Large Language Models as Digital Libraries: A Multi-Benchmark and Multi-Model Study

Mirco Cazzaro[1*] and Gianmaria Silvello[1]

[1*]Department of Information Engineering, University of Padova, Via Gradenigo 6/B, Padova, 35131, Italy.

*Corresponding author(s). E-mail(s): mirco.cazzaro@phd.unipd.it;
Contributing authors: gianmaria.silvello@unipd.it;

Abstract

Purpose: Large Language Models (LLMs) are increasingly used as flexible access layers for information-rich collections, raising the question of whether they can also support structured access to cultural heritage data as if they were digital libraries. This work investigates that question through Galois, a database-inspired system that treats the LLM as a noisy storage layer.

Methods: We evaluate six querying modes: direct natural language prompting, direct SQL prompting, and four Galois strategies that combine LLM-facing tuple retrieval with deterministic relational processing. The evaluation uses two cultural heritage benchmarks: Paintings Extended, based on paintings data, and a new benchmark based on Europeana metadata. For both benchmarks, we define fourteen query templates with ten variants each. The main experiments use GPT-4o-mini, Granite 3.3 8B, and Llama 3 70B, with an additional diagnostic analysis of two GPT-5 deployments. We complement the benchmark metrics with a stratified manual audit of 60 unique disagreements.

Results: Performance is strongly shaped by model capability, deployment behavior, dataset structure, and execution strategy. Direct natural language prompting is the strongest and most robust baseline, while direct SQL prompting becomes nearly equivalent on the strongest open models. Among the structured strategies, only GaloisA and GaloisF become meaningfully competitive, especially on the flatter Europeana benchmark. In the audited sample, 35 disagreements were model-originated failures, 21 reflected reference-coverage or metadata-normalization effects, and four remained indeterminate. The additional GPT-5 deployment analysis reveals frequent empty outputs and provider-level constraints. These diagnostic results characterize the specific tested deployments and must not be interpreted as a general ranking of GPT-5 model capability.

Conclusion: Querying LLMs as digital libraries is feasible, but its effectiveness depends on model strength, deployment conditions, dataset structure, and execution strategy. Direct prompting is already strong for exploratory cultural heritage access, while Galois remains valuable when relational discipline and controlled query execution are required.

Keywords: Large Language Models, Digital Libraries, Cultural Heritage, Europeana

1 Introduction

Large Language Models (LLMs) are currently widely used for content generation, complex reasoning, information retrieval, and knowledge access [1]. They are deployed as augmentation

layers for traditional search systems, and increasingly as direct sources of information that leverage the knowledge encoded in their parameters [2, 3], oftentimes extended through web search or Retrieval-Augmented Generation (RAG) [4]. Retrieval can enrich or update what a model produces and shift factual access from parametric recall to an external collection, but downstream reliability then depends on retrieval coverage, evidence quality, and faithful grounding as well as on the model's ability to interpret and structure the evidence [5]. When parametric extraction or evidence-grounded generation is poor – whether due to hallucination, incompleteness, or poor calibration [6, 7] – downstream outputs remain unreliable. Establishing how well a model can serve as a primary information source for a given domain and query type is therefore not merely a technical question: it has direct implications for any application that relies on Large Language Models (LLMs) to verify, substantiate, or complement factual claims with evidence drawn from the model's own parametric knowledge.

This observation motivates a growing line of research that investigates LLMs not merely as generative assistants, but as implicit knowledge repositories that can be queried in a controlled and structured manner. The most substantive contribution in this direction is Galois [8], a system that treats the LLM as a noisy storage layer and interposes a database-inspired execution framework between the user query and the model. Rather than delegating the entire reasoning task to a single model interaction, as direct natural language (NL) prompting or direct SQL prompting do, Galois decomposes query execution into logical and physical steps, combining LLM-specific scan operators for tuple retrieval with deterministic relational operators for joins, projections, and aggregations. Galois provides a principled basis for comparing several querying strategies as direct NL prompting, direct SQL prompting, and multiple structured variants ranging from GaloisWO without any database-derived optimization strategy to GaloisF employing the full suite of optimization techniques with GaloisS and GaloisA as intermediate versions.

Our goal in this work is to analyze whether these querying strategies can be meaningfully adopted in the context of digital libraries, with a specific focus on Cultural Heritage (CH) collections. CH constitutes a particularly compelling application domain because its information is heterogeneous and often stratified across multiple representation layers ranging from raw catalog metadata to curated knowledge graphs and large-scale aggregation platforms, which places it at the boundary between structured and unstructured knowledge access. At the same time, CH entities such as artworks, artists, museums, and historical periods constitute precisely the kind of well-documented, encyclopedic knowledge that LLMs are known to encode in their parameters [2, 9]. This creates a concrete opportunity: LLMs can answer exploratory queries over CH data while substantiating and verifying factual claims about entities and their properties, complementing curated sources where coverage is uneven or provenance is difficult to trace. Access to digital libraries and CH repositories traditionally relies on formal catalogs, controlled vocabularies, and structured query interfaces; LLM-based querying offers a complementary layer that can bridge the gap between informal user needs and structured data [10, 11]. Despite this potential, and to the best of our knowledge, no prior work has systematically studied whether an LLM can be queried *as if it were a digital library*, providing structured, database-like access to CH data under controlled information needs and rigorous evaluation. The present work is the first to address this gap directly, extending our preliminary study [12] from an initial proof of concept to a broader and methodologically more rigorous empirical assessment.

To support this investigation, we construct and publicly release two new benchmarks. The first, which we call *Paintings Extended*, builds on the "Famous Paintings" dataset reinterpreted as a proxy CH catalog. It comprises fourteen query families covering artists, works, museums, subjects, and temporal constraints, each instantiated into ten parameterized formulations for a total of 140 queries. The second benchmark is derived from Europeana [13], a large-scale aggregation platform for CH metadata. Acquired through the Europeana Search API and processed through a dedicated harvesting, cleaning, and normalization pipeline, the resulting dataset comprises over 66 million records across eight metadata fields, paired with fourteen manually designed query

families and their respective parameterized variants, again yielding 140 queries. Both benchmarks are released together with their schema descriptions, ingestion scripts, and query files to support reproducibility and further analysis by the community[1][2].

Using these benchmarks, we evaluate six querying strategies – NL, SQL, GaloisWO, GaloisS, GaloisA, and GaloisF – across three primary LLM deployments: gpt-4o-mini, accessed via an MS Azure-hosted deployment, and Llama 3 70B and Granite 3.3 8B, served locally through Ollama. We also extend this evaluation to two additional commercial GPT-5 deployments, namely gpt-5.2 and gpt-5.4-mini, always accessed through MS Azure, in order to further investigate how model and deployment-specific factors affect this new search paradigm. Table 1 reports deployment specifications and versioning details. These additional deployments exhibited unexpected behavior and are therefore analyzed separately through a dedicated diagnostic evaluation. Search effectiveness is measured through F1-Cell, Cardinality, Tuple Constraint, and their average (AVG-Score), following the evaluation protocol established in the Galois study [8].

The experiments lead to the following key conclusions. First, model capability is the primary determinant of performance: the gap between stable primary deployments (GPT-4o-mini, Granite 3.3 8B, Llama 3 70B) and the two GPT-5 deployments far exceeds the gap among querying strategies within the same model. Second, direct NL prompting is the strongest and most robust baseline across both benchmarks, with direct SQL becoming nearly equivalent on the strongest open models; this represents a substantially different picture from the one reported in the original Galois study, where structured mediation clearly outperformed weaker baselines. Third, among the Galois variants, only GaloisA and GaloisF are competitive; GaloisWO is consistently underperforming, and its additional interaction complexity does not translate into quality gains. Fourth, the usefulness of structured mediation is conditional on dataset structure: Galois narrows the gap to direct prompting considerably more on the flatter Europeana benchmark, where queries operate over a single large metadata table, than on the richer multi-table Paintings Extended benchmark. Fifth, the ten-variant design reveals that methods which appear acceptable on average can be considerably less stable across semantically equivalent parameterized reformulations.

The remainder of the paper is organized as follows. Section 2 provides background on LLMs as knowledge repositories, direct querying baselines, and the Galois system. Section 3 reviews related work across three lines: LLMs as knowledge bases, LLMs in digital libraries and CH, and robustness in declarative and text-to-SQL querying. Section 4 describes the construction of the two benchmarks. Section 5 details the experimental setup, including models, execution settings, and implementation. Section 6 presents and analyzes the results across four complementary views: overall performance, quality profile decomposition, category-wise behavior, and quality versus token consumption; it also includes a diagnostic analysis of the two additional GPT-5 deployments. Section 7 discusses the implications of the findings. Section 8 concludes and outlines directions for future work.

[1] https://zenodo.org/records/19727990
[2] https://zenodo.org/records/19707162

Paper label	Provider	Deployment / local tag	Access period	Inference interface	Decoding and runtime settings
GPT-4o-mini	Azure OpenAI	`gpt-4o-mini-2`	8–9 Apr. 2026	Azure OpenAI Responses API via `/openai/v1/`	temperature = 0; top_p = 1; `max_output_tokens`=100000; seed unset
GPT-5.2	Azure OpenAI	`gpt-5.2`	26 Mar. 2026	Azure OpenAI Responses API via `/openai/v1/`	temperature = 0; top_p = 1; `max_output_tokens`=100000; seed unset
GPT-5.4	Azure OpenAI	`gpt-5.4-mini`	8 Apr. 2026	Azure OpenAI Responses API via `/openai/v1/`	temperature = 0; top_p = 1; `max_output_tokens`=100000; seed unset
Granite 3.3 8B	Ollama	`granite3.3:8b`	26 Mar. 2026	OpenAI-compatible `/v1/chat/completions`	temperature = 0; other decoding and output limits unset; Ollama 0.30.5; 8.2B, Q4_K_M, GGUF
Llama 3 70B	Ollama	`llama3:70b`	31 Mar.–1 Apr. 2026	OpenAI-compatible `/v1/chat/completions`	temperature = 0; other decoding and output limits unset; Ollama 0.30.5; 70.6B, Q4_0, GGUF

Table 1 Deployment and inference metadata for the five experimental configurations. Access periods were reconstructed from the timestamps encoded in the run-directory names and their final modification dates. For Ollama, parameters not explicitly included in the requests, including context and output limits, were determined by the server defaults.

2 Background

The premise underlying this work is that pretrained language models encode factual and relational knowledge in their parameters as a byproduct of large-scale training, and that this knowledge can, in principle, be extracted through structured interaction rather than purely generative use [2, 3, 14]. Whether this extraction is reliable in practice depends on well-documented limitations: knowledge encoded parametrically is unevenly distributed, temporally unstable, and difficult to trace to explicit provenance, with hallucination and poor calibration remaining central concerns [6, 7]. The following subsections review the querying approaches that operationalize this premise and the system architecture on which the present study is built.

Natural language querying of large language models

The most immediate way to access the knowledge embedded in an LLM is to formulate an information need directly in natural language. This interaction mode requires no formal query language and generalizes across users and domains, which explains its adoption as the default access paradigm in most deployed systems. Its central limitation, however, is linguistic ambiguity. Natural language is inherently underspecified: the same surface form can express distinct information needs, multi-hop constraints are difficult to articulate without introducing vagueness, and completeness requirements are rarely made explicit. When the goal is a short, self-contained answer, this ambiguity is often manageable. When the goal is the reconstruction of a complete result table satisfying multiple simultaneous constraints, it could become a structural obstacle. Structured Query Language (SQL), by contrast, offers a precise declarative specification of projections, selections, joins, and aggregations, and the accompanying schema description gives the model explicit information about attribute names, types, and relational structure [15]. This avoids the ambiguity of natural language formulation and removes the need for the model to infer output structure from the query text alone.

Closely related work in Text-to-SQL aims to translate natural language questions into executable SQL against an existing database, and recent surveys show that LLM-based Text-to-SQL has advanced substantially when a concrete schema and an actual backend are available [15]. This setting differs conceptually from the one addressed in this work. In Text-to-SQL, the model acts as a translator: facts reside in an external database, and the model's task is to produce a query that retrieves them correctly. Here, by contrast, the model itself is the source from which facts must be extracted. Whether parametric memory is sufficient for this role depends on the domain and entity type: recent work has shown that factual reliability degrades substantially for less prominent entities, even when prominent ones are handled well [16]. This asymmetry is directly relevant to CH querying, where well-known artists and major museums coexist with long-tail entities that are less likely to be reliably encoded.

Direct SQL prompting and its limitations

A natural step beyond direct Natural Language (NL) prompting is to submit SQL itself to the model and ask it to return the result set in structured form. Compared with NL querying, this approach offers a clearer specification of projections, selections, joins, and aggregations. In practice, however, it still relies on the model to interpret the full query and to carry out the whole reasoning process within a single interaction. This means that the structure of SQL is present only in the prompt, not in the execution mechanism. As query complexity increases, the model may satisfy some constraints while ignoring others, may produce incomplete enumerations, or may return tuples that are only partially consistent with the intended schema. In other words, direct SQL prompting constrains the surface form of the request, but it does not yet provide a complete execution framework.

Galois

Galois [8], whose architecture is shown in Figure 1, is a database-inspired framework for executing SQL queries over an LLM treated as a *noisy storage layer* rather than a monolithic question-answering system. Rather than asking the model to resolve a query in a single prompt, Galois decomposes execution: the model enumerates candidate tuples, while joins, projections, duplicate elimination, grouping, ordering, and aggregations are evaluated by a standard relational engine. The LLM thus acts as a factual source; all other query processing is handled by deterministic operators.

Given an input SQL query and a logical schema, Galois builds a logical query plan over two LLM-facing scan operators: LLMScan, which asks the model to enumerate tuples for a relation without constraint, and FilterLLMScan, which asks the model to return only tuples satisfying one or more predicates [8]. This reinterprets a classical optimization problem in an LLM setting. In a standard Database Management System (DBMS), pushing selections early reduces downstream data volume; in Galois, pushing too many predicates in the prompt increases reasoning complexity and can degrade factual completeness. Query optimization must then balance token usage versus answer quality, not just minimize intermediate cardinality.

Galois generates alternative logical plans that differ in how many predicates are pushed into LLM-facing scans. In the most conservative plan, no predicate is pushed and all filtering is deferred to deterministic operators. In the most aggressive plan, all eligible predicates are pushed into the prompt. Intermediate plans push only those predicates for which the model is estimated to be reliable, with reliability encoded in a confidence catalog that the logical optimizer consults at planning time [8]. Predicates the model handles well are candidates for pushdown; those requiring complex reasoning are left to post-processing.

At the physical level, Galois selects between two scan strategies. TableScan prompts the model to return full tuples directly in structured JSON form, repeating the interaction until no new tuples are produced. KeyScan decomposes retrieval into two stages: a first prompt enumerates key values, and subsequent prompts recover the remaining attributes for each key. The physical planner selects between the two by comparing an estimated confidence in key retrieval against a threshold τ: when confidence exceeds τ, KeyScan is preferred because more focused prompts tend to improve tuple quality; otherwise, TableScan is used [8]. Retrieved tuples are materialized locally and passed to the DBMS backend.

The two design dimensions, predicate pushdown policy and physical scan strategy, define four experimental variants compared in this study (Table 2). GaloisWO disables the optimizer entirely and always uses KeyScan, establishing the unoptimized structured baseline. GaloisS also disables predicate pushdown but always uses TableScan. GaloisA pushes all eligible predicates and applies confidence-guided physical selection. GaloisF combines confidence-guided predicate pushdown with confidence-guided physical selection. Comparing these four variants makes it possible to attribute observed performance differences to specific components of the Galois design. The intended difference between GaloisA and GaloisF lies in logical rather than physical optimization. GaloisA always adopts the most aggressive logical plan, pushing all eligible predicates into the LLM-facing scans, whereas GaloisF uses the confidence estimates to decide which predicates should be pushed down. Both variants subsequently use the same confidence-guided physical-plan selection. Consequently, the two

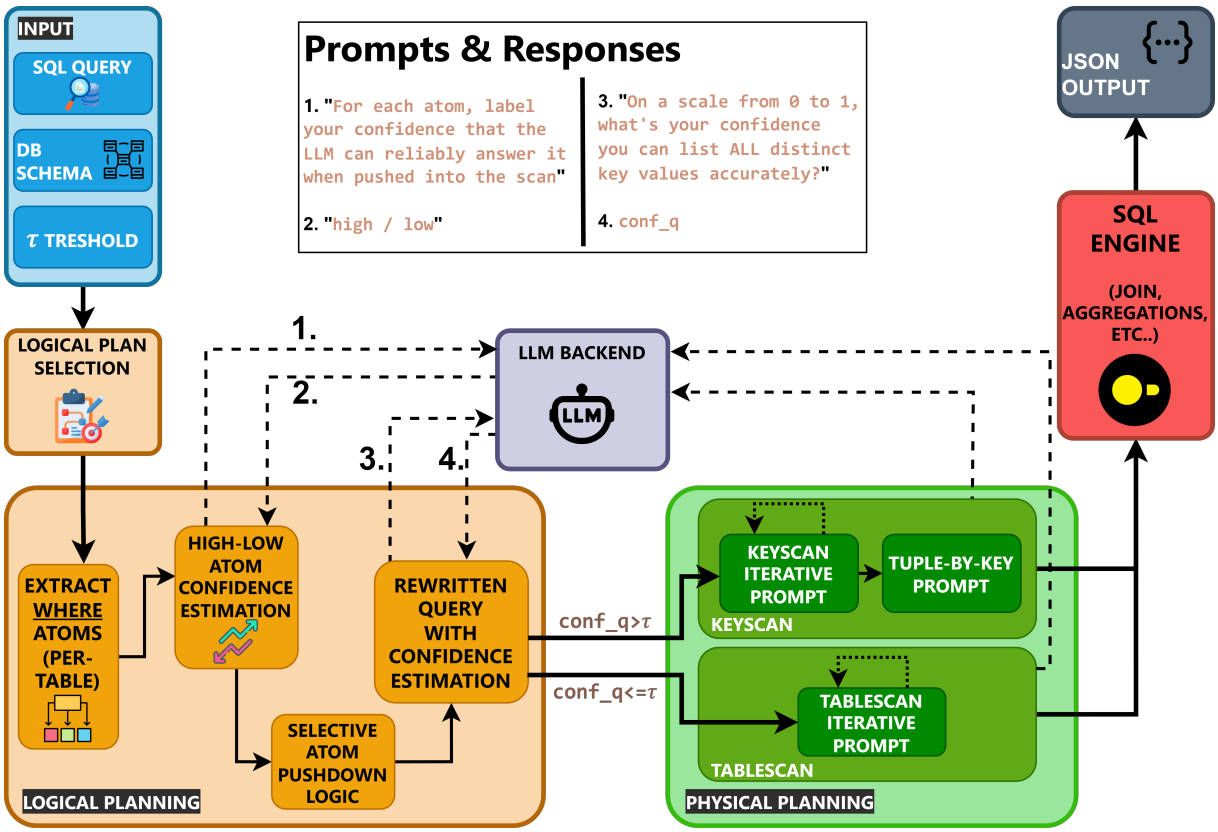

Fig. 1 Architecture representation of the Galois system. On the left, users and client applications issue SQL queries over a (possibly, self-defined) logical DB schema that describes their domain of interest. These queries enter the Galois core through the SQL front-end, which parses and analyses them to produce an initial logical plan. The logical optimizer refines this plan by deciding where query constraints should be placed, and in particular whether conditions in the WHERE clause should be pushed down into LLM-facing scan operators (FilterLLMScan) or evaluated later on materialized data (LLMScan). The confidence catalog stores precomputed estimates of how reliably the LLM can handle different attributes and predicates, and feeds this information into the optimizer. The physical planner instantiates the chosen logical plan with concrete operators, selecting among alternative scan strategies such as TableScan (direct tuple retrieval) and KeyScan (key-first retrieval with focused follow-up prompts), while all other operators (joins, projections, aggregations) are relegated at the end of the pipe to a standard DBMS.

variants become operationally equivalent whenever the confidence-guided optimizer of GaloisF selects the same predicate pushdowns as the all-pushdown strategy and the same physical scan operators.

Table 2 The four Galois variants and their defining choices along the two optimization axes.

Variant	Logical plan	Physical plan
GaloisWO	LLMScan only	KeyScan always
GaloisS	LLMScan only	TableScan always
GaloisA	FilterLLMScan	Confidence-guided
GaloisF	Confidence-guided	Confidence-guided

3 Related Work

Large language models as knowledge repositories

A first research line investigates whether LLMs can be treated as repositories of factual knowledge. A grounding research branch [2] showed that pretrained language models can store relational facts in their parameters and can be probed through prompts, thereby motivating the view of language models as implicit knowledge bases. Subsequent work [14, 17, 18] strengthened this perspective while also highlighting important limitations, including incomplete coverage, sensitivity to prompt formulation, and difficulties with more

complex knowledge access patterns such as one to many relations, temporal facts, and structured retrieval. More recently, surveys on factuality and hallucination [6, 7] showed that the central issue is not only whether models store facts, but whether they can return them reliably, consistently, and with sufficient completeness for downstream use.

This literature is important because it establishes the conceptual premise that a model can be queried as a factual source. At the same time, most of these studies focus on probing individual facts, benchmark style question answering, or general factual reliability. They do not typically study the reconstruction of structured result tables from multiple tuples and multiple constraints, nor do they focus on CH collections as a target domain.

Large language models in digital libraries and cultural heritage

A second line of work explores the adoption of LLMs in digital libraries and CH. Recent digital library work has discussed the relevance of LLMs for library services and information access [10]. In CH, prior studies have investigated LLMs for museum guidance and digital storytelling [11], multimodal search and explainable discovery over visual collections [19], and broader infrastructures aimed at multilingual and Artificial Intelligence (AI) assisted access to CH data [20].

These contributions show that LLMs are becoming increasingly relevant for access and interaction in GLAM and digital library environments. However, their primary focus is not structured query execution over the model itself. In most cases, the goal is to improve discovery, narration, recommendation, or metadata quality. As a result, the literature still lacks a systematic evaluation of whether an LLM can act as a structured access layer for CH data under database-like querying requirements.

DB and LLM querying systems and robustness

The work of Satriani et al. [8] reframes the problem of LLM-based querying: the objective is no longer to ask a model for an answer, but to execute a structured query over a probabilistic and imperfect source, separating LLM-facing retrieval from deterministic relational processing. This places the problem at the intersection of data management and LLM querying, a space that many recent systems have begun to occupy from different angles.

Palimpzest [21] explores declarative optimization for AI-powered analytics over unstructured data, demonstrating that database principles transfer productively to LLM-based computation; its scope, however, is broader than SQL-style execution over model knowledge and it does not target CH evaluation under controlled families of structured information needs.

A related but architecturally distinct direction is represented by SwellDB [22], a system that generates relational tables on the fly in response to SQL queries rather than executing queries over pre-existing stored data. SwellDB decomposes table generation into a planning phase, in which an LLM decides which columns can be derived from internal knowledge, from structured input datasets, or from web search, and a materialization phase, in which partial results from heterogeneous sources are merged in a single table via a query engine. The LLM is thus one source among many rather than the sole storage layer.

The work of Zhao et al. [23] occupies yet another point in this space, addressing *beyond-database questions*: information needs that are partially grounded in an existing relational database but cannot be answered from its content alone because some columns or tables are absent. Their system, HQDL, expands the database schema by prompting the LLM to generate the missing attribute values, which are then materialized and joined with the existing data via standard SQL. Both SwellDB and HQDL differ from the setting considered in this work along the same critical axis: in both cases an external structured or semi-structured source anchors the query, and the LLM fills gaps in that source. In the Galois system and in our analyses, no such anchor exists. Every tuple must be elicited from the model's internal knowledge, and the quality ceiling of the result is determined entirely by what the model has internalized, with no possibility of grounding retrieved values against a curated record.

Robustness to query variation is a second dimension along which the existing literature leaves a gap. In semantic parsing and Text-to-SQL, prior work has shown that models are highly sensitive to perturbations in natural language inputs or table representations, and that strong benchmark scores do not necessarily imply

robust behavior under realistic reformulations [24, 25]. More recent work has argued for evaluation protocols that systematically generate semantically equivalent but linguistically diverse formulations in order to isolate robustness to variation itself [26]. This concern applies equally and more acutely when the LLM is not translating natural language into SQL for an existing database, but is itself the source from which structured results must be extracted: if parametric retrieval is sensitive to surface reformulation, that instability propagates directly into the result table.

What remains missing, therefore, is a study that combines database-mediated LLM querying with systematic families of semantically equivalent query variants, multi-model evaluation, and a large-scale benchmark built from a CH aggregation platform. No prior work in digital libraries or CH addresses this combination. This is precisely the gap the present work fills.

4 Data Collection

This section describes the construction of the two benchmarks used in this work. The first extends the query space of the benchmark introduced in [12], while keeping the underlying data unchanged. The second introduces a new benchmark derived from Europeana metadata through a dedicated acquisition, recomposition, cleaning, and packaging pipeline. In both cases, the goal is to create new robust benchmarks to test the extraction of knowledge from LLMs in the CH domain.

4.1 Paintings Extended

The *Paintings Extended* benchmark is grounded in the Famous Paintings dataset from Kaggle, [3] a tabular collection originally released for SQL practice over data about paintings, artists, and museums. We reinterpret this resource as a proxy art catalogue and derive from it a relational schema, a curated ground truth, and a controlled set of information needs representative of typical CH querying scenarios.

[3] https://www.kaggle.com/datasets/mexwell/famous-paintings

The underlying data are organized into seven tables. The three core tables targeted by evaluation are `artist` (421 rows, 9 attributes), `museum` (57 rows, 9 attributes), and `work` (14,776 rows, 5 attributes). The `artist` table records the full name decomposed into first, middle, and last name, along with nationality, stylistic label, and birth and death years. The `museum` table describes each institution by name, city, state or region, country, postal code, phone, and URL. The `work` table links paintings to their creators and hosting institutions through foreign keys (`artist_id`, `museum_id`) and records a title and a stylistic label. Four additional tables are part of the exposed schema: `subject` (6,771 rows), which associates works with iconographic subjects from a controlled vocabulary of 29 values; `canvas_size` (200 rows) and `product_size` (110,347 rows), which record physical dimensions and sale prices; and `image_link` (14,775 rows), which maps works to digitized artifact URLs. These tables are not targeted by the evaluation queries in this study.

A key design property of this benchmark is the strict separation between the logical schema and the model's parametric knowledge. The Kaggle data serve exclusively as the source of ground-truth answers: data instances are not exposed to the LLM at query time. From the system's perspective, the database is empty and all facts must be elicited from the model through scan operators. This setup mirrors a scenario in which an institution wishes to prototype a structured access layer over an LLM without committing to a full catalog ingestion. The dataset is intentionally oriented toward relatively well-known entities: major Western museums and prominent artists are more likely to be represented in a general-purpose model's training data, making this a favorable but non-trivial setting for factual recall. At the same time, the presence of multiple artists within the same stylistic movement and multiple museums within the same country creates disambiguation challenges that expose the limits of unstructured retrieval.

The evaluation query space covers fourteen information needs reflecting recurring CH patterns: filtering by artist nationality, birth or death year range, and stylistic label; retrieving museums by country, city, or state; and joining works with their authors. Each family is instantiated into ten parameterized formulations obtained by varying

only the `WHERE` clause constants while keeping the relational structure, projections, and join topology fixed, yielding 140 queries in total. Ground-truth cardinalities range from 1 to 81 tuples, with a mean of approximately 12 rows per query. The full benchmark package, including the relational schema, the SQL ingestion script, and the query files, occupies approximately 13 MB on disk.

4.2 Europeana

The second benchmark was built from Europeana, a large scale aggregation platform for CH data [13]. To acquire the data, we used the Europeana Search API, which provides JSON responses over CH records and supports deep pagination through cursor based iteration. [4] In our acquisition pipeline, requests were sent to the `search.json` endpoint with a wildcard query. The script also stored the cursor state after each successful response, so that interrupted runs could be resumed without restarting the crawl.

The acquisition process is illustrated in Figure 2. The left block represents the first stage of the pipeline. At this stage, the JSON responses returned by the API were flattened into row oriented records. The flattened records were then written into chunked CSV files, each containing a fixed number of rows. This chunk based design was adopted to make the harvesting process robust, resumable, and easier to manage at scale.

The central block of Figure 2 corresponds to the offline cleaning and normalization phase, which was applied after chunk recomposition. In our implementation, the chunk files were first merged into an intermediate CSV dataset. This intermediate version still preserved fields that were useful during harvesting but unnecessary for benchmark querying. We therefore performed a column pruning step, removing identifiers, API links, preview and landing page fields, timestamps, and rights information. Concretely, the columns `id`, `guid`, `link`, `rights`, `edmPreview`, `edmIsShownAt`, `edmIsShownBy`, `timestamp created`, and `timestamp update` were discarded.

After column pruning, we applied value normalization to the remaining fields. Since several metadata fields returned by Europeana contain repeated or multilingual values concatenated with the separator "—", we normalized each value by trimming whitespace and retaining only the first segment before the separator. For each affected entry, variations were either identical copies of the same string or, at most, linguistic or orthographic variants referring to the same underlying value, rather than distinct metadata values. Retaining only the first segment was therefore a deliberate benchmark-design simplification introduced to obtain a regular single-valued table for controlled query evaluation; it was not intended as a lossless representation of the original multilingual metadata.

The final cleaned Europeana table contains the eight columns `title`, `creator`, `type`, `year`, `language`, `country`, `provider`, and `dataProvider`. The resulting benchmark package occupies about 8.4 GB on disk and includes a main CSV file with 66,232,921 lines, together with the schema file, the SQL ingestion script, the SQL query file, and the natural language query file. The DuckDB snapshot used to construct the reference answers contains 64,990,187 loaded rows.

4.3 Query Design and Parametrization

The query design process follows the same high level philosophy in both benchmarks. For each dataset, we manually defined fourteen reference query families aimed at reflecting meaningful structured information needs over the available attributes. In the case of *Paintings Extended*, the families continue the logic of the original benchmark, targeting artists, works, and museums. In the case of Europeana, the families were designed after manual inspection of the cleaned metadata, with the explicit goal of selecting parameter combinations that guarantee non empty and semantically meaningful answers.

Once the fourteen reference queries had been defined, each family was expanded into ten total formulations, namely the original query plus nine additional variants. The additional variants were obtained with the aid of `Claude Sonnet 4.5` LLM, but only at the level of parameter suggestion. The logical structure of the SQL query was never altered. In particular, the projections, joins and grouping structure remained fixed within each

[4] https://europeana.atlassian.net/wiki/spaces/EF/pages/2385739812/Search+API+Documentation

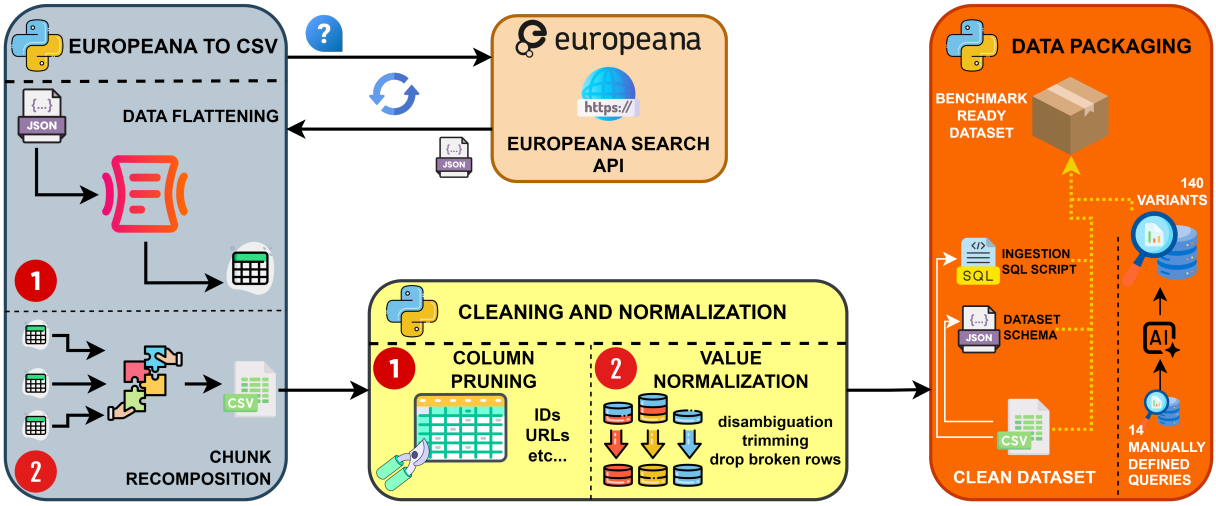

Fig. 2 Pipeline used to construct the Europeana benchmark. The process starts from iterative requests to the Europeana Search Application Programming Interface (API), whose JSON responses are flattened into chunked Comma Separated Value (CSV) files. These chunks are then recomposed into an intermediate dataset, which is processed through offline cleaning and normalization. The resulting clean dataset is packaged together with the ingestion script, schema description, and manually defined query files to obtain the final Galois ready benchmark.

family, while only the concrete values in the `WHERE` clause were changed. This design was adopted to isolate robustness with respect to parameter variation rather than logical reformulation.

This strategy yields 140 queries for *Paintings Extended* and 140 queries for *Europeana*. The resulting query families cover both simple lookups and more structured patterns involving temporal filters, conjunctions of categorical constraints, and aggregative requests. Detailed descriptions of the fourteen query families for the two datasets are reported in Table 3 and Table 4. These tables summarize the semantic intent of each family together with its main SQL characteristics, and provide the query level organization used later in the analysis of category wise performance.

5 Experimental Setup

All experiments were executed through our Python implementation of Galois[5], wrapped by a bash orchestration script that controls the dataset, execution mode, provider, output directory, and confidence threshold. The same software stack was used across all experiments and across all execution modes, namely `nl`, `sql`, `galois_wo`, `galois_s`, `galois_a`, and `galois_f`. For each query, the runner stores a structured JSON output containing the result set, the end-to-end execution time, and the total number of consumed tokens, together with a log file describing the plan and the scan level interactions.

5.1 Models and inference backends

The main evaluation focuses on three LLM deployments. GPT-4o-mini was accessed through an Azure-hosted deployment, while Llama 3 70B and Granite 3.3 8B were executed locally through Ollama. These three deployments constitute the primary experimental setting used in the main plots, quality-profile table, category-wise heatmaps, and quality–cost analysis.

We also evaluated two additional commercial GPT-5 deployments, namely GPT-5.2 and GPT-5.4, again through Azure-hosted deployments. These runs are reported separately in Section 6.6. The reason for this separation is methodological rather than merely presentational: the GPT-5.2 and GPT-5.4 deployments showed a qualitatively different behavior, characterized by frequent empty outputs and interactions with provider-level filtering mechanisms. Including them directly in the main comparative plots would therefore

[5] https://github.com/mircocazzaro/py-galois

Label	Description	SQL peculiarities
q1	Museum by country	selection + projection + join
q2	Artist nationality, birth range	selection + range filter + join + projection
q3	Artist style and nationality	selection + join + projection
q4	Museum in a USA state	selection + join + projection
q5	Museum country and city	selection + join + projection
q6	Artist style, birth before year	selection + range filter + join + projection
q7	Artist nationality, death after year	selection + range filter + join + projection
q8	Work style and museum	multi-table join + selection + projection
q9	Works by artist last name	join + equality condition + projection
q10	Work, museum country, artist nationality	multi-table join + multiple selections + projection
q11	Work subject and style	multi-table join + conjunctive selection + projection
q12	Work style and sale price	join + selection + numeric attribute projection
q13	Work image, style, museum country	multi-table join + selection + projection
q14	Artist, work subject, nationality	multi-table join + conjunctive selection + projection

Table 3 Query dictionary for the Paintings Extended benchmark. The table summarizes the fourteen manually defined query families inherited from the original paintings benchmark and extended in this work through parameter variation. For each family, we report its semantic intent and its main SQL characteristics. Each family was expanded into ten total query formulations, yielding 140 queries overall.

Label	Description	SQL peculiarities
q1	Image by country	selection + projection
q2	Text by language	selection + projection
q3	Country items since 1900	selection + range filter + projection
q4	Country image provider in the 19th century	multiple selections + range filter + projection
q5	Top countries by language	selection + grouping + counting + ordering
q6	Top providers by country	selection + grouping + counting + ordering
q7	Image by language with data provider	conjunctive selection + projection
q8	Top years by country	selection + grouping + counting + ordering
q9	Pre-1800 creator items by country	selection + range filter + projection
q10	Top languages by country	selection + grouping + counting + ordering
q11	Top creators by country	selection + grouping + counting + ordering
q12	Recent items by language	selection + ordering/range filter + projection
q13	Image by period	selection + historical period selection + projection
q14	Provider equals data provider by country	equality predicate + selection + projection

Table 4 Query dictionary for the Europeana benchmark. The table summarizes the fourteen manually defined query families used in the benchmark, reporting for each family its semantic intent and its main SQL characteristics. Each family was later expanded into ten total query formulations through parameter variation, yielding 140 Europeana queries overall.

conflate the comparison among querying strategies with a deployment-specific failure mode.

The implementation uses a unified abstraction layer for LLM access. Azure-hosted GPT models are queried through the official `openai` Python client against Azure OpenAI[6] deployments. The local models are accessed through

[6] https://learn.microsoft.com/en-us/azure/foundry/openai/how-to/responses

Ollama[7] using its OpenAI-compatible API[8] surface, which allowed us to reuse the same message-based interaction pattern across local and remote backends. This design choice keeps the Galois logic independent from the specific provider and makes the execution pipeline reproducible across heterogeneous inference settings.

The local experiments on Ollama were run on a Mac Studio equipped with an Apple M3 Ultra chip, a 32-core CPU with 24 performance cores and 8 efficiency cores, an 80-core GPU with Metal 3 support, and 512 GB of unified memory. This machine hosted the local Ollama server used for the Llama and Granite models, while the Azure-based GPT models were cloud-based. As a consequence, the reported execution times for local and remote backends include different sources of overhead. Local runs include on-device inference time, whereas Azure runs include remote service latency in addition to local orchestration. In all cases, however, the reported time corresponds to the full end-to-end runtime measured by the client from request submission to parsed output availability.

This split allowed us to compare the behavior of the Galois execution modes across both a proprietary remotely hosted model and locally served open models in the main evaluation, while still documenting the behavior of additional commercial deployments in a dedicated extended analysis.

5.2 Decoding parameters and execution settings

To reduce stochastic variability and make the comparison across models and modes more stable, all experiments were run with deterministic or near deterministic decoding settings. In particular, temperature was fixed to 0.0 throughout the experiments. When supported by the backend, we also fixed `top_p` to 1.0 and set a seed equal to 7. The confidence threshold controlling the physical choice between *KeyScan* and *TableScan* was fixed to $\tau = 0.6$, following the value adopted in the original Galois study.

The system interacts with the models through JSON oriented prompts. For the baseline `nl` and `sql` modes, the model is asked to directly return the final result set as a JSON array of records matching the expected output schema. For the Galois modes, the same JSON discipline is enforced at scan level, so that intermediate tuples can be parsed and materialized before the final relational operations are executed locally. In the local Ollama setting, requests are sent through the OpenAI compatible `/v1/chat/completions` endpoint. In the Azure setting, GPT based models are queried through Azure OpenAI deployments using the official Python client and the Responses API abstraction. As an auxiliary sensitivity check, we also examined the behavior of GaloisF under a complete sweep of τ from 0.0 to 1.0, in deltas of 0.1, using GPT-4o-mini on the Geo subset of SPIDER[9].

The runner was invoked from the command line by specifying the data root, the dataset list, the provider, the execution mode, the threshold τ, and the output directory. For each model and each benchmark, the same query set was executed independently in all the considered modes. This ensured that the only factors varying across runs were the model backend and the Galois execution strategy.

5.3 Implementation details

The experimental software was developed in Python and relies primarily on the `openai` client library for remote GPT deployments, on `requests` for HTTP based local inference through Ollama, and on DuckDB [27] based post processing to execute the final relational step over the tuples collected during the scan phase. This preserves the core Galois logic of separating LLM facing tuple acquisition from deterministic query evaluation.

6 Results

In this section, we analyze the behavior of the six querying modes across the two benchmarks. The main comparative analysis focuses on the three model deployments, namely GPT-4o-mini, Granite 3.3 8B, and Llama 3 70B. Two additional Azure-hosted deployments, GPT-5.2 and GPT-5.4, were also evaluated; however, their runs exposed a distinct deployment-related failure

[7] https://docs.ollama.com/api/openai-compatibility
[8] https://github.com/openai/openai-python
[9] https://yale-lily.github.io/spider

mode dominated by empty outputs and content-filter interactions. For this reason, they are discussed separately in Section 6.6.

The goal of the main analysis is to identify the strongest average configuration and to understand how quality varies across querying strategies, query families, datasets, and cost regimes. To this end, we report four complementary views of the parametric-only results: overall *AVG-Score*, the three underlying quality dimensions, category-wise behavior, and quality versus token consumption. Finally, we inspect the failure pattern of GPT-5.2 and GPT-5.4 deployments.

The three base metrics are adopted from the original Galois evaluation [8, 12] and operate at different levels of granularity over the result table.

F1-Cell treats each table as a set of unique, normalized cell values and computes the standard F1 score between the ground-truth cell set G and the predicted cell set P, ignoring row and column position. Precision is $|G \cap P|/|P|$ and recall is $|G \cap P|/|G|$; F1-Cell is their harmonic mean. The metric captures how many individual factual values in the expected result appear in the system output, penalizing both omissions and spurious values. Because it abstracts away row structure, a correct cell contributes equally whether or not the tuple it belongs to is fully reconstructed.

Cardinality measures the ratio between predicted and ground-truth row counts:

$$\text{Cardinality} = \frac{\min(|E|, |A|)}{\max(|E|, |A|)},$$

where $|E|$ is the number of expected tuples and $|A|$ is the number of returned tuples. The score equals 1 when the system returns exactly the right number of rows and decreases toward 0 as the discrepancy grows, regardless of which specific tuples were retrieved.

Tuple Constraint operates at the full-row level. After row normalization, it checks for each distinct ground-truth tuple whether it appears in the system output with the correct multiplicity, and reports the fraction of ground-truth tuples for which the multiplicity matches exactly. Unlike *F1-Cell*, this metric requires all attribute values of a row to be simultaneously correct: a tuple in which even one value is wrong or missing does not count toward the score.

The *AVG-Score* is the arithmetic mean of *F1-Cell*, *Cardinality*, and *Tuple Constraint*, providing a single summary indicator of table-level retrieval quality. Error bars in the grouped bar charts represent the standard error computed over the 140 query instances of each benchmark:

$$SE = \frac{\sigma}{\sqrt{n}}, \qquad \sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2},$$

where x_i is the *AVG-Score* of the i-th query instance, $\bar{x}$ is the corresponding mean, and $n = 140$.

These metrics quantify agreement with a collection-derived reference answer. They do not, by themselves, distinguish a hallucinated claim from a correct real-world fact that is absent, differently normalized, or incorrectly represented in the reference data. We therefore complement the aggregate evaluation with a manual disagreement audit.

6.1 Error Analysis and Factual Reliability

We performed a stratified manual audit of 60 unique claims returned by the three primary models but not accepted as exact or tolerance-based matches by the benchmark evaluator. The sample was fixed before manual verification and contains 20 cases per model. Within each model, it is balanced across the two datasets (10 Paintings Extended and 10 Europeana cases) and across execution families (five direct NL/SQL and five Galois cases per dataset). Identical claims returned by more than one strategy were deduplicated so that repeated outputs did not receive disproportionate weight. Each case was checked against the executed query and the corresponding reference rows; where the disagreement concerned real-world identity, attribution, location, or date, we also consulted authoritative institutional or collection sources.

To avoid treating every mismatch as a hallucination, we used five mutually exclusive categories: (i) *verified model error*, covering wrong factual relations, violated query constraints, and incorrect aggregate values; (ii) *unsupported generation*, covering placeholders and claims for which

no credible support could be found; (iii) *externally valid but outside the reference*, covering real entities or relations absent from the benchmark snapshot; (iv) *reference or normalization artifact*, covering malformed rows, aliases, title granularity, ontology differences, and Europeana provenance-role mismatches; and (v) *indeterminate*, when the available evidence did not support a reliable assignment. The first two categories are model-originated failures; the next two are reference- or scope-originated disagreements.

Table 5 shows that 35 of the 60 audited disagreements (58.3%) were model-originated failures: 27 were verifiable factual, relational, constraint, or aggregate errors, and eight were unsupported or placeholder-like generations. Conversely, 21 cases (35.0%) were attributable to reference coverage or representation rather than to an unequivocally false claim: 11 described externally valid entities or relations outside the benchmark reference, and 10 arose from reference-data or normalization artifacts. Four cases (6.7%) remained indeterminate. Thus, the strictest hallucination-like category—unsupported generation—accounts for eight audited cases, whereas the broader class of model-originated failures also includes plausible-looking but demonstrably wrong values and relations.

The two datasets exhibit different qualitative profiles in this balanced sample. In Paintings Extended, 18 of 30 disagreements were reference- or scope-originated. Examples include valid artists and museums omitted from the finite catalogue, name variants such as "J.M.W. Turner" versus the expanded benchmark form, and a malformed museum row in which the address *Palace Square, 2* shifts the State Hermitage Museum's city value to "2", thereby penalizing the factually correct answer "Saint Petersburg". In Europeana, by contrast, 23 of 30 cases were model-originated failures, mostly wrong counts in top-k aggregation queries, language labels that did not follow the benchmark's code representation, incorrect dates such as assigning 1980 to *The Scream*, and unsupported title–provider combinations. Europeana also exposed provenance-specific ambiguity: a culturally correct institution–work association may still disagree with the snapshot because `provider` and `dataProvider` encode distinct aggregation roles.

These observations change how the quantitative scores should be read. F1-Cell, Cardinality, Tuple Constraint, and AVG-Score remain valid measures of reproducibility against the released benchmark snapshots, but they should not be interpreted as direct estimates of absolute factual truth. A rejected tuple may be false, unsupported, outside collection scope, or represented differently in the reference. At the same time, the audit confirms that many mismatches are genuine reliability failures, particularly for exact aggregates and multi-attribute provenance relations. Consequently, parametric-only answers should not be presented as authoritative cultural-heritage records without source-level verification.

6.2 Overall comparative performance

The first experiment provides the broadest picture of the main results. The grouped bar charts in Figure 3 compare the average performance of all methods on the two benchmarks for the three primary models, while the error intervals indicate how stable those averages are across the 140 query instances. This analysis isolates the two main dimensions of the study, namely *model choice* and *querying strategy*, and makes it possible to see whether the same ranking remains stable across datasets.

The paired analysis refines the interpretation of the numerical gaps shown in Figure 3. For GPT-4o-mini, NL significantly outperforms SQL on both benchmarks. No statistically significant difference between NL and SQL was detected for Granite 3.3 8B or Llama 3 70B on either benchmark. Likewise, no statistically significant difference between GaloisA and GaloisF was detected for any primary model on either benchmark. The corresponding gaps should therefore be interpreted as numerical similarities rather than as evidence of superiority or statistical equivalence.

On *Paintings Extended*, direct prompting baselines dominate. Llama 3 70B reaches an AVG-Score of 0.685 with NL and 0.678 with SQL, while Granite 3.3 8B reaches 0.680 with NL and 0.670 with SQL. GPT-4o-mini follows the same broad trend, although with lower absolute values: NL reaches 0.616 and SQL 0.540. In all three cases, the best Galois variants remain below the direct baselines. GaloisA and GaloisF reach 0.402 for

Table 5 Manual classification of 60 unique benchmark disagreements. Values are counts; percentages are reported for the overall row. The stratified sample is descriptive and is not an estimate of corpus-wide error prevalence.

Model	Verified error	Unsupported	Valid outside reference	Reference/ normalization artifact	Indeterminate	Total
GPT-4o-mini	9	5	3	3	0	20
Granite 3.3 8B	12	2	2	2	2	20
Llama 3 70B	6	1	6	5	2	20
All	27 (45.0%)	8 (13.3%)	11 (18.3%)	10 (16.7%)	4 (6.7%)	60

Llama 3 70B, 0.332 for Granite 3.3 8B, and 0.441 and 0.430 respectively for GPT-4o-mini. Thus, on this benchmark, the strongest models appear to benefit more from answering the complete information need directly than from decomposed tuple acquisition followed by deterministic relational processing.

On *Europeana*, the overall ranking changes in an interesting way. NL remains the best method for GPT-4o-mini, Granite 3.3 8B, and Llama 3 70B, reaching 0.542, 0.528, and 0.535 respectively. Direct SQL is numerically close to NL for the two open models, with no statistically significant difference detected, reaching 0.527 for both Granite 3.3 8B and Llama 3 70B. However, GaloisA and GaloisF become much more competitive on this benchmark for some models. GPT-4o-mini reaches 0.467 with both GaloisA and GaloisF, reducing the gap from the NL baseline to 0.075. Llama 3 70B reaches 0.440 with both GaloisA and GaloisF, reducing the gap from the NL baseline to 0.095. This makes Europeana the setting in which Galois comes closest to recovering the performance of direct prompting.

Two conclusions can already be drawn from this comparison. The first is that direct NL prompting is a remarkably strong baseline across both benchmarks, and no statistically significant difference between direct NL and SQL is detected for the strongest open models. This differs from the pattern emphasized in the original Galois study, where the strongest structured configuration clearly outperformed weaker baselines [8]. The second is that the relative value of Galois depends on the benchmark structure. Galois remains clearly behind direct prompting on the richer multi-table *Paintings Extended* benchmark, but narrows the gap substantially on the flatter Europeana metadata benchmark.

Take home message 1. Direct NL prompting remains the strongest and most robust baseline across the main experiments. Direct SQL becomes nearly equivalent on the strongest open models, while GaloisA and GaloisF become competitive mainly on the flatter Europeana benchmark.

6.3 Quality profile decomposition

The previous experiment is intentionally aggregated, since the AVG-Score merges three different dimensions of quality. The quality-profile table therefore serves a different purpose. It reveals whether a method succeeds because it retrieves the right values, because it returns the correct number of tuples, or because it reconstructs complete tuples more faithfully. This decomposition is crucial, since methods with similar AVG-Score may fail in different ways. Table 6 collects the detailed quality profiles for the three primary models and both benchmarks.

On *Paintings Extended*, the strongest open models show that the dominance of NL and SQL is not accidental. For Granite 3.3 8B, NL reaches 0.966 in F1-Cell, 0.235 in Cardinality, and 0.841 in Tuple Constraint, for an AVG-Score of 0.680, while SQL is extremely close with 0.972, 0.210, and 0.828, yielding 0.670. Llama 3 70B exhibits the same pattern, with NL at 0.979, 0.217, and 0.857, and SQL at 0.977, 0.205, and 0.851. In other words, these models return many correct values and also reconstruct complete tuples with high fidelity. GaloisA and GaloisF improve over GaloisS and GaloisWO, but still lag especially on F1-Cell and Tuple Constraint. GPT-4o-mini displays a similar but softer version of the same pattern, with NL strongest on F1-Cell and Tuple Constraint, and SQL slightly ahead only on Cardinality.

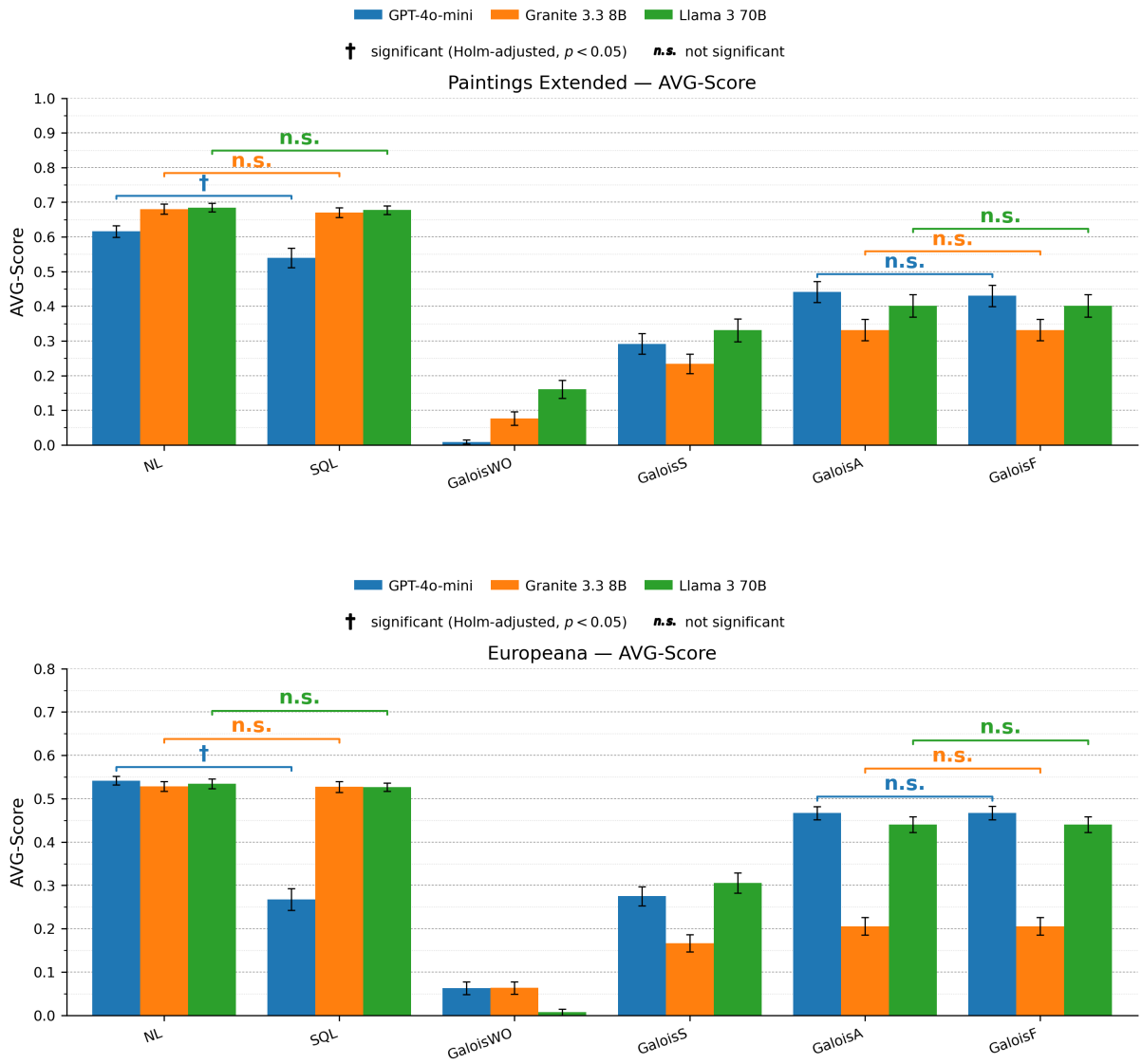

Fig. 3 Overall AVG-Score for the six querying modes on the two benchmarks for the three primary models. The top panel reports results on *Paintings Extended*, while the bottom panel reports results on *Europeana*. Each bar reports the mean AVG-Score over the 140 query instances, with error bars indicating the standard error. Within each benchmark and model, brackets identify the two pre-specified comparisons: NL versus SQL, and GaloisA versus GaloisF. Statistical significance was assessed using two-sided paired t-tests over the matched per-query AVG-Scores, with Holm correction applied separately within each benchmark across its six comparisons (three models $\times$ two contrasts; $\alpha = 0.05$). The symbol † indicates a statistically significant difference after correction, whereas *n.s.* indicates that no statistically significant difference was detected.

On *Europeana*, the decomposition is even more revealing. GPT-4o-mini achieves the best AVG-Score with NL because it simultaneously attains the highest F1-Cell, namely 0.917, and the highest Tuple Constraint, namely 0.459, even though SQL slightly wins on Cardinality with 0.266. GaloisA and GaloisF come much closer than on *Paintings Extended*, with F1-Cell values of 0.845 and 0.837 and Tuple Constraint values of 0.369 and 0.375. Their main weakness is not value correctness, but lower Cardinality. This suggests that the structured Galois variants are often able to

Model	Dataset	Method	F1-Cell	Cardinality	Tuple Constr.	AVG-Score
GPT-4o-mini	*Paintings Extended*	NL	**0.933**	0.198	**0.718**	**0.616**
		SQL	0.727	**0.210**	0.682	0.540
		GaloisWO	0.014	0.001	0.012	0.009
		GaloisS	0.417	0.135	0.324	0.292
		GaloisA	0.611	0.184	0.529	0.441
		GaloisF	0.590	0.186	0.515	0.430
	Europeana	NL	**0.917**	0.249	**0.459**	**0.542**
		SQL	0.408	**0.266**	0.128	0.267
		GaloisWO	0.121	0.026	0.042	0.063
		GaloisS	0.487	0.183	0.155	0.275
		GaloisA	0.845	0.187	0.369	0.467
		GaloisF	0.837	0.189	0.375	0.467
Granite 3.3 8B	*Paintings Extended*	NL	0.966	**0.235**	**0.841**	**0.680**
		SQL	**0.972**	0.210	0.828	0.670
		GaloisWO	0.110	0.041	0.079	0.076
		GaloisS	0.332	0.110	0.261	0.234
		GaloisA	0.450	0.152	0.393	0.332
		GaloisF	0.450	0.152	0.393	0.332
	Europeana	NL	0.909	**0.256**	0.420	**0.528**
		SQL	**0.915**	0.226	**0.440**	0.527
		GaloisWO	0.114	0.028	0.048	0.063
		GaloisS	0.305	0.105	0.089	0.166
		GaloisA	0.398	0.094	0.125	0.206
		GaloisF	0.398	0.094	0.125	0.206
Llama 3 70B	*Paintings Extended*	NL	**0.979**	**0.217**	**0.857**	**0.685**
		SQL	0.977	0.205	0.851	0.678
		GaloisWO	0.218	0.083	0.181	0.161
		GaloisS	0.416	0.209	0.367	0.331
		GaloisA	0.530	0.187	0.488	0.402
		GaloisF	0.530	0.187	0.488	0.402
	Europeana	NL	0.909	**0.256**	**0.440**	**0.535**
		SQL	**0.912**	0.256	0.412	0.527
		GaloisWO	0.007	0.007	0.007	0.007
		GaloisS	0.520	0.218	0.179	0.306
		GaloisA	0.790	0.219	0.312	0.440
		GaloisF	0.790	0.219	0.312	0.440

Table 6 Quality profiles for the three primary models on *Paintings Extended* and *Europeana*. The table reports mean *F1-Cell*, *Cardinality*, *Tuple Constraint*, and *AVG-Score* for each model, benchmark, and querying mode. Bold values denote the best score within each model–dataset block.

retrieve good tuples when they succeed, but still fail to recover enough of them to fully match the benchmark outputs. Llama 3 70B shows the same phenomenon even more clearly. NL and SQL dominate F1-Cell and Tuple Constraint, but GaloisA and GaloisF still reach 0.790 in F1-Cell and 0.312 in Tuple Constraint, which explains why their overall AVG remains much closer to the strongest baselines than on *Paintings Extended*.

A second important observation from Table 6 is that GaloisWO is consistently dominated. Across both datasets and all three primary models, it is either the weakest method or very close to the weakest one. This is especially notable because GaloisWO is the variant with the highest interaction complexity, yet the quality gains expected from deeper decomposition do not materialize in these experiments. In contrast, GaloisA and GaloisF are the only Galois variants that repeatedly become competitive. Their scores are often identical or nearly identical, particularly for Granite 3.3 8B and Llama 3 70B, indicating that the confidence-guided optimizer often selects plans that behave much like the all-pushdown strategy. The auxiliary Geo sensitivity analysis, shown in Figure 4, provides additional evidence for this interpretation. Across the complete $\tau \in [0, 1]$ sweep, the mean AVG-Score of GaloisF remained within a narrow range and showed no monotonic relationship with the threshold. The highest mean was obtained at $\tau = 0.2$ with an

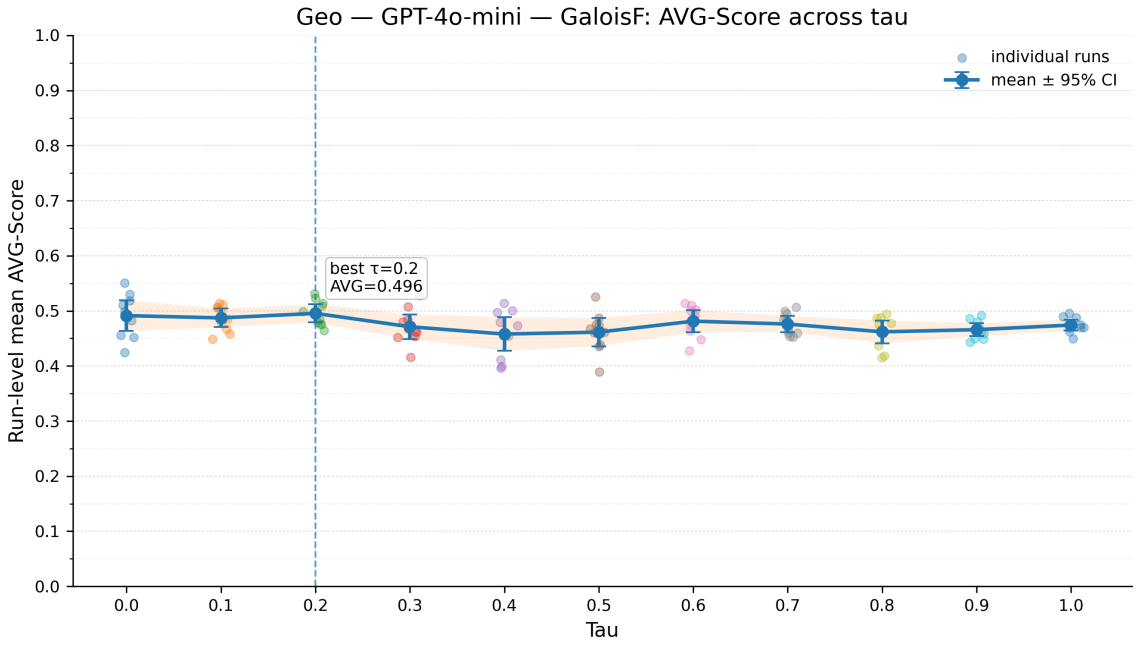

Fig. 4 Sensitivity of GaloisF to the physical-plan confidence threshold τ on the Geo subset of SPIDER, using GPT-4o-mini. Mean AVG-Score is reported for $\tau \in \{0.0, 0.1, \ldots, 1.0\}$, with shaded bands indicating 95% confidence intervals. The dashed vertical line marks $\tau = 0.6$, the value used elsewhere in the paper.

AVG-Score of 0.496, but the result at the original setting $\tau = 0.6$ remained close to it, and the 95% confidence intervals largely overlapped across threshold values. Therefore, at least in this diagnostic setting, changing the physical scan threshold does not materially alter effectiveness. This suggests that the frequent similarity between GaloisA and GaloisF is unlikely to be explained primarily by the particular choice $\tau = 0.6$. A more plausible explanation is that the confidence estimates frequently lead GaloisF to select logical plans that are identical or very close to the aggressive all-pushdown plans used by GaloisA. Since this sensitivity check was conducted on Geo rather than on the two CH benchmarks, however, it should be interpreted as supporting diagnostic evidence rather than as a direct robustness result for the present datasets.

Take home message 2. Among the structured strategies, only GaloisA and GaloisF become meaningfully competitive. GaloisWO is consistently dominated, and its additional interaction cost does not translate into better quality.

6.4 Category-wise behavior across query families

The previous analyses average over all 140 query instances of each benchmark. This is informative, but it also hides which semantic families are easier or harder. The category heatmap grid in Figure 5 addresses this issue by grouping the ten variants of each query family and displaying, for every primary model and every method, the mean AVG-Score together with its standard deviation. This view reveals whether performance differences are globally uniform or concentrated in specific classes of information needs.

Several patterns stand out. First, GaloisWO fails broadly rather than selectively. Its low scores are visible across many query families and both benchmarks, confirming that its poor overall averages are not caused by a small number of catastrophic families, but by a structural inability to sustain good performance across the benchmark. This again suggests that increasing interaction complexity through key-first decomposition is not sufficient by itself; if the scan process fails to

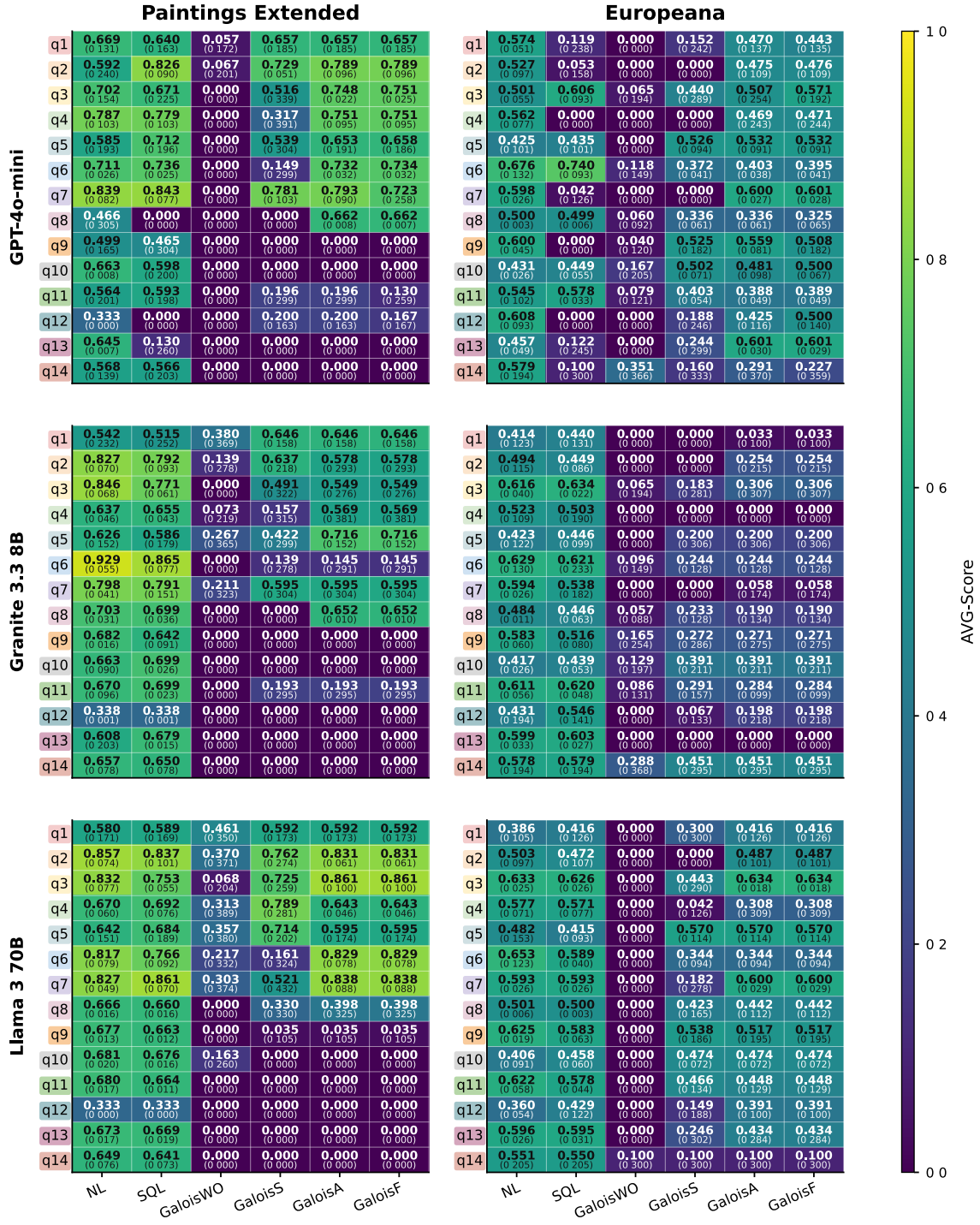

Fig. 5 Category-wise AVG-Score heatmap grid across the three primary models, datasets, and querying modes. Rows correspond to models and columns to datasets. Within each panel, rows correspond to the fourteen query families ($q1$–$q14$) and columns correspond to the six querying modes. Each heatmap cell reports the mean AVG-Score over the ten variants of the corresponding query family, together with the standard deviation in parentheses. The coloured q-labels match the query dictionaries defined for the two benchmarks.

retrieve enough correct candidates, the deterministic relational stage has little useful evidence to process.

Second, the stronger methods do not distribute their quality uniformly across categories. On *Paintings Extended*, NL and SQL are particularly stable for Granite 3.3 8B and Llama 3 70B, with high scores spread across many of the fourteen families. This is especially visible in the museum and artist-centred families, where the benchmark involves well-known entities and relatively stable attribute combinations. On Europeana, the same models remain strong, but their performance becomes more uneven for categories involving broader aggregation patterns. This is consistent with the intuition that grouping and ranking queries place stronger demands on completeness, because missing a subset of relevant tuples can strongly affect the final grouped answer.

Third, the heatmap highlights the difference between the two datasets. Europeana is structurally flatter, with most query families operating over a single large metadata table and differing mainly in the type of filtering or aggregation requested. Paintings Extended, by contrast, includes richer relational structure and many multi-table families involving artists, works, museums, subjects, prices, and images. This difference helps explain why GaloisA and GaloisF become relatively more competitive on Europeana for some models, especially GPT-4o-mini and Llama 3 70B, whereas on Paintings Extended the direct baselines remain systematically stronger. In the flatter Europeana setting, structured mediation can often recover a substantial portion of the relevant tuples, while in the richer paintings setting the baselines benefit more from end-to-end direct reasoning over well-known cultural entities.

Finally, the standard deviations printed inside the heatmap cells provide an important robustness signal. High performing methods on strong models often show moderate variability across the ten variants of the same family, whereas weaker Galois variants display much more unstable behavior. This supports the methodological value of the query-variant design itself. Had the evaluation relied on only one query per family, part of this instability would have remained invisible.

Take home message 3. The usefulness of structured mediation is heavily influenced by the semantics of the query and by the structure of the dataset. Galois narrows the gap to direct prompting much more on the flatter Europeana benchmark than on the richer multi-table Paintings Extended benchmark, where quality is distributed more unevenly across query families.

6.5 Quality versus token consumption

The final main analysis compares quality and token cost. The scatter plots in Figure 6 position every model–method pair according to its mean AVG-Score and its mean token consumption. This experiment is necessary because a method that appears competitive in quality may do so only at the expense of much higher interaction cost. Conversely, a weaker method may still be attractive if it lies on a favourable quality–cost frontier.

The plots show that the best quality–cost frontier is usually occupied by NL and SQL on the stronger models. On *Paintings Extended*, NL and SQL with Granite 3.3 8B and Llama 3 70B achieve AVG-Scores around 0.67–0.69 while remaining in the low-token region. GPT-4o-mini also offers a relatively efficient frontier position, especially in NL mode. By contrast, GaloisWO is the clearest dominated strategy. It moves to the right, sometimes far into the high-token region, without delivering competitive quality. This is particularly visible for the local open models, where GaloisWO consumes the most tokens while still staying well below the strongest baselines in AVG-Score.

The same pattern appears on Europeana, but with one important nuance. GaloisA and GaloisF for Llama 3 70B and GPT-4o-mini move closer to the frontier than they do on *Paintings Extended*. They still consume substantially more tokens than NL and SQL, but the quality they recover is now much closer to the strongest baselines, especially for Llama 3 70B. This again confirms that Europeana is the benchmark on which structured mediation becomes most promising in the current study. Nevertheless, even in this more favourable setting, the frontier is rarely won by a Galois variant. In practice, direct NL or SQL on a strong

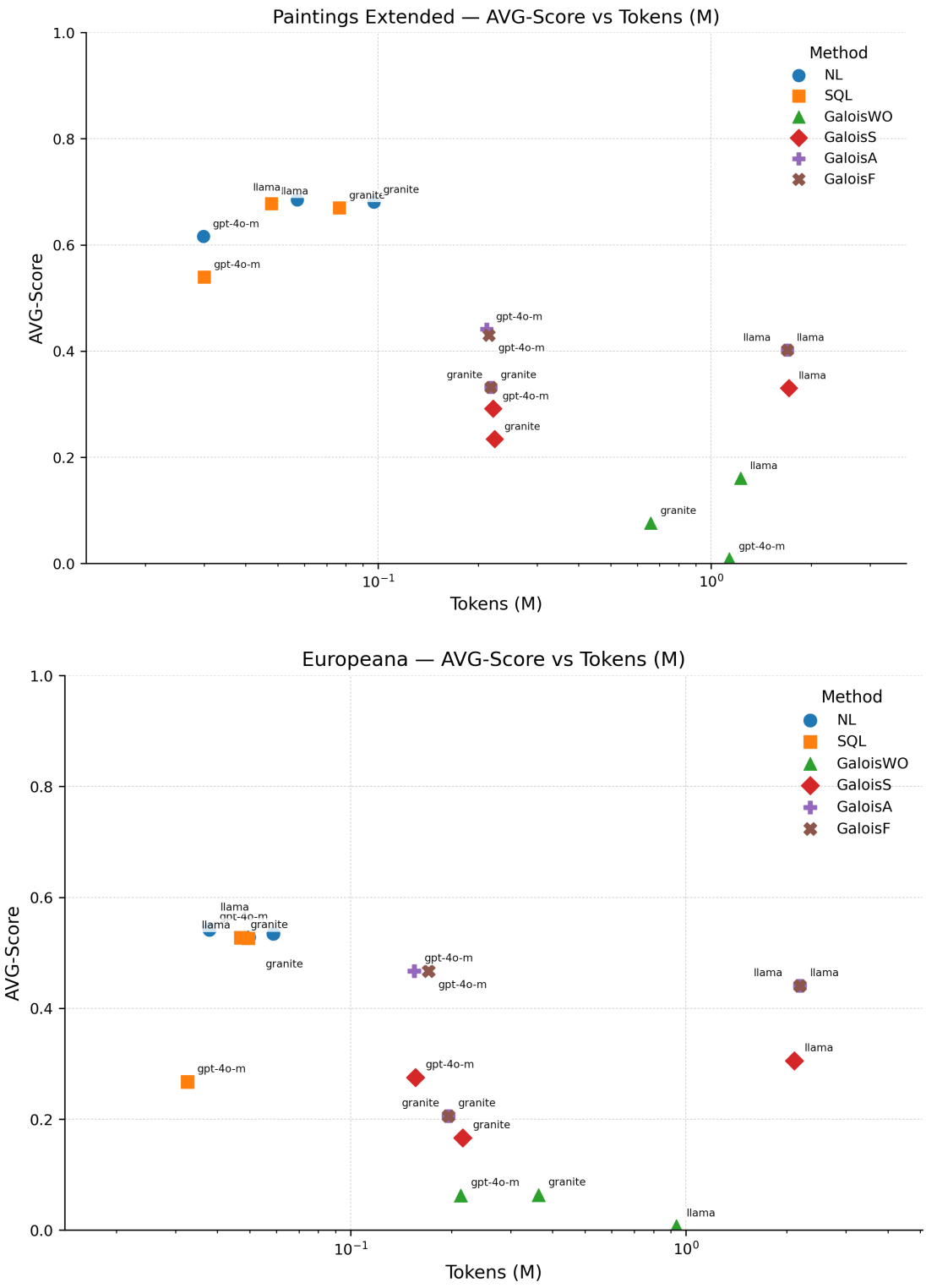

Fig. 6 AVG-Score versus token consumption for the six querying modes on the two benchmarks for the three primary models. The top panel reports *Paintings Extended*, while the bottom panel reports *Europeana*. Each point represents one model–method pair, positioned according to its mean AVG-Score and its mean token consumption in millions. Marker shape identifies the querying mode, while the adjacent label identifies the model. The figure shows the quality–cost trade-off, highlighting efficient configurations and those dominated by cheaper, more accurate ones.

model remains the most economical high-quality option in most cases.

Take home message 4. Direct NL prompting remains the strongest quality–cost baseline. Even where GaloisA and GaloisF recover competitive quality, especially on Europeana, they usually do so at a higher token cost than NL or SQL on the strongest models.

6.6 Extended experimental analysis of GPT-5 deployments

In addition to the three primary models reported in the main plots, we also evaluated two Azure-hosted GPT-5 deployments, namely GPT-5.2 and GPT-5.4. We report them separately because their behavior exposed a qualitatively different failure mode from the one observed in the main comparison. These experiments are nevertheless important, since they show that model choice alone is not the only relevant factor: the concrete deployment environment, including provider-level constraints and response behavior, can strongly affect whether structured querying succeeds.

Table 7 reports the standard quality profiles for GPT-5.2 and GPT-5.4 on both benchmarks. Table 8 complements this view with an empty-output and token-consumption diagnostic. The two tables should be read together. The quality-profile table measures how well each configuration matches the ground-truth result tables, while the diagnostic table measures whether the deployment returned any usable candidate tuples at all. This distinction is important because low AVG-Score values may arise from two different failure regimes. A model may return non-empty but noisy results, in which case low performance is caused by incorrect values, incomplete tuples, or wrong cardinalities. Alternatively, a model may return no tuples at all, in which case the evaluation score collapses because there is no candidate answer to compare with the ground truth. GPT-5.2 and GPT-5.4 frequently fall into this second regime.

The contrast with the primary models is informative. On *Paintings Extended*, GPT-4o-mini in NL mode returned empty outputs for only 4 out of 140 queries, corresponding to 2.9%, while reaching an AVG-Score of 0.616. By contrast, GPT-5.2 returned empty outputs for 121 out of 140 NL queries, and GPT-5.4 did so for 103 out of 140 NL queries. The collapse is even stronger in direct SQL and in the structured Galois modes. GPT-5.2 returned an empty result for all direct SQL queries and for all GaloisWO queries, while GPT-5.4 returned empty outputs for all GaloisWO queries and for almost all GaloisS, GaloisA, and GaloisF queries. Therefore, the weak scores of GPT-5.2 and GPT-5.4 on *Paintings Extended* are not primarily due to small factual errors or imperfect tuple reconstruction. They are largely explained by systematic empty-output behavior.

The same diagnostic pattern also appears on *Europeana*. GPT-5.2 returned empty outputs for 79 out of 140 NL queries, for all direct SQL and GaloisWO queries, and for more than 70% of GaloisA and GaloisF queries. GPT-5.4 was even more extreme: except for NL, which returned non-empty outputs for 18 queries, all other modes returned empty outputs for all 140 queries. This explains why several Europeana scores in Table 7 are close to zero despite non-negligible token consumption. In those cases, token usage does not indicate productive retrieval; rather, it often reflects interactions that terminate with empty final result sets.

This failure mode is particularly harmful in the Galois setting. In direct prompting, an empty answer directly leads to a poor final result. In Galois, however, empty outputs can also occur at intermediate scan level. When a scan over a relation returns no tuples, the downstream deterministic operators have no materialized candidates to join, filter, or aggregate. As a consequence, even simple logical queries can collapse into an empty final result. This explains why the structured modes do not rescue GPT-5.2 and GPT-5.4: the execution framework can organize and post-process retrieved tuples, but it cannot recover information when the model does not provide any candidates in the first place.

A manual inspection of the execution logs further suggests that part of this behavior may be entangled with the Azure-hosted deployment environment rather than only with model capacity. Some CH queries contain subject labels such as “Nude”, which are legitimate catalogue descriptors in art-historical metadata but may interact with deployment-level content-management mechanisms. In the affected cases, prompts involving

Model	Dataset	Method	F1-Cell	Cardinality	Tuple Constr.	AVG-Score
GPT-5.2	*Paintings Extended*	NL	0.133	0.045	0.128	0.102
		SQL	0.000	0.000	0.000	0.000
		GaloisWO	0.000	0.000	0.000	0.000
		GaloisS	0.192	0.048	0.192	0.144
		GaloisA	0.129	0.052	0.128	0.103
		GaloisF	0.139	0.055	0.126	0.107
	Europeana	NL	0.405	0.174	0.148	0.242
		SQL	0.007	0.007	0.007	0.007
		GaloisWO	0.007	0.007	0.007	0.007
		GaloisS	0.420	0.107	0.192	0.240
		GaloisA	0.246	0.081	0.090	0.139
		GaloisF	0.261	0.066	0.106	0.144
GPT-5.4	*Paintings Extended*	NL	0.261	0.082	0.250	0.198
		SQL	0.083	0.021	0.085	0.063
		GaloisWO	0.000	0.000	0.000	0.000
		GaloisS	0.007	0.003	0.007	0.006
		GaloisA	0.014	0.005	0.014	0.011
		GaloisF	0.014	0.005	0.014	0.011
	Europeana	NL	0.126	0.066	0.040	0.078
		SQL	0.007	0.007	0.007	0.007
		GaloisWO	0.007	0.007	0.007	0.007
		GaloisS	0.007	0.007	0.007	0.007
		GaloisA	0.007	0.007	0.007	0.007
		GaloisF	0.007	0.007	0.007	0.007

Table 7 Quality profiles for the two additional Azure-hosted GPT-5 deployments. The table reports mean *F1-Cell*, *Cardinality*, *Tuple Constraint*, and *AVG-Score* for GPT-5.2 and GPT-5.4 on both benchmarks. These results are reported separately from the main comparison because the two deployments exhibited qualitatively different behavior, characterized by frequent empty outputs and interactions with provider-level filtering mechanisms.

Dataset	Model	Method	Empty outputs	Non-empty outputs	Empty share	Mean tokens	Total tokens
Paintings Extended	GPT-5.2	NL	121/140	19/140	86.4%	165.4	23,151
		SQL	140/140	0/140	100.0%	151.6	21,221
		GaloisWO	140/140	0/140	100.0%	1,612.0	225,681
		GaloisS	112/140	28/140	80.0%	1,272.0	178,080
		GaloisA	122/140	18/140	87.1%	1,177.3	164,819
		GaloisF	120/140	20/140	85.7%	1,114.1	155,980
	GPT-5.4	NL	103/140	37/140	73.6%	181.5	25,406
		SQL	128/140	12/140	91.4%	162.1	22,693
		GaloisWO	140/140	0/140	100.0%	367.8	51,496
		GaloisS	139/140	1/140	99.3%	687.5	96,249
		GaloisA	138/140	2/140	98.6%	696.8	97,552
		GaloisF	138/140	2/140	98.6%	695.2	97,332
Europeana	GPT-5.2	NL	79/140	61/140	56.4%	225.2	31,531
		SQL	140/140	0/140	100.0%	164.2	22,995
		GaloisWO	140/140	0/140	100.0%	455.3	63,747
		GaloisS	77/140	63/140	55.0%	991.4	138,792
		GaloisA	102/140	38/140	72.9%	697.5	97,650
		GaloisF	101/140	39/140	72.1%	724.4	101,422
	GPT-5.4	NL	122/140	18/140	87.1%	160.0	22,397
		SQL	140/140	0/140	100.0%	166.6	23,325
		GaloisWO	140/140	0/140	100.0%	227.0	31,780
		GaloisS	140/140	0/140	100.0%	432.0	60,486
		GaloisA	140/140	0/140	100.0%	449.2	62,884
		GaloisF	140/140	0/140	100.0%	448.0	62,717

Table 8 Empty-output and token-consumption diagnostic for the two additional Azure-hosted GPT-5 deployments. For each dataset, model, and querying mode, the table reports the number of parsed final outputs whose `result_set` is an empty JSON array, the complementary number of non-empty outputs, the empty-output share, the mean token consumption per query, and the total token consumption over the 140 queries.

such labels were not handled as ordinary catalogue queries. In other cases, the model returned empty arrays or schema-shaped but semantically invalid rows, such as copying the predicate value into an identifier field. This indicates that the low scores of GPT-5.2 and GPT-5.4 deployments should be interpreted with caution: they reflect not only weak factual retrieval under the present prompting setup, but also the interaction between structured CH querying and the constraints of the specific hosted deployment.

These extended experiments reinforce the main methodological point of the paper. The quality of the underlying model is central, but so is the environment in which that model is deployed. A structured execution framework such as Galois can only operate over the tuples that the model is willing and able to return. If a deployment frequently abstains, produces empty arrays, or blocks legitimate metadata terms through guardrails, the resulting failure is not just a matter of query planning or prompting strategy: it is a compound effect involving model capability, provider configuration, and content-management policies.

Take home message 5. Model capability remains central, but deployment conditions also matter. The GPT-5.2 and GPT-5.4 runs show that structured cultural heritage querying can enter an empty-output regime when the deployed model frequently abstains, returns empty arrays, or interacts poorly with provider-level constraints.

7 Discussion

The experiments refine and extend the picture that emerged from the preliminary study. In that earlier work, direct natural language prompting had already proven much stronger than expected, while the best Galois configuration was still able to match and slightly surpass it on a smaller CH benchmark. The broader evaluation conducted here shows that this conclusion cannot be generalized uniformly. Once the analysis is extended to two datasets, three primary models, systematic query variants, and two additional GPT-5 deployments analyzed separately, direct prompting remains the dominant strategy in most stable settings, especially for the strongest open models.

A first implication concerns the role of model and deployment capability. The main results suggest that, under the current experimental setup, the underlying model is a major determinant of performance, more than the choice among several querying modes. This is visible in the clear separation between the stronger configurations, namely GPT-4o-mini, Granite 3.3 8B, and Llama 3 70B, and the weaker or unstable behavior observed in the extended GPT-5.2 and GPT-5.4 analysis. For the stronger models, the system often has enough internal factual coverage and instruction-following ability to make direct querying surprisingly effective. For weaker or more constrained deployments, no execution strategy is able to compensate for the model's limited ability or willingness to reconstruct benchmark answers. This suggests that structured orchestration can improve access to model knowledge only once a sufficient level of underlying factual competence and deployment reliability is already present.

A second implication concerns the role of direct prompting as a baseline. Across both benchmarks, direct natural language querying is not simply competitive. It is often the strongest or near strongest strategy, and direct SQL prompting becomes similarly strong for the best open models. This is an important result for two reasons. First, it confirms that recent language models can often interpret both informal and formal requests with substantial fidelity, even when the target output is tabular rather than discursive. Second, it raises the bar for database-mediated methods. In the original Galois setting, the structured execution layer was motivated by the weakness of direct baselines and by the need to enforce relational semantics over model outputs [8]. In the present study, those baselines are much stronger, which means that a structured framework must now justify itself through correctness and by offering a meaningful advantage over already capable direct interaction.

At the same time, the results do not imply that Galois has become unnecessary. Rather, they clarify where its value remains visible. Among the structured strategies, only GaloisA and GaloisF become meaningfully competitive, while GaloisWO is consistently dominated and

GaloisS provides only partial improvements. This indicates that structured mediation is useful only when the interaction pattern remains selective enough to avoid excessive prompt fragmentation. In practice, the worst performing structured variant is the one that multiplies interaction complexity without recovering enough additional accuracy. The better performing variants are instead those that preserve a tighter balance between decomposition and information density. This is a useful lesson for future database oriented querying over LLMs. More orchestration is not automatically better orchestration.

A third implication concerns dataset structure. The experiments show that the relative usefulness of Galois depends on the shape of the benchmark. On Europeana, GaloisA and GaloisF come much closer to the direct baselines than they do on Paintings Extended, especially for GPT-4o-mini and Llama 3 70B. This is likely related to the flatter nature of Europeana, where most query families operate over a single large metadata table and differ mainly in filters, grouping, and ordering. In this setting, scan-based mediation can recover a substantial portion of the relevant tuples while preserving a coherent relation level view. Conversely, Paintings Extended contains richer multi-table structure and many entity centred joins. Here the strongest models appear to benefit more from answering the whole request directly, without being forced through multiple intermediate retrieval steps. This distinction is important for digital libraries. It suggests that structured LLM querying may be more promising for broad metadata aggregations than for richer relational catalogs where entity coherence across tables is central.

A fourth implication concerns robustness. One of the main methodological contributions of this work is the move from one query per information need to ten parameterized formulations per family. The heatmaps and their within family variability show that this extension was not merely cosmetic. Some strategies that appear acceptable on average are much less stable once one inspects the dispersion across variants. Conversely, the strongest methods combine higher means with lower volatility across the ten variants of the same semantic family. This confirms that evaluation based on a single formulation can provide an overly optimistic picture of structured querying over LLMs. In the context of digital libraries, where users may express the same information need through many closely related formulations, robustness is a core property of usability.

The quality versus token analysis further sharpens the interpretation of the results. Even when GaloisA and GaloisF recover a large share of the quality of the best baselines, they usually do so at a higher token cost. This means that, under the present setup, direct natural language prompting remains the strongest quality cost option for exploratory access, while structured mediation is better viewed as a method for preserving relational discipline rather than for maximizing efficiency. This distinction matters in practice. If the goal is lightweight CH exploration over a strong model, NL prompting may already be sufficient. If instead the goal is to preserve a structured execution perspective and to control the path from predicates to tuples, then Galois still offers a principled framework, even when it is not the cheapest choice.

The near equivalence often observed between GaloisA and GaloisF is also informative. It suggests that, on these benchmarks, the adaptive planner frequently converges to decisions that are close to the more aggressive pushdown regime, or at least to plans whose downstream behavior is very similar. This may indicate that the query families used here contain many predicates for which pushdown is beneficial, or that the present confidence estimates are not yet discriminative enough to create stronger separation between the two regimes. In either case, this is not a negative result. It points to a concrete research direction, namely the refinement of the confidence catalog and of the planning heuristics so that optimization can react more sharply to dataset structure and predicate type.

These observations should also be interpreted together with the limits of the current study. The query variants change only parameter values in the `WHERE` clause and do not alter the logical structure of the queries. This was a deliberate choice, because it isolates robustness to parameter variation and avoids conflating it with logical reformulation. However, it also means that the present benchmark does not yet address broader linguistic or structural paraphrases. In addition, the experiments depend on specific deployments, providers, and prompting settings, and therefore should not

be read as universal claims about model families. The extended GPT-5.2 and GPT-5.4 deployments analysis further shows that provider-level filters and guardrails may interact with legitimate CH metadata terms, creating deployment-specific failure modes. What the study provides is therefore a comparative picture under a controlled and reproducible setup, together with evidence that both model capability and deployment conditions shape the feasibility of structured querying over LLMs. A further threat to validity concerns the status of the reference data. Both benchmarks are collection snapshots, not exhaustive factual authorities. Paintings Extended inherits omissions, naming conventions, taxonomy choices, and occasional malformed rows from its source tables. Europeana aggregates heterogeneous provider metadata and our preprocessing retains only the first value in repeated or multilingual fields; consequently, title variants, language codes, and the distinction between `provider` and `dataProvider` may affect agreement. The manual audit demonstrates that these issues are not merely hypothetical, but it is itself limited: its 60 cases were deliberately stratified across models, datasets, and execution families, rather than sampled to estimate corpus-wide prevalence. Its category counts must therefore be read descriptively, and four cases remained indeterminate. Finally, the current outputs do not provide record-level citations, so provenance was reconstructed post hoc for the audit rather than supplied by the queried model. What the study provides is a comparative picture under a controlled and reproducible setup, not a universal factuality score.

Taken together, the discussion suggests a more nuanced answer to the question posed by the paper. LLMs can indeed be queried as if they were digital libraries, but the effectiveness of this paradigm depends on three interacting factors: the underlying factual and instruction following strength of the model, the structural characteristics of the target collection, and the amount of mediation imposed by the query execution framework. The contribution of this work is precisely to make those conditions visible. Instead of asking only whether Galois works, we are now able to say more clearly when structured mediation helps, when direct prompting is already sufficient, and why these outcomes differ across datasets, models, query variants, and deployment environments.

8 Conclusion and Final Remarks

This work studied whether contemporary LLMs can be queried as if they were digital libraries, with particular attention to structured access over CH data. Starting from our preliminary study, we substantially extended both the empirical scope and the methodological strength of the evaluation. We expanded the original paintings benchmark by introducing ten parameterized formulations for each of the fourteen manually defined query families, and we introduced a second benchmark derived from Europeana metadata at larger scale. The resulting experimental setting made it possible to compare six querying modes across five models, two datasets, and multiple families of semantically equivalent queries.

The results show that the answer is neither uniformly positive nor uniformly negative. On one hand, LLMs can indeed support meaningful structured access to CH information, often with surprisingly strong performance under direct natural language prompting. On the other hand, the effectiveness of this access paradigm depends strongly on the model, the dataset, the querying strategy and the semantics of the information need. The strongest results are concentrated in GPT-4o-mini, Granite 3.3 8B, and Llama 3 70B, while weaker behavior is observed for GPT-5.2 and GPT-5.4 deployments. However, these weaker results were largely driven by deployment-specific empty-output and filtering behavior. They therefore characterize only the tested Azure deployments and must not be interpreted as a general ranking of GPT-5 model capability. Moreover, although GaloisA and GaloisF can approach the quality of direct prompting, they generally do so at a higher token cost, positioning structured mediation as a mechanism for execution control rather than efficiency. Across both benchmarks, direct natural language prompting remains the strongest overall baseline, with no statistically significant difference detected between direct NL and SQL for Granite 3.3 8B and Llama 3 70B. Among the structured strategies, only GaloisA and GaloisF become meaningfully competitive, while GaloisWO is consistently dominated and GaloisS offers more limited improvements.

A key contribution of the paper lies in showing that the usefulness of structured mediation is conditional rather than absolute. Galois narrows the gap to direct prompting much more on the flatter Europeana benchmark than on the richer multi-table Paintings Extended benchmark. This indicates that database-mediated querying over LLMs is more promising when the underlying collection resembles a broad metadata aggregation than when it requires stronger multi-relation coherence across entities and attributes. At the same time, the query-variant design reveals that robustness cannot be inferred from a single formulation per information need. Some methods that appear acceptable on average are less convincing once variability across semantically equivalent queries is taken into account.

These findings have both practical and methodological implications. From a practical perspective, they suggest that direct prompting is already a strong solution for exploratory CH access when the underlying model is sufficiently capable. From a methodological perspective, they show that structured execution frameworks such as Galois remain valuable not because they always outperform direct interaction, but because they provide a principled way to control how structured constraints are applied to an imperfect and probabilistic source. This remains important whenever the goal is to obtain an answer while also being able to reason about the execution path that leads to it.

The error audit adds an important qualification. Benchmark mismatches cannot be equated mechanically with hallucinations: in the 60 audited disagreements, 21 reflected valid out-of-reference facts or reference and normalization artifacts. Nevertheless, 35 were model-originated failures, including wrong aggregates, violated constraints, incorrect factual relations, and unsupported generations. The results therefore support exploratory use, but not unverified use as an authoritative catalogue. For high-trust applications, returned tuples require provenance and validation against collection records.

Several directions remain open for future work. A first extension would be to go beyond parameter variation and introduce broader linguistic and structural reformulations of the same logical query, thereby analyzing robustness at an even richer level. A second direction concerns the refinement of the confidence models and planning strategies used by Galois, especially in light of the frequent similarity between GaloisA and GaloisF observed in our experiments. A third direction would be to extend the evaluation to additional CH datasets and multilingual settings, which are particularly relevant for digital libraries and aggregation platforms such as Europeana.

In conclusion, this work shows that querying LLMs as if they were digital libraries is a viable and informative paradigm, but one whose success depends on carefully balancing model capability, execution strategy, information access, reference quality, and provenance requirements.

References

[1] Zhu, Y., Yuan, H., Wang, S., Liu, J., Liu, W., Deng, C., Chen, H., Liu, Z., Dou, Z., Wen, J.-R.: Large language models for information retrieval: A survey. ACM Trans. Inf. Syst. **44**(1) (2025) https://doi.org/10.1145/3748304

[2] Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., Miller, A.H.: Language models as knowledge bases? In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, pp. 2463–2473. Association for Computational Linguistics (2019). https://doi.org/10.18653/V1/D19-1250

[3] Roberts, A., Raffel, C., Shazeer, N.: How much knowledge can you pack into the parameters of a language model? In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 5418–5426. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.emnlp-main.437

[4] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Advances in Neural Information

Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, pp. 9459–9474 (2020). https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf

[5] Mallen, A., Asai, A., Zhong, V., Das, R., Khashabi, D., Hajishirzi, H.: When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 9802–9822. Association for Computational Linguistics (2023). https://doi.org/10.18653/v1/2023.acl-long.546

[6] Wang, C., Liu, X., Yue, Y., Guo, Q., Hu, X., Tang, X., Zhang, T., Jiayang, C., Yao, Y., Hu, X., Qi, Z., Gao, W., Wang, Y., Yang, L., Wang, J., Xie, X., Zhang, Z., Zhang, Y.: Survey on factuality in large language models. ACM Comput. Surv. **58**(1), 13–11337 (2026) https://doi.org/10.1145/3742420

[7] 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), 42–14255 (2025) https://doi.org/10.1145/3703155

[8] Satriani, D., Veltri, E., Santoro, D., Rosato, S., Varriale, S., Papotti, P.: Logical and physical optimizations for SQL query execution over large language models. Proc. ACM Manag. Data **3**(3), 181–118128 (2025) https://doi.org/10.1145/3725411

[9] Vasic, I., Fill, H.-G., Quattrini, R., Pierdicca, R.: Knowledge Graphs vs. Large Language Models: Competitors or Partners in Supporting Virtual Museums. J. Comput. Cult. Herit. **18**(4) (2025) https://doi.org/10.1145/3756016

[10] Hu, X.: Application of large language models for digital libraries. In: Chu, S.K., Hu, X. (eds.) Proceedings of the 24th ACM/IEEE Joint Conference on Digital Libraries, JCDL 2024, pp. 119–11192. ACM (2024). https://doi.org/10.1145/3677389.3702617

[11] Trichopoulos, G.: Large language models for cultural heritage. In: Proceedings of the 2nd International Conference of the ACM Greek SIGCHI Chapter, CHI 2023, pp. 33–1335. ACM (2023). https://doi.org/10.1145/3609987.3610018

[12] Cazzaro, M., Silvello, G.: Querying llms as if they were digital libraries. In: Proceedings of the 22nd Conference on Information and Research Science Connecting to Digital and Library Science, Modena, Italy (2026). IRCDL 2026

[13] Aloia, N., Concordia, C., Meghini, C.: Europeana: Towards the european digital library. In: Agosti, M., Esposito, F., Thanos, C. (eds.) Post-proceedings of the Fifth Italian Research Conference on Digital Libraries - IRCDL 2009, Padova, Italy, 29-30 January 2009, pp. 154–157. DELOS: an Association for Digital Libraries / Department of Information Engineering of the University of Padua (2009). http://www.dei.unipd.it/%7Eagosti/ircdl/atti-ircdl2009-finale.pdf

[14] Heinzerling, B., Inui, K.: Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In: Merlo, P., Tiedemann, J., Tsarfaty, R. (eds.) Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021, pp. 1772–1791. Association for Computational Linguistics (2021). https://doi.org/10.18653/V1/2021.EACL-MAIN.153

[15] Shi, L., Tang, Z., Zhang, N., Zhang, X., Yang, Z.: A survey on employing large language models for text-to-sql tasks. ACM Comput. Surv. **58**(2), 54–15437 (2026) https://doi.org/10.1145/3737873

[16] Sun, K., Xu, Y., Zha, H., Liu, Y., Dong, X.L.: Head-to-Tail: How Knowledgeable are Large Language Models (LLMs)? A.K.A. Will LLMs Replace Knowledge Graphs? In:

Duh, K., Gomez, H., Bethard, S. (eds.) Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pp. 311–325. Association for Computational Linguistics, Mexico City, Mexico (2024). https://doi.org/10.18653/v1/2024.naacl-long.18 . https://aclanthology.org/2024.naacl-long.18/

[17] Wang, C., Liu, P., Zhang, Y.: Can generative pre-trained language models serve as knowledge bases for closed-book qa? In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pp. 3241–3251. Association for Computational Linguistics (2021). https://doi.org/10.18653/V1/2021.ACL-LONG.251

[18] Youssef, P., Koras, O.A., Li, M., Schlötterer, J., Seifert, C.: Give me the facts! A survey on factual knowledge probing in pre-trained language models. In: Bouamor, H., Pino, J., Bali, K. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023. Findings of ACL, pp. 15588–15605. Association for Computational Linguistics (2023). https://doi.org/10.18653/V1/2023.FINDINGS-EMNLP.1043

[19] Arnold, T.B., Tilton, L.: Explainable search and discovery of visual cultural heritage collections with multimodal large language models. In: Proceedings of the Computational Humanities Research Conference 2024, Aarhus, Denmark, December 4-6, 2024. CEUR Workshop Proceedings, pp. 559–574. CEUR-WS.org (2024). https://ceur-ws.org/Vol-3834/paper1.pdf

[20] Vanallemeersch, T., Szoc, S., Meeus, L.: Ai4culture: Towards multilingual access for cultural heritage data. In: Scarton, C., Prescott, C., Bayliss, C., Oakley, C., Wright, J., Wrigley, S., Song, X., Gow-Smith, E., Forcada, M., Moniz, H.L. (eds.) Proceedings of the 25th Annual Conference of the European Association for Machine Translation (Volume 2), EAMT 2024, Sheffield, UK, June 24-27, 2024, pp. 59–60. European Association for Machine Translation (EAMT) (2024). https://aclanthology.org/2024.eamt-2.30

[21] Liu, C., Russo, M., Cafarella, M.J., Cao, L., Chen, P.B., Chen, Z., Franklin, M.J., Kraska, T., Madden, S., Shahout, R., Vitagliano, G.: Palimpzest: Optimizing ai-powered analytics with declarative query processing. In: 15th Conference on Innovative Data Systems Research, CIDR 2025, Amsterdam, The Netherlands, January 19-22, 2025. www.cidrdb.org (2025). https://vldb.org/cidrdb/2025/palimpzest-optimizing-ai-powered-analytics-with-declarative-query-processing.html

[22] Giannakouris, V., Trummer, I.: Swelldb: Dynamic query-driven table generation with large language models. In: Companion of the 2025 International Conference on Management of Data. SIGMOD/PODS '25, pp. 95–98. Association for Computing Machinery, New York, NY, USA (2025). https://doi.org/10.1145/3722212.3725136

[23] Zhao, F., Agrawal, D., El Abbadi, A.: Hybrid Querying Over Relational Databases and Large Language. In: 15th Conference on Innovative Data Systems Research, CIDR 2025. www.cidrdb.org (2025). https://vldb.org/cidrdb/2025/hybrid-querying-over-relational-databases-and-large-language-models.html

[24] Pi, X., Wang, B., Gao, Y., Guo, J., Li, Z., Lou, J.: Towards robustness of text-to-sql models against natural and realistic adversarial table perturbation. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pp. 2007–2022. Association for Computational Linguistics (2022). https://doi.org/10.18653/

V1/2022.ACL-LONG.142

[25] Saparina, I., Lapata, M.: Improving generalization in semantic parsing by increasing natural language variation. In: Graham, Y., Purver, M. (eds.) Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - Volume 1: Long Papers, St. Julian's, Malta, March 17-22, 2024, pp. 1178–1193. Association for Computational Linguistics (2024). https://aclanthology.org/2024.eacl-long.71

[26] Safarzadeh, M., Oroojlooy, A., Roth, D.: Evaluating NL2SQL via SQL2NL. In: Christodoulopoulos, C., Chakraborty, T., Rose, C., Peng, V. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025, pp. 18954–18968. Association for Computational Linguistics (2025). https://aclanthology.org/2025.findings-emnlp.1031/

[27] Raasveldt, M., Mühleisen, H.: Duckdb: an embeddable analytical database. In: Boncz, P., Manegold, S., Ailamaki, A., Deshpande, A., Kraska, T. (eds.) Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019, pp. 1981–1984. ACM (2019). https://doi.org/10.1145/3299869.3320212