PM-LLM-Benchmark: Evaluating Large Language Models on Process Mining Tasks

Alessandro Berti[1,2(✉)], Humam Kourani[1,2], and Wil M. P. van der Aalst[1,2]

[1] Process and Data Science Chair, RWTH Aachen University, Aachen, Germany
{a.berti,wvdaalst}@pads.rwth-aachen.de
[2] Fraunhofer FIT, Sankt Augustin, Germany
humam.kourani@fit.fraunhofer.de

Abstract. Large Language Models (LLMs) have the potential to semi-automate some process mining (PM) analyses. While commercial models are already adequate for many analytics tasks, the competitive level of open-source LLMs in PM tasks is unknown. In this paper, we propose *PM-LLM-Benchmark*, the first comprehensive benchmark for PM focusing on domain knowledge (process-mining-specific and process-specific) and on different implementation strategies. We focus also on the challenges in creating such a benchmark, related to the public availability of the data and on evaluation biases by the LLMs. Overall, we observe that most of the considered LLMs can perform some process mining tasks at a satisfactory level, but tiny models that would run on edge devices are still inadequate. We also conclude that while the proposed benchmark is useful for identifying LLMs that are adequate for process mining tasks, further research is needed to overcome the evaluation biases and perform a more thorough ranking of the "competitive" LLMs.

Keywords: Process Mining · Large Language Models · Evaluation Strategies · LLM Benchmarking

1 Introduction

Process mining (PM) is a branch of data science aiming to derive process-related insights from the event data recorded during the execution of a process. A wide set of automated PM techniques exist for process discovery (the automated discovery of process models starting from the event data), conformance checking (comparing event data and process models), and model enhancement (annotating a process model with metrics derived from the event data). PM could benefit significantly from the provision of domain knowledge [7]. Modern Large Language Models (LLMs) have the capability to follow the instructions contained in a given prompt and are trained on large sets of generic knowledge, including process-related knowledge. The release of OpenAI's GPT-4 has been a milestone,

A. Delgado and T. Slaats (Eds.): ICPM 2024 Workshops, LNBIP 533, pp. 610–623, 2025.
https://doi.org/10.1007/978-3-031-82225-4_45

as such LLM proved capable in different PM tasks [4] including semantic anomaly detection and root cause analysis. However, GPT-4 is a commercial LLM, and open-source LLMs were significantly behind the quality of GPT-4. Recently, many companies released good-performing open-source LLMs, which in general-purpose benchmarks approach GPT-4-level quality[1]. For example, *Llama 3*[2] by Meta, *Mixtral 8x7B* and *Mixtral 8x22B*[3] by Mistral, and *WizardLM2*[4] by Microsoft are all-rounder LLMs. Also alternative commercial models have been proposed from Antrophic (*Claude AI*[5]) and Google (*Gemini*[6]), which also approach GPT-4 levels of quality.

Given the large number of commercial and open-source LLMs, benchmarks are essential to distinguish between good and bad-for-the-purpose LLMs. While many general-purpose benchmarks exist for LLMs, there is a lack of comprehensive benchmarks in process mining (PM). Several factors contribute to the difficulty of proposing such a benchmark: multiple PM artifacts exist (e.g., traditional/object-centric event logs, procedural/declarative process models, situation tables); multiple PM types exist (e.g., process discovery, conformance checking, applications of machine learning such as anomaly detection and root cause analysis, predictive analytics); multiple PM practices exist, with different pathways followed during a process mining analysis [27]; multiple PM code libraries and query languages exist, including Python (e.g., pm4py), SQL, and non-relational languages; and the ability to propose valuable answers depends on the ability of the analyst to propose valuable inquiries/hypotheses [2].

In this paper, we propose three main contributions: i) a first *comprehensive benchmark for process mining tasks executable by LLMs*, focusing on two implementation paradigms (direct provision of insights and code generation), and including several categories of "static" prompts (stored in TXT files) requiring process-mining-specific and process-specific domain knowledge; ii) a *scalable evaluation strategy* to assess the quality of the textual/coding answers provided by LLMs; iii) the *results of the application of the benchmark* on several state-of-the-art LLMs. While the benchmark provides a score useful to rank LLMs, some caveats discussed in Sect. 3 suggest avoiding comparing the scores for highly-performing LLMs. In particular, the role and limitations of LLMs as judges for PM tasks' outputs need to be discussed along with the need for ground truth in scoring the answers. Moreover, advanced implementation paradigms (RAG, agents crew, multi-stage hypothesis generation), discussed in Sect. 5, are not assessed by the benchmark.

The rest of the paper is organized as follows. Section 2 describes the related work connecting process mining and LLMs, and the evaluation of LLMs outputs. Section 3 introduces the categories of prompts included in the benchmark and the

[1] https://chat.lmsys.org/.
[2] https://llama.meta.com/llama3/.
[3] https://mistral.ai/news/mixtral-of-experts/.
[4] https://huggingface.co/WizardLM.
[5] https://claude.ai/.
[6] https://gemini.google.com/.

proposed evaluation strategy. Section 4 discusses the results of the benchmark on state-of-the-art LLMs. Section 5 proposes some novel scenarios requiring novel benchmarks. Finally, Sect. 6 concludes the paper.

2 Related Work

Connecting LLMs to PM: Several Business Process Management tasks have been linked with LLMs [24]. For instance, process modeling exploits LLMs to create process models starting from textual descriptions [11,15,16]. Process mining tasks have been implemented on LLMs thanks to textual description of the mainstream artifacts (event logs, process models) [4]. Three implementation paradigms are identified [3]: i) *direct provision of insights*, ii) *generation of database queries* (SQL), and iii) *autonomous hypotheses generation*. The direct provision of insights requires the provision of the necessary information to the LLM. Generating database queries [13] uses LLMs to create (SQL) statements to be executed against the data source, mitigating privacy risks but not exploiting the domain knowledge of LLMs. The autonomous formulation of hypotheses combines the two methodologies allowing the LLM to create database queries and interpret their output. In [2], the capabilities required for process mining on LLMs are described: *acceptance of long prompts*, allowing for the provision of a significant amount of information to the LLM; *acceptance of visual prompts*, as visualizations allow us to easily identify process-related patterns; *coding capabilities*, including the generation of scripts and SQL statements; and *factuality*, being able to cross-check the outputs against knowledge bases or search engines. Given the absence of process-mining-specific benchmarks, [2] suggests using general-purpose benchmarks (using traditional, domain-knowledge, visual, coding, fairness, and hypotheses generation benchmarks).

Other Benchmarks of PM on LLMs: In [13], the authors assess the LLM-based translation of process mining questions proposed in [1] to SQL statements. They found that even advanced LLMs require the provision of database-specific and process-mining-specific domain knowledge, that requires prompt injection. Moreover, the questions that could be translated to SQL statements are quantitative and do not assess the process-specific knowledge of the LLMs. In [3], an initial comparison of two LLMs (GPT-4 and Google Bard) on different process mining tasks and implementation paradigms is performed. The results lay the foundations for this paper, which proposes a much more comprehensive benchmark. In [9,10], some benchmarks covering causal reasoning and explaining decision points in business processes are proposed. In [21], benchmarks for some semantics-aware process mining tasks, i.e. semantic anomaly detection and the prediction of the next activity, are provided along with strategies to fine-tune the LLMs to improve their ability to execute the tasks. Moreover, [26] proposes a benchmark at the intersection between Robotic Process Automation and Business Process Management, evaluating the ability to exploit workflows recorded in user screenshots for Business Process Management tasks.

LLMs-as-Judges: As the outputs of LLMs are mainly textual, evaluating them in an automatic way is challenging. Some metrics have been proposed to evaluate how well an answer matches a "ground truth" provided by an human analyst[7]. However, their evaluation schema cannot be adapted to open-ended answers. Another option is to let an LLM evaluate the answer (LLM-as-a-Judge) [28]. Using LLMs as judges, the evaluation can be tailored to the desired criteria and consider open-ended answers. Some studies compared the scores given by humans and LLMs to a given set of questions/answers, finding a good alignment between humans and LLMs [23,28]. However, some studies highlighted potential weaknesses due to bias [20] and difficulty in following arbitrary evaluation directives [8]. In [22], it is shown that LLMs (as also humans) suffer from the Dunning-Kruger effect, underestimating/overestimating scores. LLMs-as-Judges can be implemented with or without the provision of a ground truth [6]. If no ground truth is provided, it is necessary that the judge LLM would be able to i) respond correctly to the given inquiry; ii) identify errors and opportunities for improvement in the provided answer. In ranking different LLMs without ground truth, there are other potential biases to consider. In [25], it is highlighted how LLMs tend to prefer answers of similar LLMs. The "egocentric bias" (LLMs preferring their own answers) is highlighted in [14].

3 Benchmark

In this section, we describe *PM-LLM-Benchmark*, which is available at the address https://github.com/fit-alessandro-berti/pm-llm-benchmark. First, in Sect. 3.1, we describe the categories of prompts included in the benchmarks. Then, in Sect. 3.2, we describe the evaluation strategy (LLM-as-a-Judge).

3.1 Categories of Tasks

Our benchmark measures how much an LLM is *knowledgeable* and *capable* in process mining. The capability is measured in the correct interpretation and production of different process mining artifacts (traditional and object-centric event logs; procedural and declarative process models). We also evaluate the ability of the LLM to autonomously formulate hypotheses over the event data or the process model. Moreover, as the goal of process mining is to assist data-driven decision-making, we aim to assess how much the LLM is able to identify biases starting from the event data. We also want to assess the ability of Large Vision Language Models (LVLMs; so LLMs supporting visual prompts) to interpret popular visualizations and process mining diagrams.

The benchmark focuses on two implementation paradigms, i.e., the *direct provision of insights* and *code generation*. Moreover, specific focus is given on process-mining-specific and process-specific *domain knowledge*, which is required for the considered prompts. Other available lists of process mining inquiries, such

[7] https://mlflow.org/docs/latest/llms/llm-evaluate/index.html.

Table 1. Prompts included in the benchmark.

	Prompt	Open	Requires DK	Task	Input Abstraction	Input Dataset
C1	cat01_01_variants_bpic2020_rca	X	X	RCA	Variants	BPI2020 Domestic
C1	cat01_02_variants_roadtraffic_anomalies	X	X	Semantic AD	Variants	Road Traffic
C1	cat01_03_bpic2020_var_descr	X	X	Description	Variants	BPI2020 Domestic
C1	cat01_04_roadtraffic_var_descr	X	X	Description	Variants	Road Traffic
C1	cat01_05_bpic2020_dfg_descr	X	X	Description	DFG	BPI2020 Domestic
C1	cat01_06_roadtraffic_dfg_descr	X	X	Description	DFG	Road Traffic
C1	cat01_07_ocel_container_description	X	X	Description	OC-DFG	Logistics
C1	cat01_08_ocel_order_description	X	X	Description	OC-DFG	Order Management
C1	cat01_09_ocel_container_rca	X	X	RCA	OC-DFG	Logistics
C1	cat01_10_ocel_order_rca	X	X	RCA	OC-DFG	Order Management
C2	cat02_01_open_event_abstraction	X	X	Domain Knowledge		
C2	cat02_02_open_process_cubes	X	X	Domain Knowledge		
C2	cat02_03_open_decomposition_strategies	X	X	Domain Knowledge		
C2	cat02_04_open_trace_clustering	X	X	Domain Knowledge		
C2	cat02_05_open_rpa	X	X	Domain Knowledge		
C2	cat02_06_open_anomaly_detection	X	X	Domain Knowledge		
C2	cat02_07_open_process_enhancement	X	X	Domain Knowledge		
C2	cat02_08_closed_process_mining		X	Domain Knowledge		
C2	cat02_09_closed_petri_nets		X	Domain Knowledge		
C3	cat03_01_temp_profile_generation	X	X	Process Modeling		
C3	cat03_02_declare_generation	X	X	Process Modeling		
C3	cat03_03_log_skeleton_generation	X	X	Process Modeling		
C3	cat03_04_process_tree_generation	X	X	Process Modeling		
C3	cat03_05_powl_generation	X	X	Process Modeling		
C3	cat03_06_temp_profile_discovery	X	X	Process Discovery	Variants	Road Traffic
C3	cat03_07_declare_discovery	X	X	Process Discovery	Variants	Road Traffic
C3	cat03_08_log_skeleton_discovery	X	X	Process Discovery	Variants	Road Traffic
C4	cat04_01_bpmn_xml_tasks	X	X	Task List	BPMN XML	Running Example
C4	cat04_02_bpmn_json_description	X	X	Description	BPMN JSON	CCC19
C4	cat04_03_bpmn_simp_xml_description	X	X	Description	BPMN XML (simple)	CCC19
C4	cat04_04_declare_description	X	X	Description	DECLARE	BPI2020 Domestic
C4	cat04_05_declare_anomalies	X	X	Semantic AD	DECLARE	BPI2020 Domestic
C4	cat04_06_log_skeleton_description	X	X	Description	Log Skeleton	BPI2020 Domestic
C4	cat04_07_log_skeleton_anomalies	X	X	Semantic AD	Log Skeleton	BPI2020 Domestic
C5	cat05_01_hypothesis_bpic2020	X	X	Hypothesis Generation	Variants	BPI2020 Domestic
C5	cat05_02_hypothesis_roadtraffic	X	X	Hypothesis Generation	Variants	Road Traffic
C5	cat05_03_hypothesis_bpmn_json	X	X	Hypothesis Generation	BPMN JSON	CCC19
C5	cat05_04_hypothesis_bpmn_simpl_xml	X	X	Hypothesis Generation	BPMN XML (simple)	CCC19
C6	cat06_01_renting_attributes		X	Discrimination Factors	Situation table	Renting (Fairness)
C6	cat06_02_hiring_attributes		X	Discrimination Factors	Situation table	Hiring (Fairness)
C6	cat06_03_lending_attributes		X	Discrimination Factors	Situation table	Lending (Fairness)
C6	cat06_04_hospital_attributes		X	Discrimination Factors	Situation table	Hospital (Fairness)
C6	cat06_05_renting_prot_comp	X	X	Comparison	Variants	Renting (Fairness)
C6	cat06_06_hiring_prot_comp	X	X	Comparison	Variants	Hiring (Fairness)
C6	cat06_07_lending_prot_comp	X	X	Comparison	Variants	Lending (Fairness)
C6	cat06_08_hospital_prot_comp	X	X	Comparison	Variants	Hospital (Fairness)
C7	cat07_01_dotted_chart	X	X	Description	Visual	Road Traffic
C7	cat07_02_perf_spectrum	X	X	Description	Visual	Road Traffic
C7	cat07_03_running-example	X	X	Description	Visual	Running Example
C7	cat07_04_credit-score	X	X	Description	Visual	Credit Score
C7	cat07_05_dfg_ru	X	X	Description	Visual	Running Example
C7	cat07_06_process_tree_ru	X	X	Description	Visual	Running Example

as the ones proposed in [1], are based on the generation of SQL statements but do not require process-specific domain knowledge.

Different categories of prompts (Table 1) are contained in the benchmark:

- **C1 General-purpose qualitative tasks:** The first category assesses the ability to describe processes, detect anomalies, and analyze root causes using DFG/variants abstractions of event logs. It also includes object-centric process mining artifacts for testing object-centric comprehension.
- **C2 Open/closed process mining domain knowledge questions:** The second category evaluates the process mining domain knowledge of the LLM, with open and closed questions about process mining and Petri nets.
- **C3 Process model generation:** The third category tests the ability to generate procedural (process trees, POWLs [17]) and declarative process models (control-flow and temporal) for mainstream processes, and the ability to propose constraints given some process data.
- **C4 Process model understanding:** The fourth category assesses the understanding of proposed procedural (BPMN) and declarative (Log Skeleton and DECLARE [18]) process models.
- **C5 Hypotheses generation:** The fifth category evaluates the ability to generate hypotheses over the proposed data and process models.
- **C6 Fairness assessment:** The sixth category tests the ability to identify event log attributes sensitive for fairness and compare protected and non-protected groups [19].
- **C7 Visual prompts:** The seventh category assesses the visual capabilities (if supported) of the LLM/LVLM.

We tailored the benchmark to the level of comprehension and reasoning of currently available state-of-the-art LLMs (in particular, *gpt-4o-20240513* and *claude-3.5-sonnet*), which can perform the tasks satisfactorily. Some mainstream tasks (for instance, applying the Alpha Miner algorithm, or checking the soundness of a Petri nets) are still not supported effectively by state-of-the-art LLMs. Therefore, they have not been included in the current version of the benchmark, which is comprehensive but not complete. The prompts of the benchmark are "static" (i.e., stored in TXT files). The benchmark could be easily adapted in the future to contain more prompts and/or different categories of tasks.

The size of the prompt does not exceed $8K$ characters, ensuring their executability on any of the considered LLMs. Among the considered LLMs, *Llama3 70B Instruct* has the most restrictive context window (i.e., the number of tokens that can be provided). Some state-of-the-art LLMs, such as *Nemotron 340B* or *Phi-3*, only support a baseline context window of $4K$, but we choose not to support that as it is too restrictive for process mining tasks. For more difficult event logs or process models, a bigger context window ($32K$, $64K$, or $128K$) is preferable as it allows to encode more information in the prompt.

3.2 Evaluation Strategy

The challenge of evaluating textual outputs from LLMs in an objective manner is significant. Traditional metrics that compare answers to a human-provided

Table 2. Scores between 1.0 and 10.0 ($mean \pm stddev$) of various LLMs in the proposed benchmark (using *gpt-4o-20240513* as a judge).

Commercial LLMs	Big Open-Source LLMs	Small LLMs (≤ 8 GB)	Tiny LLMs (≤ 4 GB)
claude-3.5-sonnet 8.4 ± 0.7	Qwen v2.0 72B (instruct, fp16) 7.6 ± 1.4	Qwen v2.0 7B (instruct, Q6K) 6.5 ± 2.2	Mistral 7B v0.3 (instruct, Q3KS) 4.6 ± 2.2
gpt-4o-20240513 (self) 8.3 ± 1.0	WizardLM v2 8x22b (16b) 7.5 ± 1.7	Mistral 7B v0.3 (instruct, Q6K) 5.9 ± 2.5	Qwen v2.0 7B (instruct, Q2K) 4.6 ± 2.1
gpt-4-turbo-20240409 8.1 ± 1.0	Mixtral v0.1 8x22b (instruct, 16b) 7.5 ± 1.6	Llama 3 8B (instruct, Q6K) 5.9 ± 2.5	Gemma v1.0 2B (instruct, Q6K) 4.0 ± 2.6
claude-3-sonnet 7.8 ± 1.4	Llama 3 70B (instruct, 16b) 7.4 ± 1.5	WizardLM v2 7b (Q6K) 5.9 ± 2.3	Qwen v2.0 1.5B (instruct, Q6K) 3.8 ± 2.2
Google Gemini (20240528) 7.5 ± 1.7	Mixtral v0.1 8x7b (instruct, 16b) 6.9 ± 1.7	Gemma v2.0 9B (instruct, Q6K) 5.7 ± 2.9	Qwen v2.0 0.5B (instruct, Q6K) 3.1 ± 1.7
gpt-3.5-turbo-0125 7.1 ± 1.8	Llama 3 70B (instruct, Q4_0) 6.7 ± 2.4	CodeGemma v1.5 7B (instruct, Q6K) 4.9 ± 2.6	Qwen 4B v1.5 (text, Q6K) 2.5 ± 2.2
	Codestral 22B (Q6K) 6.7 ± 2.1	Gemma v1.0 7B (instruct, Q6K) 4.5 ± 2.6	
	Llama 3 8B (instruct, 16b) 6.6 ± 2.0		
	OpenChat 3.6 8B (16b) 6.5 ± 1.7		

ground truth are limited by their inability to be customized to specific evaluation criteria and to consider open-ended answers. This leads to the exploration of using LLMs as judges (LLMs-as-Judges), a method that offers the potential for more comprehensive evaluations.

As the output of LLMs for the proposed prompts is textual, we propose to use an advanced LLM (for instance, *gpt-4o-20240513*) as judge [28], assigning a score from 1.0 (minimum) to 10.0 (maximum) to each answer.

In this benchmark, LLMs-as-Judges are utilized without ground truth for two primary reasons:

- The open-ended nature of many inquiries, in particular *process modeling* (C3) and *automated hypotheses generation* (C5), which lack definitive answers.
- The possibility that future LLM training sets could include the benchmark inquiries and their ground truths, potentially compromising the integrity of the benchmark by allowing "cheating" during the training phase of the LLM.

Potential limitations have been summarized in Sect. 2. In particular, LLMs-as-Judges are more reliable when there is a significant performance gap in favor of the judge LLM compared to the answering LLM. Also, scoring similar-performing LLMs without ground truth is challenging and requires accounting for potential "egocentric" bias. Moreover, high-performing LLMs tend to produce verbose and detailed outputs, which can result in a bias against more concise responses from lower-performing LLMs and vice-versa.

For the evaluation, the following procedure is followed for every prompt:

1. The prompt is provided to the LLM:
   - Reported as-is for all the textual prompts.
   - For visual prompts (if supported by the given model), upload the image accompanied by the following textual prompt: *Can you describe the provided visualization?*

Table 3. Scores between 1.0 and 10.0 (*mean* ± *stddev*) for different model categories and question categories (using *gpt-4o-20240513* as a judge).

	Commercial LLMs	Big Open-Source LLMs	Small LLMs (≤ 8 GB)	Tiny LLMs (≤ 4 GB)
C1	7.6 ± 1.1	7.0 ± 1.5	5.5 ± 2.3	3.2 ± 1.6
C2	8.7 ± 0.6	8.4 ± 0.9	8.5 ± 0.9	6.5 ± 2.2
C3	7.1 ± 1.7	5.7 ± 2.0	4.7 ± 2.5	2.4 ± 1.5
C4	7.7 ± 1.5	6.7 ± 1.9	3.7 ± 2.0	2.9 ± 1.5
C5	7.8 ± 1.8	7.3 ± 1.6	6.0 ± 2.5	4.6 ± 2.2
C6	8.1 ± 1.2	7.2 ± 2.0	5.0 ± 2.2	3.1 ± 1.6
C7	8.2 ± 1.3	no supp.	no supp.	no supp.

2. The LLM's answer is persisted.
3. An expert LLM (LLM-as-a-Judge) is used to evaluate the output. Template:
   - For textual prompts, *Given the following question: ... How would you grade the following answer from 1.0 (minimum) to 10.0 (maximum)?*.
   - For visual prompts, upload the image to the LVLM and ask *Given the attached image, how would you grade the following answer from 1.0 (minimum) to 10.0 (maximum)?*.

3.3 Benchmarking Scripts

The provided benchmark can be executed manually (first, an LLM is used to answer the questions, and its answers are evaluated by another LLM). We also provide some scripts to automate the execution of the benchmark. In particular, the Python script **answer.py** can be used to execute the prompts against any LLM supporting the OpenAI APIs, while **evaluation.py** can be used to evaluate the answers using an LLM as the judge. The configuration parameters could be set up inside the two scripts.

4 Benchmark Results

We executed the proposed benchmark against current state-of-the-art LLMs. The results are collected at the address https://zenodo.org/records/13164994.

Table 4. Scores between 1.0 and 10.0 (*mean* ± *stddev*) for answering (rows) and evaluating (columns) LLM pairs.

Answ./Eval.	gpt-4o-20240513	Llama3 70B Instr.	Mixtral 8x22B	Mixtral 8x7B	Qwen2 72B Instr.
Llama3 70B Instr.	7.4 ± 1.5	**8.8 ± 0.5**	8.6 ± 1.4	8.7 ± 1.3	7.7 ± 2.4
WizardLM-2-8x22B	7.5 ± 1.7	7.2 ± 3.0	7.2 ± 2.7	8.3 ± 1.3	7.1 ± 2.9
Mixtral 8x22B	7.5 ± 1.6	7.8 ± 2.3	7.5 ± 3.1	8.5 ± 1.3	7.6 ± 2.4
Mixtral 8x7B	6.9 ± 1.7	7.7 ± 2.3	7.4 ± 2.9	8.1 ± 1.7	7.1 ± 2.8
Qwen2 72B Instr.	7.6 ± 1.4	7.9 ± 2.3	8.6 ± 1.3	8.3 ± 1.9	**8.0 ± 2.5**

Table 5. Scores between 1.0 and 10.0 ($mean \pm stddev$) for different model categories and single prompts (using *gpt-4o-20240513* as a judge).

Prompt	Comm.	Big OS	Small	Tiny
cat01_01_variants_bpic2020_rca	8.6 ± 0.4	6.9 ± 1.4	5.7 ± 2.9	3.2 ± 1.7
cat01_02_variants_roadtraffic_anomalies	6.5 ± 1.6	6.1 ± 1.8	4.3 ± 2.4	2.2 ± 0.9
cat01_03_bpic2020_var_descr	7.8 ± 1.6	7.5 ± 0.9	7.0 ± 1.3	3.4 ± 1.4
cat01_04_roadtraffic_var_descr	8.3 ± 0.6	7.4 ± 0.9	6.8 ± 1.4	3.2 ± 1.6
cat01_05_bpic2020_dfg_descr	8.2 ± 0.6	8.2 ± 0.7	7.4 ± 1.8	3.8 ± 2.1
cat01_06_roadtraffic_dfg_descr	7.8 ± 0.6	6.4 ± 1.8	5.1 ± 2.1	3.9 ± 1.9
cat01_07_ocel_container_description	7.2 ± 0.9	6.6 ± 1.5	4.9 ± 1.2	4.0 ± 1.3
cat01_08_ocel_order_description	7.7 ± 0.7	6.8 ± 1.4	4.8 ± 2.3	3.5 ± 1.6
cat01_09_ocel_container_rca	7.2 ± 1.2	7.5 ± 0.8	5.4 ± 2.1	2.6 ± 0.9
cat01_10_ocel_order_rca	7.2 ± 0.7	6.7 ± 1.8	3.5 ± 1.3	2.5 ± 1.0
cat02_01_open_event_abstraction	8.7 ± 0.4	8.4 ± 0.6	8.7 ± 0.4	6.6 ± 1.8
cat02_02_open_process_cubes	8.9 ± 0.2	8.4 ± 0.5	8.5 ± 0.5	5.8 ± 1.8
cat02_03_open_decomposition_strategies	8.7 ± 0.9	8.5 ± 0.3	8.6 ± 0.3	7.2 ± 1.2
cat02_04_open_trace_clustering	8.5 ± 0.7	8.4 ± 0.5	8.7 ± 0.7	7.7 ± 0.9
cat02_05_open_rpa	9.1 ± 0.2	8.8 ± 0.4	9.0 ± 0.5	8.2 ± 1.2
cat02_06_open_anomaly_detection	8.6 ± 0.5	8.7 ± 0.4	8.1 ± 1.0	8.0 ± 1.4
cat02_07_open_process_enhancement	8.9 ± 0.3	8.7 ± 0.4	8.7 ± 0.7	6.4 ± 1.0
cat02_08_closed_process_mining	8.8 ± 0.6	8.0 ± 1.1	8.8 ± 0.7	6.0 ± 2.7
cat02_09_closed_petri_nets	8.1 ± 0.6	7.2 ± 1.8	6.9 ± 1.3	2.9 ± 1.2
cat03_01_temp_profile_generation	9.0 ± 0.1	7.8 ± 0.9	7.8 ± 1.2	3.4 ± 1.6
cat03_02_declare_generation	6.8 ± 1.5	5.8 ± 1.2	4.8 ± 1.4	2.7 ± 0.9
cat03_03_log_skeleton_generation	8.2 ± 1.2	7.3 ± 1.2	5.8 ± 1.7	3.2 ± 1.6
cat03_04_process_tree_generation	7.4 ± 1.2	5.6 ± 2.0	5.6 ± 1.6	3.4 ± 1.4
cat03_05_powl_generation	6.6 ± 1.2	6.5 ± 1.5	6.9 ± 2.0	2.5 ± 1.4
cat03_06_temp_profile_discovery	5.5 ± 1.8	4.2 ± 1.7	2.4 ± 0.7	1.7 ± 0.7
cat03_07_declare_discovery	7.4 ± 1.3	4.3 ± 1.4	1.9 ± 1.0	1.0 ± 0.0
cat03_08_log_skeleton_discovery	5.8 ± 1.2	4.1 ± 1.4	2.2 ± 1.1	1.2 ± 0.6
cat04_01_bpmn_xml_tasks	9.2 ± 0.6	7.8 ± 3.1	1.3 ± 0.5	1.7 ± 0.7
cat04_02_bpmn_json_description	7.9 ± 1.0	6.3 ± 2.0	2.9 ± 1.5	2.8 ± 2.1
cat04_03_bpmn_simp_xml_description	8.2 ± 0.6	6.7 ± 1.2	3.6 ± 1.9	2.6 ± 1.4
cat04_04_declare_description	7.3 ± 1.1	7.1 ± 1.1	5.7 ± 1.6	3.6 ± 1.4
cat04_05_declare_anomalies	6.8 ± 2.0	5.9 ± 1.6	3.4 ± 0.9	3.2 ± 1.3
cat04_06_log_skeleton_description	7.9 ± 1.3	6.6 ± 1.4	5.2 ± 1.9	3.3 ± 1.7
cat04_07_log_skeleton_anomalies	6.5 ± 1.3	6.5 ± 1.6	4.0 ± 1.9	2.9 ± 0.7
cat05_01_hypothesis_bpic2020	8.2 ± 0.4	7.4 ± 1.4	7.6 ± 0.7	4.6 ± 1.7
cat05_02_hypothesis_roadtraffic	8.1 ± 1.0	6.6 ± 1.7	6.8 ± 2.2	4.7 ± 2.3
cat05_03_hypothesis_bpmn_json	7.5 ± 1.6	7.7 ± 1.2	5.2 ± 2.2	4.5 ± 2.3
cat05_04_hypothesis_bpmn_simpl_xml	7.2 ± 2.9	7.4 ± 1.9	4.6 ± 2.9	4.5 ± 2.4
cat06_01_renting_attributes	8.8 ± 0.6	8.2 ± 1.4	4.1 ± 2.0	2.5 ± 1.1
cat06_02_hiring_attributes	8.9 ± 0.7	8.6 ± 0.8	6.1 ± 2.4	5.0 ± 2.2
cat06_03_lending_attributes	8.3 ± 0.5	8.3 ± 1.1	7.1 ± 2.3	2.8 ± 1.2
cat06_04_hospital_attributes	8.8 ± 0.5	8.4 ± 0.7	6.3 ± 1.8	4.4 ± 2.1
cat06_05_renting_prot_comp	7.5 ± 1.2	5.8 ± 1.8	3.9 ± 1.1	2.0 ± 0.0
cat06_06_hiring_prot_comp	7.2 ± 1.4	6.1 ± 2.5	4.4 ± 1.7	2.9 ± 0.9
cat06_07_lending_prot_comp	8.3 ± 0.6	6.5 ± 1.8	3.6 ± 1.6	2.3 ± 0.5
cat06_08_hospital_prot_comp	7.0 ± 1.6	5.8 ± 1.8	4.4 ± 1.6	2.7 ± 0.9
cat07_01_dotted_chart	8.3 ± 0.7	–	–	–
cat07_02_perf_spectrum	8.5 ± 0.8	–	–	–
cat07_03_running-example	8.9 ± 0.5	–	–	–
cat07_04_credit-score	8.6 ± 0.6	–	–	–
cat07_05_dfg_ru	7.4 ± 2.3	–	–	–
cat07_06_process_tree_ru	7.8 ± 1.0	–	–	–

In Table 2, we collect the evaluation results (using *gpt-4o-20240513* as a judge) for different LLMs of different sizes. In particular, we divide between *commercial models* (usually ranging hundreds of billions of parameters), *big open-source LLMs*, *small LLMs* ($\leq$ 8 GB of RAM), and *tiny LLMs* ($\leq$ 4 GB of RAM).

We tested some LLMs with different levels of quantization. Quantization refers to the process of reducing the precision of the model's weights and activations from higher bit-widths (such as 16-bit floating point) to lower bit-widths (such as 8-bit or even lower). This technique aims to decrease the model's memory footprint and computational requirements, making it more efficient to run on hardware with limited resources. While quantization can lead to a slight loss in model accuracy, it often significantly improves the model's speed and reduces power consumption, enabling more practical and scalable deployment of LLMs.

In general, we see that commercial and big open-source models can perform process mining tasks adequately well. The winners among commercial models seem to be *gpt-4o-20240513* and *claude-3.5-sonnet*. The best of the open-source models seem to be *Qwen2 72B Instruct* (occupying 145 GB of memory at full quantization). Some small LLMs also report an overall sufficient score. For instance, *Qwen2 7B Instruct* at *Q6K* quantization is process-mining-capable while occupying just 6.3 GB of memory. However, tiny LLMs are still inadequate for process mining tasks.

Overall, we found that aggressive quantization has a severe impact on LLMs' abilities in the proposed benchmark. While we do not have precise explanations for this phenomenon, the benchmarks' prompt requires significant attention to some elements of the input prompt (e.g., the semantic anomalies), and quantization impacts such ability.

In Table 3, we report the scores for every model category and question category. In Table 5, the scores for every prompt of the benchmark are averaged over a model category.

We notice that question category **C2** (open/closed process mining domain knowledge questions) is answered adequately by all the model categories. Also, currently, only commercial models could answer to **C7** (visual prompts). Surprisingly, most of the considered LLMs could automatically generate hypotheses over the provided process models and event data (**C5**). Categories **C3** (process model generation) and **C4** (process model understanding) have high variance between the different model categories. While commercial and big open-source LLMs can perform the tasks adequately, small and tiny models fail to generate/understand the information provided in the prompt. Categories **C1** (general-purpose qualitative tasks) and **C6** (fairness assessment) have similar outcomes. Commercial and big open-source LLMs successfully execute such tasks, while small/tiny LLMs usually fail.

To further assess the validity of LLMs-as-Judges, we report in Table 4 a cross-validation based on five open-source answering models and five evaluation models (including *gpt-4o-20240513*). We see that only *Llama 3 70B Instruct* display the egocentric bias. The models tend to agree on the scores. The smallest considered model (*Mixtral 8x7B*) reports lower scores than its bigger counterpart (*Mixtral*

8x22B) and other state-of-the-art models (*Llama 3 70B Instruct* and *Qwen2 72B Instruct*). Notably, *WizardLM2 8x22B* achieves lower overall scores than *Mixtral 8x7B*. This could be explained by the model being a fine-tuned version of *Mixtral 8x22B* favoring more verbose responses. Overall, only *Qwen2 72B Instruct* and *gpt-4o-20240513* favor Qwen2 over Llama3, while Qwen2 is an overall better-performing model in general-purpose benchmarks[8]. As the models and responses become more advanced, it becomes complicated for lower-performing models to judge the differences between them. Using the less advanced *Mixtral 8x7B*, the "plateauing" of the scores becomes evident, highlighting the importance of using advanced LLMs as judges. Therefore, Table 4 justifies LLMs-as-Judges, but points to the importance of choosing advanced LLMs for the task.

5 Future Benchmarking Strategies

Benchmarking Retrieval-Augmented Generation (RAG): while LLMs have been trained on generic process-specific knowledge, the implementation of business processes in real-life organizations could be different. Therefore, Retrieval-Augmented Generation (RAG) techniques have been used to dynamically inject process-specific knowledge to the prompts provided by process mining analysts [5]. As the correct retrieval of the information is crucial for the quality of the answers, benchmarks should also assess the capabilities of the RAG pipeline.

Benchmarking LLM-Based Agent Crews: Our benchmark focuses on executing a prompt, recording the answer, and evaluating it. The *agents crew* paradigm [12] involves workflows with specialized or role-specific LLMs performing analytical tasks. For instance, estimating the level of discrimination in an event log involves identifying the protected group and comparing differences between groups. An agents crew could consist of one agent creating the SQL statement to filter the protected group and another specializing in process comparison. Each agent's performance is crucial for accurate benchmarking, as evaluating only the final output may be misleading.

Benchmarking Hypotheses Refinement: Hypotheses generation involves creating and verifying hypotheses against event data. If a hypothesis is invalid, the LLM refines it based on feedback. This process may lead to specific hypotheses that perform poorly on imseem data. Thus, it is important to test the entire hypothesis feedback and verification cycle, not just hypotheses generation.

Data Generation with Relevant Semantic Anomalies or Root Causes: Our benchmark uses publicly available event data and process models. Including these datasets in LLM training data, even without ground truth answers, can lead to higher benchmark scores. Therefore, it is important to dynamically generate novel event data for LLM benchmarking. Current simulation solutions lack the semantic understanding needed to generate data suitable for process

[8] https://qwenlm.github.io/blog/qwen2/.

mining assessment. LLMs should be considered for the goal of generating data with various semantic anomalies and root causes.

6 Conclusion

We proposed *PM-LLM-Benchmark*, a benchmark for process mining on LLMs, utilizing LLMs-as-Judges for scalable evaluation. This benchmark does not provide ground truth answers, relying instead on the "judge" LLM's capabilities, addressing the open-ended nature of the inquiries. The benchmark includes various task categories to assess LLMs' abilities in process mining, modeling, and comprehension. Applied to several commercial and open-source LLMs, we found most models perform well in process mining tasks, with bigger models achieving higher scores. Smaller models ($\leq$ 8 GB or $\leq$ 4 GB of RAM) struggle with complex tasks. We also suggest improvements in benchmarking strategies, including enhanced hypotheses generation, agent-crew-based benchmarks, and dynamic dataset generation with semantic anomalies. To our knowledge, this is the first general-purpose process mining benchmark focusing on different implementation strategies. While it has limitations in evaluating well-performing LLMs, it is a useful tool for assessing smaller LLMs and evaluating the current state of PM-on-LLMs.

References

1. Barbieri, L., Madeira, E.R.M., Stroeh, K., van der Aalst, W.M.P.: A natural language querying interface for process mining. J. Intell. Inf. Syst. **61**(1), 113–142 (2023)
2. Berti, A., Kourani, H., Hafke, H., Yun-Li, C., Schuster, D.: Evaluating large language models in process mining: capabilities, benchmarks, evaluation strategies, and future challenges. In: BPM-DS 2024 Working Conference. Springer (2024)
3. Berti, A., Qafari, M.S.: Leveraging large language models (LLMs) for process mining (technical report) (2023)
4. Berti, A., Schuster, D., van der Aalst, W.M.P.: Abstractions, scenarios, and prompt definitions for process mining with LLMs: a case study. In: BPM 2023, vol. 492, pp. 427–439. Springer (2023)
5. Chen, B., Zhang, Z., Langrené, N., Zhu, S.: Unleashing the potential of prompt engineering in large language models: a comprehensive review (2023)
6. Dhurandhar, A., Nair, R., Singh, M., Daly, E., Ramamurthy, K.N.: Ranking large language models without ground truth (2024)
7. Dixit, P.M., Buijs, J.C.A.M., van der Aalst, W.M.P., Hompes, B.F.A., Buurman, J.: Using domain knowledge to enhance process mining results. In: SIMPDA 2015, vol. 244, pp. 76–104. Springer (2015)
8. Dong, Y.R., Hu, T., Collier, N.: Can LLM be a personalized judge? arXiv preprint arXiv:2406.11657 (2024)
9. Fahland, D., Fournier, F., Limonad, L., Skarbovsky, I., Swevels, A.J.E.: How well can large language models explain business processes? (2024)
10. Fournier, F., Limonad, L., Skarbovsky, I.: Towards a benchmark for causal business process reasoning with LLMs (2024)

11. Grohs, M., Abb, L., Elsayed, N., Rehse, J.: Large language models can accomplish business process management tasks. In: BPM 2023 Workshops, vol. 492, pp. 453–465. Springer (2023)
12. Guo, T., Chen, X., Wang, Y., et al., R.C.: Large language model based multi-agents: a survey of progress and challenges (2024)
13. Jessen, U., Sroka, M., Fahland, D.: Chit-chat or deep talk: prompt engineering for process mining (2023)
14. Koo, R., Lee, M., Raheja, V., Park, J.I., Kim, Z.M., Kang, D.: Benchmarking cognitive biases in large language models as evaluators (2023)
15. Kourani, H., Berti, A., Schuster, D., van der Aalst, W.M.P.: Process modeling with large language models. In: EMMSAD 2024. Springer (2024)
16. Kourani, H., Berti, A., Schuster, D., van der Aalst, W.M.P.: ProMoAI: Process modeling with generative AI. In: IJCAI 2024 Demo Track (2024)
17. Kourani, H., van Zelst, S.J.: POWL: partially ordered workflow language. In: BPM 2023 Proceedings, vol. 14159, pp. 92–108. Springer (2023)
18. Maggi, F.M.: Declarative process mining with the declare component of ProM. In: BPM Demos sessions 2013, vol. 1021. CEUR-WS.org (2013)
19. Pohl, T., Berti, A., Qafari, M.S., van der Aalst, W.M.P.: A collection of simulated event logs for fairness assessment in process mining. In: BPM 2023 Resources. vol. 3469, pp. 87–91. CEUR-WS.org (2023)
20. Raina, V., Liusie, A., Gales, M.J.F.: Is LLM-as-a-Judge Robust? Investigating universal adversarial attacks on zero-shot LLM assessment (2024)
21. Rebmann, A., Schmidt, F.D., Glavaš, G., van der Aa, H.: Evaluating the ability of LLMs to solve semantics-aware process mining tasks. arXiv preprint arXiv:2407.02310 (2024)
22. Singh, A.K., Devkota, S., Lamichhane, B., Dhakal, U., Dhakal, C.: The confidence-competence gap in large language models: a cognitive study (2023)
23. Thakur, A.S., Choudhary, K., Ramayapally, V.S., Vaidyanathan, S., Hupkes, D.: Judging the judges: evaluating alignment and vulnerabilities in LLMs-as-judges. arXiv preprint arXiv:2406.12624 (2024)
24. Vidgof, M., Bachhofner, S., Mendling, J.: Large language models for business process management: opportunities and challenges. In: BPM 2023 Forum, vol. 490, pp. 107–123. Springer (2023)
25. Wang, P., Li, L., Chen, L., et al., D.Z.: Large language models are not fair evaluators (2023)
26. Wornow, M., Narayan, A., et al., B.V.: Do multimodal foundation models understand enterprise workflows? A benchmark for business process management tasks (2024)
27. Zerbato, F., Soffer, P., Weber, B.: Initial insights into exploratory process mining practices. In: BPM Forum 2021, vol. 427, pp. 145–161. Springer (2021)
28. Zheng, L., Chiang, W., Sheng, Y., et al., S.Z.: Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In: NeurIPS 2023 (2023)