Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

AI Skills as Structural Engineering Checking Assistants: A Framework for Conversational, Traceable Code Compliance Using ACI 318-25 and ASCE 7-22

Rinaz Riyaz Mohamed

rinaz.riyazmohamed@stantec.com

Stantec (United States)

Rahul Anand

Stantec (United States)

Jigarbhai Mansukhbhai Sonani

Stantec (United States)

Research Article

Keywords: ACI 318, code compliance, structural engineering, AI Skill, reinforced concrete, large language model, Claude, professional accountability

Posted Date: June 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10009671/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

AI Skills as Structural Engineering Checking Assistants: A Framework for Conversational, Traceable Code Compliance Using ACI 318-25 and ASCE 7-22

Rinaz Riyaz Mohamed[1,*]; Rahul Anand[2]; Jigarbhai Mansukhbhai Sonani[3]

[1] Structural Engineer, Stantec Consulting Services Inc., Houston, TX 77002, USA
[2] Structural Engineer, Stantec Consulting Services Inc., New Haven, CT 06510, USA
[3] Structural Engineer, Stiver Engineering Inc., Houston, TX 77006, USA

[*] Corresponding author: rinaz.riyazmohamed@stantec.com | ORCID: 0009-0003-2777-5279

Abstract

The integration of artificial intelligence into structural engineering practice has predominantly followed two trajectories: autonomous design generation and building information modeling integrated compliance checking. However, both share fundamental limitations. They position AI as a system that produces outputs rather than as a tool that supports an engineer's reasoning. This paper introduces a third trajectory, AI Skill as a structural engineering checking assistant. Drawing on a paradigm emerging across engineering disciplines, this study proposes a reusable instruction-set framework for reinforced concrete beam checking per ACI 318-25 and ASCE 7-22. The proposed ACI 318-25 Beam Checking Skill does not design beams but checks them. It requests all missing information before proceeding, cites the specific code section governing every calculation, flags borderline cases and assumptions, and closes every session with a mandatory disclaimer requiring review by a licensed Professional Engineer (PE). A worked example demonstrates four of the ten sequential checks in full step-by-step detail; all results are compiled in a demand-to-capacity ratio (DCR) summary table and bar chart. A comparison with traditional spreadsheet-based checking demonstrates measurable advantages in terms of assumption transparency, code traceability, borderline case detection, and interrogability. The PE accountability framework is analyzed, demonstrating that AI Skill supports rather than displaces the professional certification chain. The Skill is publicly available on Zenodo and can be used with a standard Claude subscription, requiring no programming, proprietary software, or training data, directly addressing the accessibility barriers that limit AI adoption among smaller firms, independent practitioners, and engineering educators.

Keywords: ACI 318; code compliance; structural engineering; AI Skill; reinforced concrete; large language model; Claude; professional accountability

1 Introduction

The integration of AI into structural engineering practices has been approached in two primary ways. The first focuses on AI systems that *design* structures by generating reinforcement layouts, optimizing cross-sections, and producing code-compliant beam configurations from parametric inputs using deep neural networks and reinforcement

learning [1,2]. The second involves AI systems embedded in design workflows that check Building Information Modeling (BIM) models by integrating Large Language Models (LLMs) with platforms such as Revit to perform semi-automated compliance verification against building codes [3]. Both approaches position AI upstream of engineering judgment: AI acts first, and the engineer reviews. This ordering raises a fundamental accountability question that has not yet been satisfactorily resolved [4].

This paper describes a third approach, AI Skill, as a structural engineering checking assistant. In this framework, the engineer first provides the beam design, loading conditions, and material specifications, and the AI then performs a structured, section-cited, step-by-step interrogation of that design against the current code requirements. The AI does not design. It does not approve. It checks and explains every step of that process in plain language.

The paradigm is informed by a publicly shared demonstration by the geotechnical engineer Arabel Vilas Serín, who developed a Claude-based Skill for Eurocode 7 bearing capacity checking [5]. This demonstration established that a reusable AI instruction set operating through natural language conversation rather than a parametric GUI could effectively structure and interrogate geotechnical calculations in a way that enhances rather than replaces professional judgment. This study proposes and demonstrates an analogous framework for the concrete structural engineering domain, extending the paradigm to ACI 318-25 beam checking [9].

The significance of this study is grounded in two well-documented gaps in the current landscape of AI tools for structural engineering. The first gap is accessibility: Tools capable of rigorous, code-traceable structural checking typically require proprietary software licenses costing thousands of dollars annually, placing them beyond reach for many small practices, independent consultants, and engineering educators. The second gap is trust and explainability: a 2025 AASHTO nationwide survey of state DOTs found that concerns over the trust and reliability of AI models were among the most-cited barriers to AI adoption, reported by 61.3% of respondents, and that engineering stamping enforcement, review, and certification by a licensed engineer is the most commonly implemented AI governance measure across state DOTs [13]. Most existing AI structural tools, including deep learning design systems [1,2] and BIM-integrated checking platforms [3], produce results without the step-by-step reasoning transparency required by Professional Engineer (PE) certification standards. The AI Skill paradigm directly addresses both gaps simultaneously. The skill demonstrated in this study is operationally accessible to any engineer with a Claude Pro subscription (~$20/month) and produces a fully auditable section-cited calculation record for every checking session, as supported by the broader research program on AI-driven infrastructure resilience at Lamar University's Center for Resiliency [6].

2 The AI Skill paradigm

2.1 Definition and operating principles

In the context of large language model interfaces such as Claude (Anthropic), a Skill is a reusable, versioned instruction document that configures the AI to operate as a specialized assistant for a defined workflow. Unlike a one-time prompt, a skill is a

persistent instruction set installed once and activated by name in any subsequent conversation that defines the role of the AI, the information it must gather before proceeding, the calculations it must perform in sequence, the format in which it must present results, and the boundaries of its scope. This is conceptually analogous to a quality assurance checklist or calculation template, except that it is interactive, self-explanatory, and refuses to proceed with missing or ambiguous inputs.

The distinction between a skill and a standalone prompt is operationally significant. A prompt instructs the AI on what to do in a single exchange. A Skill creates a persistent and reproducible protocol that behaves consistently across multiple sessions and multiple users. When a structural engineer installs the ACI 318-25 Beam Checking Skill, they receive a standardized, version-controlled checking workflow regardless of how they phrase their inputs or the session in which they are working. Skill's logic — its check sequence, its code citations, and its flagging rules — do not change between sessions unless the Skill document is deliberately revised and re-released under a new version number, providing the version control and auditability that professional practice requires.

Three operating principles govern AI skills as checking assistants. First, the Skill is reactive rather than generative; it responds to engineer-provided inputs and interrogates them against the code rather than producing a design from first principles. This ensures that the engineer's judgment is the primary driver for every design decision. Second, the Skill is complete rather than selective: it systematically covers all checks within its declared scope, rather than only the checks explicitly requested by the engineer, ensuring that critical code provisions are not inadvertently omitted. Third, the Skill is transparent rather than opaque: it shows every calculation step and every code citation in plain language, creating a record that a reviewing PE can trace, challenge, and certify independently. Figure 1 illustrates the five-stage operational sequence of the skill from activation through PE disclaimer.

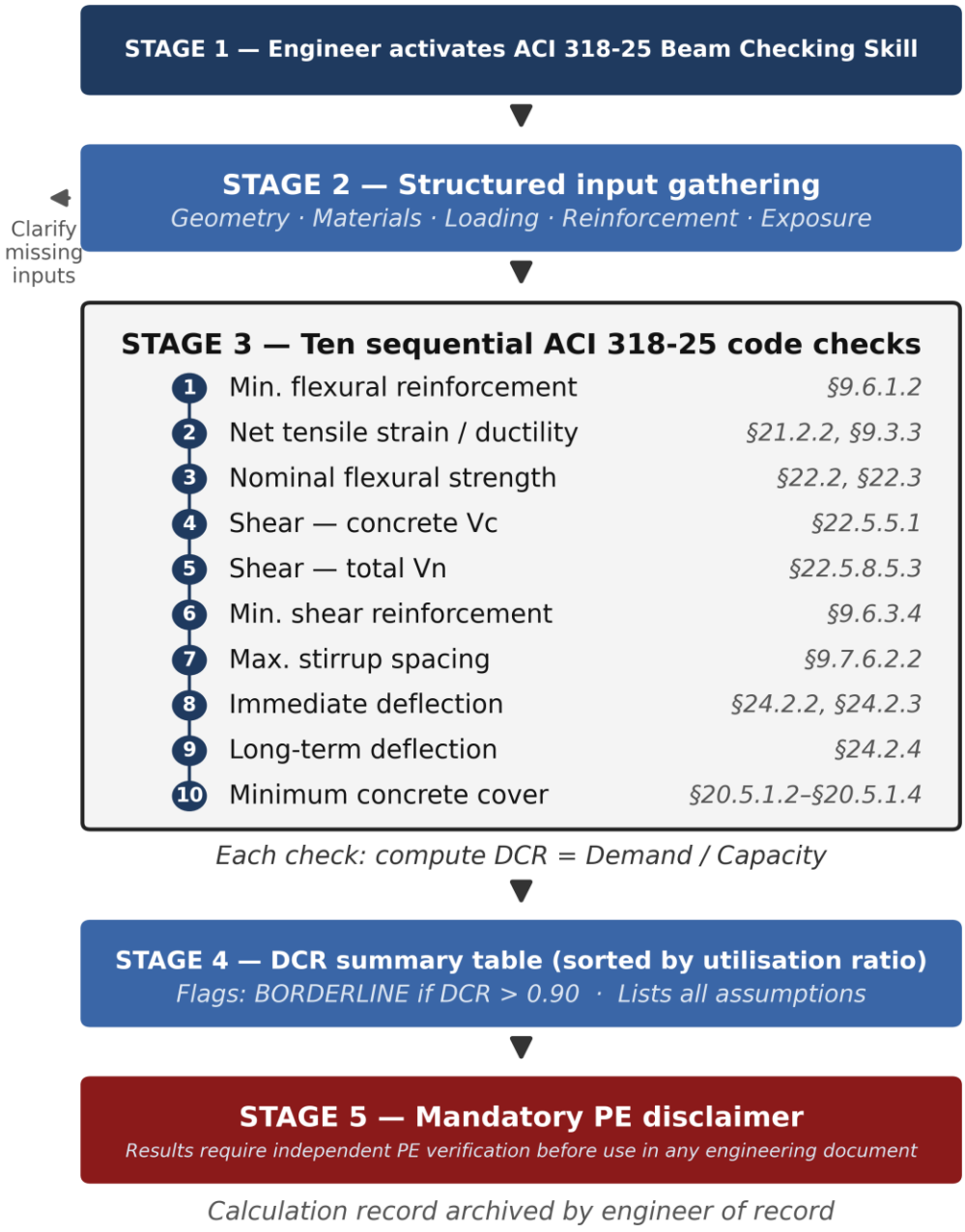

Fig 1: *Five-stage operational sequence of the ACI 318-25 Beam Checking Skill. Stage 3 shows all ten sequential code checks with their governing ACI 318-25 section citations. The clarification loop in Stage 2 ensures all inputs are confirmed before any calculation begins*

2.2 Precedents and the geotechnical connection

The AI Skill paradigm for engineering checking was publicly demonstrated for the first time in the geotechnical domain by Arabel Vilas Serín in 2025, with a Claude-based Skill for Eurocode 7 foundation bearing capacity checking [5]. The EC7 Skill does not design foundations; it systematically checks the foundation design that the geotechnical engineer

has already produced, prompting confirmation of the Design Approach selection, eccentricity logic, partial factors, and combination requirements in a conversational format that surfaces assumptions that might otherwise remain implicit in a spreadsheet. The Skill was shared publicly via GitHub and attracted direct engagement from practicing geotechnical engineers, confirming practitioner interest in this paradigm.

The structural engineering AI landscape includes related but distinct approaches. Eslamlou and Naser [7] investigated whether general-purpose chatbots could be used to analyze and design steel structures and found that while LLMs demonstrate useful qualitative awareness of structural engineering concepts, they are prone to numerical errors and unsupported assumptions when operating without a structured protocol. Liu et al. [16] confirmed this finding more rigorously by benchmarking Llama-3.3 70 B on eight beam structural analysis problems and demonstrated that, despite the apparent qualitative understanding of structural mechanics, unstructured LLMs lack the quantitative reliability and robustness required for engineering applications. These findings directly motivate the skill approach: a structured protocol with mandatory information gathering, sequential check order, explicit code citations, and a required PE review disclaimer directly addresses the numerical reliability and assumption transparency gaps that both studies identify. AI Skill is not the chatbot being asked to solve a structural problem. It is a precisely defined workflow that constrains LLM to behave as a systematic checking assistant within a controlled auditable protocol. A complementary line of work by Joffe et al. [15] demonstrated that a retrieval-augmented LLM framework can answer building code and standards queries with accurate citations using the National Building Code of Canada, validating the broader paradigm of LLM-assisted code referencing in engineering practice. Similarly, the responsible AI framework proposed by Plevris and Hosamo [8] identifies transparency and professional accountability as the two non-negotiable requirements for AI adoption in structural engineering practice, requirements that the Skill paradigm satisfies by design.

2.3 Cost-accessibility and practice democratization

A practical dimension of the AI Skill paradigm that distinguishes it from the existing structural AI tools is its radical cost accessibility. Commercial structural analysis platforms capable of performing rigorous, code-traceable beam checking, including moment-shear-deflection checks with section-cited code provisions, typically require annual licenses of $2,000 to $10,000 or more per seat, placing them beyond reach for independent practitioners, small structural consultants, and engineering students in developing countries. Academic or student licenses may be cheaper but are restricted to non-commercial use and expire upon graduation.

The AI Skill approach requires only a Claude Pro subscription, which is priced at approximately $20 per month as of the time of writing. There are no installations, compatibility requirements, seat licenses, or vendor lock-ins. The Skill itself is a plain-text document, distributed freely via Zenodo under an open license [12], and any engineer can download, review, modify it to meet their jurisdiction's code requirements, and use it immediately. This combination of near-zero cost, open access, no technical prerequisites, and immediately actionable outputs represents a qualitative shift in the accessibility of structured code-checking tools to the global structural engineering community. It is

particularly relevant for small practices and individual consultants who constitute the majority of licensed structural engineers in both the United States and internationally and who currently have the least access to advanced checking software.

2.4 Distinction from AI design and BIM-checking approaches

AI Skill is distinguished from the two dominant paradigms of structural AI along several dimensions that are important for understanding its appropriate role in engineering practice.

AI *design* systems [1,2] use machine learning to generate structural designs from parametric inputs, typically without intermediate calculations. Their outputs are design proposals that require subsequent code checking by an engineer. They are optimized for speed and automated generation, rather than transparency and traceability. The AI Skill is the conceptual inverse: it takes the engineer's design as input and systematically interrogates it against code provisions, showing every step of every check. AI design systems reduce the engineer's role in the design generation phase, and the AI Skill amplifies the engineer's ability to perform and document the code-checking phase.

BIM-integrated AI checking systems [3] perform code compliance checks within model-based design environments, typically flagging noncompliant elements based on model properties. Although powerful in the context of large building projects with established BIM workflows, these systems require BIM platform access, model structuring according to vendor specifications, and integration configuration. They are typically enterprise tools rather than practitioner tools. An intermediate approach explored by Liang et al. [17] uses LLMs to parse structural descriptions and translate them into executable finite element analysis scripts, thereby automating the analysis while still requiring domain-specific prompt engineering and software integration. AI Skill requires only a conversational interface and can be used for any beam in a BIM model, sketch, or field observation. The engineer can provide geometric, material, and loading inputs in a structured form.

The complementarity of the three approaches is noteworthy. An AI design system can generate an initial beam cross-section. A BIM platform may embed a section in a building model. The AI Skill can then be used by the engineer to conduct a detailed, section-cited, step-by-step interrogation of that section's code compliance, the kind of rigorous individual check that neither the design generator nor the BIM checker typically provides in an explicit, auditable form. Therefore, these three approaches are best understood as operating at different levels of the engineering workflow rather than as competitors.

3 The ACI 318-25 Beam Checking Skill

3.1 Design philosophy

The ACI 318-25 Beam Checking Skill was developed based on four design principles derived directly from the requirements of PE accountability and the characteristics of routine concrete beam-checking practices.

Principle 1: No assumption of silence. The Skill must explicitly identify every assumption it makes and requires engineer confirmation of any input that is missing or ambiguous before proceeding. In a traditional spreadsheet, a cell formula that assumes

normal-weight concrete at 150 pcf may not visibly display this assumption in the output. In an AI Skill session, the assumption was stated in plain language and flagged for affirmative engineer confirmation before any dependent calculation was performed. This principle directly addresses the finding of Orris and Ssegawa [11] that assumed transparency is one of the most critical gaps in current AI tools for structural engineering.

Principle 2: Full calculation transparency. The Skill must show every step of every calculation, including substituting numerical values into the governing formula and not just the final result. This enables the review of PE to trace each computed quantity to its inputs, code equation, and numerical values in a single reading of the session output, creating an audit trail that is, in practice, more transparent than most commercial software outputs, which typically display only final results without intermediate steps.

Principle 3: Mandatory code traceability. Every check must cite the specific ACI 318-25 section number that governs it. This is not optional. This is a structural element of the PE accountability chain. The engineer reviewing the output must be able to locate the cited section in their copy of ACI 318-25 and verify that the Skill's implementation is correct. This also makes Skill's outputs immediately instructive for less-experienced engineers and engineering students, who can follow the ACI section citations directly to the source code provisions.

Principle 4: Scope honesty. The Skill must explicitly identify conditions that fall outside its scope and decline to apply an inappropriate methodology rather than proceeding silently. When an engineer inputs a deep beam (span-to-depth ratio less than 4), a prestressed section, or a condition involving torsion, the skill identifies the out-of-scope condition and halts, directing the engineer to appropriate specialized resources. This prevents the most dangerous category of AI checking error: a check that appears to pass but has been applied to a condition for which it was not designed.

3.2 Information gathering protocol

The first action upon Skill activation is a structured input request covering four categories: (1) geometrical properties: span, support conditions, web width, total depth, effective depth (with default $d = h - 2.5$ in if not provided), flange geometry for T-beams; (2) material properties: concrete compressive strength f'c, steel yield strength fy, unit weight, exposure category for cover requirements; (3) loading: superimposed dead load, live load, load combination per ASCE 7-22 §2.3.1 [10]; and (4) reinforcement details: tension bars, compression bars if any, stirrup size, spacing, and number of legs, clear cover to stirrups.

This Skill will not proceed to any code check until all required inputs have been received and confirmed by the engineer. When inputs are missing, the skill explicitly asks for them: When inputs are internally inconsistent -for example, if the provided clear cover and bar sizes are geometrically incompatible with the stated web width- the Skill flags the inconsistency and asks the engineer to resolve it before continuing. When the effective depth d is not provided, Skill assumes $d = h - 2.5$, in explicitly states this assumption, and requires engineer confirmation. These checks ensured that the calculation session started with a verified and consistent set of inputs.

3.3 Code checks performed

This Skill performs ten sequential checks per ACI 318-25 [9]. Table 1 shows each check and its governing code provisions. The complete Skill instruction document is provided as Supplementary Information.

Table 1: *Ten sequential code checks performed by the ACI 318-25 Beam Checking Skill*

#	Check name	ACI 318-25 section	Demand parameter	Capacity parameter
1	Minimum flexural reinforcement	§9.6.1.2	As,provided (in²)	As,min (in²)
2	Net tensile strain / ductility	§21.2.2, §9.3.3	εt,min = 0.004	εt,computed
3	Nominal flexural strength	§22.2, §22.3	Mu (in-kip)	φMn (in-kip)
4	Shear strength - concrete Vc	§22.5.5.1, Table 22.5.5.1	Vu,critical (kip)	φVc (kip)
5	Shear strength - total Vn	§22.5.8.5.3	Vu,max (kip)	φVn (kip)
6	Minimum shear reinforcement	§9.6.3.4	Av,min/s (in²/in)	Av,prov/s (in²/in)
7	Maximum stirrup spacing	§9.7.6.2.2	s,provided (in)	smax (in)
8	Immediate deflection	§24.2.2, §24.2.3	Δimm,LL (in)	L/360 (in)
9	Long-term deflection	§24.2.4	ΔLT + Δimm,LL (in)	L/240 (in)
10	Minimum concrete cover	§20.5.1.2 - §20.5.1.4	cover,req (in)	cover,prov (in)

DCR = demand-to-capacity ratio. Any check with DCR > 0.90 is flagged BORDERLINE regardless of pass/fail status.

The check sequence reflects a deliberate scope decision: to cover the complete set of provisions, a practicing engineer would verify for a standard gravity-loaded beam under

ACI 318-25, while explicitly excluding checks requiring specialized expertise beyond routine beam design–seismic provisions (Chapter 18), torsion (§22.7), deep beams (§9.9), and prestress. The shear check implementation (Check 4) deserves specific mention because ACI 318-25 introduces a significant departure from the pre-2019 editions. The earlier simplified Vc equations were replaced by the table-based method of §22.5.5.1, which requires the longitudinal reinforcement ratio ρw and includes the size modification factor λs, a provision that many practicing engineers trained under pre-2019 ACI code editions may not yet be incorporated into their checking routines. This Skill implements this updated method and explicitly flags the change for the engineer.

3.4 Results presentation and engineer review flags

Following the individual checks, Skill presents a summary table sorted by DCR in descending order, immediately directing the engineer's attention to the most critical checks. The table includes the check name, demand value, capacity value, DCR, and status (PASS, FAIL, or BORDERLINE). Any DCR exceeding 0.90 is flagged BORDERLINE and highlighted for specific engineer attention, regardless of whether the check technically passes. This BORDERLINE threshold is a deliberate conservatism: a DCR of 0.93 technically passes but leaves only 7% reserve within the range of normal construction tolerance variation and calculation rounding, warranting explicit PE awareness.

Following the DCR table, Skill compiles all assumptions made during the session into an explicit list requiring engineer confirmation, identifies any conditions that fall outside its scope, flags any detected input inconsistencies, and closes with the mandatory PE disclaimer. The complete session record -inputs, calculations, DCR table, assumptions, and disclaimer–constitutes an auditable calculation document that the reviewing PE can archive alongside other project calculations.

4 Worked example: simply supported rectangular beam

4.1 Beam specifications

An example is presented for a simply supported rectangular concrete beam, demonstrating Skill's operation for the four selected checks to illustrate the most instructive aspects of the framework: Check 4 (concrete shear, which implements the updated ACI 318-25 §22.5.5.1 provisions), Check 3 (flexural strength), Check 9 (long-term deflection, which demonstrates the multistep Ie calculation), and Check 10 (cover, which demonstrates BORDERLINE flagging at DCR = 1.00). Table 2 presents the complete input parameters confirmed by the engineer to the Skill.

Table 2: *Confirmed input parameters for the worked example beam*

Category	Parameter	Value
Geometry	Span L / support conditions	10 ft, simply supported
	Web width bw	15 in

	Total depth h	20 in
	Effective depth d	17 in (engineer-provided)
Materials	Concrete strength f'c	4,000 psi (normal weight, 150 lb/ft³)
	Steel yield strength fy	60,000 psi (Grade 60)
	Exposure	Not exposed to weather or earth
Loading	Superimposed dead load wD	1.200 kip/ft
	Live load wL	1.500 kip/ft
	Load combination	ASCE 7-22 §2.3.1: U = 1.2D + 1.6L
Reinforcement	Tension steel	5 - #5 bars (As = 1.55 in²)
	Compression steel	2 - #4 bars (A's = 0.40 in²)
	Stirrups	#3 @ 6 in, 2 legs (Av = 0.22 in²)
	Clear cover to stirrups	1.5 in

Self-weight auto-calculated by the Skill: Wsw = 150 × (15/12) × (20/12) = 0.260 kip/ft. Total factored load: Wu = 1.2(1.460) + 1.6(1.500) = 4.152 kip/ft. Mu = 622.8 in-kip; Vu,critical = 14.88 kip at d from support face.

4.2 Detailed code checks

4.2.1 Check 3 - Nominal flexural strength [ACI §22.2, §22.3]

The flexural strength check computes the depth of the equivalent rectangular stress block, neutral axis depth, net tensile strain for φ-factor determination, and design moment capacity:

Depth of equivalent stress block [§22.2.2]:
$a = As \times fy / (0.85 \times f'c \times bw) = 1.55 \times 60{,}000 / (0.85 \times 4{,}000 \times 15)$
a = 93,000 / 51,000 = 1.824 in

β_1 [§22.2.2.3], f'c = 4,000 psi]: β_1 = 0.85
c = a / β_1 = 1.824 / 0.85 = 2.146 in

Net tensile strain [§21.2.2]: $\varepsilon t = (dt/c - 1) \times 0.003$
= (17/2.146 − 1) × 0.003 = 6.921 × 0.003 = 0.02076
εt = 0.02076 ≥ 0.005 → Tension-controlled → φ = 0.90

Design flexural strength [§22.3]:

$$Mn = As \times fy \times (d - a/2) = 1.55 \times 60{,}000 \times (17 - 0.912)$$

Mn = 93,000 × 16.088 = 1,496,184 in-lb = 1,496.2 in-kip
φMn = 0.90 × 1,496.2 = 1,346.6 in-kip

Demand: Mu = 622.8 in-kip
DCR = Mu / φMn = 622.8 / 1,346.6 = 0.46 → PASS

The large reserve capacity (DCR = 0.46) reflects the relatively modest span-to-depth ratio of the beam. The Skill notes that while this beam is well within its flexural limit, the low DCR on flexure combined with the higher DCR on shear (Check 4, DCR = 0.84) suggests that the design is shear-critical rather than moment-critical, a pattern the Skill explicitly draws the engineer's attention to in its summary.

4.2.2 Check 4 - Shear strength: concrete contribution [ACI §22.5.5.1]

This check demonstrates Skill's implementation of the updated ACI 318-25 Vc equation, the most significant change from the pre-2019 ACI editions. The new table-based method in §22.5.5.1 requires the longitudinal reinforcement ratio and includes a size modification factor absent from the earlier simplified equations:

Longitudinal reinforcement ratio:

$$\rho w = As / (bw \times d) = 1.55 / (15 \times 17) = 1.55 / 255 = 0.00608$$

Size modification factor λs [§22.5.5.1.3]:

$$\lambda s = min(1.0, \sqrt{(2 / (1 + 0.004 \times d))})$$

= min(1.0, √(2 / (1 + 0.004 × 17)))
= min(1.0, √(2 / 1.068)) = min(1.0, 1.368) = 1.0 (capped)

λ = 1.0 (normal-weight concrete) [§19.2.4]
(ρw)^(1/3) = (0.00608)^(1/3) = 0.1826

Vc [Table 22.5.5.1, Nu = 0]:

$$Vc = [8\lambda s \times \lambda \times (\rho w)\text{^}(1/3) \times \sqrt{f'c}] \times bw \times d$$

= [8 × 1.0 × 1.0 × 0.1826 × √4,000] × 15 × 17
= [8 × 0.1826 × 63.25] × 255
= 92.48 × 255 = 23,582 lb = 23.58 kip

φVc = 0.75 × 23.58 = 17.69 kip
Vu,critical = 14.88 kip (at d from face of support [§9.4.3.2])
DCR = 14.88 / 17.69 = 0.84 → PASS

The DCR of 0.84 on concrete shear alone is the highest in this example and is the single most important result for the engineer to see. The Skill flags this explicitly: without stirrups, this beam would be operating at 84% of its concrete shear capacity at the critical section. The stirrups increase the total design shear strength to φVn = 45.74 kip (Check 5, DCR = 0.45), providing substantial additional reserve, but the bare concrete shear utilization is sufficient to warrant engineering attention. The Skill's practice of presenting the DCR table sorted by utilization ratio rather than by check number ensures that this result is immediately visible.

4.2.3 Check 9 - Long-term deflection [ACI §24.2.4]

The long-term deflection check demonstrates Skill's multistep calculation sequence involving the effective moment of inertia, cracking moment, and time-dependent multiplier:

Elastic modulus [§19.2.2.1(a)]: $Ec = 57{,}000 \times \sqrt{f'c} = 57{,}000 \times 63.25 = 3{,}605{,}250\ psi$

Gross moment of inertia: $Ig = bw \times h^3 / 12 = 15 \times (20)^3 / 12 = 10{,}000\ in^4$

Modulus of rupture [§24.2.3.5]: $fr = 7.5\lambda\sqrt{f'c} = 7.5 \times 1.0 \times 63.25 = 474.3\ psi$

Cracking moment: $Mcr = fr \times Ig / yt = 474.3 \times 10{,}000 / 10 = 474{,}300\ in-lb = 474.3\ in-kip$

Service moment (D + L unfactored):

Wa = 1.460 + 1.500 = 2.960 kip/ft

Ma = 2.960 × (10)² / 8 = 37.0 ft-kip = 444.0 in-kip

Since Ma (444.0 in-kip) < Mcr (474.3 in-kip): section is UNCRACKED

→ Ie = Ig = 10,000 in⁴

Immediate dead-load deflection:

wD,in = 1,460/12 = 121.67 lb/in

Δimm,D = 5 × 121.67 × (120)⁴ / (384 × 3,605,250 × 10,000) = 0.0911 in

Immediate live-load deflection:

wL,in = 1,500/12 = 125.0 lb/in

Δimm,LL = 5 × 125.0 × (120)⁴ / (384 × 3,605,250 × 10,000) = 0.0936 in

Allowable (L/360) = 120/360 = 0.333 in → DCR = 0.0936/0.333 = 0.28 → PASS

Long-term multiplier [§24.2.4.1.1]:

ξ = 2.0 (sustained load duration ≥ 5 years — confirm with engineer)

$\rho' = A's / (bw \times d) = 0.40 / (15 \times 17) = 0.001569$

$$\lambda\Delta = \xi / (1 + 50\rho') = 2.0 / (1 + 50 \times 0.001569) = 2.0 / 1.0784 = 1.855$$

ΔLT = λΔ × Δimm,D = 1.855 × 0.0911 = 0.169 in
Δtotal = ΔLT + Δimm,LL = 0.169 + 0.0936 = 0.263 in
Allowable (L/240) = 120/240 = 0.500 in → DCR = 0.263/0.500 = 0.52 → PASS

The uncracked section condition (Ma < Mcr) is a result that the skill identifies explicitly. In a spreadsheet, the effective moment of inertia Ie might be computed by a formula without the engineer noticing that the result equals Ig, meaning the cracking check was effectively bypassed. The Skill states this condition clearly: the section is uncracked, Ie = Ig, and the cracking moment check is satisfied by a margin of only 6.4%, a narrow margin that the skill flags as noteworthy, even though the deflection check itself passes comfortably.

4.2.4 Check 10 - Minimum concrete cover [ACI §20.5.1.2]

The cover check demonstrates Skill's BORDERLINE flagging at DCR = 1.00, technically passing but warranting explicit engineering attention:

Exposure: Not exposed to weather or earth
Required cover to stirrups [§20.5.1.2, bars ≤ #5]: 1.5 in
Provided cover to stirrups: 1.5 in
DCR = 1.5 / 1.5 = 1.00 → PASS

ENGINEER FLAG: Provided cover equals exactly the code minimum.
Zero construction tolerance for bar placement error.
Evaluate whether specifying 1.75 in or 2.0 in is warranted for constructability.

This flag illustrates a category of engineering judgment in which a simple PASS/FAIL check cannot be captured. The cover requirement satisfies the millimeter, but field bar placement involves tolerance variation, and specifying the exact minimum with no reserve means that any positive bar displacement toward the form face produces a code violation. The Skill's automatic flagging of DCR = 1.00 ensures that this condition is always visible to the reviewing engineer.

4.3 Summary DCR table and figure

Table 3 presents the complete DCR summary as produced by the Skill following all ten checks, sorted by DCR from highest to lowest. Figure 2 presents the same results as a horizontal bar chart, providing immediate visual identification of the most critical checks and the proximity of each result to the code limit and BORDERLINE threshold.

Table 3: *Complete DCR summary for the worked example beam - all ten checks, sorted by DCR (descending)*

#	Check name	Demand	Capacity	DCR	Status
10	Concrete cover [§20.5.1.2]	1.50 in	1.50 in	1.00	PASS - FLAG
4	Shear - Vc [§22.5.5.1]	14.88 kip	17.69 kip	0.84	PASS
7	Max stirrup spacing [§9.7.6.2.2]	6.0 in	8.5 in	0.71	PASS
1	Min. flexural steel [§9.6.1.2]	0.850 in²	1.55 in²	0.55	PASS
9	Long-term deflection [§24.2.4]	0.263 in	0.500 in	0.52	PASS
3	Flexural strength [§22.2, §22.3]	622.8 in-kip	1,346.6 in-kip	0.46	PASS
5	Shear - total Vn [§22.5.8.5.3]	20.76 kip	45.74 kip	0.45	PASS
6	Min. shear reinf. [§9.6.3.4]	0.01250 in²/in	0.03667 in²/in	0.34	PASS
8	Immediate deflection [§24.2.2]	0.0936 in	0.333 in	0.28	PASS
2	Net tensile strain [§21.2.2]	εt,min = 0.004	εt = 0.02076	0.19	PASS

All ten checks: PASS. No BORDERLINE flags (DCR > 0.90) except Check 10 at DCR = 1.00. Section citations reference ACI 318-25. Results require PE review before use in any engineering document or decision.

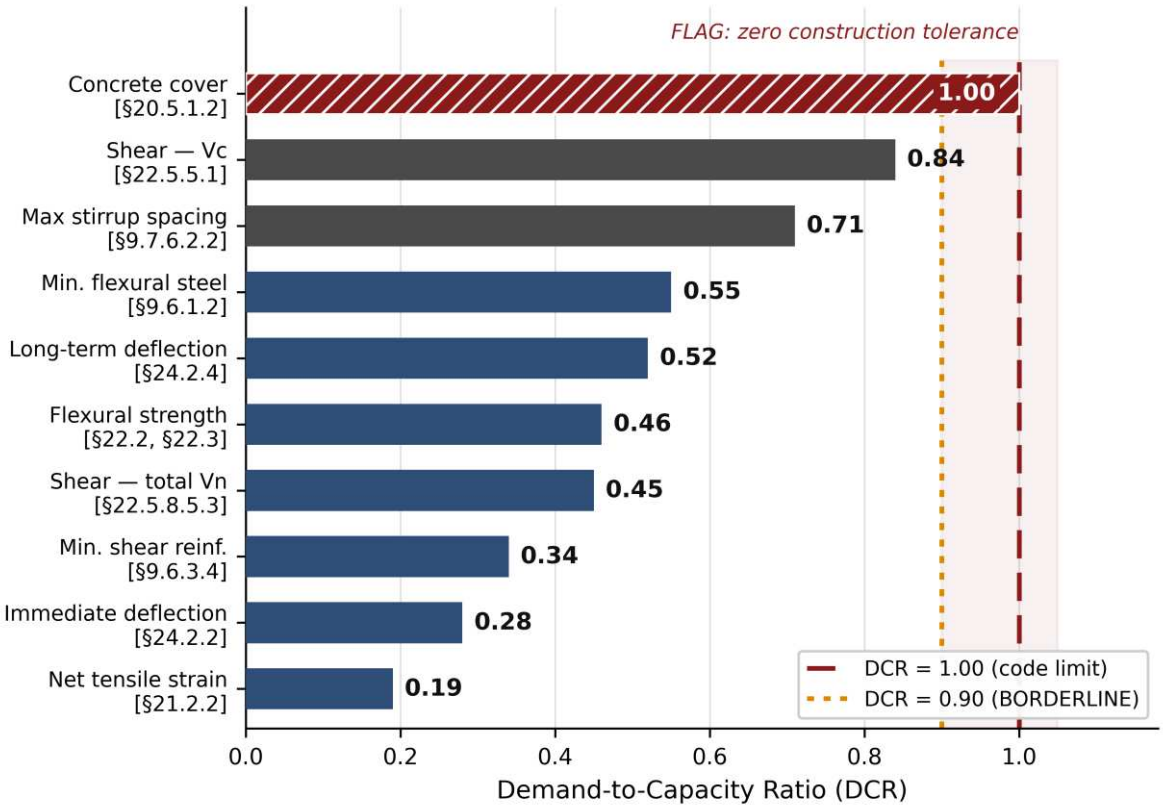

***Fig 2**: DCR values for all ten code checks for the worked example beam. Checks sorted by utilization ratio (descending). Dark red dashed line marks DCR = 1.00 (code limit); orange dotted line marks DCR = 0.90 (BORDERLINE threshold). Check 10 (concrete cover) at DCR = 1.00 triggers a constructability flag (hatched bar). Check 4 (shear Vc) at DCR = 0.84 is the highest structural utilization*

5 Comparison with traditional spreadsheet-based checking

Table 4 presents a structured comparison of AI Skill checking against traditional spreadsheet-based checking across the eight dimensions of engineering practice quality and accountability. This comparison is grounded in the specific characteristics of the ACI 318-25 Beam Checking Skill and the worked example results presented in Section 4, rather than in generalized claims about AI versus spreadsheets.

***Table 4:** Comparison of AI Skill checking and traditional spreadsheet checking across eight practice dimensions*

Dimension	Traditional spreadsheet	AI Skill (this work)
Assumption transparency	Assumptions embedded in cell formulas; often implicit or undocumented in output	Every assumption stated explicitly and confirmed by engineer before calculation

		proceeds; all assumptions listed in session summary
Code traceability	Code section may appear in a text cell; often absent from printed output	Every check cites the specific ACI 318-25 section number governing it - mandatory, not optional, per Skill design principle 3
Missing data handling	Spreadsheet may auto-calculate with default or zero values without engineer awareness	Skill stops and requests clarification; will not proceed until all required inputs are confirmed
Borderline case detection	No automatic flagging - engineers must notice DCR proximity to limit	All checks with DCR > 0.90 flagged BORDERLINE; DCR = 1.00 on cover (this example) triggers constructability flag regardless of pass/fail
Interrogability	Engineers must trace cell references across sheets to understand logic	Engineers can ask 'Why does the shear check show 0.84?' and receive ACI §22.5.5.1 explanation with full intermediate values
PE accountability	Calculation printed and signed - static document	Output explicitly requires PE review; dynamic calculation record - each step traceable to input, formula, and code section
Scope boundary	Spreadsheets may silently apply outside intended scope if inputs are changed	Skill states its scope explicitly and flags out-of-scope conditions rather than proceeding silently

Version control	Spreadsheet file version unclear; cells can be accidentally modified	Skill version documented in Zenodo DOI [12]; instruction set fixed and auditable per version

DCR = demand-to-capacity ratio.

The most significant difference between the two approaches from a professional practice perspective is interrogability. In the worked example presented in Section 4, the most critical structural check is shear at DCR = 0.84 (visible at a glance in Figure 2). In a spreadsheet, a reviewer who wants to understand why this value is 0.84 must trace through the cell formula for Vc, locate the values of ρw and λs, verify that the table-based §22.5.5.1 method was used rather than the legacy simplified formula, and confirm that the correct d was applied. This tracing exercise typically requires familiarity with the structure of a specific spreadsheet and may require speaking with the original author.

In an AI Skill session, the reviewing engineer type: 'Why is the shear DCR 0.84?' The Skill responds with the complete §22.5.5.1 calculation with ρw, λs, the table-based Vc formula, the substituted values, the final Vc = 23.58 kip, and the critical section location at d from the support face per §9.4.3.2 in less time than it takes to open the spreadsheet's formula bar. This is not a cosmetic advantage; it is the specific capability that directly addresses the trust and reliability concerns that AASHTO survey respondents identified as a leading barrier to AI adoption [13] and that current AI structural tools, including both design generators and BIM checkers, do not provide.

A second dimension worth analyzing in detail is the handling of missing data. Traditional spreadsheets are vulnerable to silent defaults; a cell that was manually entered for one project may persist into the next project unchecked. In the worked example, the critical assumption that the long-term deflection multiplier ξ = 2.0, corresponding to a sustained load duration of five years or more per ACI (Table R24.2.4.1.3), is explicitly stated by Skill and flagged for engineer confirmation. If the actual project load duration is two years rather than five, ξ = 1.4 applies, and the long-term deflection DCR changes from 0.52 to approximately 0.43. A spreadsheet that hardcodes ξ = 2.0 without labeling the assumption may produce a different result for the same beam without the engineer noticing. The Skill's explicit flagging makes this sensitivity visible.

The version control dimension is also practically significant. The ACI 318-25 Beam Checking Skill is distributed as a version of a permanent Zenodo DOI [12], meaning that any engineering calculation citing the skill can reference a specific, permanently retrievable, human-readable document that defines exactly what logic was applied. If a dispute arises in a design review, forensic investigation, or code compliance audit, the version of the skill used can be retrieved and reviewed by any party. This level of auditability is difficult to achieve with internal firm spreadsheets, which are often not version-controlled and can be modified by multiple engineers over time.

It is important to note that this comparison does not argue that AI Skills should replace the calculation spreadsheets. For repetitive, well-established beam designs in which the

engineer is highly familiar with the calculation logic, a well-maintained spreadsheet remains an efficient and appropriate tool. The AI Skill adds the most value in three specific scenarios: (1) when a practitioner is working outside the area of highest familiarity. For example, a bridge engineer checking a building beam or vice versa needs step-by-step guidance through the code provisions; (2) when an engineer checks someone else's design and needs to interrogate the rationale behind specific values; and (3) when an engineering student or early career engineer is developing their code checking competency and benefits from the section-cited calculation trial.

6 Accountability framework and PE certification compatibility

6.1 The PE certification requirement

In the United States and most international jurisdictions, any structural engineering recommendation that affects public safety must be certified by a licensed Professional Engineer. The certifying PE must be able to independently verify, challenge, and take personal professional accountability for the engineering judgments expressed in the document. This requirement is not a bureaucratic formality; it is the mechanism by which engineering responsibility is anchored to an identifiable individual who has demonstrated competency and is subject to professional discipline if their judgments are negligent or incorrect.

AI tools that generate structural designs or check structural compliance create an accountability challenge: If the AI produces a result that is incorrect, who bears professional responsibility? The answer, under current licensing frameworks in all US jurisdictions, is unambiguous: the PE who signed and sealed the document bears responsibility regardless of whether an AI was involved in producing the calculation. This accountability assignment does not change when AI is used, but it changes the PE's verification burden. Because the PE must now verify not only that the calculation is correct but also that the AI tool's methodology is appropriate, its code citations are accurate, and its assumptions are acceptable for this specific application.

The AI Skill paradigm is designed specifically to support this verification burden rather than obscure it. The session record produced by the skill contains everything the PE needs to perform their verification: the complete input set with all assumptions stated, the full calculation for each check with intermediate values, the specific ACI 318-25 section number governing each check, and a flagged list of borderline cases and conditions requiring special attention. The PE does not need to trust the Skill's results; they are provided with everything necessary to independently verify those results step by step. Complementary governance approaches are emerging in the civil engineering literature: Poudel and Surface propose post-hoc audits that compare LLM advice against locally validated empirical risk models [19]; the Skill paradigm addresses the same misalignment risk structurally, by constraining the model to deterministic, section-cited check logic before any output reaches the engineer.

6.2 Comparison with commercial software accountability

The professional accountability structure that AI Skill supports is directly analogous to the accountability structure that has governed the use of commercial structural analysis

software, including RISA-3D, ETABS, SAP2000, and STAAD. Pro in licensed engineering practice over the past three decades. Licensed engineers routinely sign and seal documents whose results are produced by these software packages, and they take full professional accountability for those results without having written, reviewed, or even fully understood the source code of the software. The professional standard is not that the PE must write the calculation from first principles; it is that the PE must understand the methodology, verify that it is appropriate for the application, and check the output for reasonableness.

AI Skill satisfies this standard more transparently than most commercial software because it produces a step-by-step calculation record rather than a result summary. When a commercial analysis package reports a moment demand of 622.8 in-kip, the PE is typically provided with a summary table, not with the intermediate calculation showing Wu = 4.152 kip/ft, L = 10 ft, and Mu = Wu × L²/8 = 51.90 ft-kip. The AI Skill shows all the three steps. This is not a theoretical advantage; it is a practical improvement in the reviewability of engineering calculations that directly reduces the verification effort required to certify the PE.

Mangalathu et al. [14] demonstrated in the context of machine learning-based structural assessment that PE adoption of AI tools depends fundamentally on the engineer's ability to understand and independently assess analytical reasoning, which they termed interpretability at the level of individual prediction. Their SHAP-based framework for ML model interpretation provides a useful conceptual analogy for the AI Skill's step-by-step calculation display: just as SHAP which allows a reviewer to understand how each input feature contributed to an ML prediction. Skill's sequential check format allows a PE to understand how each input parameter drives each DCR value. The DCR chart in Figure 2 provides a visual equivalent of an immediate, single-glance overview of the relative utilization of every code provision, enabling the rapid identification of the checks that warrant the most careful PE attention.

6.3 The PE disclaimer as a design feature

The mandatory PE disclaimer that closes every AI Skill session is not a legal hedge; it is a design feature that is fundamental to the framework's professional accountability structure. The disclaimer explicitly states that the session output has not been stamped, sealed, or certified by a licensed engineer; that the results must be independently verified by a PE before use in any engineering decision or document; and that the engineer of record assumes full professional responsibility for all design decisions. This disclaimer appears in every session regardless of how confident the results appear, how straightforward the beam geometry is, or how experienced the engineer using the skill is.

The reason this is a design feature rather than a disclaimer is that it actively reconfigures the engineer's relationship with the Skill's output. An engineer who reads 'All 10 checks: PASS' without the disclaimer might be tempted to treat that result as a certification. An engineer who reads 'All 10 checks: PASS - requires independent PE verification before use in any engineering document' is structurally reminded that they are responsible for verifying the result, not merely for recording it. This distinction is behaviorally significant: it maintains the PE accountability chain in the engineer's mental model of the calculation process, not merely in the legal fine print of a terms-of-service agreement.

The framework described in this paper therefore represents not merely a technical tool but a professional practice protocol, in which AI enhances the quality, transparency, and traceability of structural code checking while deliberately preserving and strengthening the PE accountability structure that makes licensed engineering practice trustworthy to the public it serves.

7 Limitations and future development

The current ACI 318-25 Beam Checking Skill has several limitations that define its appropriate use. The skill covers simply supported and continuous rectangular T-beams under standard gravity loading. Seismic provisions (Chapter 18), torsion (§22.7), deep beams (§9.9), prestressed members, and connections were outside the scope. These limitations are explicitly documented in the skill and are flagged when out-of-scope inputs are detected. The broader landscape of AI applications in civil engineering surveyed comprehensively by Nyokum and Tamut [18] confirms that scope specificity and disciplinary focus are characteristics shared by the most effective current AI tools for engineering practice; general-purpose tools without domain constraints consistently underperform domain-specific tools in high-stakes applications.

This skill also depends on the mathematical reasoning accuracy of the underlying LLM. While current LLMs perform reliably on the sequential arithmetic required for routine beam checking, engineers should independently verify the numerical results for critical calculations, particularly deflection computations, where sequential rounding can accumulate. As Orris and Ssegawa [11] found and as Liu et al. [16] confirmed quantitatively, LLMs without a structured protocol are prone to numerical errors in structural calculations; Skill's structured protocol substantially mitigates but does not eliminate this risk. A formal validation study comparing Skill outputs against commercial software results across a statistically representative sample of beam configurations is planned to establish quantitative error bounds for the framework.

Future development directions include extension to additional structural elements (columns, slabs, connections), additional codes (AISC 360 for steel and AASHTO LRFD for bridge beams), and metric unit systems for international engineering practice. The AASHTO LRFD extension is of particular significance: highway bridge girder checking under AASHTO provisions follows a similar sequential check structure to the ACI 318-25 beam checks demonstrated here, and a bridge girder checking skill directly connects this framework to the infrastructure resilience research context that motivates this work [6].

8 Conclusion

This paper introduced AI Skill as a structural engineering checking assistant: a reusable, practitioner-accessible, and PE-accountable framework for conversational, section-cited code compliance checking of reinforced concrete beams. The framework is built on four design principles: no assumption of silence, full calculation transparency, mandatory code traceability, and scope honesty that directly address the accountability and explainability requirements of licensed structural engineering practices.

The worked example presented in Section 4 demonstrates Skill's operation on a simply supported 15 in × 20 in rectangular beam, with four checks shown in full step-by-step

detail and all ten checks summarized in Table 3 and visualized in Figure 2. The most critical check, shear at DCR = 0.84, was immediately visible as the dominant bar in the DCR chart. The cover check at DCR = 1.00 demonstrated the skill's BORDERLINE flagging capability, identifying a constructability concern that a pass/fail check would not surface. The deflection check demonstrated Skill's multistep Ie calculation, including the explicit identification of the uncracked section condition with a 6.4% margin above the cracking moment.

The comparison in Section 5 demonstrates measurable advantages over traditional spreadsheet checking across eight practice dimensions, with interrogability and assumption transparency identified as the most consequential. The accountability framework analysis in Section 6 demonstrates that the skill is compatible with, and designed to strengthen, the PE certification chain, offering a level of calculation transparency that exceeds that provided by most commercial structural software. The AASHTO findings that trust and reliability concerns rank among the leading barriers to AI adoption among state DOTs and that engineering stamping enforcement is the most commonly implemented AI governance measure [13] directly validate these as the correct dimensions for evaluating structural AI tools.

The ACI 318-25 Beam Checking Skill is publicly available via Zenodo (DOI: https://doi.org/10.5281/zenodo.20216886) [12]. The Skill can be installed and activated immediately by engineers using Claude Pro or an equivalent LLM interface. The authors welcome peer review of the Skill's methodology and scope boundaries and invite collaboration on extensions to additional structural elements, codes, and engineering disciplines.

Supplementary information

The ACI 318-25 Beam Checking Skill instruction document (Version 3.0, Imperial Units) is provided as Supplementary Information and is permanently archived at: https://doi.org/10.5281/zenodo.20216886

Author contributions

Rinaz Riyaz Mohamed: Conceptualization, Methodology, Investigation, Formal analysis, Software, Writing - original draft.

Rahul Anand: Writing - review and editing, Formal analysis, Data curation.

Jigarbhai Mansukhbhai Sonani: Writing - review and editing, Validation.

Funding

The authors declare that they received no financial support for the research, authorship, or publication of this article.

Data availability

The ACI 318-25 Beam Checking Skill instruction document is publicly available through Zenodo (DOI: https://doi.org/10.5281/zenodo.20216886). The worked example inputs and outputs are self-contained and reproducible using the publicly available Skill.

Declarations

Ethics approval and consent to participate: Not applicable.

Consent for publication: Not applicable.

Competing interests: The authors declare no competing interests.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process: During the preparation of this work, the authors used Claude (Anthropic) to assist with language refinement and paragraph structuring based on author-provided directions. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article. All research concepts, framework design decisions, engineering judgments, ACI code analyses, and intellectual content are the sole responsibilities of the authors.

References

1. Khan J, Paka S, Bukaita W. A multi-output deep learning framework for automated reinforced concrete beam design and detailing. Am J Civil Eng. 2025;13(6). https://doi.org/10.11648/j.ajce.20251306.13.
2. Hosseinzadeh A, Dehestani M. Intelligent low carbon reinforced concrete beam design optimization via deep reinforcement learning. Sci Rep. 2025;15(1):12443. https://doi.org/10.1038/s41598-025-18543-4.
3. He Z, Wang Y. H, Zhang J. Generative AI-BIM: An automatic and intelligent structural design pipeline integrating BIM and generative AI. Inf Fusion. 2025;114:102654. https://doi.org/10.1016/j.inffus.2024.102654.
4. Reid R. What do civil engineers need to know about artificial intelligence? Civil Eng Mag ASCE. 2024. https://www.asce.org/publications-and-news/civil-engineering-source/civil-engineering-magazine/issues/magazine-issue/article/2024/11/what-do-civil-engineers-need-to-know-about-artificial-intelligence.
5. Vilas Serín A. EC7 Bearing Capacity Review Skill - Demo Version [Software]. GitHub; 2025. https://github.com/Arabelvilas/ec7-bearing-capacity-claude-skill-demo.
6. Selvaratnam T, Mohamed RR, Eren-Tokgoz B, Haselbach L, Gummelt G, Williams BD, Pyne MI. A data-driven approach for decision making in a regional interdisciplinary resiliency center. Int J Disaster Risk Sci. 2025;16(3):513–520. https://doi.org/10.1007/s13753-025-00643-4.
7. Dadras Eslamlou A, Naser MZ. Can chatbots design and analyze steel structures? J Struct Des Constr Pract ASCE. 2026. https://doi.org/10.1061/JSDCCC.SCENG-2103.

8. Plevris V, Hosamo H. Responsible AI in structural engineering: a framework for ethical use. Front Built Environ. 2025;11:1612575. https://doi.org/10.3389/fbuil.2025.1612575.
9. American Concrete Institute. ACI 318-25: Building Code Requirements for Structural Concrete and Commentary. Farmington Hills, MI: ACI; 2025.
10. American Society of Civil Engineers. ASCE/SEI 7-22: Minimum Design Loads and Associated Criteria for Buildings and Other Structures. Reston, VA: ASCE; 2022.
11. Orris C, Ssegawa P. Assessment of the potential of emerging large language models in structural analysis: a case study on beam analysis [dissertation]. Makerere University; 2025. https://dissertations.mak.ac.ug/items/aebc2546-2194-44c8-adbc-1747807814af.
12. Mohamed RR, Anand R, Sonani JM. ACI 318-25 Concrete Beam Checking Assistant - A Claude AI Skill for Structural Engineering Code Compliance (Version 3.0, Imperial Units) [Software]. Zenodo; 2026. https://doi.org/10.5281/zenodo.20216886.
13. American Association of State Highway and Transportation Officials, Committee on Transportation System Operations. Artificial intelligence adoption and implementation in state DOT transportation operations: a whitepaper. Washington, DC: AASHTO; 2026. https://transportation.org/systemoperations/wp-content/uploads/sites/2/2026/01/AI_Whitepaper_2026.pdf.
14. Mangalathu S, Hwang SH, Jeon JS. Failure mode and effects analysis of RC members based on machine-learning-based Shapley additive exPlanations. Eng Struct. 2020;219:110927. https://doi.org/10.1016/j.engstruct.2020.110927.
15. Joffe I, Felobes G, Elgouhari Y, Talebi Kalaleh M, Mei Q, Chui YH. The framework and implementation of using large language models to answer questions about building codes and standards. J Comput Civ Eng. 2025;39(4):05025004. https://doi.org/10.1061/JCCEE5.CPENG-6037.
16. Liu J, Geng Z, Cao R, Cheng L, Bocchini P, Cheng M. A large language model-empowered agent for reliable and robust structural analysis. Struct Infrastruct Eng. 2026. https://doi.org/10.1080/15732479.2026.2630123.
17. Liang H, Talebi Kalaleh M, Mei Q. Integrating large language models for automated structural analysis. arXiv preprint arXiv:2504.09754; 2025. https://doi.org/10.48550/arXiv.2504.09754.
18. Nyokum T, Tamut Y. Artificial intelligence in civil engineering: emerging applications and opportunities. Front Built Environ. 2025;11:1622873. https://doi.org/10.3389/fbuil.2025.1622873.
19. Poudel A, Surface T. A governance audit framework for large language model advice in civil infrastructure decision-making illustrated with local risk models in Sugar Land, Texas. Discov Civ Eng. 2026;3:91. https://doi.org/10.1007/s44290-026-00474-2.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- ACIBeamCheckerv3.docx