 2: Factual Errors	−0.543	0.232	−0.160	−0.134	0.030
Category 3: Logical Errors	−0.800	0.170	−0.699	−0.129	−0.322
Category 4: Technical Errors	−0.533	0.124	−0.225	−0.129	−0.016
1a Instruction Override	−0.629	0.238	−0.277	−0.131	−0.269
1b Context Omission	−0.666	0.123	−0.547	−0.142	−0.644
1c Prompt Contradiction	−0.591	0.311	−0.343	−0.159	0.053
2a Concept Fabrication	−0.410	0.190	−0.161	−0.047	0.109
2b Spurious Numeric	−0.522	0.257	−0.157	−0.150	−0.008
2c False Citation	−0.141	−0.046	0.030	−0.075	−0.047
3a Unsupported Leap	−0.811	0.173	−0.701	−0.125	−0.376
3b Self-Contradiction	−0.430	0.078	−0.415	−0.111	0.075
3c Circular Reasoning	0.023	0.045	−0.016	0.037	0.243
4a Syntax Error	−0.421	0.210	0.077	−0.144	0.026
4b Model Semantics Breach	−0.546	0.162	−0.319	−0.195	−0.040
4c Visual Description Mismatch	0.023	−0.245	−0.194	0.220	−0.023

Notes: Negative correlations with the benchmark score indicate that higher PM-LLM-Benchmark performance is associated with fewer detected hallucinations. Only cells with $|r| \geq 0.50$ are tinted.

Impact of Reasoning Orientation and Model Recency. Reasoning-oriented models exhibit systematically lower hallucination rates, particularly for logical errors ($r = -0.699$ with the "reasoning model" indicator) and input misalignment ($r = -0.559$). At the subtype level, the reductions are most visible for *unsupported leaps* and *context omissions*. Model recency also helps: newer models show fewer input-alignment issues overall ($r = -0.582$) and fewer context omissions specifically ($r = -0.644$), with a moderate reduction of logical errors. In contrast, recency yields limited changes for factual and technical subtypes, and circular reasoning does not improve, suggesting a residual challenge that persists across generations (Table 2).

Size and Openness Effects. Model size correlates only weakly with hallucination counts, and the open-source indicator shows mixed associations. Open-

source models tend to have slightly higher rates in alignment and logic on average, while exhibiting somewhat fewer *visual-description mismatches*. These differences are small compared to the advantages conferred by reasoning specialization and recency, reinforcing that architectural and training choices targeting reasoning quality matter more than raw scale alone (Table 2).

Task-Family Hotspots. Hallucination profiles concentrate in specific PM task families. *Unsupported leaps* peak in model generation and modification (C3: 2155 counts), with substantial incidence also in process querying and context understanding; overall, logical errors sum to 10,550 occurrences across tasks. *Context omission* dominates input misalignment (total 4926), with the highest counts in context understanding (C1: 957), optimization (C8: 849), and process querying (C4: 790). Factual errors are led by *spurious numerics*, most prominent in conformance and anomaly detection (C2: 716) and also elevated in modeling (C3: 501) and optimization (C8: 403). Technical errors show two clear hotspots: *model-semantics breaches* concentrate in modeling tasks (C3: 791), while *syntax errors* are most frequent during hypothesis and question generation (C5: 200). Visual interpretation (C7) has comparatively fewer total hallucinations but the highest concentration of *visual-description mismatches* (249). Aggregates per family, reported in Table 3, confirm that C3 contributes the largest share of hallucinations overall, followed by C1 and C4.

Co-occurrence of Failure Modes. Hallucination families tend to co-occur rather than trade off. The strongest off-diagonal association links input misalignment with logical errors ($r = 0.717$), indicating that answers that underuse or contradict context often also contain unjustified inferences. Factual and technical errors are likewise coupled ($r = 0.572$), and logical errors correlate moderately with technical issues ($r = 0.553$). These relationships, summarized in Table 4,

Table 3: Aggregated hallucination counts across all LLMs, organized by PM-LLM-Benchmark categories. *Bold numbers* indicate, for each hallucination type, the PM category with the highest count in that row (ties are bolded). The *Total* row is fully bolded for readability.

Hallucination Type	**C1 PCo**	**C2 CC**	**C3 PMo**	**C4 PQ**	**C5 HG**	**C6 FA**	**C7 VI**	**C8 OPT**	**Total**
1a. Instruction Override	**304**	86	156	147	113	121	1	43	971
1b. Context Omission	**957**	724	286	790	377	592	351	849	4926
1c. Prompt Contradiction	**179**	84	95	91	22	38	13	8	530
2a. Concept Fabrication	95	45	105	205	47	113	55	**220**	885
2b. Spurious Numeric	361	**716**	501	153	191	177	108	403	2610
2c. False Citation	12	21	6	**37**	6	15	0	9	106
3a. Unsupported Leap	1406	1384	**2155**	1711	1227	1098	449	1120	10550
3b. Self Contradiction	**317**	186	309	183	56	128	26	30	1235
3c. Circular Reasoning	6	4	5	0	3	1	0	**14**	33
4a. Syntax Error	39	12	186	77	**200**	63	0	14	591
4b. Model Semantics Breach	151	61	**791**	156	95	165	123	7	1549
4c. Visual Description Mismatch	105	26	39	68	21	20	**249**	20	548
Total	**3932**	**3349**	**4634**	**3618**	**2358**	**2531**	**1375**	**2737**	**24534**

Table 4: Inter-category correlation matrix. Cell shading encodes magnitude (white = 0, blue = 1); diagonal cells (1.000) are shown in gray; bold marks the strongest off-diagonal correlation per row.

Category	Cat. 1 Input Misalign.	Cat. 2 Factual Errors	Cat. 3 Logical Errors	Cat. 4 Technical Errors
Cat. 1 Input Misalignment	1.000	0.236	**0.717**	0.290
Cat. 2 Factual Errors	0.236	1.000	0.489	**0.572**
Cat. 3 Logical Errors	**0.717**	0.489	1.000	0.553
Cat. 4 Technical Errors	0.290	**0.572**	0.553	1.000

Notes: Diagonal entries are 1.000 by definition. Off-diagonal values quantify how often categories co-occur across models.

suggest shared underlying causes: improving context fidelity and enforcing disciplined reasoning are likely to reduce several error families simultaneously.

Practical Reading of the Audit. Taken together, the results indicate that the judge's numeric score is a strong first filter for reliability, while the hallucination audit pinpoints the residual risks that persist even at moderate scores. Emphasis on reasoning-oriented configurations and recent model versions yields the largest risk reductions, especially for the dominant pairs of failure modes—context omissions and unsupported leaps—whereas technical and factual hygiene require complementary safeguards in tasks that demand structured outputs or quantitative claims.

5 Discussion

The analysis presented in this study provides an interpretable and structured risk profile for each evaluated LLM by translating the judge's qualitative feedback into quantitative hallucination metrics. Two key insights emerge clearly: first, models that score higher on the PM-LLM-Benchmark demonstrate significantly fewer hallucinations, especially regarding logical reasoning and alignment with prompt instructions. Second, hallucination categories tend to co-occur, suggesting common underlying mechanisms that produce errors. These findings underscore that improving contextual understanding and reasoning abilities can mitigate multiple categories of hallucinations simultaneously, thus enhancing overall model reliability [32,30,33].

Practical Consequences We identify three main suggestions for practical use.

First, use the PM-LLM-Benchmark score as a coarse filter and the hallucination audit as a fine-grained risk profile for deployment. In practice, teams should (i) shortlist models by overall score to ensure baseline task competence, then (ii) gate deployment on *per-category* and *per-severity* thresholds from the audit (e.g., no high/critical "unsupported leap" or "context omission" events in C2–C4, no syntax or modeling-semantics breaches in C3). Because our results show that newer, reasoning-oriented models reduce input-misalignment and logical errors more than size alone, practitioners should prefer recent, reasoning-focused models when two candidates tie on the headline score. This two-step selection reduces the chance that seemingly "good" answers still carry high-impact failure modes.

Second, map safeguards directly to the dominant failure modes revealed by the audit and to the PM task family at hand. To reinforce *context fidelity*, structure prompts and outputs so claims are explicitly tied to provided artifacts (log/-model/diagram snippets or retrieved evidence) [20,4,34]. To enforce *disciplined reasoning*, require structured justifications and self-checks to prevent unsupported leaps and contradictions [32,30,21,24]. To ensure *structured-output hygiene*, validate JSON/queries with grammars or constrained decoding and check BPMN/Petri-net semantics automatically [15,8,13,26,1,12,2]. Make these controls task-specific: e.g., for C3 (model generation/modification) run notation/semantics validators; for C7 (visual interpretation) require that textual references correspond to nodes/edges present in the diagram; for C2 (conformance/anomalies) forbid causal attributions that are not grounded in the supplied evidence. Targeting the audit's top error types yields broad gains because these categories frequently co-occur.

Third, operationalize continuous monitoring and governance around the same taxonomy used for selection. Periodically re-run the hallucination audit on production samples and model updates; track counts by type and severity, set alert thresholds, and route high-severity cases to human review. Maintain a living regression suite of prompts per process mining category (with expected structured outputs and validator checks) [28], and record drift when providers roll new versions. Because improvements to context fidelity and reasoning tend to reduce multiple error families simultaneously, focus monitoring and remediation on these levers first, then address rarer but impactful issues (e.g., fabricated citations) through targeted checks. This closes the loop from benchmark to deployment, keeps risk visible, and prevents silent degradation over time.

Model Selection: When selecting models, prioritizing reasoning capabilities and recent advancements yields substantial improvements in reducing hallucinations compared to merely selecting larger models. While larger models offer incremental improvements, reasoning-oriented and newly developed models significantly outperform older or general-purpose models, regardless of their open-source or proprietary status [32,30,33,23].

Limitations: Our analysis carries inherent limitations. The annotations rely solely on the judge's narrative completeness and accuracy [22,35,9,16,31]. Consequently, any oversight or omission by the judge directly affects the taxonomy's

comprehensiveness. Additionally, the automated extraction process, while efficient, may blur subtle distinctions between certain error types, such as circular versus unsupported reasoning. Finally, correlation-based findings demonstrate associations rather than causal relationships.

Future Work: Looking ahead, the robustness of hallucination detection can be enhanced through multi-judge evaluations of LLM answers [35,10,22], which would help assess the stability and generalizability of risk profiles. Additionally, integrating structured guardrails directly into PM workflows (such as mandatory evidence referencing, structured reasoning prompts, and automated validation tools) can further elevate the reliability of LLM-generated answers [20,4,21,15,26]. Together, these measures provide a practical framework for managing and mitigating hallucination risks in process mining tasks using large language models.

6 Related Work

In this section, we situate our hallucination audit within prior work on LLM hallucinations and BPM/PM evaluation, highlighting how our taxonomy, cross-task scope, and empirical findings extend and generalize existing approaches.

6.1 Hallucination Taxonomies

General taxonomies and surveys. Recent syntheses organize hallucinations into a small set of recurring families: *input misalignment/faithfulness*, *factual errors*, *logical errors*, and *technical or domain-specific errors*. These surveys argue for sub-types that make failure modes actionable and comparable across tasks, which we adopt and tailor to process mining use cases [17,25].

Faithfulness and input use. Work on faithfulness distinguishes instruction inconsistency (violating explicit task constraints) and context inconsistency (ignoring or contradicting supplied evidence) [17,25]. In PM prompts, these map directly to *instruction override*, *context omission*, and *prompt contradiction*, which we use to flag when answers drift from format requirements or fail to use provided logs, models, or diagrams.

Factual fabrication and references. Factual hallucinations include invented concepts, ungrounded numbers, and fabricated or misattributed citations [17]. Our *concept fabrication*, *spurious numeric*, and *false citation* sub-types reflect these patterns; the last one is motivated by studies showing models can produce authoritative-sounding but non-existent references [3].

Logical consistency and reasoning errors. Surveys also emphasize internal coherence: models may make unsupported inferences, contradict themselves within one answer, or argue in circles [17,18]. We capture these as *unsupported leap*, *self-contradiction*, and *circular reasoning*, which are particularly harmful in analytic PM tasks (e.g., causal claims without evidence in the log or model).

Technical and domain-specific violations. As LLMs emit structured outputs or describe visuals, additional failure modes arise: malformed JSON/code, notation/schema breaches (e.g., BPMN/Petri-net),

and visual misreadings [36,27,14,11]. These motivate *syntax error*, *model-semantics breach*, and *visual-description mismatch*, which we audit explicitly in PM settings.

Process-mining–oriented frameworks. Domain treatments for PM consolidate these ideas into practical taxonomies and pipelines that combine automatic checks (format/schema validators, factual spot checks) with model- or human-in-the-loop review of subtler reasoning flaws [29]. Our work follows this line but extracts the signals from the judge's narratives in PM-LLM-Benchmark [5], producing per-model hallucination profiles aligned with PM task families.

6.2 Existing Hallucination Benchmarks in BPM

Empirical studies of hallucinations in Business Process Management (BPM) are limited. A recent line of work on process *modeling* introduces *knowledge-driven hallucinations*, where an LLM's domain priors override the provided specification, yielding plausible but unsupported elements (e.g., extra activities/gateways/flows) ([19]). These errors primarily reflect *Input Misalignment* (ignoring or contradicting the given model: 1b/1c) and *Logical Errors* (unsupported leaps: 3a), and often co-occur with *Technical Errors* when invented fragments violate notation semantics (4b).

Our audit generalizes this perspective across eight process mining task families (C1–C8). We observe the same prior-over-evidence mechanism beyond modeling: large counts for *context omission* in context understanding and optimization, *spurious numerics* in conformance/anomaly detection, and concentration of *unsupported leaps* and *model-semantics breaches* in model generation/modification (see Tables 3–4). Moreover, reasoning-oriented and newer models show fewer omissions and unsupported leaps (Table 2), suggesting that disciplined inference reduces these prior-driven deviations. Practically, safeguards that enforce context fidelity, structured reasoning, and notation/schema validation operationalize the main lesson from prior BPM work: do not trust domain priors over authoritative artifacts (logs, reference models, diagrams).

7 Conclusion

This paper introduced a structured hallucination audit that converts the judge narratives of the PM-LLM-Benchmark into process-mining–specific annotations. By mapping issues to a four-family taxonomy and aggregating them across models and task families, we obtain an interpretable risk profile that complements the benchmark score.

Our analysis shows that although higher PM-LLM-Benchmark scores generally correlate with fewer hallucinations, non-zero, and often co-occurring, hallucinations remain across input alignment and logical reasoning (Table 2, Table 4). This residual error profile is not merely cosmetic: in high-stakes PM scenarios such as conformance checking and anomaly detection (C2), fairness assessment (C6), and optimization or change recommendations (C8), hallucinations

can translate into regulatory misstatements, unfair decisions, or costly operational actions. Consequently, hallucinations are a primary factor limiting the applicability of current LLMs as *autonomous* decision-makers in these settings. Until incidence and severity are demonstrably low for the specific deployment distribution, LLM outputs should be treated as *untrusted suggestions* that require verification before action.

Implications for high-stakes deployment. The audit suggests a risk-driven deployment pattern in which model autonomy is conditioned on measured hallucination risk:

1. *Classify task criticality and cap autonomy.* Use LLMs in assistive or propose-and-review modes for high-impact tasks; reserve autonomous actions for low-risk utilities (e.g., drafting, retrieval).
2. *Enforce evidence grounding.* Require answers to explicitly reference provided logs, models, or diagrams; suppress unverifiable numbers and claims; prefer prompts that elicit structured citations to the given context.
3. *Validate structure and semantics.* Gate outputs through schema/grammar checks (e.g., JSON, queries) and domain validators (e.g., BPMN/Petri-net semantics), rejecting or auto-correcting malformed content.
4. *Add disciplined reasoning and self-checks.* Use structured justifications, self-consistency/reflection, and cross-checking judges; trigger abstention or escalation when checks disagree or confidence is low.
5. *Mandate human-in-the-loop and auditability.* Require expert review for high-stakes actions, maintain logs, and continuously sample for hallucination audits post-deployment.

Outlook. All artifacts are openly released to enable replication, extension, and continuous monitoring. Future work should (i) incorporate multi-judge setups to improve robustness, (ii) calibrate severities to downstream harm for task- and domain-specific risk limits, and (iii) integrate the audit into PM toolchains as a standing safety layer. In summary, while recent models reduce hallucinations, the remaining failure modes identified by our audit currently bound the safe applicability of LLMs in high-stakes process mining; rigorous verification and risk-aware deployment are therefore prerequisites, not add-ons.

References

1. Business Process Model and Notation (BPMN), Version 2.0.2 (December 2013), oMG Document Number: formal/13-12-09
2. van der Aalst, W.M.: Data science in action. In: Process mining: Data science in action, pp. 3–23. Springer (2016)
3. Agrawal, A., Suzgun, M., Mackey, L., Kalai, A.: Do Language Models Know When They're Hallucinating References? In: EACL (Findings). pp. 912–928. Association for Computational Linguistics (2024)
4. Asai, A., Wu, Z., Wang, Y., Sil, A., Hajishirzi, H.: Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection (2023)
5. Berti, A., Kourani, H., van der Aalst, W.M.P.: PM-LLM-Benchmark: Evaluating Large Language Models on Process Mining Tasks. In: ICPM Workshops. Lecture Notes in Business Information Processing, vol. 533, pp. 610–623. Springer (2024)

6. Berti, A., Kourani, H., Häfke, H., Li, C., Schuster, D.: Evaluating Large Language Models in Process Mining: Capabilities, Benchmarks, and Evaluation Strategies. In: BPMDS/EMMSAD@CAiSE. Lecture Notes in Business Information Processing, vol. 511, pp. 13–21. Springer (2024)
7. Berti, A., Kourani, H., Park, G., van der Aalst, W.M.: Configuring Large Reasoning Models using Process Mining: A Benchmark and a Case Study. Authorea Preprints (2025)
8. Beurer-Kellner, L., Fischer, M., Vechev, M.: Prompting Is Programming: A Query Language for Large Language Models. Proceedings of the ACM on Programming Languages **7**(PLDI), 1946–1969 (Jun 2023)
9. Chen, G.H., Chen, S., Liu, Z., Jiang, F., Wang, B.: Humans or LLMs as the Judge? A Study on Judgement Biases (2024)
10. Chiang, W.L., Zheng, L., Sheng, Y., Angelopoulos, A.N., Li, T., Li, D., Zhang, H., Zhu, B., Jordan, M., Gonzalez, J.E., Stoica, I.: Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference (2024)
11. Cossio, M.: A comprehensive taxonomy of hallucinations in Large Language Models. arXiv preprint arXiv:2508.01781 (2025)
12. Dijkman, R.M., Dumas, M., Ouyang, C.: Semantics and analysis of business process models in BPMN. Information and Software Technology **50**(12), 1281–1294 (2008)
13. Dong, Y., Ruan, C.F., Cai, Y., Lai, R., Xu, Z., Zhao, Y., Chen, T.: XGrammar: Flexible and Efficient Structured Generation Engine for Large Language Models (2025)
14. Dugar, R.: Crafting Structured JSON Responses: Ensuring Consistent Output from any LLM, updated 2024-10-26
15. Geng, S., Josifoski, M., Peyrard, M., West, R.: Grammar-Constrained Decoding for Structured NLP Tasks without Finetuning (2024)
16. Gu, J., Jiang, X., Shi, Z., Tan, H., Zhai, X., Xu, C., Li, W., Shen, Y., Ma, S., Liu, H., Wang, S., Zhang, K., Wang, Y., Gao, W., Ni, L., Guo, J.: A Survey on LLM-as-a-Judge (2025)
17. Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., Liu, T.: 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)
18. Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y., Madotto, A., Fung, P.: Survey of Hallucination in Natural Language Generation. ACM Comput. Surv. **55**(12), 248:1–248:38 (2023)
19. Kourani, H., Antonov, A., Berti, A., van der Aalst, W.M.: Knowledge-Driven Hallucination in Large Language Models: An Empirical Study on Process Modeling (Sep 2025)
20. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., tau Yih, W., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (2021)
21. Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., Cobbe, K.: Let's Verify Step by Step (2023)
22. Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., Zhu, C.: G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment (2023)
23. Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Gupta, S., Majumder, B.P., Hermann, K., Welleck, S., Yazdanbakhsh, A., Clark, P.: Self-Refine: Iterative Refinement with Self-Feedback (2023)

24. Manakul, P., Liusie, A., Gales, M.J.F.: SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models (2023)
25. Maynez, J., Narayan, S., Bohnet, B., McDonald, R.T.: On Faithfulness and Factuality in Abstractive Summarization. In: ACL. pp. 1906–1919. Association for Computational Linguistics (2020)
26. Rajpal, S.: Guardrails AI (2023), guardrails AI Framework
27. Ravuri, C., Amarasinghe, S.: Eliminating Hallucination-Induced Errors in LLM Code Generation with Functional Clustering. CoRR **abs/2506.11021** (2025)
28. Ribeiro, M.T., Wu, T., Guestrin, C., Singh, S.: Beyond Accuracy: Behavioral Testing of NLP models with CheckList (2020)
29. Shankar, S., Zamfirescu-Pereira, J.D., Hartmann, B., Parameswaran, A.G., Arawjo, I.: Who Validates the Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences. In: UIST. pp. 131:1–131:14. ACM (2024)
30. Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., Zhou, D.: Self-Consistency Improves Chain of Thought Reasoning in Language Models (2023)
31. Wataoka, K., Takahashi, T., Ri, R.: Self-Preference Bias in LLM-as-a-Judge (2025)
32. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., Zhou, D.: Chain-of-Thought Prompting Elicits Reasoning in Large Language Models (2023)
33. Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T.L., Cao, Y., Narasimhan, K.: Tree of Thoughts: Deliberate Problem Solving with Large Language Models (2023)
34. Ye, X., Sun, R., Arik, S., Pfister, T.: Effective Large Language Model Adaptation for Improved Grounding and Citation Generation (2024)
35. Zheng, L., Chiang, W.L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E.P., Zhang, H., Gonzalez, J.E., Stoica, I.: Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena (2023)
36. Zhou, Y., Cui, C., Yoon, J., Zhang, L., Deng, Z., Finn, C., Bansal, M., Yao, H.: Analyzing and Mitigating Object Hallucination in Large Vision-Language Models. In: ICLR. OpenReview.net (2024)