Scilight

Bulletin of Computational Intelligence
https://www.sciltp.com/journals/bci



Review

Scope, Challenges, and Future of Agentic AI in Geotechnical and Earthquake Engineering: A Critical Review

Ishwor Thapa [1], Rojan Subedi [2], Garima Shrestha [2], Sundar Adhikari [3], Pratik Rana Magar [4] and Bishwash Poudel [2,*]

[1] Institute of Forestry, Hetauda Campus, Tribhuvan University, Hetauda 44107, Nepal
[2] Madan Bhandari College of Engineering, Urlabari, Morang 56604, Nepal
[3] School of Engineering, Faculty of Science and Technology, Pokhara University, Pokhara 33700, Nepal
[4] Department of Civil Engineering, Nepal Engineering College, Changunarayan, Bhaktapur 44800, Nepal
* Correspondence: bishwashpoudel@mbman.edu.np

How To Cite: Thapa, I.; Subedi, R.; Shrestha, G.; et al. Scope, Challenges, and Future of Agentic AI in Geotechnical and Earthquake Engineering: A Critical Review. *Bulletin of Computational Intelligence* **2026**, *2*(4), 291–317. https://doi.org/10.53941/bci.2026.100016

Received: 3 September 2026
Revised: 15 October 2026
Accepted: 27 October 2026
Published: 4 November 2026

Abstract: Agentic artificial intelligence (agentic AI) refers to systems that pursue goals by planning, invoking external tools, maintaining memory and acting on the results with a defined degree of autonomy. This review examines the scope, challenges and future directions of agentic AI in geotechnical and earthquake engineering, with an explicit distinction between capabilities that have been demonstrated and those that remain conceptual. We first propose a taxonomy that separates predictive machine learning, generative AI, retrieval-augmented language-model systems, tool-using agents, multi-agent systems and closed-loop autonomous agents, and we distinguish within-episode feedback and memory updating from online learning, reinforcement learning and model retraining. We then synthesize 22 studies (2023–2026) in which large language models (LLMs) or agents perform geotechnical, earthquake or adjacent structural engineering tasks, together with two closed-loop exemplars from chemistry, and classify each by architecture, tools, memory, validation, uncertainty treatment, human involvement and evidence level. All civil-engineering studies located are prototypes or proof-of-concept systems; no field-validated agentic deployment in geotechnical or earthquake practice was found, and no peer-reviewed study of an agent operating a commercial geotechnical finite-element package was identified. We discuss the enabling layers that engineering agents require—retrieval-augmented generation over codes and site records, knowledge graphs and data standards (AGS4, DIGGS, IFC), interoperability with OpenSees, PLAXIS and OpenQuake, and physics-informed modeling—and report indicative cost and latency figures. An illustrative proof-of-concept of the tool layer for CPT-based liquefaction assessment shows how deterministic triggering analysis, Monte Carlo uncertainty propagation and human approval gates can be combined. Finally, we assess eight challenges with an explicit ordinal rubric and present a conditional short-, medium- and long-term roadmap linked to eight research gaps. Agentic AI can add value as a supervised orchestration layer over validated engineering tools; claims of autonomous engineering practice are not yet supported by evidence.

Keywords: agentic AI; large language models; geotechnical engineering; earthquake engineering; retrieval-augmented generation; knowledge graphs; physics-informed machine learning; liquefaction

Copyright: © 2026 by the authors. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).
Publisher's Note: Scilight stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Geotechnical and earthquake engineering have historically relied on a synthesis of empirical correlations, analytical mechanics, numerical simulation and expert judgment to address problems characterized by profound uncertainty, spatial variability and potentially catastrophic consequences [1]. Soil–structure interaction under seismic loading remains one of the most complex multi-physics problems in civil engineering, governed by nonlinear constitutive behavior, pore-pressure generation, strain localization and dynamic site amplification. Conventional AI in these fields has largely been confined to supervised learning for property prediction or for pattern recognition in monitoring data [2–4], including recent work on pile bearing capacity [5], stabilized-soil strength [6] and slope stability in mountainous terrain [7].

Large language models (LLMs) have recently been embedded in software systems that plan multi-step tasks, call external tools and retain information across steps; such systems are commonly described as LLM-based agents [8–10]. In principle, an agent of this kind could read borehole logs and cone penetration test (CPT) soundings, query design codes, run a finite-element model and summarize the result for an engineer. Whether, and under what conditions, this is achievable in safety-critical geotechnical and earthquake practice is an open question. Early studies show that LLMs can assist with geotechnical calculations and reporting but also make conceptual and arithmetic errors that require expert checking [11–13].

This review distinguishes capabilities that have been demonstrated from those that remain prospective. Its objectives are to (i) define agentic AI precisely through a taxonomy that does not treat predictive and agentic systems as a binary; (ii) synthesize and classify the existing studies of LLM-based and agentic systems in geotechnical, earthquake and adjacent structural engineering by evidence level; (iii) describe the enabling layers—retrieval, knowledge representation, interoperability and physics-informed modeling—that engineering agents require; (iv) illustrate the workflow with a reproducible proof-of-concept; and (v) derive research gaps and a conditional roadmap. Throughout, statements about capability are labeled as demonstrated (with citation) or prospective.

2. Review Method and Evidence Classification

2.1. Literature Search

This is a critical, narrative review supported by a structured literature search rather than a formal systematic review. Searches were conducted in Google Scholar, Crossref, OpenAlex, publisher platforms (ASCE, Elsevier, Springer, Taylor & Francis, Wiley, MDPI, Nature Portfolio, J-STAGE) and arXiv, up to October 2026. Search strings combined agent-related terms ("large language model", "LLM agent", "agentic", "multi-agent", "retrieval-augmented", "tool use") with domain terms ("geotechnical", "soil", "foundation", "slope", "liquefaction", "earthquake", "seismic", "structural analysis", "structural health monitoring", "damage assessment"). Studies were included in the comparative synthesis (Section 4) when an LLM, vision-language model or software agent performed a geotechnical, earthquake or structural engineering task and the study reported some form of evaluation. Preprints were retained because the field is moving quickly, but they are flagged as such. Foundational references on agents, retrieval, knowledge graphs, physics-informed machine learning and the relevant engineering methods were added to provide context. Every bibliographic record was checked against its publisher, DOI, arXiv or proceedings page.

2.2. Evidence Levels and Degree of Autonomy

Each application discussed in Sections 4–6 is assigned one of three evidence levels: (i) demonstrated—used in real engineering practice or real physical experiments with documented outcomes; (ii) prototype or proof-of-concept—implemented and evaluated on benchmarks, textbook problems, laboratory data or retrospective case studies, but not deployed in practice; and (iii) conceptual—proposed or argued for, without an implemented and evaluated system in the domain. The degree of autonomy demonstrated in a study is classified, following the progression in recent taxonomies [14–16], as LLM assistant, tool-using agent, multi-agent system or closed-loop autonomous agent (Section 3.1). These labels describe what a study demonstrated, not what its authors anticipated.

3. Foundations of Agentic AI

3.1. Definition and Taxonomy

In the classical sense, an agent is a system that perceives its environment and acts upon it [17]. In current usage, "agentic AI" denotes systems in which a generative model—usually an LLM—decomposes a goal into steps, selects and calls external tools, observes the results and decides what to do next, with varying amounts of human oversight [8,9,14]. The distinction from predictive machine learning is a matter of degree rather than a binary.

Predictive models are not necessarily stateless—recurrent and sequence models carry state, and online-learning systems update continuously—and agents do not necessarily learn. Likewise, chain-of-thought prompting [18] is a prompting technique available to any LLM; it is not a defining architectural property of agentic systems. Table 1 therefore distinguishes six categories by their defining capability.

Table 1. Taxonomy of AI system categories relevant to geotechnical and earthquake engineering. Latency values are indicative orders of magnitude reported in the cited studies, not benchmarks.

Category	Defining Capability	State/Memory	Indicative Latency Per Task	Geotechnical Example
Predictive ML	Learns a mapping from inputs to a target (regression, classification)	Fixed after training; sequence models carry internal state; online learning possible	Milliseconds–seconds per prediction	Pile bearing capacity from soil and pile parameters [5]
Generative AI (single call)	Generates text, code or images from a prompt	Context window only	Seconds	ChatGPT drafting a geotechnical data report [11]
LLM system with retrieval (RAG)	Grounds generation in retrieved documents	External document index; no planning	Seconds	Geotechnical tutor answering from course material [19]
Tool-using agent	Plans steps and calls external tools (code, solvers, databases) in a loop	Context plus optional vector store	Seconds–minutes; e.g., 7–17 min per FEA problem [20]	ReAct agent calling bearing-capacity and DIGGS-search tools [12]
Multi-agent system	Several role-specialized agents coordinate (planner, executor, checker)	Shared conversation or structured memory	Minutes; e.g., ≈2 min per design scenario [21]	Router-based agents for shallow and pile foundation design [22]
Closed-loop autonomous agent	Acts on the physical or operational environment and uses the observed outcome to decide the next action	Persistent memory; may update policies	Dominated by physical actions (experiments, field operations)	None demonstrated in geotechnics; laboratory exemplar in chemistry [23]

A concrete engineering example clarifies the difference between goal-directed behavior and tool use. Consider a pile-group design task with the goal "confirm that the group meets a 25 mm settlement limit with a target reliability index of 3.0, adjusting pile length if necessary". Tool use is the mechanism: the agent calls a load-transfer (t–z) analysis written in OpenSeesPy [24] and a reliability routine, and receives numerical results. Goal-directed behavior is the control logic: the agent decides which analyses are needed, notices that the computed reliability index is below the target, increases pile length, re-runs the analyses, stops when both criteria are met, and submits the design and its evidence to the engineer for approval. A system that only calls the t–z tool when instructed is a tool-using assistant; a system that decomposes the goal and iterates toward it is goal-directed.

3.2. Architectural Components and Choice of Language Models

A typical agentic framework for geotechnical and earthquake engineering comprises six interacting components (Figure 1). The perception layer converts heterogeneous inputs—borehole logs, CPT soundings, laboratory results, imagery and accelerometer streams—into structured representations, using format parsers, optical character recognition, vision-language models and signal processing [9]. The planning and reasoning core consists of one or more LLMs or vision-language models that decompose tasks, select tools and critique intermediate results [8,10]. Memory comprises short-term working memory (the context window and scratchpad) and long-term stores such as project databases, design precedents and case histories, often indexed in vector databases [25]. A knowledge layer provides retrieval over codes, reports and site records and, where available, a knowledge graph (Sections 3.4 and 3.5). The tool layer exposes engineering software—OpenSees and OpenSeesPy [24,26], PLAXIS through its remote-scripting interface [27], OpenQuake [28], GIS, probabilistic engines, trained surrogates and validated code-check scripts—through typed function calls [29,30] or Model Context Protocol (MCP) servers [31]. Finally, the engineer of record sets goals and approves safety-critical actions; outputs such as reports, design updates and alerts are released only after approval.

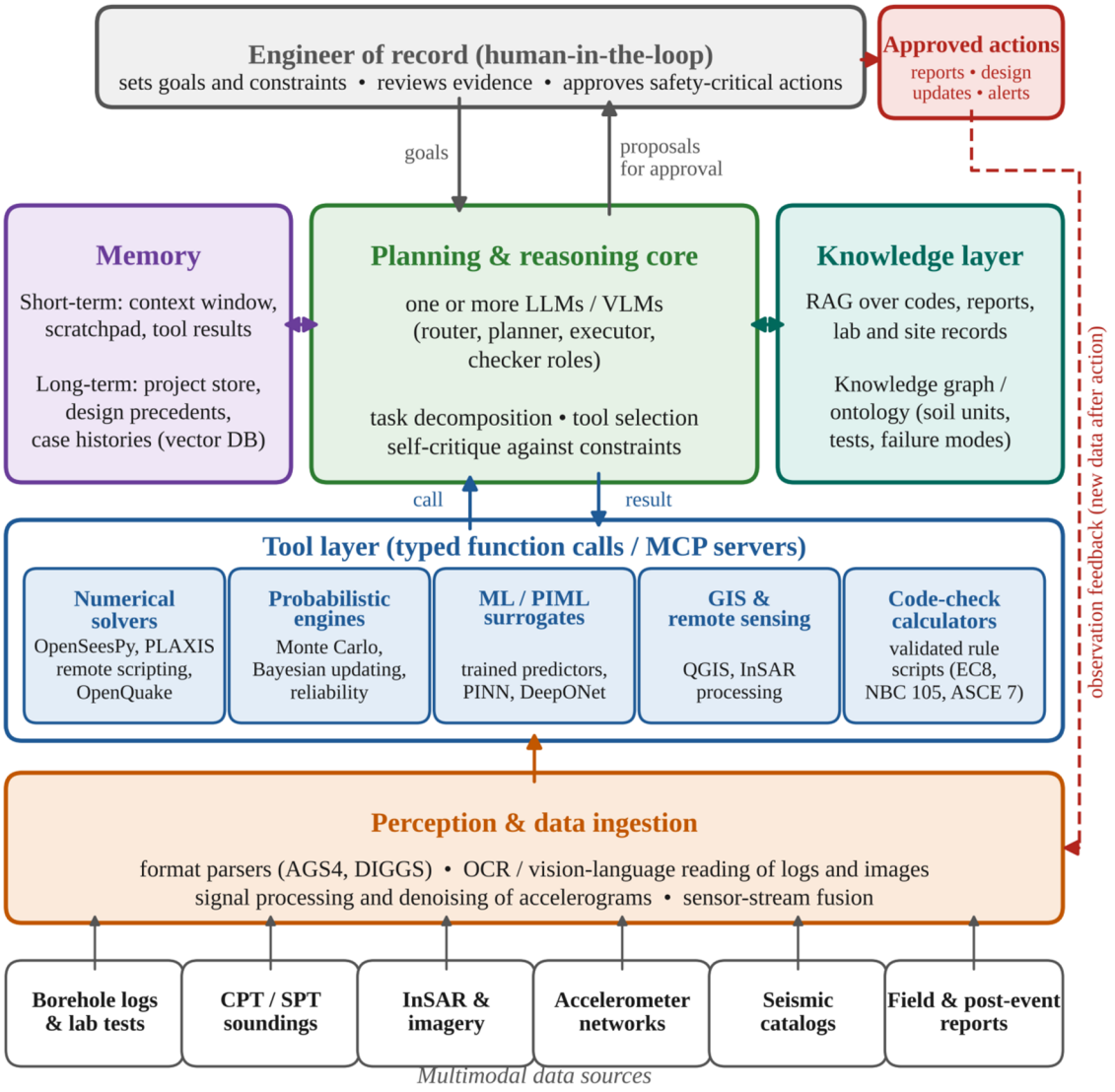


Figure 1. Architecture of an agentic system for geotechnical and earthquake engineering. The planning and reasoning core calls engineering tools through typed interfaces, draws on memory and a knowledge layer, and submits proposals to the engineer of record; approved actions generate new observations that re-enter through the perception layer.

One LLM or several? The reviewed studies show that the choice matters. In a 20-problem structural-analysis benchmark, accuracy ranged from 100% for GPT-4o to 30% for Llama-3.3 under the same prompting scheme [32]. Conversely, a five-agent system built on the open-weight Llama-3.3-70B outperformed single-model baselines built on larger proprietary models for 2D frame analysis [33], and a router-based multi-agent system that assigns foundation-design tasks to specialized experts reached 95.0% (shallow) and 90.6% (pile) accuracy on 27 cases [22]. Surveys of LLM-based multi-agent systems describe how such agents are profiled, communicate and acquire capabilities [34]. A single general-purpose LLM is therefore sufficient for assistant-type tasks, whereas multi-step engineering workflows benefit from role specialization (planner, executor, checker) and may combine models: a capable model for planning and checking, a smaller or open-weight model for routine sub-tasks, a vision-language model for imagery [35,36] and an embedding model for retrieval. Open-weight models such as Llama 3 [37] and Qwen2.5 [38] can be hosted on premises, which is relevant where site data are confidential.

3.3. Feedback, Memory and Adaptation

The presence of a feedback loop is sometimes taken to make an agent self-improving; this is not the case. A feedback loop may only correct a decision within one task; it does not by itself change the model. Table 2 distinguishes five mechanisms. In the workflow of Figure 2, the loop from “reflect and update” back to “perceive” represents within-episode decision correction and memory updating; online learning, reinforcement learning and retraining are separate processes that must be scheduled, versioned and re-validated.

Table 2. Feedback and adaptation mechanisms in agentic systems and their implications for verification and validation (V&V).

Mechanism	What Changes	Timescale	Condition for Calling the Agent Adaptive	V&V Implication
Decision correction	Next action, given an observation (e.g., re-run a failed analysis) [10]	Within one task	Not adaptive across tasks; behavior on the next task is unchanged	Validate the control logic and stopping criteria
Memory updating	Content of a persistent store (lessons, case histories, verbal reflections); model weights unchanged [25,39]	Across tasks	Adaptive if retrieved memories measurably change behavior on recurring tasks	Audit memory contents; guard against storing erroneous lessons
Online learning	Model parameters updated incrementally from streaming data [40]	Continuous	Adaptive; self-improving only if held-out performance improves	Monitor drift and catastrophic forgetting [41,42]
Reinforcement learning	Policy updated from rewards through interaction [43]	Episodes	Adaptive; requires a safe reward and environment (often a simulator)	Validate reward design; restrict exploration in physical systems
Retraining/fine-tuning	Model replaced by a new version trained on new data	Scheduled, offline	Adaptive by design; a new model version	Full re-validation before release; version control

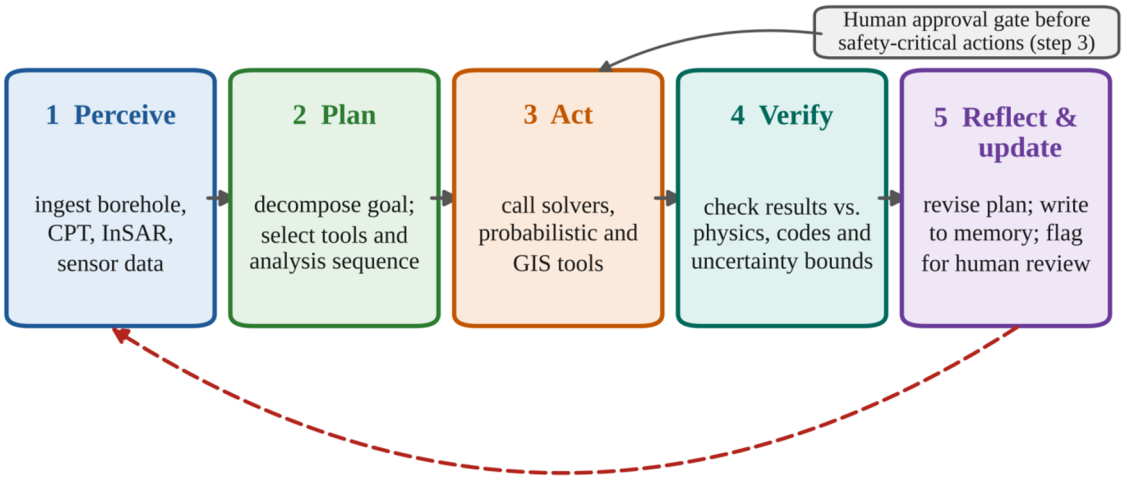


Figure 2. Agentic workflow for geotechnical decision-making. The dashed loop denotes within-episode decision correction and memory updating; safety-critical actions pass a human approval gate.

We therefore call an agent adaptive only when stored information or parameter updates change its behavior on recurring tasks, and self-improving only when that change produces a measurable improvement on a fixed, held-out benchmark. Neither property has yet been demonstrated for a geotechnical agent. The closest example is a non-LLM agentic monitoring framework that adapts its policy through Bayesian inference and reinforcement learning on simulated and benchmark structures [44].

3.4. Retrieval-Augmented Generation

Engineering agents cannot rely on the parametric knowledge of a language model alone. LLMs hallucinate—produce fluent but unsupported content—and their training data have a cut-off date and rarely include project-

specific records [45,46]. Retrieval-augmented generation (RAG) addresses this by retrieving relevant passages from an external corpus at query time and conditioning the response on them [47,48]. For geotechnical and earthquake engineering, the corpus can include design codes (e.g., EN 1998-5 [49], ASCE 7-22 [50] and NBC 105:2020 [51]), geotechnical investigation reports, laboratory databases, site-investigation records, previous case histories and validated calculation notes.

Early evidence is encouraging but limited. Kumar used embedding-based retrieval over the DIGGS schema within a ReAct agent [12]; a RAG-supported undergraduate geotechnical tutor built on GPT-4 reached 95% accuracy at a temperature setting of 0.1 [19]; and RAG over building codes combined with multi-agent cross-checking reached 95.5% and 91.3% accuracy on precast-slab and retaining-wall calculations [52]. Retrieval quality and domain fine-tuning both affect answer quality on building-code questions [53], and an educational geotechnical chatbot restricted to instructor-provided material has been reported [54]; a recent review compares prompting, RAG, domain-adaptive pre-training and fine-tuning for geotechnics [55]. RAG reduces but does not eliminate unsupported output: retrieved passages may be outdated, incomplete or misread, and in one damage-assessment study a RAG variant achieved perfect scores that the author attributed to likely memorization of the test images [56]. Good practice therefore includes citing the retrieved clause in every answer, restricting retrieval to version-controlled sources, and evaluating retrieval and generation separately.

3.5. Knowledge Graphs and Engineering Ontologies

A knowledge graph represents entities and their relationships explicitly—for example, borehole BH-3 → intersects → alluvial unit A2 → has test → CPT-07 → indicates → loose silty sand—so that facts can be queried, checked for consistency and reasoned over [57]. Combining LLMs with knowledge graphs supplies the explicit, verifiable structure that LLMs lack, while LLMs can help to populate and query the graph [58]; graph-based retrieval (GraphRAG) can answer questions that require aggregation across a corpus [59]. Geoscience knowledge graphs are an active field [60], and GeoSciML provides a standard model for geological units and structures [61].

Geotechnical practice already has the starting points for an ontology: AGS4 [62] and DIGGS [63] standardize the exchange of borehole, in-situ and laboratory data, and IFC 4.3 (ISO 16739-1:2024) extends building information models toward infrastructure and ground [64]. A geotechnical ontology linking soil and rock units, test types, derived parameters, failure mechanisms, design parameters, code clauses and infrastructure components would allow an agent to reconcile heterogeneous logs, check that a design parameter was derived from an appropriate test, and trace every recommendation to its evidence. The Geo-Intelligence system, which couples an ontology-based knowledge graph of borehole and stratigraphy data with an LLM through GraphRAG and Web-GIS, is the first geotechnical prototype of this kind that we identified [65]. No shared geotechnical ontology yet exists; this is identified as research gap G2 (Section 9).

3.6. Integration with Engineering Software

Connecting agents to engineering software raises several practical issues. **Access and licensing:** OpenSees, OpenSeesPy and OpenQuake are open source and scriptable [24,26,28], whereas PLAXIS exposes a remote-scripting server and Python wrapper that require a licensed installation [27]; the COMSOL-based FEABench study likewise notes the need for a commercial license [20]. Licence terms may restrict automated or cloud-hosted use and must be checked. **Data exchange:** ground data can be exchanged through AGS4 or DIGGS [62,63], geological models through GeoSciML or the Open Mining Format (OMF) [61,66], and infrastructure models through IFC [64]. Coupled simulations—for example, a soil model and a structural model—can be packaged as Functional Mock-up Units under the Functional Mock-up Interface (FMI) standard for tool-independent co-simulation [67]. **Agent–tool interface:** each tool should be wrapped with a typed schema that states its inputs, units, valid ranges and outputs, so that the agent cannot call it outside its domain of validity; MCP provides a common protocol for such wrappers [31], and a seismology agent built on MCP servers has been used to configure, run and visualize SPECFEM wave-propagation simulations [68]. LLM assistants that convert natural-language or image input into geomechanics simulation scripts have been described in a preprint [69], but we found no peer-reviewed study in which an agent operates PLAXIS or another commercial geotechnical finite-element package (gap G3).

3.7. Building an Engineering Agent: Practical Steps, Cost and Latency

Based on the reviewed implementations, an engineering agent can be built in seven steps: (1) define the goal, scope and the decisions that must remain with the engineer; (2) wrap each engineering tool with a typed schema, unit checks and validity limits; (3) build the knowledge layer (RAG corpus and, where possible, a knowledge graph); (4) choose an orchestration pattern—ReAct-style single agent [10], plan-and-execute, router with experts [22], or

designer–checker multi-agent teams [70,71]—using a framework such as LangChain/LangGraph [72,73], AutoGen [74] or CrewAI (used in [35]); (5) add guardrails: physics and code checks on every tool output, deterministic settings for reproducibility, and approval gates for safety-critical actions; (6) build an evaluation harness with held-out problems, repeated runs to measure run-to-run variability and reporting of cost and latency [75]; and (7) log every prompt, retrieved document, tool call and result for audit.

Cost and latency are rarely reported in civil-engineering agent studies. Table 3 collects the published figures and adds an indicative estimate. Kapoor et al. showed that accuracy-only benchmarks reward needlessly expensive agents and argued for evaluating agents on a joint cost–accuracy frontier [75]. Because API prices change frequently, the estimate in Table 3 is expressed per price tier and dated.

Table 3. Published cost, token and latency figures for LLM agents, and an indicative per-run cost estimate for a geotechnical agent workflow.

Source/Scenario	Reported Figure	Relevance
SWE-agent, software repair [76]	Average US$1.59 (GPT-4 Turbo) and US$2.59 (Claude 3 Opus) per resolved issue; US$4 budget per issue	Order of magnitude of API cost for a multi-step tool-using agent
MetaGPT vs. ChatDev [77,78]	31,255 vs. 19,292 tokens per software project; 541 s vs. 762 s runtime	Token budgets of multi-agent systems
FEABench, COMSOL agent [20]	≈12 min per problem (range 7–17 min)	Latency when an agent drives a finite-element package
MASSE, structural design (preprint) [21]	≈2 min vs. ≈2 h for engineers per design scenario	Time saving reported for a multi-agent design workflow
DisasTeller, post-disaster [35]	"A few minutes" end-to-end; ≈35 output tokens s^{-1} (GPT-4o)	Latency for post-earthquake reporting
Code-compliant RC design [70]	90% time saving relative to manual design	Productivity rather than monetary cost
Indicative estimate (this study): 20 LLM calls × (8000 input + 1000 output tokens) = 160k input + 20k output tokens per workflow run	Frontier-tier models (US$4–5 input/US$20–30 output per million tokens): ≈US$1.0–1.4 per run. Mid-tier (US$2/US$10–12): ≈US$0.5–0.6. Small-tier (US$0.10–0.20/US$0.50–1.20): ≈US$0.03–0.06. List prices accessed 10 October 2026 [79–81]	Tool computation (FE runs, Monte Carlo) and engineer review time are excluded; self-hosted open-weight models replace API fees with hardware cost [37,38]

4. State of the Evidence: LLM-Based and Agentic Systems in Geotechnical and Earthquake Engineering

Tables 4 and 5 summarize the 22 civil-engineering studies identified by the search (Section 2.1) and two closed-loop exemplars from chemistry. Table 4 describes each system (domain, architecture, inputs and outputs, external tools, degree of autonomy and memory); Table 5 describes how it was evaluated (validation and dataset size, uncertainty treatment, human involvement, limitations and evidence level). Figure 3 maps the studies onto application areas and degrees of autonomy.

Four observations follow. First, every civil-engineering study is at evidence level (ii) or below: we found no field-validated agentic deployment in geotechnical or earthquake practice. Second, the most autonomous systems are found in structural analysis and design, where multi-agent systems drive OpenSeesPy or code-checking tools [21,32,33,52,70,82]; geotechnical studies remain mostly at the assistant level [11,13,19,83,84], with tool use demonstrated for data search and bearing capacity [12] and site-investigation knowledge graphs [65]; a router-based multi-agent system has also been applied to shallow and pile foundation design [22], although no external tools were reported for it. Third, most evaluations are small (fewer than about 30 problems or cases, with a few larger benchmarks), uncertainty is rarely treated beyond repeated runs, and humans remain responsible for checking results in every study. Fourth, for soil–structure interaction, probabilistic seismic hazard analysis with OpenQuake and landslide early warning, we found no implemented agentic system; these applications remain conceptual (level iii). The only closed-loop systems with physical action are the chemistry exemplars Coscientist [23] and ChemCrow [85], which show what level (i) evidence for an agent looks like: real experiments, with outcomes checked by independent analytical methods.

Table 4. Comparative summary of LLM-based and agentic studies—system characteristics. Autonomy: A = LLM assistant; T = tool-using agent; M = multi-agent system; C = closed-loop autonomous. "NR" = not reported in the sources consulted. * = preprint.

#	Study	Domain	Architecture/LLM(s)	Input → Output	External Tools	Aut.	Memory
1	Kumar [12]	Soil classification; DIGGS search; bearing capacity	GPT-3.5/ChatGPT; single ReAct agent	Prompts, soil report, DIGGS schema → code, XML, classification, bearing load	Custom bearing-capacity tools; FAISS search	T	Vector-store RAG
2	Shoemaker et al. [11]	Data report; bearing; infinite slope; liquefaction	ChatGPT-4, single chat	Text prompts → report template, code, calculations	None	A	Conversation
3	Wu et al. [84]	Four geotechnical problems (workshop)	ChatGPT-class LLMs	Multimodal data, text → prototype solutions	Code execution; text embedding	A–T	Embeddings (one task)
4	Chen et al. [13]	Geotechnical problem solving/education	GPT-4; zero-shot, CoT, custom prompts	Textbook questions → solutions	None	A	Conversation
5	Tophel et al. [19]	Geotechnical tutoring	GPT-4, Llama-3 via API, with RAG	391 textbook questions → answers	Retrieval	A	Vector-store RAG
6	Youwai et al. [22]	Shallow and pile foundation design	Router + expert agents; DeepSeek R1, GPT-4 Turbo, Grok 3, Gemini 2.5 Pro	Design problems → calculations	NR	M	NR
7	Goto et al. [65]	Site investigation decision support	LLM with GraphRAG over ontology	Borehole XML/CSV/JSON, core images → site answers, Web-GIS	Knowledge graph; spatial search; Web-GIS	T	Knowledge-graph RAG
8	Kamran et al. [83]	Mine slope stability classification	Gemini as coding assistant	Slope dataset → ML pipeline, predictions	Python/Colab (LLM-written code)	A	Conversation
9	Ni & Buehler [71]	Elasticity/FE mechanics	GPT-4 with AutoGen; 2-agent and 7-agent teams	Text tasks → FEniCS code, results, plots	Python execution; FEniCS	M	Conversation
10	Liu et al. [82]	Beam structural analysis	Llama-3.3-70B; code generation with CoT and few-shot	Text problems → OpenSeesPy code, results	OpenSeesPy	T	Prompt examples
11	Geng et al. [33] *	2D frame analysis	Llama-3.3-70B; five agents	Text problems → OpenSeesPy model, results	OpenSeesPy	M	NR
12	Liang et al. [32] *	Structural analysis problems	GPT-4o (vs. GPT-4, Gemini 1.5 Pro, Llama-3.3)	Text → OpenSeesPy scripts, results	OpenSeesPy	T	In-context only

Table 4. *Cont.*

#	Study	Domain	Architecture/LLM(s)	Input → Output	External Tools	Aut.	Memory
13	Liang et al. (MASSE) [21] *	Racking-system structural design	GPT-4o; analyst, engineer and management teams	Project documents → loads, analysis, capacity verdict	OpenSeesPy; RAG over codes	M	Structured JSON memory + RAG
14	Chen & Bao [70]	Code-compliant RC beam design	Coordinated multi-agent LLM	Natural-language queries → designs, explanations	Pre-approved calculation tools	M	NR
15	Choi et al. (RAMASC) [52]	Structural calculation incl. retaining walls	Retrieval-augmented multi-agent LLM	Design queries → executable Python calculations	Code-provision retrieval; Python	M	RAG
16	Mudur et al. (FEABench) [20] *	Multiphysics FEA benchmark	Gemini-1.5-Pro controller, corrector, verifier, tool-lookup agents	Physics problems → COMSOL API calls, target value	COMSOL API; snippet retrieval	M	Snippet RAG
17	Jiang et al. (SDA-Chat) [36]	Post-earthquake building damage	Fine-tuned vision-language model	Damage images, questions → damage interpretation	None	A	Conversation
18	Chen et al. (DisasTeller) [35]	Post-earthquake response and reporting	GPT-4o controlling four VLM agents (CrewAI); Gemma3-27B tested	Images, web data, EMS-98 → grading, alert maps, allocation, reports	Image analysis; web search; map annotation	M	Inter-agent memory
19	Estêvão [56]	EMS-98 damage grading	GPT-4o, GPT-4o mini, Gemini 1.5 Pro/Flash	Photos → damage grade, structural system	None	A	RAG variant tested
20	Nguyen et al. [86]	Vibration-based SHM	Prompt-engineered LLM	Vibration signals → insights, damage-detection program	In-house program	A–T	NR
21	Sharma & Bhowmik [44]	SHM (non-LLM agentic)	Perceive–reason–act–reflect loop; Bayesian inference; RL/POMDP	Vibration response → damage state, policy adaptation	Frequency-domain decomposition	C (simulated)	Policy state
22	Ren et al. [68] *	Seismic wave simulation	LLM agent with MCP server suite	Conversational intent → SPECFEM runs, plots	SPECFEM 2D/3D	T–C	NR
E1	Boiko et al. (Coscientist) [23]	Chemistry (exemplar)	GPT-4 planner with modules	Goals → protocols, robot commands, analyses	Web/doc search; Python; liquid-handling robot; cloud lab	C	Documentation retrieval
E2	Bran et al. (ChemCrow) [85]	Chemistry (exemplar)	GPT-4 with LangChain ReAct loop	Tasks → syntheses, analyses	18 expert tools; robotic synthesis platform	T–C	NR

Table 5. Comparative summary of LLM-based and agentic studies—evaluation characteristics. Evidence levels: (i) demonstrated in practice or real physical experiments; (ii) prototype/proof-of-concept; (iii) conceptual. "NR" = not reported in the sources consulted. * = preprint.

#	Study	Validation and Dataset Size	Uncertainty Treatment	Human Involvement	Main Limitations	Evidence
1	Kumar [12]	Worked examples; one hand-checked bearing case	None	Expert designs prompts and tools	Hallucination; arithmetic errors; context limits	(ii) demos in forum article
2	Shoemaker et al. [11]	4 tasks checked by hand; several calculation errors	None	Full human QC	Calculation errors	(ii)
3	Wu et al. [84]	4 problem demonstrations	NR	Expert oversight	Accuracy on specialized tasks	(ii)
4	Chen et al. [13]	Textbook answers; 28.9% zero-shot, 34% CoT, 67% custom prompts	None	Verification essential	Higher-order tasks; figures	(ii)
5	Tophel et al. [19]	391 questions (20-question hard subset); GPT-4 95% at temperature 0.1	Temperature sensitivity	Tutor setting	Formula handling	(ii)
6	Youwai et al. [22]	27 cases × 3 runs; 95.0% shallow, 90.6% piles	Repeated runs	Oversight for safety-critical work	Small sample	(ii)
7	Goto et al. [65]	Practical queries; GraphRAG vs. conventional RAG	Search guardrails	Engineer as user	NR	(ii)
8	Kamran et al. [83]	Standard ML metrics (99% accuracy reported)	NR	Human prompts each step	Accuracy claim needs independent check	(ii)
9	Ni & Buehler [71]	≈5 classical elasticity tasks	None	Hint needed to fix a conceptual error	Accepted a physically implausible result	(ii)
10	Liu et al. [82]	8-problem benchmark, repeated runs; >99%	Run-to-run reliability	User supplies problem	Beams only; one LLM	(ii)
11	Geng et al. [33] *	20 problems × 10 trials; >80% in most cases	Repeated trials	NR	NR	(ii)
12	Liang et al. [32] *	20 problems; GPT-4o 100%, Llama-3.3 30%	Generative stability examined	NR	Small benchmark	(ii)
13	Liang et al. (MASSE) [21] *	100 scenarios from real projects; 11-engineer timing study	NR	Restricted to research/education	Single domain	(ii)
14	Chen & Bao [70]	30 beams vs. SAP2000; 97% accuracy	Agent cross-checking	Engineer makes final decision	Beams only	(ii)
15	Choi et al. (RAMASC) [52]	95.5% (precast slabs), 91.3% (retaining walls)	Multi-agent cross-checking	NR	Two element types	(ii)

Table 5. *Cont.*

#	Study	Validation and Dataset Size	Uncertainty Treatment	Human Involvement	Main Limitations	Evidence
16	Mudur et al. (FEABench) [20] *	15 gold + 200 large problems; 2/15 valid targets	NR	None during runs	Licence; simple geometry	(ii)
17	Jiang et al. (SDA-Chat) [36]	8195 image–text pairs; 83.0% QA accuracy	NR	NR	NR	(ii)
18	Chen et al. (DisasTeller) [35]	5 runs per model; expert and LLM scoring; Wajima case	Non-determinism acknowledged	Human-in-the-loop decision support	Error propagation; weak spatial reasoning	(ii)
19	Estêvão [56]	EMS-98 images; 28.6–75.0% accuracy	Mean grade error	NR	Tiny dataset; possible leakage	(ii)
20	Nguyen et al. [86]	2 case studies from literature	NR	User knowledge in prompts	Discussed by authors	(ii)
21	Sharma & Bhowmik [44]	Simulated 5-storey building; ASCE benchmark	Bayesian inference	NR	Simulation only	(ii)
22	Ren et al. [68] *	Case studies vs. standard baselines	NR	Optional human-in-the-loop mode	NR	(ii)
E1	Boiko et al. [23]	6 tasks; couplings confirmed by GC-MS	Comparison with Bayesian optimization	Manual plate handling; one guiding prompt	Dual-use risk; subjective scoring	(i)–(ii)
E2	Bran et al. [85]	Robotic syntheses; 14 expert-rated use cases	NR	Expert evaluation	Tool quality; LLM evaluator unreliable	(i)–(ii)

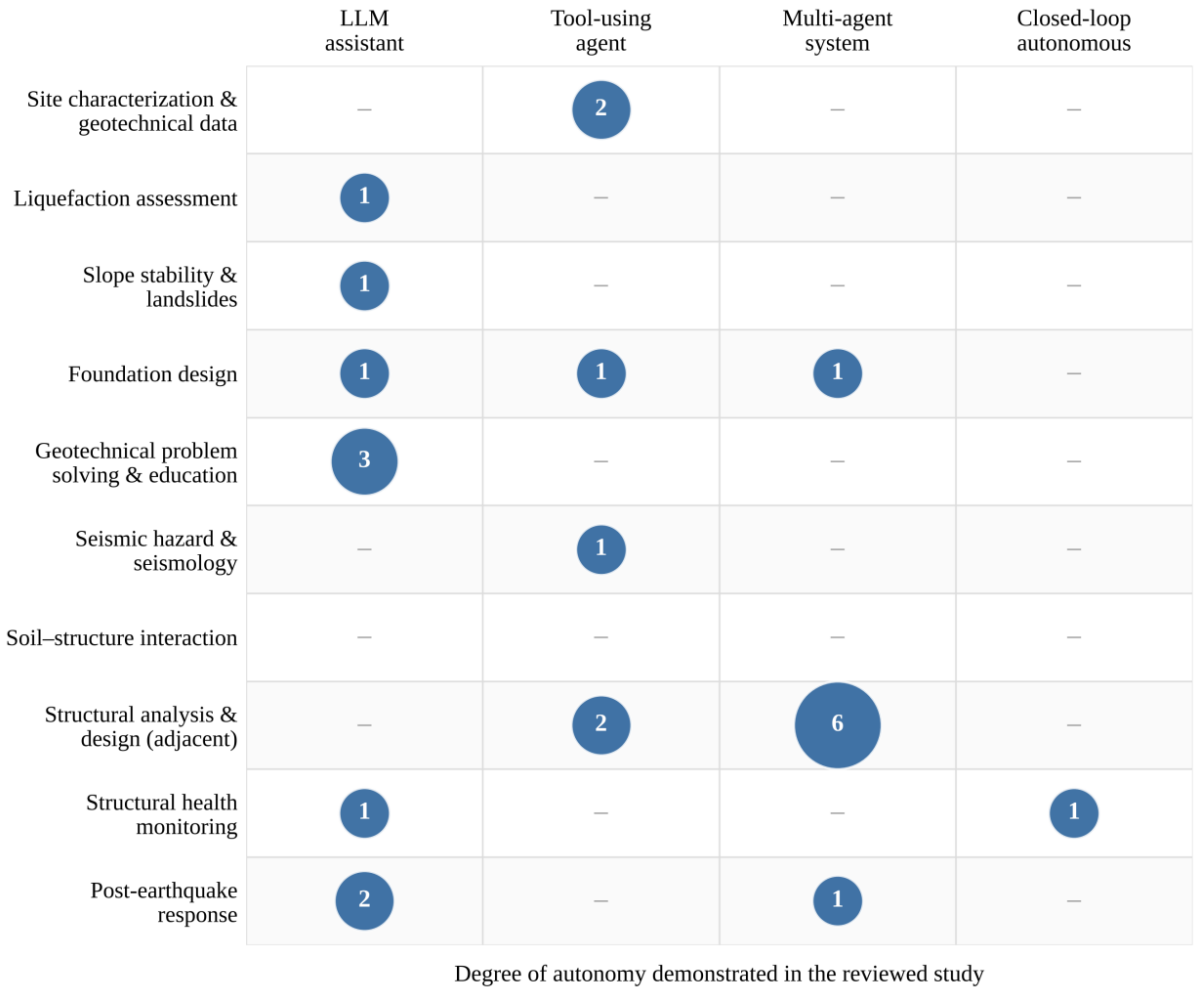


Figure 3. Evidence map of the reviewed studies by application area and degree of autonomy demonstrated. Numbers are counts of studies in Tables 4–5 (a study addressing two areas is counted in both). All are prototype/proof-of-concept evidence (level ii); "—" indicates that no peer-reviewed or preprint study was located and the capability remains conceptual (level iii).

5. Scope in Geotechnical Engineering

5.1. Site Characterization

Site investigation is an information-gathering problem conducted under budget and time constraints [1,87]. Geotechnical data are typically multivariate, uncertain, sparse, incomplete and sometimes corrupted, which complicates purely data-driven interpretation [87,88]. An agent could assist by comparing a prior geological model with stratigraphy inferred from sparse borings, proposing locations for additional CPTs, and interpreting test data through combined language and vision pipelines [65,84].

The established approach to the "active learning" step is Bayesian: a probabilistic model of the soil property field is updated with each measurement, and the next sounding is placed where an acquisition function is maximized. Bayesian compressive sensing reconstructs non-stationary soil profiles from sparse data and provides posterior uncertainty at unsampled locations [89]. Using the information entropy of that posterior as the acquisition function, Zhao et al. selected CPT locations within a multilayer cross-section that outperformed random placement in at least 90% of cases [90]; Wang and Li combined Voronoi diagrams with Bayesian compressive sampling to decide both how many samples to take and where [91]. Site-specific multivariate distributions learned from a generic database and sparse site data supply priors for such updating [92]. In an agentic workflow, the agent would not replace these methods; it would call them as tools, explain the proposed locations and their expected reduction in uncertainty, and submit the plan for approval. **Evidence status:** the Bayesian methods are established; tool-using agents for geotechnical data search and site-investigation queries are at level (ii) [12,65]; agent-driven exploration planning is conceptual (iii).

5.2. Liquefaction Assessment

Liquefaction triggering analysis applies empirical frameworks—from the simplified procedure of Seed and Idriss [93] to CPT-based methods [94–96]—under substantial parameter and model uncertainty. Machine-learning models have also been trained to predict liquefaction and lateral spreading from case histories [97,98]. An agentic workflow could cross-reference CPT-based soil behavior type with borehole grain-size data to resolve ambiguous classifications, run Monte Carlo simulations through probabilistic tools, generate explanations with sensitivity results, and list ground-improvement options for the engineer to evaluate. **Evidence status:** a general-purpose LLM has performed liquefaction triggering calculations with errors requiring human correction [11]; no tool-using liquefaction agent has been reported (iii). Section 7 provides an illustrative proof-of-concept of the tool layer.

5.3. Slope Stability and Landslide Forecasting

Landslide early-warning systems already combine monitoring data with alert thresholds [99], and methods for forecasting the time of slope failure from displacement data are well reviewed [100]. Bayesian updating can incorporate in-situ measurements into slope reliability in spatially variable soils [101,102], and Gaussian-process surrogates can replace costly limit-equilibrium analyses in system reliability assessment [103]. Interpretable closed-form factor-of-safety models have been developed for stabilized soils in mountainous terrain [7], and a physics-informed network embedding Newmark sliding-block physics has been proposed for earthquake-induced landslide susceptibility [104]. An agent could orchestrate these components—assimilating InSAR displacement and rainfall data, updating factor-of-safety estimates, and proposing staged alerts—but would need validated thresholds and human authority over warnings. **Evidence status:** an LLM has been used as a coding assistant to build a slope-classification model [83] (ii); agentic landslide forecasting is conceptual (iii).

5.4. Foundation Design

Deep-foundation design requires iterative checks of bearing capacity, settlement and structural capacity under several load cases. Predictive models such as optimized neural networks for pile capacity [5] can serve as tools within an agent, alongside load–settlement analyses and code checks. Agents can parse investigation reports to populate design parameters, iterate between analyses and code-compliance checks, and rank alternatives by reliability, constructability and embodied carbon. **Evidence status:** a router-based multi-agent system has performed shallow- and pile-foundation calculations with 90–95% accuracy on 27 cases [22], and an agent has called bearing-capacity tools [12] (ii). Retaining-wall calculations with retrieval-augmented multi-agent cross-checking are reported in the adjacent structural literature [52]. Design optimization over the full trade-off space remains conceptual (iii).

6. Scope in Earthquake Engineering

6.1. Seismic Hazard Characterization

Probabilistic seismic hazard analysis integrates seismological, geodetic and paleoseismic data to characterize earthquake recurrence, rupture and ground-motion attenuation [28]. Deep learning is now established in seismology for earthquake detection and phase picking [105,106]. An agent could query earthquake catalogs and fault databases, assemble logic trees of ground-motion models, run OpenQuake to compute hazard curves and uniform hazard spectra, and document each modeling choice. **Evidence status:** an LLM agent using MCP servers has configured and run SPECFEM wave-propagation simulations (ii) [68]; we found no agent that performs probabilistic seismic hazard analysis (iii).

6.2. Soil–Structure Interaction

Soil–structure interaction analysis couples geotechnical and structural models whose assumptions must be consistent [26]. Physics-informed and physics-guided networks have learned nonlinear structural seismic response from limited data [107,108], and nonlinear single-degree-of-freedom dynamics, including negative-stiffness behavior, have been examined as a basis for machine-learning models [109]. LLM agents already generate and run OpenSeesPy models of beams and frames [32,33,82]. Extending this to soil–structure interaction would require the agent to translate structural demands into soil springs or continuum meshes, run parametric sweeps of soil stiffness and damping, and identify governing scenarios. **Evidence status:** structural analysis agents are at level (ii); agentic soil–structure interaction modeling is conceptual (iii).

6.3. Structural Health Monitoring and Signal Processing

Structural health monitoring (SHM) generates continuous vibration data in which changes in modal properties may indicate damage; machine learning and computer vision are now widely used for damage detection and inspection [110,111]. A prerequisite for any automated assessment is reliable signal processing: strong-motion and shaking-table records require filtering and baseline correction, whose choices affect response spectra and displacements [112,113]. Deep-learning denoising separates earthquake signal from noise [114], convolutional networks have automated the selection of high-pass cut-off frequencies [115], and Bayesian-optimized LSTM networks have been benchmarked against tuned signal-processing pipelines for noise suppression in shaking-table acceleration records [116]. Such validated processing steps are natural tools for an agent's perception layer. **Evidence status:** a prompt-engineered LLM framework has been demonstrated for vibration-based SHM on two case studies [86], and a non-LLM agentic framework with policy adaptation has been tested on simulated and benchmark structures [44] (ii). Agents that autonomously dispatch inspection resources remain conceptual (iii).

6.4. Post-Earthquake Response and Resilience Planning

After an earthquake, decision makers must combine remote sensing, field reports and sensor data under severe time pressure; datasets such as xBD support building-damage assessment from satellite imagery [117]. Vision-language models fine-tuned on damage images reached 83% question-answering accuracy [36], but general-purpose models graded EMS-98 damage with only 28.6–75.0% accuracy [56]. The multi-agent DisasTeller system produced damage grading, alert maps, resource allocation and recovery reports for an earthquake case within minutes, but its authors report error propagation between agents and weak spatial reasoning [35]. **Evidence status:** (ii). Agentic systems for inspection-route prioritization and recovery simulation remain conceptual (iii).

7. Illustrative Proof-of-Concept: Tool Layer for CPT-Based Liquefaction Assessment

7.1. Purpose and Scope

To make the proposed workflow concrete, we implemented the tool layer of an agentic liquefaction assessment and executed it on a synthetic CPT profile. The purpose is to show what the agent would be asked to do, which tools it would call, what they return, and where human approval gates sit. The orchestration logic is scripted (rule-based) to represent the plan an LLM planner would follow; the LLM component itself is not evaluated here, and the results are not a benchmark of agent performance. The site is hypothetical.

7.2. Data and Tools

The synthetic CPT profile extends to 15 m with readings every 0.05 m (300 readings) and the groundwater table at 1.5 m. It comprises a silty clay crust (0–1.5 m), loose silty sand (1.5–6.0 m, $q_t \approx 3$–5 MPa), a transitional sandy silt (6.0–6.8 m), clayey silt (6.8–8.5 m) and medium-dense to dense sand (8.5–15 m, $q_t \approx 16$–22 MPa), with log-normal spatial noise (coefficient of variation ≈ 8% on q_t and 15% on friction ratio). The design scenario is a moment magnitude $M_w = 7.8$ event with peak ground acceleration $a_{max} = 0.25$ g. Four tools were implemented in Python: (1) a soil behavior type index (I_c) calculator with stress-normalization iteration [118]; (2) the Boulanger and Idriss [96] CPT triggering procedure, including fines-content estimation from I_c, the depth-reduction factor, magnitude scaling factor and overburden correction; (3) a Monte Carlo engine; and (4) the liquefaction potential index (LPI) of Iwasaki et al. [119]. Soils with $I_c > 2.6$ were screened as not susceptible to liquefaction, and I_c between 2.4 and 2.6 was flagged as ambiguous. The Monte Carlo analysis (N = 5000) treated a_{max} as log-normal (median 0.25 g, logarithmic standard deviation 0.30), groundwater depth as uniform between 1.0 and 2.5 m, and model uncertainty in cyclic resistance as normal in log space with $\sigma_{lnR} = 0.20$, using the median (50% probability) triggering curve [96].

7.3. Workflow Trace and Results

Table 6 lists the steps, tools and outputs, and Figure 4 shows the profiles. The deterministic analysis identified a continuous liquefiable layer from 1.55 to 6.05 m (all 90 readings with factor of safety FS < 1; minimum 0.56), liquefiable readings within the transitional layer (minimum FS 0.47), and two isolated readings with FS < 1 in the dense sand at 8.80 and 9.45 m. The deterministic LPI was 13.1. With uncertainty propagated, the median LPI was 6.3 (10th–90th percentile 0.1–18.7); the probability that LPI exceeds 5 was 0.55 and that it exceeds 15 was 0.19. The deterministic value exceeds the Monte Carlo median because the deterministic curve corresponds to a 16%

probability of liquefaction, whereas the Monte Carlo analysis uses the median curve with explicit model uncertainty [96]. The full analysis, including 5000 realizations, ran in about 4 s on a standard laptop processor.

Table 6. Workflow trace of the illustrative proof-of-concept. Steps 3 and 7 are human approval gates; the agent proposes but does not decide.

Step	Agent Task	Tool	Output on the Synthetic Profile	Gate
1	Ingest and check CPT data	Parser and quality checks	300 readings, 0–15 m; no gaps; groundwater table 1.5 m	—
2	Classify soil behavior	I_c calculator [118]	Mean I_c: 2.10 (1.5–6.0 m), 2.57 (6.0–6.8 m), 2.91 (6.8–8.5 m), 1.59 (8.5–15 m); 0.45 m cumulative with $2.4 < I_c \leq 2.6$	—
3	Resolve ambiguous classification	Rule check; borehole cross-reference	Transitional layer flagged; agent recommends sampling for fines content and plasticity instead of assuming susceptibility	Engineer: treat as susceptible pending laboratory data
4	Deterministic triggering	Boulanger–Idriss CPT procedure [96]	FS < 1 continuously at 1.55–6.05 m (min 0.56) and within the transitional layer (min 0.47); two isolated readings in dense sand flagged as possible thin-layer or transition effects; LPI = 13.1 [119]	—
5	Propagate uncertainty	Monte Carlo engine (N = 5000)	LPI median 6.3 (P10–P90: 0.1–18.7); P(LPI > 5) = 0.55; P(LPI > 15) = 0.19	—
6	Verify	Applicability and consistency checks	q_{c1Ncs} within method limits; deterministic–probabilistic difference explained; FS capped at 2.0 for reporting	—
7	Report and propose options	Report generator	Ground-improvement options for 1.5–6.8 m listed with the evidence and its uncertainty; no design change executed	Engineer: select option or request further investigation

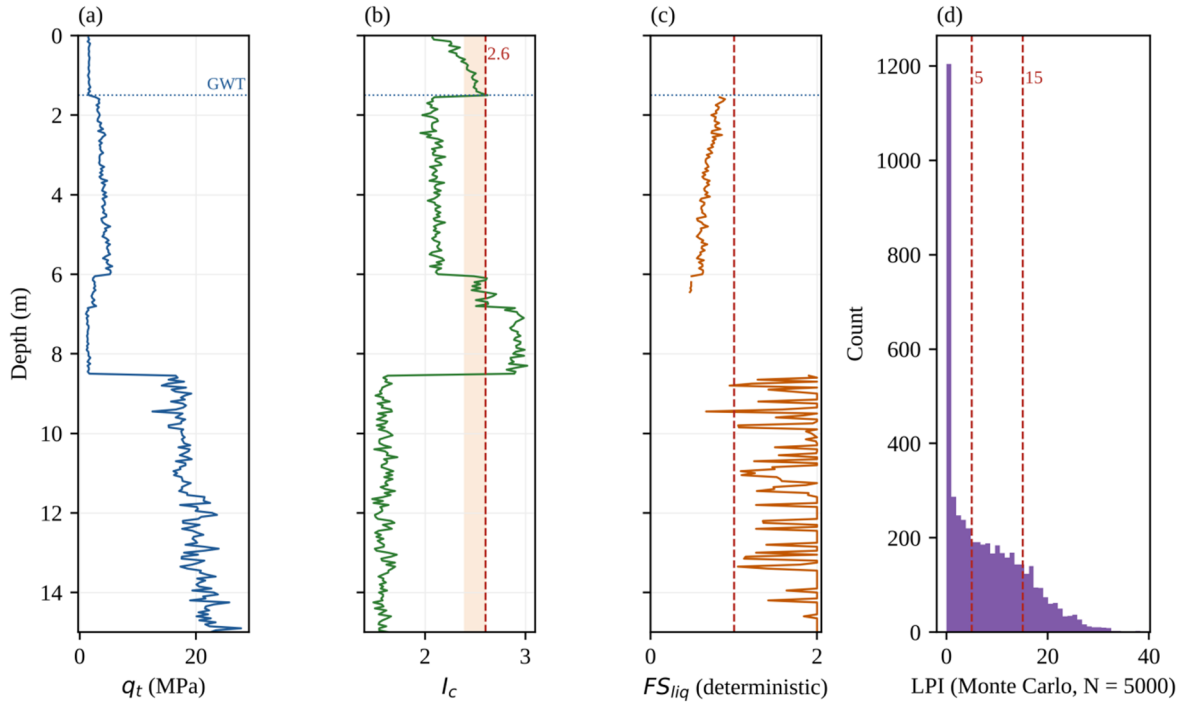


Figure 4. Illustrative proof-of-concept on a synthetic CPT profile: (**a**) corrected cone resistance; (**b**) soil behavior type index with the 2.6 susceptibility limit and the 2.4–2.6 ambiguous band; (**c**) deterministic factor of safety against liquefaction for susceptible soils below the groundwater table (GWT); (**d**) distribution of LPI from 5000 Monte Carlo realizations, with 5 and 15 marked.

The example illustrates three points. First, every quantitative result comes from a deterministic, verifiable tool; the role of the language model would be limited to planning, routing, explanation and reporting. Second, the uncertainty analysis changes the message for the engineer: the deterministic LPI suggests high severity, whereas the probabilistic result indicates roughly even odds of exceeding moderate severity. An agent that reports only one of these would mislead. Third, two decisions—how to treat the ambiguous layer and which mitigation to adopt—remain with the engineer. A full evaluation of an LLM planner on this workflow, using real CPT soundings with documented field performance, is proposed as future work (gaps G1 and G3).

8. Critical Challenges

8.1. Data Heterogeneity and Scarcity (C1)

Geotechnical data are sparse, site-specific and multimodal [87]. Unlike computer-vision datasets, geotechnical records lack consistent ontologies, borehole log formats vary between organizations, and measurements are often missing or censored, as when cone refusal limits penetration depth. The high cost of in-situ and laboratory testing means that many projects have little subsurface information [2]. Data standards (AGS4, DIGGS) [62,63], knowledge graphs (Section 3.5), federated learning, Bayesian methods that combine generic databases with site data [92], and data-centric practices [88] provide partial solutions.

8.2. Epistemic Uncertainty and Model-Form Error (C2)

Geomechanics involves ill-posed inverse problems in which constitutive model selection and spatial variability introduce large epistemic uncertainty [1]. An agent must represent uncertainty in model choice (for example, Mohr–Coulomb versus hardening-soil models), in spatial variability through random-field or Bayesian methods [89,101], and in empirical correlations such as liquefaction triggering curves [96]. Monte Carlo robustness checks of predictive models, as in recent soil-stabilization work [6], are one component. The illustrative example in Section 7 shows how omitting uncertainty can overstate severity. LLMs also introduce their own uncertainty—run-to-run variability and sensitivity to prompt wording and temperature [19,82]—which must be measured rather than assumed away. Where hazards must be ranked under uncertainty, structured multi-criteria frameworks, such as that developed for rockburst in underground construction [120], give agents a transparent and auditable basis for prioritization.

8.3. Physics Compliance, Hallucination and Physics-Informed Modeling (C3)

LLMs can produce physically inconsistent outputs: in the MechAgents study, an agent plotted the wrong stress component and accepted a physically implausible result until a human intervened [71], and general-purpose LLMs made calculation errors in geotechnical tasks [11,13]. The principal safeguard is architectural: quantitative results should come from verified tools, and every tool output should be checked against physical bounds and code requirements. Physics-informed machine learning offers complementary tools [121,122], but the term covers distinct approaches with different strengths (Table 7).

Table 7. Physics-informed and hybrid modeling approaches and their role in geotechnical and earthquake-engineering agents.

Approach	Principle	Advantages	Limitations	Examples in Geotechnical/ Earthquake Engineering
Physics-informed neural networks (PINNs)	Governing PDE residuals added to the loss function [123]	Forward and inverse problems with sparse data; parameter identification	Training difficulty for stiff, multiscale, nonlinear problems; cost per new geometry	1D consolidation and coefficient identification [124]; elastodynamics [125]; wave propagation and inversion [126]; landslides (preprint) [104]
Physics-guided ML	Physical knowledge in architecture, loss or features [127,128]	Better extrapolation and data efficiency than black-box models	Constraints often soft; guarantees limited	PhyCNN and PhyLSTM for structural seismic response [107,108]

Table 7. *Cont.*

Approach	Principle	Advantages	Limitations	Examples in Geotechnical/ Earthquake Engineering
Hybrid mechanistic–data-driven models	Mechanistic model plus learned correction or Bayesian updating [128]	Retains interpretability of mechanics; corrects model-form error	Requires a reasonable mechanistic baseline	Bayesian updating of slope reliability with measurements [101]
Surrogate models	Fast emulator trained on simulator runs (GPR, neural operators)	Enables Monte Carlo and optimization at low cost	Valid only within training domain; no physics guarantee	GPR for slope system reliability [103]; DeepONet for nonlinear FE failure response [129]
Differentiable simulators	Numerical solver with automatic differentiation [130]	Exact physics; gradients for inversion and design	Few geotechnical constitutive models implemented	No geotechnical application identified
Neuro-symbolic approaches	Neural learning combined with symbolic rules or knowledge [131]	Explicit constraints and traceable reasoning	Knowledge engineering effort; scalability	LLM with ontology-based knowledge graph for site investigation [65]

For agentic workflows, these approaches play different roles: surrogates and physics-guided models are fast tools that an agent can call many times; PINNs and differentiable simulators support inverse analysis and parameter identification; hybrid models allow monitoring data to correct mechanistic predictions; and neuro-symbolic methods constrain the agent's reasoning itself. Interpretable closed-form models, such as those developed for stabilized slopes [7], are also valuable because their outputs can be checked by hand. None of these approaches alone guarantees physics-compliant agent behavior.

8.4. Verification, Validation and Trust (C4)

Civil infrastructure decisions carry life-safety implications that demand rigorous verification and validation (V&V) [132]. Verification asks whether computations are implemented correctly; validation asks whether predictions agree with experimental and field observations; uncertainty quantification asks whether stated confidence is calibrated. For agentic AI these concepts must be extended to the agent's decision logic, tool selection and stopping criteria. Practical elements include held-out benchmark problems with known answers, repeated runs to measure variability [22,82], independent checking agents or tools [52,70], complete audit logs, and comparison against independent software, as in the comparison of agent designs with SAP2000 [70]. Field validation against documented case histories has not yet been reported.

8.5. Ethical, Legal and Safety Considerations (C5)

Autonomous design agents challenge professional liability frameworks that assume direct human responsibility. We therefore recommend that the engineer of record remain accountable for every design decision and that agents be configured to propose, not to approve. Three further issues arise. **Accountability in post-earthquake response:** decisions on building occupancy, evacuation or resource allocation must be traceable to a named authority; agent outputs should be advisory, logged and reversible, as in the human-in-the-loop design of DisasTeller [35]. **Compliance with design codes:** agent outputs must comply with applicable standards—for example EN 1998-5 for foundations and geotechnical aspects in Europe [49], ASCE 7-22 in the United States [50] and NBC 105:2020 in Nepal [51]—and codes should be implemented as validated, version-controlled calculation tools rather than paraphrased by the language model. **Risk-management and regulatory frameworks:** the NIST AI Risk Management Framework [133] and the EU Artificial Intelligence Act, which imposes human-oversight obligations on high-risk systems [134], provide reference points. Equitable deployment in resource-constrained regions and protection against adversarial manipulation—poisoned data, prompt injection or compromised tool interfaces—are further concerns [8].

8.6. Model Drift and Lifecycle Maintenance (C8)

Deployed agents face drift at several levels: the statistical properties of incoming data may change (new sensors, new soil conditions, post-earthquake changes in structural behavior); the underlying language model may be updated or withdrawn by its provider; and the retrieval corpus may change as codes are revised. Concept-drift detection and adaptation methods [41,42] should monitor input distributions and tool-output residuals, and any change of model version should trigger re-validation on the benchmark suite (Table 2). Pinning model versions and archiving prompts and corpora are necessary for reproducibility.

8.7. Computational Cost, Latency and Interoperability (C6, C7)

Workflows involving repeated finite-element simulations, stochastic sampling and LLM inference can be expensive at scale. Reported latencies range from minutes for design and reporting workflows to about 12 min per problem when an agent drives a finite-element package (Table 3), which is acceptable for design but not for earthquake early warning, where decisions are needed within seconds. Time-critical functions should therefore be implemented as pre-validated, low-latency models, with the agent operating in a supervisory role outside the critical path. Interoperability and licensing issues are discussed in Section 3.6.

8.8. Challenge Assessment

Table 8 and Figure 5 summarize the eight challenges. The scores are ordinal values defined by an explicit rubric, with a justification for each. They are the authors' qualitative assessment informed by the literature reviewed, not measured quantities.

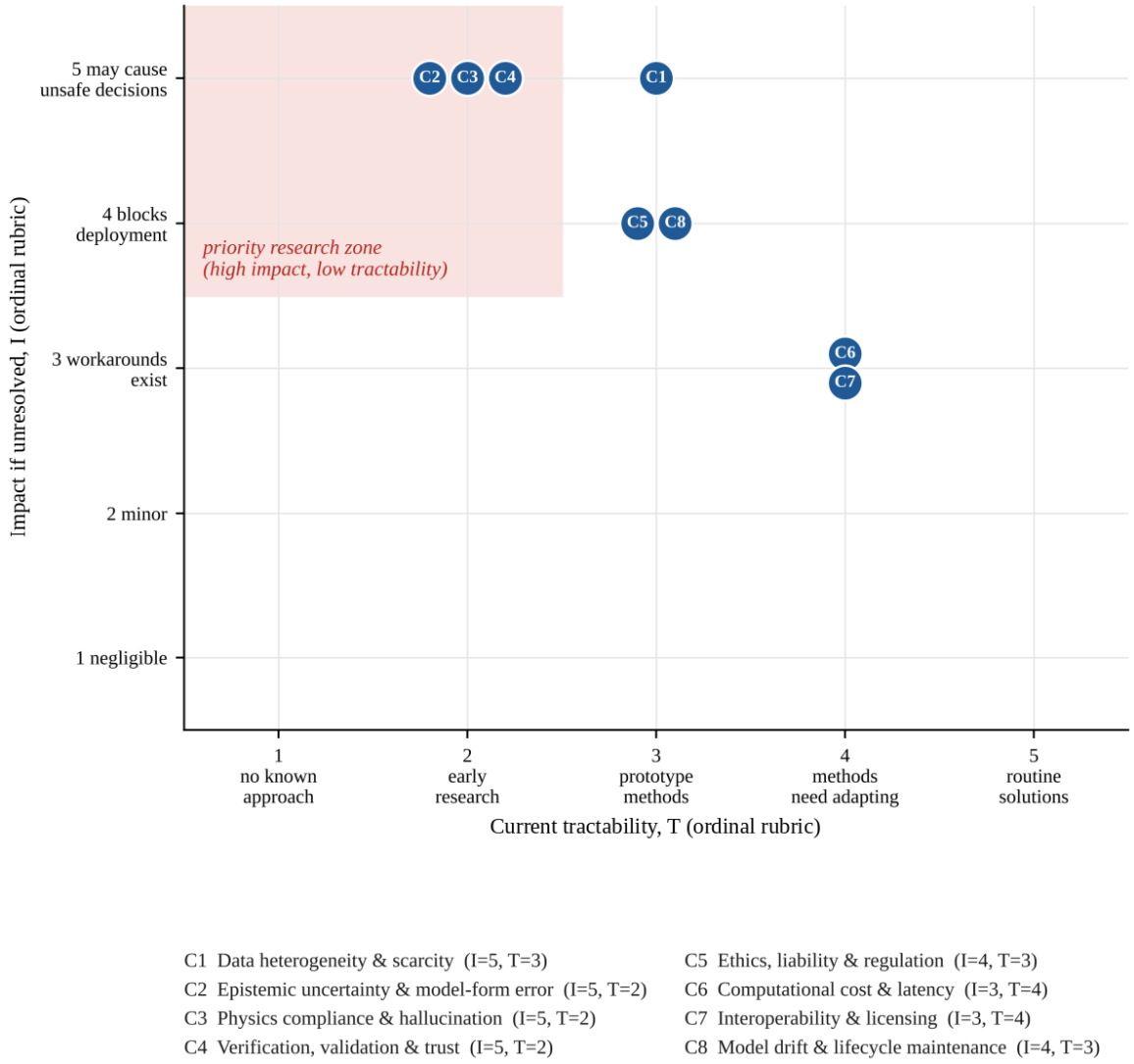


Figure 5. Ordinal assessment of challenges (impact if unresolved versus current tractability) using the rubric of Table 8. Scores are the authors' qualitative assessment, not measurements.

Table 8. Ordinal assessment of challenges. Impact (I): 1 negligible; 2 minor; 3 increases effort but workarounds exist; 4 blocks deployment in practice; 5 could directly cause an unsafe engineering decision. Tractability (T): 1 no known approach; 2 early research; 3 methods demonstrated in prototypes; 4 methods exist but need adaptation; 5 routine solutions.

Challenge	I	T	Justification
C1 Data heterogeneity and scarcity	5	3	Sparse or misread data propagate directly into design parameters; standards and Bayesian methods exist but are not linked to agents [62,63,87]
C2 Epistemic uncertainty and model-form error	5	2	Overconfident outputs are unsafe (Section 7); uncertainty propagation through agent decisions is not yet studied
C3 Physics compliance and hallucination	5	2	Physically implausible results accepted by agents [71]; PIML methods immature for multiphysics problems [122]
C4 Verification, validation and trust	5	2	No V&V protocol or field validation for engineering agents; benchmarks are small (Table 5)
C5 Ethics, liability and regulation	4	3	Liability frameworks absent, but governance frameworks exist [133,134]
C6 Computational cost and latency	3	4	Costs of US$0.03–1.4 per run (Table 3); latency limits real-time use but supervisory roles are feasible
C7 Interoperability and licensing	3	4	APIs and exchange standards exist [24,27,67]; integration effort and licence terms remain
C8 Model drift and lifecycle maintenance	4	3	Drift methods exist [41,42]; provider model updates are outside the user's control

9. Research Gaps and a Conditional Roadmap

Table 9 lists eight research gaps derived from Sections 3–8. Figure 6 arranges them into three horizons separated by readiness gates. Following technology-readiness practice [135,136] and scenario planning [137], the roadmap is a conceptual scenario: progression to the next horizon is conditional on meeting each gate, not on calendar dates, and its pace will depend on regulation, certification and infrastructure readiness.

Table 9. Research gaps, linked challenges and roadmap horizon.

Gap	Description	Challenge	Evidence (Section)	Horizon
G1	No open benchmarks or curated case-history datasets for agentic geotechnical reasoning	C1, C4	4, 7	Short term
G2	No shared geotechnical ontology linking AGS4/DIGGS data to knowledge graphs and RAG corpora	C1, C7	3.4, 3.5	Short term
G3	No field-validated agent deployment; no agent operating a commercial geotechnical FE package	C4, C7	3.6, 4	Short → medium
G4	Uncertainty is not propagated through agent decisions	C2	7, 8.2	Medium term
G5	No systematic physics-consistency checking of agent outputs	C3	8.3	Medium term
G6	No verification, validation or certification protocol for engineering agents; no drift monitoring	C4, C8	8.4, 8.6	Medium → long
G7	No liability and governance framework specific to engineering agents	C5	8.5	Long term
G8	Cost and latency rarely reported; real-time use unsupported	C6	3.7, 8.7	Short term

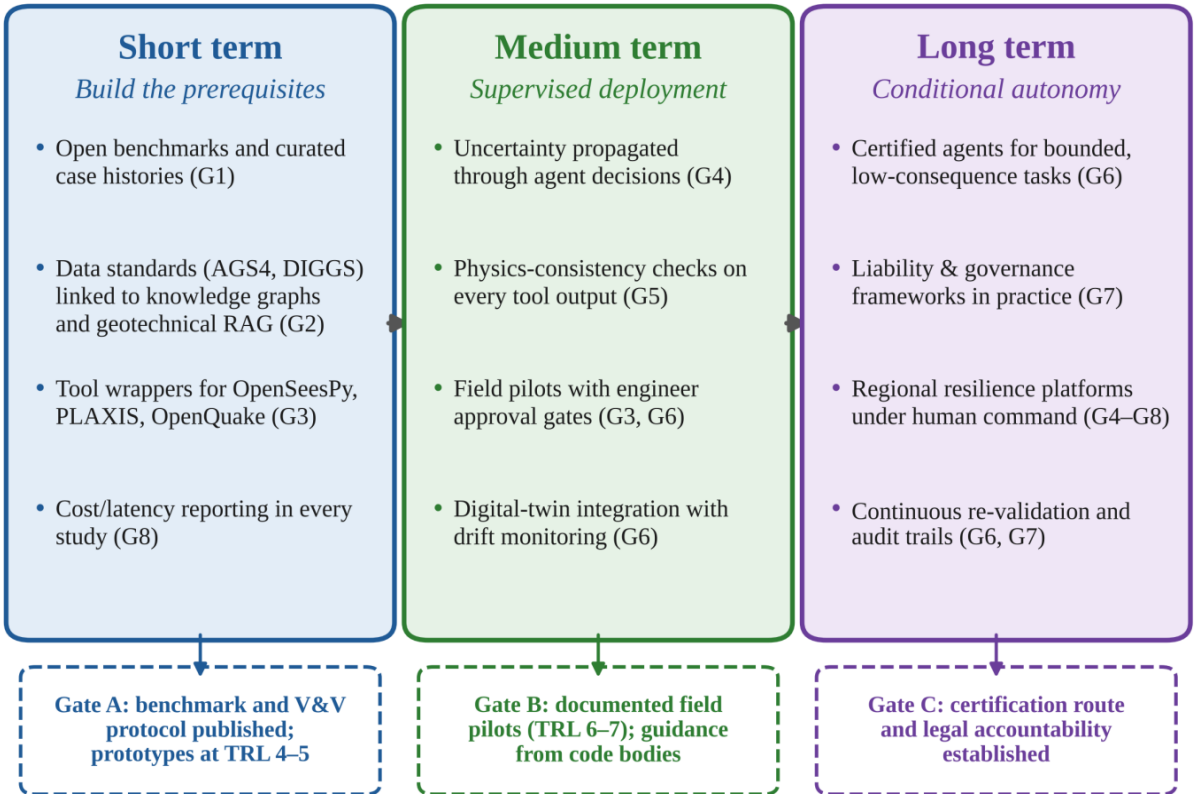


Figure 6. Conditional roadmap for agentic AI in geotechnical and earthquake engineering, organized by short-, medium- and long-term horizons linked to gaps G1–G8 and separated by readiness gates. G1–G8 are the research gaps listed in Table 9. The roadmap is a conceptual scenario, not a forecast; progression between horizons is conditional on passing each readiness gate.

9.1. Short Term: Building the Prerequisites

The field needs standardized benchmarks for agentic geotechnical reasoning, analogous to those in software engineering and multiphysics simulation [20,76], built from curated case histories with documented field performance (G1). Data standards should be connected to knowledge graphs and RAG corpora (G2), and validated tool wrappers should be developed for OpenSeesPy, PLAXIS and OpenQuake (G3). Every study should report cost and latency alongside accuracy (G8) [75]. Domain-specific foundation models for geomechanics are a possible longer-term development, but the evidence reviewed suggests that retrieval, tools and domain adaptation of existing models [55] offer a nearer path. *Gate A:* a public benchmark and V&V protocol exist, and prototypes reach technology readiness levels (TRL) 4–5.

9.2. Medium Term: Supervised Deployment

In this horizon, agents would operate in field pilots under engineer approval gates. Uncertainty should be propagated through agent decisions (G4), physics-consistency checks applied to every tool output (G5), and agents integrated with geotechnical digital twins that assimilate monitoring data [138–140], with drift monitoring (G6). Human–agent collaborative design—agents proposing, critiquing and documenting alternatives while engineers decide—is the expected mode of use. *Gate B:* documented field pilots (TRL 6–7) and guidance from code bodies on the use of AI tools.

9.3. Long Term: Conditional Autonomy

Only after these gates are passed would certified agents operate with greater autonomy, and then only for bounded, low-consequence tasks (G6). The vision of regional resilience platforms that assimilate seismic, geodetic and structural data in real time and support emergency decisions remains plausible, but it requires sensor infrastructure, communication networks, legal accountability frameworks (G7) and continuous re-validation; such platforms should remain under human command. *Gate C:* an established certification route and legal accountability for agent-supported decisions.

10. Conclusions

This review has examined the scope, challenges and future directions of agentic AI in geotechnical and earthquake engineering, distinguishing demonstrated capabilities from conceptual ones. The main conclusions are as follows.

- Agentic AI is best defined by a graded taxonomy—from predictive ML through retrieval-augmented and tool-using systems to multi-agent and closed-loop agents—rather than a binary contrast with predictive AI. Chain-of-thought prompting is not a defining property, and a feedback loop does not by itself make an agent self-improving.
- All 22 civil-engineering studies identified are prototypes or proof-of-concept systems (evidence level ii). The most autonomous systems are in structural analysis and design; geotechnical applications are mainly assistant-level, with tool use demonstrated for data search, site-investigation queries and foundation design. Agentic seismic hazard analysis, soil–structure interaction modeling and landslide forecasting remain conceptual.
- Reliable engineering agents depend on enabling layers: retrieval over version-controlled codes and records, knowledge graphs built on existing data standards, typed interfaces to validated engineering software, and physics-informed or physics-checked tools. Quantitative results should come from verified tools, not from the language model.
- Reported cost and latency are modest for design tasks (indicatively US$0.03–1.4 per workflow run in API fees, minutes per task) but rule out agents from the critical path of real-time warning systems.
- The proof-of-concept shows that uncertainty propagation can materially change the conclusion of a liquefaction assessment and that approval gates can be placed where engineering judgment is required.
- The highest-priority gaps are open benchmarks and case-history datasets, uncertainty propagation through agent decisions, physics-consistency checking, and verification, validation and certification protocols.

In its current state, agentic AI can add value as a supervised orchestration layer that retrieves evidence, runs validated analyses and documents its reasoning for engineers. Claims of autonomous geotechnical or seismic engineering practice are not yet supported by evidence; progress toward greater autonomy should be conditional on the readiness gates proposed here.

Author Contributions

I.T.: literature work; R.S. and G.S.: conceptualization; S.A.: methodology and investigation; P.R.M.: original draft preparation; B.P.: original draft preparation, review, editing and finalization. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

No new experimental data were generated. The synthetic CPT profile in Section 7 is fully specified in Section 7.2 (random seed 2026), and the Python scripts used for the proof-of-concept and figures are available from the corresponding author on request.

Acknowledgments

The authors gratefully acknowledge the data repositories, digital libraries, and bibliographic databases that provided access to the scholarly literature essential for conducting this review. Without the comprehensive indexing and archiving services provided by these organizations, the identification and retrieval of relevant studies would not have been possible.

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

During the preparation of this work, the authors used large language model (LLM)-based tools to improve the clarity, quality and grammatical accuracy of the English text. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

1. Phoon, K.K.; Zhang, W. Future of machine learning in geotechnics. *Georisk Assess. Manag. Risk Eng. Syst. Geohazards* **2023**, *17*, 7–22. https://doi.org/10.1080/17499518.2022.2087884.
2. Zhang, W.; Zhang, R.; Wu, C.; et al. State-of-the-art review of soft computing applications in underground excavations. *Geosci. Front.* **2020**, *11*, 1095–1106. https://doi.org/10.1016/j.gsf.2019.12.003.
3. Zhang, W.; Li, H.; Li, Y.; et al. Application of deep learning algorithms in geotechnical engineering: A short critical review. *Artif. Intell. Rev.* **2021**, *54*, 5633–5673. https://doi.org/10.1007/s10462-021-09967-1.
4. Xie, Y.; Sichani, M.E.; Padgett, J.E.; et al. The promise of implementing machine learning in earthquake engineering: A state-of-the-art review. *Earthq. Spectra* **2020**, *36*, 1769–1801. https://doi.org/10.1177/8755293020919419.
5. Khatti, J.; Kontoni, D.P.N. Assessment of Bearing Capacity of Concrete Piles in Alluvial Soils Using Bio and Swarm-Optimized Artificial Neural Network Models. *Bull. Comput. Intell.* **2025**, *1*, 53–75. https://doi.org/10.53941/bci.2025.100004.
6. Thapa, I.; Ghani, S. AI-Enabled Sustainable Soil Stabilization for Resilient Urban Infrastructure: Advancing SDG 9 and SDG 12 through Hybrid Deep Learning and Environmental Assessment. *Bull. Comput. Intell.* **2025**, *1*, 3–30. https://doi.org/10.53941/bci.2025.100002.
7. Thapa, I.; Ghani, S. Development of Interpretable Regression Models for Slope Stability of Nano-Silica Stabilised Soils in Mountainous Terrain. *Bull. Comput. Intell.* **2026**, *2*, 103–133. https://doi.org/10.53941/bci.2026.100006.
8. Xi, Z.; Chen, W.; Guo, X.; et al. The rise and potential of large language model based agents: A survey. *Sci. China Inf. Sci.* **2025**, *68*, 121101. https://doi.org/10.1007/s11432-024-4222-0.
9. Wang, L.; Ma, C.; Feng, X.; et al. A survey on large language model based autonomous agents. *Front. Comput. Sci.* **2024**, *18*, 186345. https://doi.org/10.1007/s11704-024-40231-1.
10. Yao, S.; Zhao, J.; Yu, D.; et al. ReAct: Synergizing reasoning and acting in language models. In Proceedings of the International Conference on Learning Representations (ICLR 2023), Kigali, Rwanda, 1–5 May 2023.
11. Shoemaker, T.A.; Beaino, C.; Centella, D.; et al. Generative AI: The New Geotechnical Assistant? *J. Geotech. Geoenviron. Eng.* **2023**, *149*, 02823004. https://doi.org/10.1061/jggefk.gteng-11859.
12. Kumar, K. Geotechnical Parrot Tales (GPT): Harnessing Large Language Models in Geotechnical Engineering. *J. Geotech. Geoenviron. Eng.* **2024**, *150*, 02523001. https://doi.org/10.1061/jggefk.gteng-11828.
13. Chen, L.; Tophel, A.; Hettiyadura, U.; et al. An Investigation into the Utility of Large Language Models in Geotechnical Education and Problem Solving. *Geotechnics* **2024**, *4*, 470–498. https://doi.org/10.3390/geotechnics4020026.
14. Sapkota, R.; Roumeliotis, K.I.; Karkee, M. AI Agents vs. Agentic AI: A Conceptual taxonomy, applications and challenges. *Inf. Fusion* **2026**, *126*, 103599. https://doi.org/10.1016/j.inffus.2025.103599.
15. Feng, K.J.K.; McDonald, D.W.; Zhang, A.X. Levels of autonomy for AI agents. *arXiv* **2025**, arXiv:2506.12469.
16. Morris, M.R.; Sohl-Dickstein, J.; Fiedel, N.; et al. Position: Levels of AGI for Operationalizing Progress on the Path to AGI. In Proceedings of the International Conference on Machine Learning, Vienna, Austria, 21–27 July 2024; pp. 36308–36321.
17. Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*, 4th ed.; Pearson: Hoboken, NJ, USA, 2020.
18. Wei, J.; Wang, X.; Schuurmans, D.; et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022; Volume 35, pp. 24824–24837. https://doi.org/10.52202/068431-1800.
19. Tophel, A.; Chen, L.; Hettiyadura, U.; et al. Towards an AI tutor for undergraduate geotechnical engineering: A comparative study of evaluating the efficiency of large language model application programming interfaces. *Discov. Comput.* **2025**, *28*, 76. https://doi.org/10.1007/s10791-025-09580-8.
20. Mudur, N.; Cui, H.; Venugopalan, S.; et al. FEABench: Evaluating language models on multiphysics reasoning ability. *arXiv* **2025**, arXiv:2504.06260.
21. Liang, H.; Zhou, Y.; Kalaleh, M.T.; et al. Automating Structural Engineering Workflows with Large Language Model Agents. *arXiv* **2025**, arXiv:2510.11004.
22. Youwai, S.; Phim, D.; Murcia, V.G.; et al. Large language model-based multi-agent systems for automated foundation design: Router-driven task classification and expert selection framework. *AI Civ. Eng.* **2026**, *5*, 5. https://doi.org/10.1007/s43503-026-00088-8.

23. Boiko, D.A.; MacKnight, R.; Kline, B.; et al. Autonomous chemical research with large language models. *Nature* **2023**, *624*, 570–578. https://doi.org/10.1038/s41586-023-06792-0.
24. Zhu, M.; McKenna, F.; Scott, M.H. OpenSeesPy: Python library for the OpenSees finite element framework. *SoftwareX* **2018**, *7*, 6–11. https://doi.org/10.1016/j.softx.2017.10.009.
25. Park, J.S.; O'Brien, J.; Cai, C.J.; et al. Generative Agents: Interactive Simulacra of Human Behavior. In Proceedings of the UIST'23: The 36th Annual ACM Symposium on User Interface Software and Technology, San Francisco, CA, USA, 29 October–1 November 2023; pp. 1–22. https://doi.org/10.1145/3586183.3606763.
26. McKenna, F. OpenSees: A Framework for Earthquake Engineering Simulation. *Comput. Sci. Eng.* **2011**, *13*, 58–66. https://doi.org/10.1109/mcse.2011.66.
27. Bentley Systems. Using PLAXIS Remote Scripting with the Python Wrapper. 2024. Available online: https://communities.bentley.com/products/geotech-analysis/w/wiki/46005/using-plaxis-remote-scripting-with-the-python-wrapper (accessed on 10 October 2026).
28. Pagani, M.; Monelli, D.; Weatherill, G.; et al. OpenQuake Engine: An Open Hazard (and Risk) Software for the Global Earthquake Model. *Seismol. Res. Lett.* **2014**, *85*, 692–702. https://doi.org/10.1785/0220130087.
29. Schick, T.; Dwivedi-Yu, J.; Dessi, R.; et al. Toolformer: Language Models Can Teach Themselves to Use Tools. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 10–16 December 2023; Volume 36, pp. 68539–68551. https://doi.org/10.52202/075280-2997.
30. OpenAI. Function Calling and Other API Updates. 2023. Available online: https://openai.com/index/function-calling-and-other-api-updates/ (accessed on 10 October 2026).
31. Anthropic. Introducing the Model Context Protocol. 2024. Available online: https://www.anthropic.com/news/model-context-protocol (accessed on 10 October 2026).
32. Liang, H.; Kalaleh, M.T.; Mei, Q. Integrating Large Language Models for Automated Structural Analysis. *arXiv* **2025**, arXiv:2504.09754.
33. Geng, Z.; Liu, J.; Cao, R.; et al. A lightweight large language model-based multi-agent system for 2D frame structural analysis. *arXiv* **2025**, arXiv:2510.05414.
34. Guo, T.; Chen, X.; Wang, Y.; et al. Large Language Model Based Multi-agents: A Survey of Progress and Challenges. In Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24), Jeju, Republic of Korea, 3–9 August 2024; pp. 8048–8057. https://doi.org/10.24963/ijcai.2024/890.
35. Chen, Z.; Asadi Shamsabadi, E.; Jiang, S.; et al. Integration of large vision language models for efficient post-disaster damage assessment and reporting. *Nat. Commun.* **2026**, *17*, 1481. https://doi.org/10.1038/s41467-025-68216-z.
36. Jiang, Y.; Wang, J.; Shen, X.; et al. Large language model for post-earthquake structural damage assessment of buildings. *Comput.-Aided Civ. Infrastruct. Eng.* **2025**, *40*, 6324–6342. https://doi.org/10.1111/mice.70010.
37. Grattafiori, A.; Dubey, A.; Jauhri, A.; et al. The Llama 3 herd of models. *arXiv* **2024**, arXiv:2407.21783.
38. Yang, A.; Yang, B.; Zhang, B.; et al. Qwen2.5 technical report. *arXiv* **2024**, arXiv:2412.15115.
39. Shinn, N.; Cassano, F.; Gopinath, A.; et al. Reflexion: Language agents with verbal reinforcement learning. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 10–16 December 2023; Volume 36, pp. 8634–8652. https://doi.org/10.52202/075280-0377.
40. Wang, L.; Zhang, X.; Su, H.; et al. A Comprehensive Survey of Continual Learning: Theory, Method and Application. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 5362–5383. https://doi.org/10.1109/tpami.2024.3367329.
41. Gama, J.; Žliobaitė, I.; Bifet, A.; et al. A survey on concept drift adaptation. *ACM Comput. Surv.* **2014**, *46*, 44. https://doi.org/10.1145/2523813.
42. Lu, J.; Liu, A.; Dong, F.; et al. Learning under Concept Drift: A Review. *IEEE Trans. Knowl. Data Eng.* **2019**, *31*, 2346–2363. https://doi.org/10.1109/tkde.2018.2876857.
43. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018.
44. Sharma, A.; Bhowmik, B. Autonomous agentic AI with policy adaptation for physics-informed spectral learning in Structural Health Monitoring. *Adv. Eng. Inform.* **2026**, *70*, 104224. https://doi.org/10.1016/j.aei.2025.104224.
45. Ji, Z.; Lee, N.; Frieske, R.; et al. Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.* **2023**, *55*, 248. https://doi.org/10.1145/3571730.
46. Huang, L.; Yu, W.; Ma, W.; et al. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Trans. Inf. Syst.* **2025**, *43*, 42. https://doi.org/10.1145/3703155.
47. Lewis, P.; Perez, E.; Piktus, A.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–12 December 2020; Volume 33, pp. 9459–9474.
48. Gao, Y.; Xiong, Y.; Gao, X.; et al. Retrieval-augmented generation for large language models: A survey. *arXiv* **2023**, arXiv:2312.10997.
49. *EN 1998-5:2004*; Eurocode 8: Design of Structures for Earthquake Resistance—Part 5: Foundations, Retaining Structures and Geotechnical Aspects. European Committee for Standardization: Brussels, Belgium, 2004.

50. ASCE. *Minimum Design Loads and Associated Criteria for Buildings and Other Structures*; American Society of Civil Engineers: Reston, VA, USA, 2022. https://doi.org/10.1061/9780784415788.
51. Government of Nepal. *Nepal National Building Code NBC 105:2020, Seismic Design of Buildings in Nepal*; Department of Urban Development and Building Construction: Kathmandu, Nepal, 2020.
52. Choi, K.; Jeong, M.; Kim, T.; et al. RAMASC: A retrieval-augmented multi-agent framework for automated structural calculation. *Adv. Eng. Inform.* **2026**, *74*, 104698. https://doi.org/10.1016/j.aei.2026.104698.
53. Aqib, M.; Hamza, M.; Mei, Q.; et al. Fine-Tuning Large Language Models and Evaluating Retrieval Methods for Improved Question Answering on Building Codes. *arXiv* **2025**, arXiv:2505.04666.
54. Fagan, A.; Ahmed, A. Retrieval-Augmented Generation for Educational AI Chatbot in Geotechnical Engineering. In Proceedings of the 2026 ASEE-NE Section Conference, Hooksett, NH, USA, 27–28 March 2026. https://doi.org/10.18260/1-2--58597.
55. Fan, L.; Liu, F.; Chen, C. Domain adaptation of large language models for geotechnical applications. *arXiv* **2025**, arXiv:2507.05613.
56. Estêvão, J.M.C. Effectiveness of Generative AI for Post-Earthquake Damage Assessment. *Buildings* **2024**, *14*, 3255. https://doi.org/10.3390/buildings14103255.
57. Hogan, A.; Blomqvist, E.; Cochez, M.; et al. Knowledge Graphs. *ACM Comput. Surv.* **2022**, *54*, 71. https://doi.org/10.1145/3447772.
58. Pan, S.; Luo, L.; Wang, Y.; et al. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 3580–3599. https://doi.org/10.1109/tkde.2024.3352100.
59. Edge, D.; Trinh, H.; Cheng, N.; et al. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. *arXiv* **2024**, arXiv:2404.16130.
60. Ma, X. Knowledge graph construction and application in geosciences: A review. *Comput. Geosci.* **2022**, *161*, 105082. https://doi.org/10.1016/j.cageo.2022.105082.
61. Open Geospatial Consortium. *OGC Geoscience Markup Language 4.1 (GeoSciML)*; Open Geospatial Consortium, Inc.: Arlington, VA, USA, 2017. https://doi.org/10.62973/16-008.
62. Association of Geotechnical and Geoenvironmental Specialists (AGS). AGS4 Data Format. Edition 4.2. 2025. Available online: https://www.ags.org.uk/data-format/ (accessed on 10 October 2026).
63. ASCE Geo-Institute. DIGGS—Data Interchange for Geotechnical and Geoenvironmental Specialists, Version 3. 2024. Available online: https://www.geoinstitute.org/special-projects/diggs (accessed on 10 October 2026).
64. *ISO 16739-1:2024*; Industry Foundation Classes (IFC) for Data Sharing in the Construction and Facility Management Industries—Part 1: Data Schema. International Organization for Standardization, Geneva, Switzerland, 2024.
65. Goto, Y.; Okuhara, C.; Horita, T.; et al. Geo-Intelligence: An integrated decision support system for geotechnical investigation using knowledge graphs and LLMs. *Artif. Intell. Data Sci.* **2026**, *7*, 323–330. https://doi.org/10.11532/jsceiii.7.1_323.
66. Global Mining Guidelines Group (GMG). omf-python: Python Library for Working with OMF Files. 2024. Available online: https://github.com/gmggroup/omf-python (accessed on 10 October 2026).
67. Blochwitz, T.; Otter, M.; Arnold, M.; et al. The Functional Mockup Interface for Tool independent Exchange of Simulation Models. In Proceedings of the 8th International Modelica Conference, Dresden, Germany, 20–22 March 2011; Volume 63, pp. 105–114. https://doi.org/10.3384/ecp11063105.
68. Ren, Y.; Yu, S.; Chen, K.; et al. Seismology modeling agent: A smart assistant for geophysical researchers. *arXiv* **2025**, arXiv:2512.14429.
69. Bekele, Y.W. GeoSim.AI: AI assistants for numerical simulations in geomechanics. *arXiv* **2025**, arXiv:2501.14186.
70. Chen, J.; Bao, Y. Multi-agent large language model framework for code-compliant automated design of reinforced concrete structures. *Autom. Constr.* **2025**, *177*, 106331. https://doi.org/10.1016/j.autcon.2025.106331.
71. Ni, B.; Buehler, M.J. MechAgents: Large language model multi-agent collaborations can solve mechanics problems, generate new data, and integrate knowledge. *Extrem. Mech. Lett.* **2024**, *67*, 102131. https://doi.org/10.1016/j.eml.2024.102131.
72. Chase, H. LangChain. 2022. Available online: https://github.com/langchain-ai/langchain (accessed on 10 October 2026).
73. LangChain AI. LangGraph: Low-Level Orchestration Framework for Building Stateful Agents. 2024. Available online: https://github.com/langchain-ai/langgraph (accessed on 10 October 2026).
74. Wu, Q.; Bansal, G.; Zhang, J.; et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. In Proceedings of the Conference on Language Modeling (COLM 2024), Philadelphia, PA, USA, 7–9 October 2024.
75. Kapoor, S.; Stroebl, B.; Siegel, Z.S.; et al. AI agents that matter. *Trans. Mach. Learn. Res.* 2025. Available online: https://openreview.net/pdf?id=Zy4uFzMviZ (accessed on 10 October 2026).
76. Yang, J.; Jimenez, C.; Wettig, A.; et al. SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 10–15 December 2024; Volume 37, pp. 50528–50652. https://doi.org/10.52202/079017-1601.

77. Hong, S.; Zhuge, M.; Chen, J.; et al. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024; pp. 23247–23275.
78. Qian, C.; Liu, W.; Liu, H.; et al. ChatDev: Communicative Agents for Software Development. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, Bangkok, Thailand, 11–16 August 2024; Volume 1, pp. 15174–15186. https://doi.org/10.18653/v1/2024.acl-long.810.
79. OpenAI. API Pricing. 2026. Available online: https://openai.com/api/pricing/ (accessed on 10 October 2026).
80. Anthropic. Claude API Pricing. 2026. Available online: https://platform.claude.com/docs/en/about-claude/pricing (accessed on 10 October 2026).
81. Google. Gemini Developer API Pricing. 2026. Available online: https://ai.google.dev/gemini-api/docs/pricing (accessed on 10 October 2026).
82. Liu, J.; Geng, Z.; Cao, R.; et al. A large language model-empowered agent for reliable and robust structural analysis. *Struct. Infrastruct. Eng.* **2026**, 1–16. https://doi.org/10.1080/15732479.2026.2630123.
83. Kamran, M.; Faizan, M.; Wang, S.; et al. Generative AI with prompt engineering in construction: Enhancing predictive slope stability modelling for safe, sustainable, climate-smart mining practices. *Geosci. Front.* **2025**, *16*, 102163. https://doi.org/10.1016/j.gsf.2025.102163.
84. Wu, S.; Otake, Y.; Mizutani, D.; et al. Future-proofing geotechnics workflows: Accelerating problem-solving with large language models. *Georisk Assess. Manag. Risk Eng. Syst. Geohazards* **2025**, *19*, 307–324. https://doi.org/10.1080/17499518.2024.2381026.
85. Bran, A.M.; Cox, S.; Schilter, O.; et al. Augmenting large language models with chemistry tools. *Nat. Mach. Intell.* **2024**, *6*, 525–535. https://doi.org/10.1038/s42256-024-00832-8.
86. Nguyen, T.T.; Dang, V.H.; Pham, T.T. Development of an automatic and knowledge-infused framework for structural health monitoring based on prompt engineering. *Front. Struct. Civ. Eng.* **2024**, *18*, 1752–1774. https://doi.org/10.1007/s11709-024-1118-7.
87. Phoon, K.K.; Ching, J.; Shuku, T. Challenges in data-driven site characterization. *Georisk Assess. Manag. Risk Eng. Syst. Geohazards* **2022**, *16*, 114–126. https://doi.org/10.1080/17499518.2021.1896005.
88. Phoon, K.K.; Ching, J.; Cao, Z. Unpacking data-centric geotechnics. *Undergr. Space* **2022**, *7*, 967–989. https://doi.org/10.1016/j.undsp.2022.04.001.
89. Wang, Y.; Zhao, T. Interpretation of soil property profile from limited measurement data: A compressive sampling perspective. *Can. Geotech. J.* **2016**, *53*, 1547–1559. https://doi.org/10.1139/cgj-2015-0545.
90. Zhao, T.; Wang, Y.; Xu, L. Efficient CPT locations for characterizing spatial variability of soil properties within a multilayer vertical cross-section using information entropy and Bayesian compressive sensing. *Comput. Geotech.* **2021**, *137*, 104260. https://doi.org/10.1016/j.compgeo.2021.104260.
91. Wang, Y.; Li, P. Data-driven determination of sample number and efficient sampling locations for geotechnical site investigation of a cross-section using Voronoi diagram and Bayesian compressive sampling. *Comput. Geotech.* **2021**, *130*, 103898. https://doi.org/10.1016/j.compgeo.2020.103898.
92. Ching, J.; Phoon, K.K. Constructing Site-Specific Multivariate Probability Distribution Model Using Bayesian Machine Learning. *J. Eng. Mech.* **2019**, *145*, 04018126. https://doi.org/10.1061/(asce)em.1943-7889.0001537.
93. Seed, H.B.; Idriss, I.M. Simplified Procedure for Evaluating Soil Liquefaction Potential. *J. Soil Mech. Found. Div.* **1971**, *97*, 1249–1273. https://doi.org/10.1061/jsfeaq.0001662.
94. Robertson, P.K.; Wride, C.E. Evaluating cyclic liquefaction potential using the cone penetration test. *Can. Geotech. J.* **1998**, *35*, 442–459. https://doi.org/10.1139/t98-017.
95. Moss, R.E.; Seed, R.B.; Kayen, R.E.; et al. CPT-Based Probabilistic and Deterministic Assessment of *In Situ* Seismic Soil Liquefaction Potential. *J. Geotech. Geoenviron. Eng.* **2006**, *132*, 1032–1051. https://doi.org/10.1061/(asce)1090-0241(2006)132:8(1032).
96. Boulanger, R.W.; Idriss, I.M. *CPT and SPT Based Liquefaction Triggering Procedures*; Report No. UCD/CGM-14/01; Center for Geotechnical Modeling, Department of Civil and Environmental Engineering, University of California: Davis, CA, USA, 2014.
97. Hanna, A.M.; Ural, D.; Saygili, G. Neural network model for liquefaction potential in soil deposits using Turkey and Taiwan earthquake data. *Soil Dyn. Earthq. Eng.* **2007**, *27*, 521–540. https://doi.org/10.1016/j.soildyn.2006.11.001.
98. Durante, M.G.; Rathje, E.M. An exploration of the use of machine learning to predict lateral spreading. *Earthq. Spectra* **2021**, *37*, 2288–2314. https://doi.org/10.1177/87552930211004613.
99. Intrieri, E.; Gigli, G.; Mugnai, F.; et al. Design and implementation of a landslide early warning system. *Eng. Geol.* **2012**, *147–148*, 124–136. https://doi.org/10.1016/j.enggeo.2012.07.017.
100. Intrieri, E.; Carlà, T.; Gigli, G. Forecasting the time of failure of landslides at slope-scale: A literature review. *Earth-Sci. Rev.* **2019**, *193*, 333–349. https://doi.org/10.1016/j.earscirev.2019.03.019.

101. Jiang, S.H.; Papaioannou, I.; Straub, D. Bayesian updating of slope reliability in spatially variable soils with in-situ measurements. *Eng. Geol.* **2018**, *239*, 310–320. https://doi.org/10.1016/j.enggeo.2018.03.021.
102. Straub, D.; Papaioannou, I. Bayesian Updating with Structural Reliability Methods. *J. Eng. Mech.* **2015**, *141*, 04014134. https://doi.org/10.1061/(asce)em.1943-7889.0000839.
103. Kang, F.; Han, S.; Salgado, R.; et al. System probabilistic stability analysis of soil slopes using Gaussian process regression with Latin hypercube sampling. *Comput. Geotech.* **2015**, *63*, 13–25. https://doi.org/10.1016/j.compgeo.2014.08.010.
104. Dahal, A.; Lombardo, L. Towards physics-informed neural networks for landslide prediction. *arXiv* **2024**, arXiv:2407.06785.
105. Mousavi, S.M.; Ellsworth, W.L.; Zhu, W.; et al. Earthquake transformer—An attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nat. Commun.* **2020**, *11*, 3952. https://doi.org/10.1038/s41467-020-17591-w.
106. Mousavi, S.M.; Beroza, G.C. Deep-learning seismology. *Science* **2022**, *377*, eabm4470. https://doi.org/10.1126/science.abm4470.
107. Zhang, R.; Liu, Y.; Sun, H. Physics-informed multi-LSTM networks for metamodeling of nonlinear structures. *Comput. Methods Appl. Mech. Eng.* **2020**, *369*, 113226. https://doi.org/10.1016/j.cma.2020.113226.
108. Zhang, R.; Liu, Y.; Sun, H. Physics-guided convolutional neural network (PhyCNN) for data-driven seismic response modeling. *Eng. Struct.* **2020**, *215*, 110704. https://doi.org/10.1016/j.engstruct.2020.110704.
109. Chatzikonstantinou, N.; Makarios, T. Dynamic Response of SDoF System with Negative Stiffness—A Relevant Key-Point for Machine Learning. *Bull. Comput. Intell.* **2026**, *2*, 31–53. https://doi.org/10.53941/bci.2026.100003.
110. Avci, O.; Abdeljaber, O.; Kiranyaz, S.; et al. A review of vibration-based damage detection in civil structures: From traditional methods to Machine Learning and Deep Learning applications. *Mech. Syst. Signal Process.* **2021**, *147*, 107077. https://doi.org/10.1016/j.ymssp.2020.107077.
111. Spencer, B.F.; Hoskere, V.; Narazaki, Y. Advances in Computer Vision-Based Civil Infrastructure Inspection and Monitoring. *Engineering* **2019**, *5*, 199–222. https://doi.org/10.1016/j.eng.2018.11.030.
112. Boore, D.M.; Bommer, J.J. Processing of strong-motion accelerograms: Needs, options and consequences. *Soil Dyn. Earthq. Eng.* **2005**, *25*, 93–115. https://doi.org/10.1016/j.soildyn.2004.10.007.
113. Wang, R.; Schurr, B.; Milkereit, C.; et al. An Improved Automatic Scheme for Empirical Baseline Correction of Digital Strong-Motion Records. *Bull. Seismol. Soc. Am.* **2011**, *101*, 2029–2044. https://doi.org/10.1785/0120110039.
114. Zhu, W.; Mousavi, S.M.; Beroza, G.C. Seismic Signal Denoising and Decomposition Using Deep Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 9476–9488. https://doi.org/10.1109/tgrs.2019.2926772.
115. Liu, B.; Zhou, B.; Kong, J.; et al. The Cut-Off Frequency of High-Pass Filtering of Strong-Motion Records Based on Transfer Learning. *Appl. Sci.* **2023**, *13*, 1500. https://doi.org/10.3390/app13031500.
116. Kouris, L.A.S.; Penna, A.; Magenes, G. Optimized Noise Suppression of Acceleration Records from Shaking-Table Tests Using Signal Processing and Deep Learning Techniques. *Bull. Comput. Intell.* **2026**, *2*, 164–180. https://doi.org/10.53941/bci.2026.100009.
117. Gupta, R.; Goodman, B.; Patel, N.; et al. Creating xBD: A Dataset for Assessing Building Damage from Satellite Imagery. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–20 June 2019; pp. 10–17.
118. Robertson, P.K. Interpretation of cone penetration tests—A unified approach. *Can. Geotech. J.* **2009**, *46*, 1337–1355. https://doi.org/10.1139/t09-065.
119. Iwasaki, T.; Tokida, K.; Tatsuoka, F. Soil liquefaction potential evaluation with use of the simplified procedure. In Proceedings of the International Conference on Recent Advances in Geotechnical Earthquake Engineering and Soil Dynamics, St. Louis, MO, USA, 26 April–3 May 1981.
120. Hosseini, S.; Jasimi, N.; Mehdizadeh, B.; et al. An Integrated Multi-Criteria Risk-Assessment Framework for Rockburst in Underground Construction. *Bull. Comput. Intell.* **2026**, *2*, 181–195. https://doi.org/10.53941/bci.2026.100010.
121. Karniadakis, G.E.; Kevrekidis, I.G.; Lu, L.; et al. Physics-informed machine learning. *Nat. Rev. Phys.* **2021**, *3*, 422–440. https://doi.org/10.1038/s42254-021-00314-5.
122. Yuan, B.; Choo, C.S.; Yeo, L.Y.; et al. Physics-informed machine learning in geotechnical engineering: A direction paper. *Geomech. Geoengin.* **2025**, *20*, 1128–1159. https://doi.org/10.1080/17486025.2025.2502029.
123. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045.
124. Bekele, Y.W. Physics-informed deep learning for one-dimensional consolidation. *J. Rock Mech. Geotech. Eng.* **2021**, *13*, 420–430. https://doi.org/10.1016/j.jrmge.2020.09.005.
125. Rao, C.; Sun, H.; Liu, Y. Physics-Informed Deep Learning for Computational Elastodynamics without Labeled Data. *J. Eng. Mech.* **2021**, *147*, 04021043. https://doi.org/10.1061/(asce)em.1943-7889.0001947.

126. Rasht-Behesht, M.; Huber, C.; Shukla, K.; et al. Physics-Informed Neural Networks (PINNs) for Wave Propagation and Full Waveform Inversions. *J. Geophys. Res. Solid Earth* **2022**, *127*, e2021JB023120. https://doi.org/10.1029/2021jb023120.
127. Karpatne, A.; Atluri, G.; Faghmous, J.H.; et al. Theory-Guided Data Science: A New Paradigm for Scientific Discovery from Data. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2318–2331. https://doi.org/10.1109/tkde.2017.2720168.
128. Willard, J.; Jia, X.; Xu, S.; et al. Integrating Scientific Knowledge with Machine Learning for Engineering and Environmental Systems. *ACM Comput. Surv.* **2023**, *55*, 66. https://doi.org/10.1145/3514228.
129. Drosopoulos, G.A.; Stavroulakis, G.E. DeepONet for the Prediction of Failure Response of a Two-Dimensional Fibre-Reinforced Composite Plate. *Bull. Comput. Intell.* **2025**, *1*, 76–88. https://doi.org/10.53941/bci.2025.100005.
130. Xue, T.; Liao, S.; Gan, Z.; et al. JAX-FEM: A differentiable GPU-accelerated 3D finite element solver for automatic inverse design and mechanistic data science. *Comput. Phys. Commun.* **2023**, *291*, 108802. https://doi.org/10.1016/j.cpc.2023.108802.
131. Garcez, A.D.A.; Lamb, L.C. Neurosymbolic AI: The 3rd wave. *Artif. Intell. Rev.* **2023**, *56*, 12387–12406. https://doi.org/10.1007/s10462-023-10448-w.
132. Oberkampf, W.L.; Roy, C.J. *Verification and Validation in Scientific Computing*; Cambridge University Press: Cambridge, UK, 2010. https://doi.org/10.1017/cbo9780511760396.
133. NIST. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*; NIST AI 100-1; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2023.
134. European Union. *Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act)*; European Union: Brussels, Belgium, 2024.
135. *ISO 16290:2013*; Space Systems—Definition of the Technology Readiness Levels (TRLs) and Their Criteria of Assessment. International Organization for Standardization: Geneva, Switzerland, 2013.
136. Mankins, J.C. Technology readiness assessments: A retrospective. *Acta Astronaut.* **2009**, *65*, 1216–1223. https://doi.org/10.1016/j.actaastro.2009.03.058.
137. Amer, M.; Daim, T.U.; Jetter, A. A review of scenario planning. *Futures* **2013**, *46*, 23–40. https://doi.org/10.1016/j.futures.2012.10.003.
138. Callcut, M.; Cerceau Agliozzo, J.P.; Varga, L.; et al. Digital Twins in Civil Infrastructure Systems. *Sustainability* **2021**, *13*, 11549. https://doi.org/10.3390/su132011549.
139. Jiang, F.; Ma, L.; Broyd, T.; et al. Digital twin and its implementations in the civil engineering sector. *Autom. Constr.* **2021**, *130*, 103838. https://doi.org/10.1016/j.autcon.2021.103838.
140. Wang, Y.; Rezaei, A.; Hicks, S. Digital twin applications in geotechnical engineering: A systematic review. *Mach. Learn. Data Sci. Geotech.* **2025**, *1*, 78–92. https://doi.org/10.1108/mlag-02-2025-0005.