

Technical Note

Not peer-reviewed version

TADI: Tool-Augmented Drilling Intelligence via Agentic LLM Orchestration over Heterogeneous Wellsite Data

[Rong Lu](#) *

Posted Date: 27 April 2026

doi: 10.20944/preprints202604.1820.v1

Keywords: agentic AI; drilling operations; tool-augmented LLM; WITSML; retrieval-augmented generation; operational intelligence; volve field



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC, OpenAlex.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

TADI: Tool-Augmented Drilling Intelligence via Agentic LLM Orchestration over Heterogeneous Wellsite Data

Rong Lu

Independent Researcher, USA; rlu@mines.edu

Abstract

We present TADI (Tool-Augmented Drilling Intelligence), an agentic AI system that transforms drilling operational data into evidence-based analytical intelligence. Applied to the Equinor Volve Field dataset, TADI integrates 1,759 daily drilling reports, selected WITSML real-time objects, 15,634 production records, formation tops, and perforations into a dual-store architecture: DuckDB for structured queries over 12 tables with 65,447 rows, and ChromaDB for semantic search over 36,709 embedded documents. Twelve domain-specialized tools, orchestrated by a large language model via iterative function calling, support multi-step evidence gathering that cross-references structured drilling measurements with daily report narratives. The system parses all 1,759 DDR XML files with zero errors, handles three incompatible well naming conventions, and is backed by 95 automated tests plus a 130-question stress-question taxonomy spanning six operational categories. We formalize the agent's behavior as a sequential tool-selection problem and propose the Evidence Grounding Score (EGS) as a simple grounding-compliance proxy based on measurements, attributed DDR quotations, and required answer sections. The complete 6,084-line, framework-free implementation is reproducible given the public Volve download and an API key, and the case studies and qualitative ablation analysis suggest that domain-specialized tool design, rather than model scale alone, is the primary driver of analytical quality in technical operations.

Keywords: agentic AI; drilling operations; tool-augmented LLM; WITSML; retrieval-augmented generation; operational intelligence; volve field

1. Introduction

The upstream oil and gas industry generates vast quantities of operational data during drilling campaigns. A single well produces hundreds of daily drilling reports (DDRs), thousands of real-time sensor records, directional surveys, mud property measurements, and formation evaluation logs. In the Equinor Volve field alone, the publicly released dataset comprises approximately 40,000 files spanning 26 wellbore sections drilled between 1993 and 2016 [1]. Extracting actionable intelligence from this corpus remains a manual, time-intensive process: a drilling engineer reviewing a 60-day well must read 60 separate reports, cross-reference activity timelines with drilling parameters, correlate operational issues with geological formations, and synthesize findings across multiple data modalities.

Existing tools address fragments of this workflow. Electronic drilling recorders capture high-frequency sensor data but provide no integration with narrative reports. Drilling analytics platforms offer parametric dashboards but require manual configuration and cannot reason about free-text operational descriptions. More recently, large language models (LLMs) have been applied to drilling data for classification [2], summarization [3], and DDR digitization [4], but these applications treat the LLM as a passive processor of pre-selected data rather than as an active agent that autonomously plans which data to retrieve and how to analyze it.

We argue that a paradigm shift is needed: from passive dashboards to *agentic reasoning systems* that autonomously navigate heterogeneous data sources, invoke domain-specific analytical tools, and

produce structured, evidence-backed answers. This paper introduces TADI (Tool-Augmented Drilling Intelligence), an agentic AI system that implements this paradigm for drilling operational intelligence. TADI combines three capabilities that existing tools keep separate: (1) structured data querying via SQL over an analytical database, (2) semantic text search over embedded drilling report narratives, and (3) multi-step reasoning through an LLM orchestrator that plans, retrieves, cross-references, and synthesizes evidence from both modalities.

The contributions of this work are:

1. **The TADI architecture:** A complete, reproducible agentic AI system for drilling operational intelligence that integrates a dual-store backend (DuckDB + ChromaDB), 12 domain-specialized tools with OpenAI function-calling schemas, and a 168-line domain-aware system prompt—all in 6,084 lines of framework-free Python code.
2. **Formal problem formulation:** We define the agent’s evidence-gathering process as a sequential tool-selection problem over a finite tool set \mathcal{T} and propose the Evidence Grounding Score (EGS) as a quantitative metric for measuring dual-source citation quality.
3. **A drilling operational question taxonomy:** A structured ontology of 130 stress test questions organized into six categories with subcategories, serving as a reusable evaluation benchmark for drilling AI systems.
4. **Evaluation framework and case studies:** Three detailed case studies (multi-phase drilling analysis, operational issue diagnosis, cross-well benchmarking), a 130-question taxonomy, and qualitative ablation/baseline analyses for stress testing the architecture.
5. **Operational Volve data integration:** Zero-error parsing of 1,759 DDR XML files and integration of selected WITSML real-time objects (161 BHA runs, 2,882 mudlog intervals, 4,217 trajectory stations, 11,134 messages), production records, formation tops, and perforations into a 12-table analytical database with 65,000+ rows and a 36,709-document semantic search index.

The remainder of this paper is organized as follows. Section 2 surveys related work across four research areas. Section 3 describes the Volve Field dataset and its challenges. Section 4 presents the TADI system architecture. Section 5 details the evaluation framework, case studies, and qualitative ablation analysis. Section 6 discusses findings, limitations, and broader implications. Section 7 concludes.

2. Related Work

TADI sits at the intersection of four active research areas. We survey each, identifying how our work extends the state of the art.

2.1. Agentic AI and Tool-Calling LLM Systems

The ReAct framework [5] established the paradigm of interleaving reasoning traces with task-specific actions, demonstrating superior performance on knowledge-intensive and interactive decision-making tasks. Toolformer [6] showed that language models can learn to call external APIs in a self-supervised manner, while Gorilla [7] advanced API-calling accuracy through retriever-aware training on over 1,600 APIs. HuggingGPT [8] demonstrated four-stage multi-tool orchestration (task planning, model selection, execution, response generation), establishing a blueprint for agent architectures. ToolLLM [9] and API-Bank [10] expanded this line of work with large-scale real-API training and evaluation corpora, while StableToolBench [11] emphasized reproducible benchmarking of large-scale tool use. Comprehensive surveys by Qin et al. [12] and Wang et al. [13] have provided unified frameworks for tool learning and autonomous agent construction, respectively. Multi-agent collaboration mechanisms have been surveyed by Chen et al. [14] and Tao et al. [15].

TADI extends this line of work by implementing tool-augmented reasoning in a domain where tool design requires deep petroleum engineering knowledge—each of our 12 tools encapsulates a specific drilling analysis algorithm (phase detection, NPT classification, difficulty indexing) that would be impractical for the LLM to derive independently within a bounded tool-calling budget.

2.2. LLMs and NLP for Drilling Operations

NLP applications to drilling data have evolved from classical text mining to LLM-powered analysis. Antoniak et al. [16] pioneered logistic regression classifiers for extracting structured information from free-text drilling risk descriptions. Hoffmann et al. [17] extended this with deep learning for sentence classification in drilling reports. Recent LLM applications include Kumar and Kathuria [3], who applied fine-tuning and prompt engineering to drilling text tasks; Yi et al. [18], who demonstrated LLM-assisted well construction planning across 200+ wells; and Ferrigno et al. [2], who deployed LLM-based WITSML classification in drilling control rooms, reducing analysis time by over 50×. Domain-specific QA systems have also begun to emerge: Bhatia et al. [4] focused on DDR digitization, Pacis et al. [19] studied zero-shot LLM question answering over drilling literature, Zhang et al. [20] built a cloud-free QA assistant over internal drilling knowledge bases, and Ogundare et al. [21] evaluated ChatGPT on practical oil and gas problems.

These works treat the LLM as a processor of pre-selected data. TADI's contribution is the agentic architecture: the LLM autonomously decides *which* data to retrieve, *which* tools to invoke, and *how* to cross-reference structured measurements with narrative evidence.

2.3. Retrieval-Augmented Generation for Technical Domains

RAG [22] addresses the limitation of LLMs' static parametric knowledge by incorporating external retrieval. Gao et al. [23] categorized RAG paradigms into Naive, Advanced, and Modular RAG, and Singh et al. [24] provided a comprehensive evolution survey. Recent adaptive retrieval work such as Self-RAG [25] has further blurred the boundary between retrieval and reasoning. HybridRAG [26] demonstrated that combining structured and unstructured retrieval outperforms either alone. The MTEB benchmark [27] established systematic evaluation of text embeddings.

TADI implements what we term *Structured-Semantic Hybrid RAG*: SQL queries over 12 DuckDB tables (structured retrieval) combined with ChromaDB vector search over 36,709 embedded documents (semantic retrieval), with a SQL keyword fallback for environments without embedding API access. This goes beyond standard RAG by enabling the agent to formulate analytical SQL queries (aggregations, groupings, window functions) alongside semantic similarity search, an increasingly important text-to-SQL capability for LLM systems [28].

2.4. The Volve Field Dataset

The Volve dataset, released by Equinor in 2018 [1], is the most comprehensive publicly available drilling dataset. Tunkiel et al. [29] provided the first systematic exploration of the real-time drilling portion, characterizing processing obstacles and data contents. Nikitin et al. [30] applied hybrid ML approaches for field development optimization, while Ng et al. [31] achieved $R^2 > 0.94$ in production forecasting using neural networks. Al-Ali et al. [32] constructed geomechanical models for the Hugin Formation, and Oloruntobi et al. [33] predicted petrophysical properties from seismic attributes. To our knowledge, TADI is the first system to integrate the operational Volve modalities used in this study—DDR XML, selected WITSML real-time objects, production records, formation tops, and perforations—into a unified, LLM-queryable analytical framework.

3. The Volve Field Dataset

3.1. Field Overview

The Volve field lies in Production License PL 046, Block 15/9 of the Norwegian North Sea, approximately 200 km west of Stavanger. Water depth in the field is approximately 91 m. The primary producing interval is the Middle Jurassic Hugin Formation, a marginal-marine sandstone at 2,700–3,100 m TVD, trapped by the overlying Upper Jurassic Draupne Formation marine shale [32]. Discovered in 1993 by the 15/9-19 S exploration well, the field produced approximately 63 million barrels of oil from the Maersk Inspirer jack-up platform between February 2008 and September 2016.

3.2. Data Sources and Formats

The released dataset [1] comprises approximately 40,000 files. Table 1 summarizes the data sources parsed by TADI and their yields.

Table 1. Data ingestion pipeline output. All 1,759 DDR XML files parsed with zero errors. Selected WITSML objects provide real-time drilling measurements across multiple wellbores.

Data Source	DuckDB Table	Rows
DDR XML (1,759 files)	ddr_status	1,759
	ddr_activities	23,447
	ddr_fluids	2,271
	ddr_surveys	1,726
	wellbore_info	1,759
WITSML Real-Time (selected objects)	witsml_bha_runs	161
	witsml_mudlog	2,882
	witsml_trajectory	4,217
	witsml_messages	11,134
Production Excel	production	15,634
Geological Data	formation_tops	409
	perforations	48
Total (12 tables)		65,447

Daily Drilling Reports. The 1,759 DDR XML files follow the WITSML 1.4.0.0 schema with namespace <http://www.witsml.org/schemas/1series> [34]. Each file encodes a 24-hour reporting period with: status information (measured depth, true vertical depth, hole diameter, 24-hour narrative summary), timestamped activities with proprietary classification codes, fluid property measurements (mud type, density, plastic viscosity, yield point), and directional survey stations. The DDR activity records use a two-level classification system (category - subcategory), with 30+ observed codes spanning productive operations (drilling-drill, cementing-casing), non-productive time (interruption-repair, interruption-waiting on weather), and well control events (well_control-kick).

WITSML Real-Time Data. Selected WITSML 1.4.1.1 files provide high-resolution drilling data: 161 BHA run configurations, 2,882 mudlog intervals with depth-indexed drilling parameters (ROP, WOB, torque, RPM, mud weight, ECD, d-exponent, gas readings, lithology), 4,217 trajectory stations, and 11,134 operational messages. Unit conversions are applied during ingestion: ROP from m/s to m/hr ($\times 3600$), WOB from N to kN ($\times 0.001$), RPM from c/s to RPM ($\times 60$), and angles from radians to degrees [35].

Production and Geological Data. Daily production records cover 7 wells from September 2007 to December 2016. Formation top picks (409 records) define the stratigraphic column, and 48 perforation intervals—all targeting the Hugin Formation—confirm the sole producing reservoir.

3.3. Data Challenges

Three challenges make the Volve dataset a rigorous testbed for AI systems:

Naming heterogeneity. Well names appear in three incompatible formats: DDR filenames use underscores (15_9_F_11_T2), WITSML headers use slash notation with prefix (NO 15/9-F-11 T2), and production data uses a third variant (15/9-F-11). TADI normalizes all names to the underscore format at ingestion via a dedicated function that strips prefixes, replaces delimiters, and collapses multiple underscores.

Sentinel values. DDR fields use -999.99 as a sentinel for missing data, while WITSML uses -999.25 and -9999 . Quality filters must also exclude physically implausible values (ROP > 200 m/hr, RPM > 300 , WOB > 500 kN).

Data density variation. The 26 wellbore sections span three decades and vary dramatically in reporting detail: the exploration-era wells (15/9-19 series, 1980s–1990s) have sparser reporting with fewer activity entries per day, while the 2007–2016 development wells (F-series) provide rich multi-source coverage. Well F-11 main bore has only 17 DDRs, whereas F-11 B has 90 and F-12 has 165.

4. System Architecture

4.1. Design Philosophy

TADI embodies three design principles aligned with the competition’s guidance that “complexity alone will not be rewarded”:

Framework-free simplicity. The system uses the raw OpenAI Python SDK for LLM interaction, DuckDB for analytical SQL, and ChromaDB for vector search. No heavyweight orchestration frameworks (LangChain, LlamaIndex) are employed. The core `ask_question()` agent loop spans 132 lines within a single 250-line orchestrator file, and the total production codebase is 6,084 lines across 33 files. This deliberate choice ensures full transparency of the agent’s decision-making process and eliminates framework-imposed abstractions between the code and the API.

Domain knowledge in tools, not models. Rather than fine-tuning a domain-specific LLM, TADI encapsulates drilling engineering knowledge in 12 deterministic tools—phase detection algorithms, NPT classification rules, difficulty indices, risk scores—that produce reproducible outputs regardless of LLM sampling variance. The LLM’s role is to *select and compose* these tools, not to independently derive drilling analysis logic within a bounded context window.

Dual-source evidence by design. The system prompt requires every answer to cite both structured drilling measurements (depths, durations, rates) and free-text DDR narrative quotes (with well name and date attribution). This expectation is reinforced at three levels: the system prompt mandates cross-referencing, the tool selection guide repeatedly routes the agent toward `get_ddr_narrative`, and an output validator warns when measurements or quoted DDR passages are missing.

4.2. Formal Problem Formulation

We formalize the agent’s behavior as a sequential tool-selection problem. Let $\mathcal{T} = \{t_1, t_2, \dots, t_{12}\}$ be the set of available tools, q be the user’s natural-language question, and s be the system prompt encoding domain knowledge and behavioral rules. At each step $i \in \{1, 2, \dots, N_{\max}\}$, the agent observes the accumulated evidence $E_{1:i-1} = \{(t_{j_k}, r_k)\}_{k=1}^{i-1}$, where r_k is the result of invoking tool t_{j_k} with arguments a_k , and selects the next action:

$$(t_{j_i}, a_i) = \pi_{\theta}(q, s, E_{1:i-1}) \quad (1)$$

where π_{θ} is the LLM policy parameterized by model weights θ . The process terminates when π_{θ} produces a terminal response \hat{y} (finish reason `stop`) instead of a tool call, or when $i > N_{\max}$ (set to 10 in our implementation). The agent’s objective is to produce an answer \hat{y} that maximizes the Evidence Grounding Score (defined in Section 5.1.1).

This formulation highlights a key distinction from standard RAG: the agent performs *iterative, adaptive retrieval* where each tool selection depends on the results of all prior steps, rather than a single retrieval pass. The bounded horizon $N_{\max} = 10$ creates a planning pressure: the agent must select an efficient tool sequence rather than exhaustively querying all data sources.

4.3. Data Ingestion Pipeline

The ingestion pipeline transforms raw Volve data files into the dual-store backend in two stages.

Stage 1: Parsing and structuring. Four specialized parsers handle the heterogeneous input formats:

- **DDR XML Parser** (315 lines): Processes 1,759 files using `lxml` with explicit WITSML namespace handling. Well names and dates are extracted from filenames via the pattern `{WELL}_{YYYY}_{MM}_{DD}.xml`. The sentinel value `-999.99` is filtered by checking $f < -990.0$. Each file yields status

records, activity records (with proprietaryCode classification), fluid measurements, and survey stations.

- **WITSML Parser** (341 lines): Traverses the nested directory structure {well}/{section}/{data_type}/*.xml, parsing four data object types (bhaRun, mudLog, trajectory, message). Applies unit conversions and quality filters.
- **Production Parser** (55 lines): Reads the Excel workbook using pandas/openpyxl, normalizing column names and well identifiers.
- **Geological Parser** (131 lines): Parses fixed-width format files for formation tops (409 records from the well pick file) and perforations (48 records).

Stage 2: Loading. Parsed DataFrames are loaded into a 12-table DuckDB database via INSERT INTO ... SELECT * FROM df. Simultaneously, 26,965 DDR text documents (activity comments, 24-hour summaries, forecasts) and 9,744 qualifying WITSML messages are embedded using OpenAI text-embedding-3-small and indexed in ChromaDB with cosine similarity, yielding 36,709 searchable documents.

4.4. Dual-Store Architecture

The architecture pairs two complementary data stores, each optimized for a different query modality.

DuckDB (structured queries) [36] serves as an in-process columnar analytical database. Its SQL interface supports the aggregate queries (GROUP BY, window functions, CTEs) that dominate the tool suite's query patterns. The database is a single file (volve.duckdb) rebuilt on each ingestion, requiring zero configuration. Tool functions open read-only connections to prevent accidental writes.

ChromaDB (semantic queries) stores text embeddings with metadata (well, date, depth, document type, activity code) enabling scoped semantic search via \$and-composed filters. A complete SQL-based keyword search fallback (LIKE queries on summary_24hr and comments columns) ensures the system functions even without an embedding API key, degrading gracefully from semantic to keyword matching.

This dual-store design implements the Structured-Semantic Hybrid RAG pattern validated by Sarmah et al. [26], where combining structured and unstructured retrieval outperforms either approach alone. The critical insight is that drilling operational questions require *both* quantitative aggregation ("What was the average ROP in the 12.25-inch section?") and narrative discovery ("What did the DDR say about stuck pipe events?").

4.5. Agent Orchestration

The orchestrator implements the iterative tool-calling loop defined in Equation 1. The implementation spans 132 lines with three layers of resilience:

Transient error retry. Rate limit (HTTP 429), server errors (500/502/503), and timeout/connection errors are retried up to 3 times with exponential backoff (2^{attempt} seconds).

Model compatibility fallbacks. If the API rejects the reasoning_effort parameter, the system permanently disables it for all subsequent calls, falling back to temperature=0.1. Similarly, max_completion_tokens falls back to the older max_tokens parameter.

Token management. Tool results are truncated to 15,000 characters. SQL queries without explicit LIMIT clauses receive automatic LIMIT 200 injection. DDR narrative retrievals cap at 15 summaries and 15 activities per call.

When invoked with the --trace flag, the orchestrator records metadata for each tool call (step number, tool name, arguments, result length, execution duration, first 500 characters of result), producing a structured evidence trace that enables post-hoc analysis of the agent's reasoning path.

4.6. Tool Suite Design

The 12 tools registered with the LLM via OpenAI function-calling schemas are summarized in Table 2. We organize them into four functional categories.

Table 2. The 12 TADI tools registered with OpenAI function-calling. Each tool encapsulates a domain-specific analysis algorithm and returns structured text evidence. Tools are dispatched by the LLM orchestrator based on question category.

Category	Tool	Function	Data Source
Data Access	query_drilling_data	Arbitrary read-only SQL on all 12 tables; auto-LIMIT injection	DuckDB
	search_daily_reports	Semantic similarity search; SQL keyword fallback	ChromaDB
Well Analysis	get_well_overview	Metadata, hole sections, formations, activity distribution	DuckDB
	get_drilling_phases	Hole-size-primary, activity-code-secondary phase detection	DuckDB
	compute_efficiency_metrics	NPT breakdown by cause, productive time, ROP by section	DuckDB
Cross-Reference	compare_wells	Side-by-side comparison of two wells across all metrics	DuckDB
	get_bha_configurations	BHA analysis with WITSML parameters and ranking	DuckDB
	identify_operational_issues	Issue categorization with multi-source root cause correlation	DuckDB
	get_field_benchmarks	5 ranking modes: progress, difficulty, gas, risk, production	DuckDB
Evidence	get_formation_context	Geological formation lookup at any depth with fallback	DuckDB
	generate_depth_time_plot	Depth-vs-time visualization with section overlays	DuckDB
	get_ddr_narrative	Attributable DDR text retrieval by date/depth range	DuckDB

Several tools deserve detailed description for their algorithmic content:

Phase Detection (`get_drilling_phases`, 273 lines) implements a two-tier algorithm. The *primary method* detects major phase boundaries from hole diameter changes in chronological DDR status records: 36"/30" = Conductor, 26" = Surface, 17.5" = Intermediate, 12.25" = Production, 8.5" = Reservoir. The *secondary method* classifies sub-phases within each hole section using a 29-entry dictionary mapping proprietaryCode values to 17 phase categories (Drilling, Tripping, Reaming, Cementing, Casing, Completion, Logging, Well Control, etc.). Depth progression validation detects reversals exceeding 10 m. Confidence assessment uses three simple tiers: HIGH (multiple hole-size phases, > 20 activities, and DDR summaries available), MEDIUM (activities and summaries available but major hole-size boundaries incomplete), LOW otherwise.

Issue Detection (`identify_operational_issues`, 349 lines) performs multi-source root cause correlation. It extracts problem activities using broad criteria (state = "problem", interruption/well-control activity codes, specific state details, and 6 well-control keywords in comments), heuristically categorizes each into one of 10 issue types via a hierarchical classifier, then cross-references each category with depth distribution, hole section, formation context (via `formation_tops`), mud properties during issue dates, and ROP context at issue depths compared to the well average. A temporal trend analysis splits the issue timeline at the midpoint, classifying trends as INCREASING (> 1.3×), DECREASING (< 0.7×), or STABLE. Statistical mud property comparison reports percentage differences between problem and normal days.

Field Benchmarks (`get_field_benchmarks`, 651 lines) implements five ranking modes. The `section_performance` mode computes a composite *difficulty index* using population z-scores:

$$D_{\text{section}} = z(\overline{\text{WOB}}) + z(\overline{\text{Torque}}) - z(\overline{\text{ROP}}) \quad (2)$$

where higher D indicates harder drilling. The *risk* mode scores wells using a weighted composite: severe comment mentions ×3 + well-control codes ×5 + fishing mentions ×2 + generic interruptions ×0.05 + perforation presence ×1, reflecting drilling engineering judgment that well-control events carry the highest operational significance.

DDR Narrative (`get_ddr_narrative`, 161 lines) solves a critical evidence gap. While semantic search (`search_daily_reports`) may fail to return relevant passages for every query, `get_ddr_narrative` uses direct SQL queries and deterministically returns DDR text whenever records exist in the specified date/depth range. The system prompt instructs the agent to call this tool at the end of every reasoning chain (“EVERY question type—ALWAYS end with `get_ddr_narrative`”), ensuring that DDR quotations with source attribution are available for inclusion in every answer.

4.7. Domain-Aware Prompt Engineering

The 168-line system prompt is structured into 12 sections that encode domain knowledge directly into the agent’s context:

1. **Role definition:** “You are a senior drilling engineer AI assistant analyzing the Equinor Volve Field dataset.”
2. **Complete schema reference:** All 12 table names and columns, embedded directly in the prompt to eliminate the need for a separate schema-lookup tool.
3. **Well naming conventions:** Underscore format rules and wells with rich WITSML coverage.
4. **Drilling domain knowledge:** Standard hole sizes and operational meaning (5 entries from 36" to 8.5"), activity code vocabulary (17 categories), Volve formations from shallowest to deepest (8 entries, with Hugin identified as PRIMARY RESERVOIR), and BHA component vocabulary.
5. **Tool selection guide:** Category-specific tool-calling sequences for each of the 6 question categories plus a global rule that every question should include `get_ddr_narrative` to retrieve attributable DDR text.
6. **Mandatory output format:** Six required sections (Answer, Evidence from Drilling Data, Evidence from Daily Reports, Reasoning, Assumptions, Confidence & Uncertainty) with explicit formatting guidance.
7. **Cross-referencing rule:** Every conclusion must cite at least one measurement from structured data AND at least one direct DDR quote.
8. **Confidence calibration:** Quantitative criteria for HIGH (> 50 DDR records, multiple independent sources), MEDIUM (data available but gaps), and LOW (< 10 records, conflicting evidence).

This approach—domain knowledge via prompt engineering rather than model fine-tuning—aligns with findings that prompt engineering combined with RAG achieves competitive performance with fine-tuned models while maintaining greater flexibility [37,38]. The key advantage for reproducibility is that the system prompt is a version-controlled text artifact, not a trained model checkpoint.

4.8. Structured Output Format

The output validator checks three aspects of every answer: (1) *section presence*—regex search for each of 6 mandatory section headers; (2) *measurement check*—regex for a number followed by any drilling unit (m, m/hr, kN, sg, g/cm³, mPa, Pa, ppm, hrs, days, %); and (3) *DDR quote check*—patterns for “DDR” followed by a well identifier and year, a date reference, or a quoted string. Missing elements trigger format warnings but do not suppress the answer, following a “warn rather than block” philosophy. These checks are intentionally lightweight compliance heuristics rather than semantic verification, consistent with broader interest in benchmarking structured LLM outputs [39].

5. Evaluation

5.1. Evaluation Framework Design

Evaluating an agentic system for drilling operational intelligence requires assessing multiple quality dimensions beyond simple answer accuracy. We evaluate TADI along four axes: (1) *evidence grounding*—do answers cite specific data from both structured and unstructured sources? (2) *reasoning quality*—are conclusions logically connected to cited evidence? (3) *domain correctness*—are drilling-specific interpretations technically sound? (4) *coverage*—can the system handle the full spectrum of operational question types? Tool-use benchmarks such as StableToolBench [11] highlight the value

of reproducible multi-step evaluation; our 130-question taxonomy plays a similar role within the narrower drilling domain.

5.1.1. The Evidence Grounding Score

We propose the Evidence Grounding Score (EGS) as a quantitative *grounding-compliance proxy* for measuring how well an agent's answer cites specific data. Let \hat{y} be the generated answer with sections $\{s_1, \dots, s_6\}$. We define:

$$\text{EGS}(\hat{y}) = \frac{1}{3} \left(\mathbb{1}[\text{has_measurement}] + \mathbb{1}[\text{has_ddr_quote}] + \frac{|\text{sections_present}|}{6} \right) \quad (3)$$

where $\mathbb{1}[\text{has_measurement}]$ is 1 if the answer contains at least one numeric value with a drilling unit, $\mathbb{1}[\text{has_ddr_quote}]$ is 1 if the answer contains at least one attributed DDR quotation (well name + date + quoted text), and $|\text{sections_present}|$ is the count of the 6 mandatory sections detected. $\text{EGS} \in [0, 1]$, with 1.0 indicating a fully grounded answer. The current shipped validator checks these constituent signals separately; EGS is the corresponding scalar summary we propose for paper-level evaluation and future large-scale benchmarking, not a substitute for human judgment of answer correctness.

5.1.2. Question Taxonomy

We developed a structured taxonomy of 130 stress test questions organized into six categories that span the full spectrum of drilling operational intelligence (Table 3). The taxonomy was designed to probe both single-well factual retrieval and multi-well analytical synthesis, with questions spanning five complexity dimensions: well scope (single vs. cross-well vs. field-wide), data sources (DDR-only vs. multi-source), time scope (single event vs. full campaign), analysis type (factual retrieval vs. root cause inference), and output type (quantitative metric vs. narrative synthesis).

Table 3. Distribution of 130 stress test questions across the 6 drilling operational question categories. The taxonomy emphasizes operational issues and synthesis, reflecting their higher analytical complexity.

Question Category	Count
Cat. 1: Phase Identification & Validation	20
Cat. 2: Time & Efficiency Analysis	21
Cat. 3: Section & ROP Performance	21
Cat. 4: BHA & Configuration Effectiveness	20
Cat. 5: Operational Issues & Root Causes	26
Cat. 6: Synthesis, Comparison & Recommendations	22
Total	130

Representative subcategories include: phase boundary detection and cross-well phase comparison (Cat. 1), NPT decomposition and drilling efficiency trends (Cat. 2), formation-level ROP analysis and drilling parameter correlation (Cat. 3), BHA run failure analysis and cross-well BHA comparison (Cat. 4), equipment reliability and well control events (Cat. 5), and best practices extraction and drilling-production integration (Cat. 6). The most challenging questions combine multiple complexity dimensions; for example, "Integrate drilling and production data for the producing wells" requires field-wide scope, multi-source data integration, and cross-domain inference.

5.2. System Metrics

Table 4 summarizes the system's quantitative characteristics.

Test coverage spans 95 test methods across 4 test files: configuration and well name normalization (18 tests), DDR XML parsing with exact count validation (16 tests), WITSML parsing with unit conversion verification (17 tests), and all 12 tools individually tested with edge cases (44 tests). The 130-question stress test suite provides a reusable integration-oriented stress corpus. Unless otherwise noted, the sub-second timings reported in the case studies refer to deterministic tool execution only and exclude LLM generation latency.

Table 4. TADI system component summary. The architecture avoids heavy frameworks in favor of direct OpenAI SDK integration with DuckDB and ChromaDB.

Component	Technology	Key Metric
LLM Orchestrator	OpenAI SDK (gpt-5.4-mini)	10 rounds max, 3 retries
Structured Database	DuckDB (in-process)	12 tables, 65K+ rows
Vector Store	ChromaDB (persistent)	36,709 documents
Embeddings	text-embedding-3-small	26,965 DDR + 9,744 WITSML
XML Parsing	lxml (WITSML 1.4)	1,759 DDR + 15 WITSML wells
Visualization	matplotlib	Depth-vs-time plots
CLI	Typer	ingest / ask / demo
Tests	pytest	95 tests, 4 modules

5.3. Case Study 1: Multi-Phase Drilling Analysis

Question: “Identify and label the major drilling phases for well 15/9-F-11 T2, including the evidence used for each phase.”

Context. Well 15/9-F-11 T2 is a sidetrack with 53 DDR reports spanning 2013-03-24 to 2013-05-15, rich WITSML data, and multiple hole sections, making it ideal for demonstrating multi-source evidence integration.

Agent execution trace (from live system run). The orchestrator, guided by the system prompt’s Category 1 tool chain, executed four tool calls in sequence:

1. `get_drilling_phases(well="15_9_F_11_T2")` — automated phase detection returned three major phases via hole diameter transitions in 0.17s (1,982 chars).
2. `get_ddr_narrative(well, date_from="2013-03-24", date_to="2013-04-14")` — 22 DDR summaries and 208 activities for Phase 1 (7,916 chars).
3. `get_ddr_narrative(well, date_from="2013-04-14", date_to="2013-04-29")` — 16 DDR summaries and 142 activities for Phase 2 (7,391 chars).
4. `get_ddr_narrative(well, date_from="2013-04-29", date_to="2013-05-15")` — 17 DDR summaries and 161 activities for Phase 3 (7,977 chars).

On the traced run, tool execution took approximately 0.2s across all four calls (excluding LLM generation). The phase detection tool identified three major phases: (1) the **26" surface section** from 306 m to 1,400 m MD (2013-03-24 to 2013-04-14, 22 DDR days, 208 activities with 53% drilling); (2) the **17.5" intermediate section** from 1,400 m to 2,907 m MD (2013-04-14 to 2013-04-29, 16 DDR days, 142 activities with 59% drilling); and (3) the **8.5" reservoir section** beginning at 2,907 m MD on 2013-04-29, reaching a maximum of 4,562 m MD before later post-drilling backtrack reports (17 DDR days, 161 activities with 68% drilling). Depth reversals on 2013-05-11 (4562 → 4104 m) and 2013-05-13 (4104 → 2569 m) were correctly identified as post-drilling operations rather than new phases.

The agent retrieved DDR narratives for each transition, yielding real evidence: DDR 2013-03-24: “Transferred over from well 15/9-F-11. Oriented drilled 26” hole from 257 m to 266 m. Observed 30 tons overpull.”; DDR 2013-04-14: “Drilled shoetrack and 3 meter new formation. Perform FIT. Drill ahead.”; DDR 2013-04-29: “RIH 8 1/2” steerable BHA on 5 1/2” DP to bottom at 2577 m. Drilled and orientated 8 1/2” hole to 2907 m.”. The final answer satisfied all six required sections and included both structured measurements and dated DDR quotations.

Design insight. The dual-method phase detection (hole-size-primary, activity-code-secondary) is more robust than either method alone. Hole sizes define unambiguous macro boundaries, while activity codes reveal the operational structure within each section.

5.4. Case Study 2: Operational Issue Diagnosis

Question: “Identify key operational issues encountered while drilling 15/9-F-11 T2 and propose likely contributing factors.”

Agent execution trace (from live system run). The agent completed this analysis in just two tool calls, with sub-second tool execution on the traced run:

1. `identify_operational_issues(well="15_9_F_11_T2")` — returned 11,870 chars identifying 119 problem/NPT activities out of 493 total (24.1%).
2. `get_ddr_narrative(well, date_from="2013-03-24", date_to="2013-05-15")` — 53 DDR summaries covering the full drilling campaign.

The issue detection tool heuristically categorized problems into: **Equipment Repair** (49 occurrences, largest category, spanning 7–2,570 m MD across all hole sections), **Weather Delay** (affecting the 17.5" and 26" sections), and **Operational Difficulty** (concentrated in the shallow 26" section). The most significant single event was a **crown block fast-line sheave bearing seizure at 454 m MD on 2013-03-25**, which forced POOH and multi-day rig maintenance. The agent cited the DDR directly: “*Found out crown block fast-line sheave bearings had failed. Prepared to displace hole to 1.35 sg KCl mud and POOH 26" BHA from 454 m.*” The statistical mud property comparison revealed mud weight was similar on problem (1.323 g/cm³) versus normal days (1.336 g/cm³), ruling out mud weight as a primary contributor. ROP on equipment repair days averaged 14.4 m/hr versus the well average of 22.1 m/hr, quantifying the efficiency impact.

Design insight. The system’s ability to simultaneously extract structured issue statistics (119/493 activities, ROP differentials) and match them with specific DDR narrative evidence (the crown block failure quote) demonstrates the value of the dual-store architecture. The statistical mud property comparison (problem vs. normal days) reveals correlations not apparent from individual reports.

5.5. Case Study 3: Cross-Well Benchmarking

Question: “Compare the drilling phase distribution of 15/9-F-11 with 15/9-F-1 C and explain key differences.”

Agent execution trace. The agent executes a layered workflow:

1. `get_field_benchmarks(mode="section_performance")` for field-wide context and difficulty indices.
2. `compare_wells(well1="15_9_F_11", well2="15_9_F_1_C")` for structured side-by-side metrics (date ranges, depth ranges, activity distributions, hole section breakdowns, production volumes).
3. `compute_efficiency_metrics` for each well individually (productive/NPT time breakdowns, ROP by section).
4. `get_ddr_narrative` for both wells at key transition dates.

This layered approach—field-wide screen, then pairwise comparison, then individual deep-dives, then narrative evidence—ensures that the comparison is contextualized within field norms. The composite difficulty index (Eq. 2) provides a principled metric for comparing section difficulty across wells with different geological and operational contexts.

Design insight. The data asymmetry between F-11 (17 DDRs, main bore only) and F-1 C (98 DDRs, rich WITSML data) demonstrates the system’s ability to produce calibrated answers: the agent assigns MEDIUM confidence for F-11 due to sparse data, while providing HIGH confidence for F-1 C conclusions.

5.6. Expected Failure Modes Under Ablation

To reason about the contribution of each major component, we define five ablation scenarios (Table 5). The effects listed here are qualitative expectations rather than executed benchmark results.

SQL-only agent. Replacing all 12 tools with a single `query_drilling_data` forces the agent to derive phase detection logic, NPT classification rules, and difficulty metrics as raw SQL queries. We predict severe degradation for Category 5 (Operational Issues) and Category 6 (Synthesis) questions, which require the most cross-referencing. Category 1 (Phase Identification) may be partially preserved since hole-size queries are straightforward SQL.

No vector store. Disabling ChromaDB forces text search to use SQL keyword fallback (LIKE queries). Semantic search should improve answer quality for questions requiring discovery of related

narratives that do not share exact keywords: a query about “stuck pipe” should find DDR comments mentioning “tight hole,” “pack-off,” or “differential sticking” via semantic similarity but not via keyword matching.

Generic prompt. Replacing the 168-line domain prompt with “You are a helpful assistant” removes tool selection guidance, domain knowledge, and output format requirements. The generic agent would make suboptimal tool selections, fail to normalize well names (using “15/9-F-11 T2” instead of “15_9_F_11_T2” in SQL), and produce unstructured answers missing required sections.

No cross-referencing enforcement. Removing the mandatory DDR quote rule tests whether the agent voluntarily retrieves narrative evidence. Without enforcement, we predict answers that rely primarily on structured data without supporting DDR quotes, directly violating the evaluation criterion of dual-source evidence.

DDR-only (no WITSML). Removing the 4 WITSML tables eliminates interval-level drilling parameters (ROP, WOB, torque, RPM per depth interval). BHA analysis falls back to DDR-estimated daily progress; gas response analysis becomes unavailable; formation-ROP correlation loses depth resolution. We predict severe degradation for Category 3 (ROP Performance) and Category 4 (BHA Effectiveness).

Table 5. Ablation scenario design. Each variant removes or degrades one component to reason about its expected contribution to answer quality across the 130-question stress test suite.

Variant	Component Removed	Expected Impact
SQL-only agent	11 specialized tools	Severe: no pre-implemented algorithms; round budget exhaustion on complex queries (Cat. 5, 6 most affected)
No vector store	ChromaDB semantic search	Moderate: keyword fallback misses semantically related passages (e.g., “stuck pipe” vs. “tight hole”)
Generic prompt	Domain system prompt (168 lines)	Severe: suboptimal tool selection, missing output sections, well name format errors
No cross-ref enforcement	Mandatory DDR quote rule	Moderate: inconsistent evidence from daily reports
DDR-only (no WITSML)	4 WITSML tables	Severe for Cat. 3–4: no interval-level ROP, WOB, RPM; only coarse daily progress available

5.7. Qualitative Baseline Expectations

To contextualize the full system, we define two simpler baseline configurations (Table 6). These are qualitative expectation baselines rather than executed benchmark results.

Table 6. Qualitative expectation comparison across three system configurations. H = High, M = Medium, L = Low, N = None. The full TADI system’s advantage lies in combining structured SQL queries with semantic search and domain-specialized tools.

System	Accuracy	Evidence	Reasoning	Completeness
Direct LLM (no tools)	L	N	M	L
Simple RAG (retrieve + generate)	M	M	L–M	M
Full TADI (12 tools)	H	H	H	H

Baseline 1: Direct LLM. The question is sent to the LLM with only the system prompt’s domain knowledge (no tool access, no data retrieval). The LLM can produce generic North Sea drilling statements from parametric knowledge but cannot cite specific measurements, dates, depths, or DDR quotes.

Baseline 2: Simple RAG. All 36,709 documents are embedded; the top- k ($k = 20$) documents are retrieved for each question, concatenated into the LLM context, and an answer is generated. This can find relevant DDR passages but cannot compute aggregate statistics, perform SQL-level filtering, cross-reference structured measurements with narratives, or apply domain-specific algorithms. Questions

requiring ranking (“which well drilled fastest?”), computation (“break down NPT by cause”), or multi-step derivation are fundamentally beyond its capability.

The key differentiator of TADI is the ability to combine structured analytical queries (SQL aggregations over 65,000+ rows) with unstructured text retrieval (semantic search over 36,709 documents) in a single, iterative reasoning chain of up to 10 steps.

6. Discussion

6.1. Key Findings

Our experience designing and evaluating TADI yields several findings relevant to the broader deployment of agentic AI in technical domains.

Tool design is the bottleneck, not model capability. The LLM’s ability to reason about drilling operations is adequate for tool *selection* and evidence *synthesis*, but insufficient for independently deriving complex domain algorithms (phase detection, difficulty indexing, risk scoring) within a 10-round tool-calling budget. The 12 specialized tools encode approximately 2,800 lines of drilling engineering logic that the LLM invokes but does not generate. This finding aligns with the broader observation that tool-augmented LLMs excel when tools encapsulate domain-specific computation [12].

The system prompt is load-bearing infrastructure. The 168-line system prompt is not a cosmetic wrapper; it is the mechanism through which domain knowledge, tool selection policies, output format requirements, and quality standards are injected into every agent interaction. The ablation removing the domain prompt is predicted to cause the most severe quality degradation. This finding reinforces the growing recognition that prompt engineering is a systematic discipline [37] rather than an ad hoc practice.

Cross-referencing prevents evidence monoculture. Without the explicit rule requiring both structured data citations and DDR quotes in every answer, the agent tends toward “evidence monoculture”—producing answers grounded in only one modality. The `get_ddr_narrative` tool, designed specifically to return attributable DDR text via SQL rather than relying only on semantic search, makes narrative evidence reliably available regardless of the semantic search’s relevance to a particular query.

6.2. Advantages of the TADI Approach

Versus traditional drilling analytics platforms. Typical drilling analytics platforms provide sensor data visualization and parametric dashboards but lack natural-language query capability and cannot reason about free-text operational descriptions. TADI’s integration of structured data querying, unstructured text comprehension, and multi-source reasoning via LLM orchestration enables queries that span data modalities: “What operational issues occurred when drilling through the Draupne Formation in well F-11 B, and did the mud properties change in response?”

Versus simple RAG. Standard RAG retrieves text passages and generates answers but cannot perform analytical computation. TADI’s tool suite enables aggregate queries (total NPT by cause, average ROP by section, field-wide difficulty rankings) that require SQL-level processing of structured data—operations fundamentally beyond passage retrieval.

Versus fine-tuned domain models. Domain-specific LLM development and fine-tuning [40] require training data, compute resources, and model management overhead. TADI achieves domain adaptation through prompt engineering and tool design, maintaining the flexibility to upgrade the underlying LLM without retraining. The domain knowledge resides in version-controlled code artifacts (system prompt, tool implementations, ingestion parsers), not in model weights.

6.3. Limitations and Failure Modes

Sparse-data wells. The exploration wells (15/9-19 series, 1980s–1990s) and the F-11 main bore (17 DDRs) produce lower-quality answers due to insufficient data for reliable phase detection and statistical analysis. The system appropriately assigns LOW confidence but cannot overcome the fundamental data limitation.

Ambiguous question scope. When users do not specify a wellbore, time window, or ranking criterion, the current agent does not ask clarifying follow-up questions. Instead, it states assumptions in the output and lowers confidence when multiple interpretations remain plausible.

Hypothetical and recommendation questions. Questions like “What BHA configuration would you recommend for a future well?” require the agent to synthesize historical performance into forward-looking recommendations. While the system retrieves relevant historical data, the recommendation quality depends on the LLM’s ability to reason over drilling engineering principles—a capability that is not formally validated.

Field-wide ranking with nuance. Rankings based on criteria not directly implemented in the five benchmark modes (e.g., “rank wells by drilling complexity considering geology, weather, and equipment”) may produce incomplete answers because the agent must compose multiple tool calls and synthesize qualitative factors that resist simple SQL aggregation.

Tool selection errors. The agent may select overly specific tools for broad questions, fail to follow up with narrative evidence tools, or formulate incorrect SQL queries that consume tool-calling rounds. The 10-round limit creates a planning pressure that the agent occasionally fails to navigate optimally for complex, multi-step queries.

Heuristic issue labels. Several operational issue categories rely partly on keyword matches in DDR comments. This improves recall but can produce lexical false positives in drilling jargon (for example, “kick off” versus a true well-control kick) unless cross-checked with surrounding context.

LLM arithmetic limitations. Multi-step quantitative derivations (e.g., “Estimate the cost impact of NPT assuming a \$500K/day spread rate”) require the agent to perform arithmetic across tool results. While the LLM can handle basic calculations, complex multi-step computations may accumulate errors.

6.4. Implications for Drilling Operations

TADI demonstrates the feasibility of AI-assisted operational intelligence that goes beyond simple data retrieval. The system’s ability to synthesize weeks of operations into thematic summaries, correlate equipment failures with formation intervals, and benchmark wells against field norms suggests three practical applications:

Post-well review acceleration. Synthesizing an entire well’s operational history into structured analyses of phase durations, NPT root causes, and configuration effectiveness—tasks that currently require an engineer to read every DDR individually.

Cross-well learning. Enabling structured comparison across wells drilled from the same platform over a decade, surfacing patterns in equipment reliability, formation-specific challenges, and operational improvements that would otherwise require extensive manual analysis.

Knowledge preservation. Encoding operational lessons in a queryable format that persists regardless of personnel turnover also aligns with broader oil-and-gas digital twin efforts that seek to externalize and reuse operational knowledge and analytics workflows [41].

6.5. Broader Applicability

The TADI architecture is not specific to drilling operations. Any domain that combines structured sensor/measurement data with free-text operational reports—subsea inspection, power plant maintenance, aviation maintenance logs, clinical trial monitoring—could benefit from the same design pattern: a dual-store backend (analytical database + vector store), domain-specialized tools encapsulating expert algorithms, and an LLM orchestrator with a domain-aware system prompt enforcing structured, evidence-based output.

7. Conclusions and Future Work

We have presented TADI (Tool-Augmented Drilling Intelligence), an agentic AI system that transforms heterogeneous wellsite data into evidence-based operational intelligence. Applied to the Equinor Volve Field dataset, TADI parses all 1,759 DDR XML files with zero errors, integrates selected

WITSML real-time objects plus production and geological records into a 12-table analytical database with 65,000+ rows and a 36,709-document semantic search index, and answers natural-language drilling questions through iterative tool-calling with dual-source evidence by design. The system's 12 domain-specialized tools encode approximately 2,800 lines of drilling engineering logic that the LLM orchestrates but does not independently derive. A 130-question stress-question taxonomy spanning six operational categories supports systematic evaluation of coverage, while three detailed case studies demonstrate evidence integration across structured measurements and narrative reports.

Our key finding is that domain-specialized tool design, not model scale or fine-tuning, is the primary driver of analytical quality in technically demanding domains. The 6,084-line, framework-free implementation demonstrates that effective agentic AI systems can be built with minimal infrastructure when domain knowledge is properly factored into deterministic tools, a carefully engineered system prompt, and a structured output format. The system remains reproducible given the public Volve dataset download and an API key.

Future work includes: (1) quantitative evaluation of all 130 stress test questions with human expert scoring to validate the EGS metric against subjective quality judgments; (2) deployment with streaming real-time data to support operational decision-making during active drilling campaigns; (3) multi-agent extensions where specialized sub-agents handle different question categories in parallel; (4) integration of wellbore stability models and real-time pore pressure estimation for proactive well control decision support; and (5) systematic comparison with fine-tuned domain LLMs to quantify the trade-off between prompt engineering and model adaptation approaches.

Acknowledgments: The author thanks Equinor for making the Volve Field dataset publicly available, enabling open research on drilling operational intelligence. The author also thanks the SPE Gulf Coast Section for organizing the 2026 ML Challenge, which motivated this work.

Appendix A. Test Coverage Details

Table A1. Automated test coverage across the system's four test modules. Tests validate both component-level correctness and end-to-end tool behavior.

Test Module	Tests	Coverage Focus
test_config.py	18	Well name normalization, display, round-trip
test_parse_ddr.py	16	DDR XML parsing, filename extraction, full corpus
test_parse_witsml.py	17	WITSML parsing, unit conversions, field structure
test_tools.py	44	All 12 tools, registry, output validation, edge cases
Total	95	Plus 130 stress test questions (stress corpus)

Appendix B. Key Wells and Data Availability

Table A2. Data availability for representative Volve wells. DDR count indicates daily drilling reports. WITSML coverage includes BHA runs, mudlog intervals, and trajectory stations. Production data is available for 7 wells.

Well	DDRs	Date Range	WITSML	Prod.	Fm. Tops
15/9-F-12	165	2007–2016	–	✓	✓
15/9-F-14	134	2007–2016	–	✓	✓
15/9-F-4	130	2007–2016	–	✓	✓
15/9-F-5	103	2007–2016	–	✓	–
15/9-F-15 D	99	various	✓	✓	✓
15/9-F-1 C	98	various	✓	–	✓
15/9-F-11 B	90	2013–2016	✓	–	✓
15/9-F-11 T2	53	2013	✓	–	✓
15/9-F-11	17	2013	✓	✓	✓

Appendix C. Question Taxonomy Subcategories

The 130-question stress test suite is organized into subcategories that probe specific analytical capabilities. Selected subcategories per category:

- **Cat. 1** (20 questions): Phase boundary detection (7), phase ambiguity resolution (4), cross-well phase comparison (5), phase sequence reconstruction (4).
- **Cat. 2** (21 questions): NPT decomposition (5), flat-time and stall analysis (3), drilling efficiency trends (4), field-wide efficiency ranking (3), trip time analysis (3), invisible lost time (2), efficiency-weather correlation (1).
- **Cat. 3** (21 questions): Formation-level ROP analysis (5), gas response analysis (4), section difficulty ranking (4), drilling parameter correlation (4), lithology-ROP relationship (2), d-exponent analysis (1), DDR-WITSML consistency (1).
- **Cat. 4** (20 questions): Best BHA run identification (4), BHA run failure analysis (3), cross-well BHA comparison (4), BHA durability trends (2), drilling parameter sensitivity (3), BHA recommendation (2), DDR-WITSML consistency (2).
- **Cat. 5** (26 questions): Equipment reliability (5), well control events (4), weather impact (4), mud losses and circulation (4), stuck pipe and tight hole (3), wellbore instability (2), cross-well issue comparison (2), field-wide risk ranking (2).
- **Cat. 6** (22 questions): Best practices extraction (3), lessons learned (3), drilling program recommendations (3), shift handover summaries (2), well-on-well improvement (3), drilling-production integration (3), future well planning (3), executive summary (1), cost estimation (1).

Appendix D. System Prompt Excerpt

The following excerpt shows the tool selection guide from the 168-line system prompt, illustrating how domain knowledge is encoded as category-specific tool-calling sequences:

Listing 1: Tool selection guide excerpt from the TADI system prompt.

```

1 ## Tool Selection Guide
2
3 EVERY question type -- ALWAYS end with this step:
4 - get_ddr_narrative(well, date_from, date_to)
5
6 Category 1 (Phase Identification):
7   get_drilling_phases -> query_drilling_data -> get_ddr_narrative
8
9 Category 2 (Time & Efficiency):
10  compute_efficiency_metrics -> get_ddr_narrative
11  -> search_daily_reports
12
13 Category 3 (ROP Performance):
14  query_drilling_data -> get_bha_configurations
15  -> get_ddr_narrative
16
17 Category 4 (BHA Effectiveness):
18  get_bha_configurations -> query_drilling_data
19  -> get_ddr_narrative
20
21 Category 5 (Operational Issues):
22  identify_operational_issues -> get_ddr_narrative
23  -> query_drilling_data
24
25 Category 6 (Synthesis / Comparison):
26  get_field_benchmarks -> compare_wells
27  -> compute_efficiency_metrics -> get_ddr_narrative

```

References

1. Equinor. Volve Field Data Set. Equinor Open Data, 2018. Multi-terabyte dataset from the Volve field, Norwegian North Sea, comprising approximately 40,000 files.
2. Ferrigno, E.; Rodriguez, M.; Davidsson, E. Revolutionizing Drilling Operations: Next-gen LLM-AI for Real-time Support in Well Construction Control Rooms. In Proceedings of the SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana, USA, 2024. SPE-220798-MS.
3. Kumar, P.; Kathuria, S. Large Language Models (LLMs) for Natural Language Processing (NLP) of Oil and Gas Drilling Data. In Proceedings of the SPE Annual Technical Conference and Exhibition, San Antonio, Texas, USA, 2023.
4. Bhatia, G.; Yadav, A.; Nanda, D.; Goyal, D.; Perumalla, S.; Shinde, A.; Jha, B.C.; Upreti, D. Digitization of Daily Drilling Reports Using LLMs. In Proceedings of the SPE Middle East Oil, Gas and Geosciences Show, Manama, Bahrain, 2025. SPE-227059-MS, <https://doi.org/10.2118/227059-MS>.
5. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. ReAct: Synergizing Reasoning and Acting in Language Models. In Proceedings of the International Conference on Learning Representations (ICLR), 2023, [arXiv:cs.CL/2210.03629].
6. Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Zettlemoyer, L.; Cancedda, N.; Scialom, T. Toolformer: Language Models Can Teach Themselves to Use Tools. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2023, [arXiv:cs.CL/2302.04761].
7. Patil, S.G.; Zhang, T.; Wang, X.; Gonzalez, J.E. Gorilla: Large Language Model Connected with Massive APIs. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2024, [arXiv:cs.CL/2305.15334].
8. Shen, Y.; Song, K.; Tan, X.; Li, D.; Lu, W.; Zhuang, Y. HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2023, [arXiv:cs.CL/2303.17580].
9. Qin, Y.; Liang, S.; Ye, Y.; Zhu, K.; Yan, L.; Lu, Y.; Lin, Y.; Cong, X.; Tang, X.; Qian, B.; et al. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs. *arXiv preprint arXiv:2307.16789* 2023, [arXiv:cs.AI/2307.16789]. <https://doi.org/10.48550/arXiv.2307.16789>.
10. Li, M.; Zhao, Y.; Yu, B.; Song, F.; Li, H.; Yu, H.; Li, Z.; Huang, F.; Li, Y. API-Bank: A Comprehensive Benchmark for Tool-Augmented LLMs. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 2023; pp. 3102–3116. <https://doi.org/10.18653/v1/2023.emnlp-main.187>.
11. Guo, Z.; Cheng, S.; Wang, H.; Liang, S.; Qin, Y.; Li, P.; Liu, Z.; Sun, M.; Liu, Y. StableToolBench: Towards Stable Large-Scale Benchmarking on Tool Learning of Large Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024, Bangkok, Thailand, 2024; pp. 11143–11156. <https://doi.org/10.18653/v1/2024.findings-acl.664>.
12. Qin, Y.; Hu, S.; Lin, Y.; Chen, W.; Ding, N.; Cui, G.; Zeng, Z.; Huang, Y.; Xiao, C.; Han, C.; et al. Tool Learning with Foundation Models. *arXiv preprint arXiv:2304.08354* 2023, [arXiv:cs.AI/2304.08354]. <https://doi.org/10.48550/arXiv.2304.08354>.
13. Wang, L.; Ma, C.; Feng, X.; Zhang, Z.; Yang, H.; Zhang, J.; Chen, Z.; Tang, J.; Chen, X.; Lin, Y.; et al. A Survey on Large Language Model based Autonomous Agents. *Frontiers of Computer Science* 2024, 18, 186345, [arXiv:cs.AI/2308.11432]. <https://doi.org/10.1007/s11704-024-40231-1>.
14. Tran, K.T.; Dao, D.; Nguyen, M.D.; Pham, Q.V.; O'Sullivan, B.; Nguyen, H.D. Multi-Agent Collaboration Mechanisms: A Survey of LLMs. *arXiv preprint arXiv:2501.06322* 2025, [arXiv:cs.AI/2501.06322].
15. Guo, T.; Chen, X.; Wang, Y.; Chang, R.; Pei, S.; Chawla, N.V.; Wiest, O.; Zhang, X. Large Language Model Based Multi-Agents: A Survey of Progress and Challenges. *arXiv preprint arXiv:2402.01680* 2024, [arXiv:cs.AI/2402.01680].
16. Antoniuk, M.; Dalglish, J.; Verkruyse, M.; Lo, J. Natural Language Processing Techniques on Oil and Gas Drilling Data. In Proceedings of the SPE Intelligent Energy International Conference and Exhibition, Aberdeen, Scotland, UK, 2016. SPE-181015-MS, <https://doi.org/10.2118/181015-MS>.
17. Hoffmann, J.; Mao, Y.; Wesley, A.; Taylor, A. Sequence Mining and Pattern Analysis in Drilling Reports with Deep Natural Language Processing. In Proceedings of the SPE Annual Technical Conference and Exhibition, Dallas, Texas, USA, 2018; [arXiv:cs.CL/1712.01476]. SPE-191505-MS.
18. Yi, M.; Ceglinski, K.; Ashok, P.; Behounek, M.; White, S.; Peroyea, T.; Thetford, T. Applications of Large Language Models in Well Construction Planning and Real-Time Operation. In Proceedings of the IADC/SPE International Drilling Conference and Exhibition, Galveston, Texas, USA, 2024. IADC/SPE-217700-MS.

19. Pacis, F.J.; Alyaev, S.; Pelfrene, G.; Wiktorski, T. Enhancing Information Retrieval in the Drilling Domain: Zero-Shot Learning with Large Language Models for Question-Answering. In Proceedings of the IADC/SPE International Drilling Conference and Exhibition, Galveston, Texas, USA, 2024. IADC/SPE-217671-MS, <https://doi.org/10.2118/217671-MS>.
20. Zhang, L.; Pacis, F.J.; Alyaev, S.; Wiktorski, T. Cloud-Free Question Answering from Internal Knowledge Bases: Building an AI for Drilling Applications. *First Break* **2025**, *43*, 43–49. <https://doi.org/10.3997/1365-2397.fb2025012>.
21. Ogundare, O.; Madasu, S.; Wiggins, N. Industrial Engineering with Large Language Models: A Case Study of ChatGPT's Performance on Oil & Gas Problems. *arXiv preprint arXiv:2304.14354* **2023**, [[arXiv:cs.CL/2304.14354](https://arxiv.org/abs/2304.14354)]. <https://doi.org/10.48550/arXiv.2304.14354>.
22. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.t.; Rocktäschel, T.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2020, Vol. 33, pp. 9459–9474, [[arXiv:cs.CL/2005.11401](https://arxiv.org/abs/2005.11401)].
23. Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Wang, M.; Wang, H. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv preprint arXiv:2312.10997* **2023**, [[arXiv:cs.CL/2312.10997](https://arxiv.org/abs/2312.10997)]. <https://doi.org/10.48550/arXiv.2312.10997>.
24. Gupta, S.; Ranjan, R.; Singh, S.N. A Comprehensive Survey of Retrieval-Augmented Generation (RAG): Evolution, Current Landscape and Future Directions. *arXiv preprint arXiv:2410.12837* **2024**, [[arXiv:cs.CL/2410.12837](https://arxiv.org/abs/2410.12837)].
25. Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; Hajishirzi, H. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. *arXiv preprint arXiv:2310.11511* **2023**, [[arXiv:cs.CL/2310.11511](https://arxiv.org/abs/2310.11511)]. <https://doi.org/10.48550/arXiv.2310.11511>.
26. Sarmah, B.; Hall, B.; Rao, R.; Patel, S.; Pasquali, S.; Mehta, D. HybridRAG: Integrating Knowledge Graphs and Vector Retrieval Augmented Generation for Efficient Information Extraction. *arXiv preprint arXiv:2408.04948* **2024**, [[arXiv:cs.IR/2408.04948](https://arxiv.org/abs/2408.04948)].
27. Muennighoff, N.; Tazi, N.; Magne, L.; Reimers, N. MTEB: Massive Text Embedding Benchmark. In Proceedings of the Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Dubrovnik, Croatia, 2023; pp. 2014–2037, [[arXiv:cs.CL/2210.07316](https://arxiv.org/abs/2210.07316)].
28. Liu, X.; Shen, S.; Li, B.; Ma, P.; Jiang, R.; Zhang, Y.; Fan, J.; Li, G.; Tang, N.; Luo, Y. A Survey of Text-to-SQL in the Era of LLMs: Where Are We, and Where Are We Going? *arXiv preprint arXiv:2408.05109* **2024**, [[arXiv:cs.CL/2408.05109](https://arxiv.org/abs/2408.05109)].
29. Tunkiel, A.T.; Wiktorski, T.; Sui, D. Drilling Dataset Exploration, Processing and Interpretation Using Volve Field Data. In Proceedings of the Proceedings of the ASME 2020 39th International Conference on Ocean, Offshore and Arctic Engineering (OMAE), Virtual, Online, 2020; Vol. 11, p. V011T11A076. <https://doi.org/10.1115/OMAE2020-18151>.
30. Nikitin, N.O.; Revin, I.; Hvatov, A.; Vychuzhanin, P.; Kalyuzhnaya, A.V. Hybrid and Automated Machine Learning Approaches for Oil Fields Development: The Case Study of Volve Field, North Sea. *Computers & Geosciences* **2022**, *161*, 105061. <https://doi.org/10.1016/j.cageo.2022.105061>.
31. Ng, C.S.W.; Ghahfarokhi, A.J.; Nait Amar, M. Well Production Forecast in Volve Field: Application of Rigorous Machine Learning Techniques and Metaheuristic Algorithm. *Journal of Petroleum Science and Engineering* **2022**, *208*, 109468. <https://doi.org/10.1016/j.petrol.2021.109468>.
32. Saha, S.; Vishal, V.; Mahanta, B.; Pradhan, S.P. Geomechanical Model Construction to Resolve Field Stress Profile and Reservoir Rock Properties of Jurassic Hugin Formation, Volve Field, North Sea. *Geomechanics and Geophysics for Geo-Energy and Geo-Resources* **2022**, *8*, 59. <https://doi.org/10.1007/s40948-022-00359-5>.
33. Oloruntobi, O.; et al. Petrophysical Property Prediction from Seismic Inversion Attributes Using Rock Physics and Machine Learning: Volve Field, North Sea. *Applied Sciences* **2024**, *14*, 1345. <https://doi.org/10.3390/app14041345>.
34. Energistics. WITSML Data Standards. Energistics Consortium, 2011. Version 1.4.1.1. Wellsite Information Transfer Standard Markup Language.
35. Geekiyana, S.C.H.; Tunkiel, A.T.; Sui, D. Drilling Data Quality Improvement and Information Extraction with Case Studies. *Journal of Petroleum Exploration and Production Technology* **2021**, *11*, 819–837. <https://doi.org/10.1007/s13202-020-01024-x>.

36. Raasveldt, M.; Mühleisen, H. DuckDB: An Embeddable Analytical Database. In Proceedings of the Proceedings of the 2019 International Conference on Management of Data (SIGMOD), Amsterdam, Netherlands, 2019; pp. 1981–1984. <https://doi.org/10.1145/3299869.3320212>.
37. Schulhoff, S.; Ilie, M.; Balepur, N.; Kahadze, K.; Liu, A.; Si, C.; Li, Y.; Gupta, A.; Han, H.; Schulhoff, S.; et al. The Prompt Report: A Systematic Survey of Prompting Techniques. *arXiv preprint arXiv:2406.06608* **2024**, [arXiv:cs.CL/2406.06608]. <https://doi.org/10.48550/arXiv.2406.06608>.
38. Sahoo, P.; Singh, A.K.; Saha, S.; Jain, V.; Mondal, S.; Chadha, A. A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications. *arXiv preprint arXiv:2402.07927* **2024**, [arXiv:cs.CL/2402.07927].
39. Geng, S.; et al. JSONSchemaBench: A Rigorous Benchmark of Structured Outputs for Language Models. *arXiv preprint arXiv:2501.10868* **2025**, [arXiv:cs.CL/2501.10868].
40. Lin, B.; Jin, Y.; Cao, Q.; Meng, H.; Pang, H.; Wei, S. Developing a Large Language Model for Oil- and Gas-Related Rock Mechanics: Progress and Challenges. *Natural Gas Industry B* **2025**, *12*, 110–122. <https://doi.org/10.1016/j.ngib.2025.03.007>.
41. Meza, E.B.M.; Souza, D.G.B.d.; Copetti, A.; Sobral, A.P.B.; Silva, G.V.; Tammela, I.; Cardoso, R. Tools, Technologies and Frameworks for Digital Twins in the Oil and Gas Industry: An In-Depth Analysis. *Sensors* **2024**, *24*, 6457. <https://doi.org/10.3390/s24196457>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.