

Performing Structural Analysis and Design of 3D Space Frame in STAAD.Pro Using LLM's: A Practical Framework for Automated Structural Engineering Workflows

Syed Munawar Ali¹

¹Lead Structural Engineer, EGIS, Saudi Arabia

Publication Date: 2026/10/22

Abstract: Generative artificial intelligence has moved quickly into text and software work, but its place in everyday structural engineering is still poorly defined. This paper sets out a practical framework in which a large language model (LLM), Claude, takes a steel structure from natural-language prompts and existing calculation documents to a complete STAAD.Pro input file, code-based member checks, foundation, base plate, anchor bolt and connection designs, and a calculation report. It was applied to a representative industrial canopy: a 25.7 m × 9.25 m double-layer space truss with 493 joints and 1,820 members on two inclined struts and two vertical columns. From short prompts, the model produced the joint coordinates, member incidences, properties and supports, 15 primary load cases to ASCE 7-10 and 80 ASD and LRFD combinations. Results were verified by global equilibrium and hand checks. All members satisfied AISC 360-16, with maximum utilisation ratios of 0.74 (ASD) and 0.71 (LRFD); roof deflection was span/928. The governing ratios for footing flexure, base plate, anchorage and the critical truss node were 0.94, 0.97, 0.80 and 0.94. Total effort was estimated to fall from about 208 h to 92 h (56 %), with most of the saving in geometry modelling and reporting, while checking time rose by a quarter. Seven errors of AI origin were caught by the verification gates and during detailing. The LLM removed most transcription work, but only within a workflow in which the engineer fixes the design basis and checks every output.

Keywords: Large Language Models; Claude AI; Structural Engineering; STAAD.Pro; Artificial Intelligence; Structural Analysis; Steel Design; Foundation Design; Engineering Automation.

How to Cite: Syed Munawar Ali (2026) Performing Structural Analysis and Design of 3D Space Frame in STAAD.Pro Using LLM's: A Practical Framework for Automated Structural Engineering Workflows. *International Journal of Innovative Science and Research Technology*, 11(10), 1333-1339. <https://doi.org/10.38124/ijisrt/26oct831>

I. INTRODUCTION

Civil engineering has passed through several waves of digital change, from matrix analysis to building information modelling (BIM) [1,2]. Analysis is now fast and reliable; the work around it has changed far less. An engineer setting up a STAAD.Pro model still turns drawings into joint coordinates, numbers members, assigns sections and supports, derives wind and seismic loads, types load combinations, and copies results into calculation sheets and a report. For a space truss this becomes the main job: the input file for the structure in this paper exceeds 3,000 lines. Typing errors in such files are dangerous because the solver cannot see them; a wrong coordinate or combination factor still gives a model that runs and produces believable results.

Computational structural engineering has used artificial intelligence (AI) for decades, in neural networks, expert systems and smart design systems for member sizing [3–7], but these tools needed training data or a ready-made

analysis model and few reached routine office use. LLMs built on the transformer architecture [8,9] are different in kind: they take plain-language instructions, write code and structured text, and need no training on the user's data. Controlled studies of generative AI report time savings of about 40 % on professional writing tasks [10], and the first studies in structural analysis show LLMs producing solver input from verbal descriptions [11,12].

What is missing is an account from practice of the whole chain from a prompt to an issuable design, including the substructure and connections. This paper defines a five-step framework for using an LLM with STAAD.Pro, applies it to a real-scale industrial canopy with all project identifiers removed, and reports the results, the estimated time and the errors the AI made, so that others can judge where the method helps and where it needs control.

II. LITERATURE REVIEW

➤ AI in Structural Engineering

Reviews by Adeli [3] and by Salehi and Burgueño [4] trace AI in structural engineering from expert systems and neural networks to pattern recognition. Later reviews [5,13] show machine learning (ML) predicting member capacities, seismic response and damage with good accuracy, but each model is valid only within its training data and rarely connects to the documents a design office must issue.

➤ Machine Learning and Optimisation in Design

Metaheuristic methods such as ant colony optimisation have sized space trusses against code constraints for twenty years [6,7]. They reduce weight but assume the analysis model already exists, and building that model is what consumes the engineer's time.

➤ Natural Language Processing in Engineering

Natural language processing (NLP) has been applied to specifications and codes, and LLM assistants retrieve information from BIM models through conversational queries [14]. In construction management, AI serves mainly scheduling, safety and document handling [2].

➤ AI-Assisted Finite Element Modelling

AI-assisted finite element (FE) modelling has mostly meant surrogate models that replace the solver. Liang et

al. [11] and Liu et al. [12] reversed this: the LLM writes the input for an established solver, which remains the source of truth.

➤ LLMs in Engineering Calculations

LLMs can produce fluent but wrong numbers, and people tend to over-trust automated output that is usually right [15]. Published LLM studies in structural analysis use benchmark problems of modest size; none covers the code design of foundations, anchors and connections or reports the errors found in a real design. This study addresses both gaps.

III. METHODOLOGY

The framework is shown in Fig. 1. One rule governs it: *the LLM produces things whose correctness can be checked mechanically, and the engineer supplies every design value.* A script that writes 1,820 member lines can be checked by counting and by running the file; a combination line can be compared with the code factor; a number recalled from memory cannot be checked, so none was accepted. Four verification gates (G1–G4) sit between the steps, and a failed gate returns the work to the engineer, who corrects the prompt or data and logs the defect.

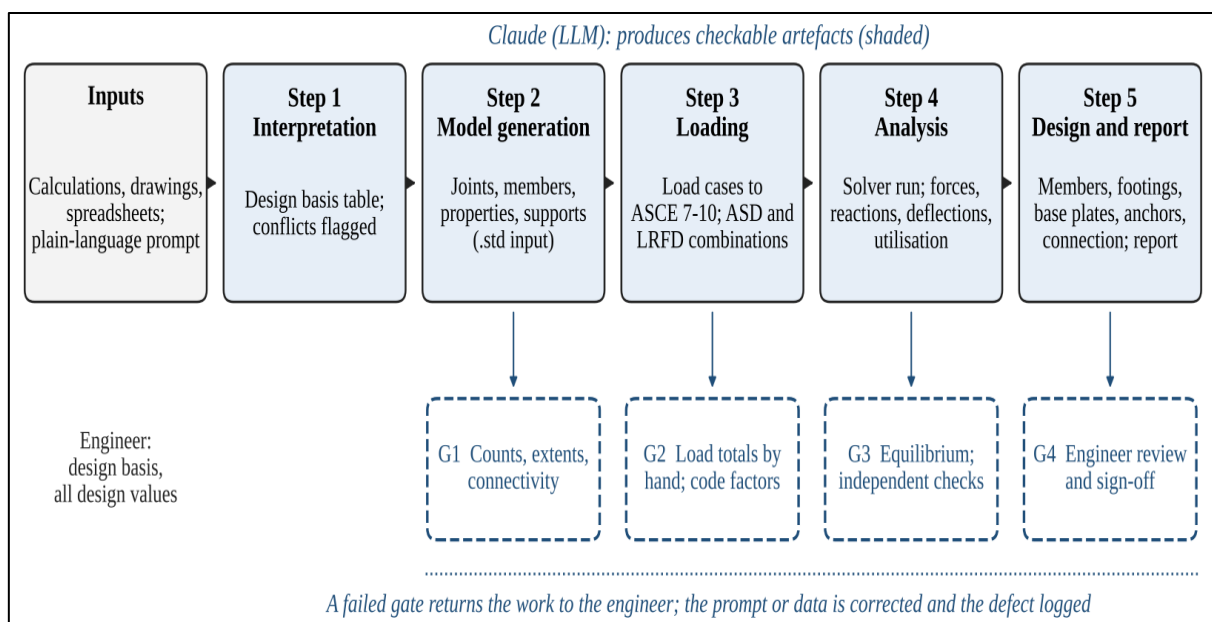


Fig 1 Prompt-to-Design Framework. The LLM Converts Documents and Prompts into Checkable Artefacts (Input Files, Workbooks, Report Text); the Engineer Owns the Design Basis and Every Verification Gate.

➤ Step 1: Interpretation of Engineering Documents

The inputs were an existing structural calculation for a similar canopy, the architectural plan and section, and the engineer's instructions for the foundation, base plate, anchor bolt and connection design. Claude returned a design basis as a table of parameters, values and source documents (Table 1). The step proved more useful than expected

because the model reported inconsistencies as well as values: the wind and seismic parameters of the reference calculation belonged to a different site, and the specified bearing pressure was a sound-rock value with no geotechnical report behind it. Both were kept as stated assumptions pending confirmation.

Table 1 Design Basis Extracted in Step 1 and the Check Applied by the Engineer.

Parameter	Value adopted	Source	Engineer's check
Plan, module, levels	25.685 × 9.250 m; 1.027 m grid; top +7.050 to +6.368 m	Architectural section	Dimensions re-measured
Sections, steel	Chords 73.0 × 5.16; webs 48.3 × 3.68; CHS 508 × 16; $F_y = 250$ MPa	Reference model	Section tables; ASTM minima
Wind	$V = 46$ m/s, Exposure D, $K_d = 0.85$	Reference calculation	Site Location : Confirmed
Seismic	$S_S = 0.435$ g, $S_1 = 0.124$ g, Site Class D, $R = 1.25$	Reference calculation	Site Location : Confirmed
Foundation	$f'_c = 35$ MPa, $f_y = 420$ MPa, bearing as specified	Engineer's instruction	Confirmed from geotechnical report

➤ *Step 2: Generation of the STAAD.Pro Model*

The model was requested in ordinary engineering language. A generic request such as “Generate a STAAD.Pro model of a steel canopy consisting of columns, rafters, purlins and bracing” yields a portal-frame model; because the case-study roof is a space truss, the prompt described its geometry directly (Table 2). Claude wrote a

parametric script that emits the STAAD.Pro input, so the 493 joints and 1,820 members follow from rules rather than typing. Top-layer joint i of row k lies at with the bottom layer offset by half a module ($\Delta_x = 1.0274$ m, $\Delta_z = 1.0278$ m). Gate G1 checked the counts in every group, the extents, the member lengths (1.027 m for chords, 1.66–2.09 m for webs) and the connectivity of every joint.

$$x = i \Delta_x, \quad y_t = 7.050 - 0.02655 x, \quad z = -k \Delta_z; \quad y_b = 5.086 + 0.01051 x \quad (1)$$

Table 2 Representative Prompt and Extract of the Generated STAAD.Pro input (Condensed).

Engineer's prompt	Claude output (extract of the generated .std file)
“Generate a STAAD.Pro input for the canopy in the attached calculation. Roof: double-layer square- on-square space truss, 25.685 m × 9.250 m, 25 × 9 bays. Top layer falls from +7.050 m to +6.368 m along X; bottom layer offset half a module, +5.086 m to +5.356 m. Two inclined struts (base X = 3.570 m, head X = 1.027 m) and two vertical columns at X = 24.658 m, CHS 508 × 16, fixed bases. Chords PIPS25; webs and head branches PIPS15, pin-ended. Units kN and m. Number top joints first, then bottom joints, heads and bases. Print the counts per group and the shortest and longest member.”	JOINT COORDINATES 1 0.0000 7.0500 0.0000; 2 1.0274 7.0227 0.0000; ... 261 0.5137 5.0914 -0.5139; ... 490 3.5700 0.0000 -1.0278; MEMBER INCIDENCES 1 1 2; ... 485 261 262; ... 901 261 1; ... 1817 490 486; DEFINE MATERIAL START ... E 2.05e+08 ... END DEFINE MATERIAL MEMBER PROPERTY AMERICAN 1 TO 900 TABLE ST PIPS25 901 TO 1816 TABLE ST PIPS15 SUPPORTS 490 TO 493 FIXED MEMBER TRUSS 901 TO 1816 * 493 joints, 1820 members; chords 1.027 m, webs 1.66-2.09 m

➤ *Step 3: Generation of Loading and Combinations*

Loads followed ASCE 7-10 [16]: self-weight plus 0.25 kN/m² for roofing and purlins on the top chords, roof live load 0.58 kN/m², and a 1.334 kN maintenance load at the worst node. Wind on the open monoslope roof uses the net pressure coefficients C_N of the free-roof provisions:

$$q_h = 0.613 K_h K_{zt} K_d V^2 = 1.279 \text{ kN/m}^2, p = q_h G C_N \quad (2)$$

With $G = 0.85$ and $C_N = +1.2/+0.3$ (pattern A, downward) and $-1.1/-0.1$ (pattern B, uplift). Seismic load uses the equivalent lateral force procedure, and a uniform temperature change of ± 26 °C completes the set. The engineer approved each derivation before Claude converted pressures to joint loads; at gate G2 the resultant of every case in the file was compared with a hand total. The combinations came from one short prompt (Table 3) and were generated from a factor matrix, which removes the commonest error here, a wrong factor on the wrong case.

$$C_s = \frac{S_{DS}}{R/I_e} = \frac{0.464}{1.25/1.0} = 0.371, V = C_s W = 0.371 \times 215.0 = 79.8 \text{ kN} \quad (3)$$

Table 3 Load Cases, Combinations and the Combination Prompt.

Item	Content
Primary cases (15)	E: 1–4 ($\pm X$, $\pm Z$); W pattern A: 5–8; D: 9 (232.4 kN); Lr: 10 (137.9 kN); T: 11; W pattern B: 13–16
Prompt	“Create the ASD and LRFD combinations of ASCE 7-10 Sections 2.4.1 and 2.3.2 for these cases; one wind or seismic direction per combination; ASD from 100, LRFD from 200.”
Output (extract)	LOAD COMB 112 COMB - 1 D + 0.75 LR + 0.45 W5 $\rightarrow$ 9 1 10 0.75 5 0.45; LOAD COMB 207 COMB - 1.2 D + 0.5 LR + 1 W5 $\rightarrow$ 9 1.2 10 0.5 5 1
Result	43 ASD (100–143) and 37 LRFD (200–236) combinations, 80 in total

➤ Step 4: Structural Analysis

The structure is analysed by the direct stiffness method, with chords, struts and columns as 3-D frame elements, webs and head branches as pin-ended truss elements, and fixed bases:

$$K u = P, \quad K = \sum_e T_e^T k_e T_e \quad (4)$$

Where k_e is the local element stiffness and T_e the transformation matrix. Claude wrote the scripts that extract forces, reactions and displacements for all 80 combinations, check equilibrium, compare deflection with span/240 and drift with $h/150$, and take each member's utilisation as the largest axial, interaction ($H1-1a/b$) or shear ratio over all combinations. The results reported here come from a direct-stiffness implementation written with Claude to the same model definition as the generated STAAD.Pro file; this second solver closed gate G3 with an equilibrium error below 10^{-10} kN. Running the .std file in STAAD.Pro is the confirming step, with tolerances fixed in advance: 0.1 % on total reactions, 1 % on deflection, 2 % on the first frequency and ± 0.01 on the governing ratio.

➤ Step 5: Structural Design

- **Steel members.** Members were checked to AISC 360-16 [17] in both ASD and LRFD. Compression uses Chapter E, and combined force uses Equations H1-1a and H1-1b, with $K = 1.0$ for the lattice and $K = 2.0$ for the cantilevering supports. In conventional terms, the struts and columns act as columns, the longitudinal chords as rafters, the transverse chords as purlins and the webs as bracing.

$$F_{cr} = [0.658^{F_y/F_e}] F_y \text{ for } \frac{KL}{r} \leq 4.71 \sqrt{\frac{E}{F_y}}, F_e = \frac{\pi^2 E}{(KL/r)^2} \quad (5)$$

- **Foundations.** From a prompt naming the reactions to import, ACI 318-14 [18], $f'_c = 35$ MPa and the sizing rules, Claude built a workbook with live formulas. In every ASD combination the footing must stay in full contact and have safety factors of at least 1.5:

$$e = \frac{M}{P+W_f} \leq \frac{L}{6}, FS_{OT} = \frac{M_R}{M_O} \geq 1.5, FS_{sl} = \frac{\mu(P+W_f)}{H} \geq 1.5 \quad (6)$$

- **Base plates and anchor bolts.** Base plates follow the large-moment method of AISC Design Guide 1 [19], with bearing stress $f_p = \phi 0.85 f'_c \sqrt{A_2/A_1} = 27.3$ MPa. Anchors follow ACI 318-14 Chapter 17:

$$\phi N_{sa} = 0.75 A_{se} f_{uta}, N_b = k_c \lambda_a \sqrt{f'_c} h_{ef}^{1.5}, t_{req} = \sqrt{\frac{4 M_{pl}}{\phi F_y}} \quad (7)$$

- **Critical connection.** The most heavily loaded truss node was checked along its load path: tube, tube-to-cone fillet weld, 20 mm cone end plate and M24 class 10.9 node bolt. Each design was delivered as a workbook with editable inputs and imported analysis values, so the engineer could check any cell and test alternatives.

IV. CASE STUDY

The representative steel structure case study is an industrial canopy structure at a confidential industrial facility (Fig. 2, Table 4). The roof is a square-on-square offset double-layer space truss that slopes 2.66 % and tapers from 1.964 m deep at the high end to 1.012 m at the low end. Two inclined struts at about 58° to the horizontal provide longitudinal stiffness; two fixed-base columns at the low end resist transverse wind and seismic forces. The steel weighs 17.6 t (74 kg/m²) in only three tube sizes, as is usual for bolted ball-node systems.

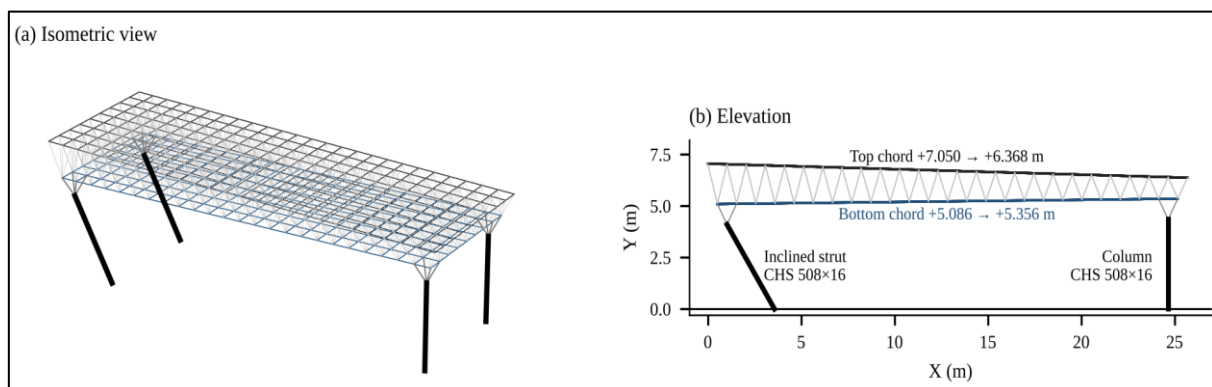


Fig 2 Representative Steel Structure Case Study: (a) Isometric View of the Analysis Model; (b) Elevation Showing the Sloping Double-Layer Roof, Inclined Struts and Vertical Columns.

Table 4 Summary of the Case-Study Structure.

Item	Description
Structural system	Free-standing steel canopy; double-layer space-truss roof on four tubular supports
Columns	2 inclined struts (4.82 m) and 2 vertical columns (4.35 m), CHS 508 × 16, fixed bases
Roof beams	Top and bottom chords, 900 members, pipe 73.0 × 5.16 mm, 1.027 m modules
Secondary members and bracing	900 web diagonals and 16 column-head branches, pipe 48.3 × 3.68 mm, pin-ended; struts brace the roof longitudinally
Analysis	STAAD.Pro (generated input; 493 joints, 1,820 members); independent direct-stiffness solver for verification
Design standards	ASCE 7-10; AISC 360-16; AISC Design Guide 1; ACI 318-14 (incl. Ch. 17)
Materials	Tubes F_y 250 MPa; plates ASTM A36; anchors ASTM F1554 Gr. 55; bolts class 10.9; E70XX welds; concrete 35 MPa

V. RESULTS AND DISCUSSION

➤ Analysis and Design Results

All 1,820 members pass in both formats (Table 5). The highest ratio, 0.743 (ASD), is in a column-head branch buckling in compression; 1,415 members are below 0.25 and none exceeds 0.75, which is typical of a space truss built

from a few standard sections. The $D + L_r$ deflection is 25.5 mm (span/928) against 98.5 mm, drift is 11.0 mm (h/372) against 27.3 mm, and the first frequency is 3.56 Hz, so strength governs rather than stiffness. The largest action on the substructure is the strut base moment, 306.5 kN·m (ASD) and 428.9 kN·m (LRFD), which drives the left-hand foundation design (Table 6).

Table 5 Governing Utilisation of the Steel Members (AISC 360-16).

Member group (role)	Section (mm)	Governing member	ASD ratio	LRFD ratio	Governing mode
Column-head branches (bracing)	48.3 × 3.68	1808	0.743	0.705	Compression, $KL/r = 78$
Web diagonals (bracing)	48.3 × 3.68	1708	0.673	0.638	Compression, $KL/r = 128$
Bottom chords, longitudinal (rafters)	73.0 × 5.16	498	0.580	0.538	Tension
Inclined struts (columns)	508 × 16	1818	0.547	0.508	Flexure, Eq. H1-1b
Top chords, longitudinal (rafters)	73.0 × 5.16	40	0.535	0.498	Compression
Transverse chords (purlins)	73.0 × 5.16	471	0.247	0.232	Compression
Vertical columns (columns)	508 × 16	1820	0.150	0.138	Flexure

Table 6 Foundation, Base Plate, Anchor Bolt and Critical Connection Design (Governing Values from the Design Workbooks).

Element	Strut base (left)	Column base (right)	Governing check
Footing $L \times B \times h$	5.0 × 2.5 × 0.6 m; T16 @ 175	3.2 × 3.1 × 0.6 m; T16 @ 175	Flexure 0.94; $e/kern$ 0.98 / 0.99
Stability	FS_{OT} 3.07; FS_{sl} 5.64	FS_{OT} 3.02; FS_{sl} 4.15	Full contact governs size
Base plate (A36)	850 × 850 × 55; t_{req} 53.3 mm	850 × 850 × 30; t_{req} 28.2 mm	0.968 / 0.940
Anchors, 8 No. F1554	M30, h_{ef} 700 mm; $T = 494$ kN	M24, h_{ef} 700 mm; $T = 113$ kN	Breakout, hairpins (left) 0.80
Tube-to-plate weld	Complete-joint-penetration	6 mm fillet	0.62 / 0.63
Critical truss node	Bottom chord, $T = 129.3$ kN: bolt 0.63, weld 0.86	–	End plate (20 mm) 0.942

At the strut base, Claude ran the ACI 318-14 Chapter 17 checks, found that concrete breakout ($\phi N_{cbg} = 195$ kN) could not resist the 494 kN anchor tension, and sized hairpin reinforcement (1,568 mm² required; 4 No. 25 bars provided). It also set out the alternatives of a stiffened base instead of a 55 mm plate, and a complete-joint-penetration weld where a fillet would need a 14.5 mm leg on a 16 mm wall. Choosing between such options is a fabrication and cost judgement that stays with the engineer.

➤ Model Generation Efficiency

Table 7 compares the effort for the same deliverable: a verified model, member and substructure designs, four design workbooks and a calculation report. The conventional hours are the author's estimate from comparable projects and the Claude-assisted hours an estimate of the effort spent; no formal time log was kept, so the figures indicate magnitude rather than a measured productivity factor.

Table 7 Estimated Effort, Conventional and Claude-Assisted Workflow (Single Case, Author's Estimates).

Task	Traditional (h)	Claude-assisted (h)	Change
Geometry modelling (joints, members, properties, supports)	36	8	−78 %
Load definition (15 cases, 80 combinations)	28	10	−64 %
Analysis and design (members, substructure, connection)	40	20	−50 %
Reporting (calculation report, design workbooks)	80	24	−70 %
Checking and verification	24	30	+25 %
Total	208	92	−56 %

Geometry modelling and reporting, the tasks that produce large volumes of structured data or text, account for 84 of the 116 hours saved; analysis and design saved less because judgement takes the same time either way. Checking grew, to a third of the Claude-assisted effort against about an eighth of the conventional effort. If every saving were only half as large, the overall reduction would still be 28 %.

➤ *Reduction of Manual Errors*

Because no joint, member or combination line was typed, transcription errors in the model were eliminated. The LLM introduced a different kind of error, and seven were logged: two arithmetic slips (a compact-section limit written as 574 instead of 57.4, and a buckling table using a wrong member length), two report sentences that contradicted the computed numbers, and three inconsistencies between stages (0.03 mm length differences that would have created extra fabrication marks, a 700 mm anchor embedment exceeding the 650 mm pedestal, and anchor nuts fouling the strut footprint). The gates caught the first five; the last two surfaced in shop detailing and were recorded as hold points. None looked wrong at a glance; output must be compared with an independent reference, not simply read [15].

➤ *Productivity, Documentation and Verification*

The larger gain is the low cost of change: the generator is parametric, so a new geometry or load means editing one parameter and re-running the gates, and the report tables update from the same data. Documentation was more consistent because no table was retyped, and verification became more thorough (all 15 cases in equilibrium, five hand checks, a second solver, eight sensitivity runs) because generated output is cheap, not because the LLM is more accurate. Four minimum verification requirements follow: count, extent and connectivity checks against independently predicted numbers; load totals confirmed by hand; equilibrium and governing members checked by a calculation the LLM did not produce; and review of every report table against its source by the engineer of record.

VI. LIMITATIONS

Human verification is required at every step, since the LLM states wrong values as confidently as right ones, and design responsibility remains with the licensed engineer who sets the design basis, accepts each gate and signs the calculation. Prompt quality affects the result; vague prompts gave plausible but incomplete models, and the best prompts read like short specifications with inputs, units, numbering rules and the checks to print. Code compliance must be checked independently, as the model may mix editions or

clauses. In this study the confirming STAAD.Pro analysis model, site information, and geotechnical input parameters have been verified based on the approved Geotechnical Investigation Report, the evidence comes from one structure, the times are estimates, and the author was both designer and evaluator.

VII. FUTURE APPLICATIONS

Smart engineering assistants linked to the solver's programming interface would remove the file exchange and run the gates automatically, and AI-generated STAAD.Pro models and calculation packages that trace each value to its source would make review faster. BIM integration would let geometry flow from the architectural model into analysis without re-entry [1], and digital twins could keep the design model synchronised with fabrication and monitoring data [20]. Autonomous engineering workflows are within reach for routine structures, but should develop as supervised automation with verification gates, not as a replacement for the engineer.

VIII. CONCLUSION

Claude successfully assisted a complete structural analysis and design workflow driven by natural-language prompts: on a 1,820-member industrial canopy it interpreted the input documents, generated the STAAD.Pro geometry, properties, supports, 15 load cases and 80 combinations, and supported the code design of members, foundations, base plates, anchors and the critical connection, all satisfying AISC 360-16 and ACI 318-14. Estimated effort fell by more than half, from about 208 h to 92 h, with the largest gains in STAAD.Pro model generation and reporting. Errors moved from typing to plausible slips in arithmetic, wording and coordination between stages; that shift is the main lesson for practice. LLMs can transform how structural engineers work by taking over the mechanical production of models and documents, provided the engineer keeps control of the design basis and verifies each output independently. Within that discipline, LLM-assisted STAAD.Pro workflows are ready for careful use on real projects.

➤ *Declaration of AI Use*

Claude (Anthropic) was the subject of this study and was also used to help draft and edit the manuscript. The author reviewed all content and takes full responsibility for it. [Author to state the Claude model version and confirm this statement.]

REFERENCES

- [1]. R. Sacks, C. Eastman, G. Lee, and P. Teicholz, *BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers*, 3rd ed. Hoboken, NJ, USA: Wiley, 2018.
- [2]. Y. Pan and L. Zhang, "Roles of artificial intelligence in construction engineering and management: A critical review and future trends," *Automation in Construction*, vol. 122, Art. no. 103517, 2021.
- [3]. H. Adeli, "Neural networks in civil engineering: 1989–2000," *Computer-Aided Civil and Infrastructure Engineering*, vol. 16, no. 2, pp. 126–142, 2001.
- [4]. H. Salehi and R. Burgueño, "Emerging artificial intelligence methods in structural engineering," *Engineering Structures*, vol. 171, pp. 170–189, 2018.
- [5]. H.-T. Thai, "Machine learning for structural engineering: A state-of-the-art review," *Structures*, vol. 38, pp. 448–491, 2022.
- [6]. C. V. Camp and B. J. Bichon, "Design of space trusses using ant colony optimization," *Journal of Structural Engineering*, vol. 130, no. 5, pp. 741–751, 2004.
- [7]. A. Kaveh, *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*. Cham, Switzerland: Springer, 2017.
- [8]. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.
- [9]. T. B. Brown *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 1877–1901.
- [10]. S. Noy and W. Zhang, "Experimental evidence on the productivity effects of generative artificial intelligence," *Science*, vol. 381, no. 6654, pp. 187–192, 2023.
- [11]. H. Liang, M. Talebi Kalaleh, and Q. Mei, "Integrating large language models for automated structural analysis," arXiv:2504.09754, 2025.
- [12]. J. Liu, Z. Geng, R. Cao, L. Cheng, P. Bocchini, and M. Cheng, "A large language model-empowered agent for reliable and robust structural analysis," arXiv:2507.02938, 2025.
- [13]. H. Sun, H. V. Burton, and H. Huang, "Machine learning applications for building structural design and performance assessment: State-of-the-art review," *Journal of Building Engineering*, vol. 33, Art. no. 101816, 2021.
- [14]. J. Zheng and M. Fischer, "BIM-GPT: A prompt-based virtual assistant framework for BIM information retrieval," arXiv:2304.09333, 2023.
- [15]. R. Parasuraman and D. H. Manzey, "Complacency and bias in human use of automation: An attentional integration," *Human Factors*, vol. 52, no. 3, pp. 381–410, 2010.
- [16]. American Society of Civil Engineers, *Minimum Design Loads for Buildings and Other Structures*, ASCE/SEI 7-10. Reston, VA, USA: ASCE, 2010.
- [17]. American Institute of Steel Construction, *Specification for Structural Steel Buildings*, ANSI/AISC 360-16. Chicago, IL, USA: AISC, 2016.
- [18]. ACI Committee 318, *Building Code Requirements for Structural Concrete (ACI 318-14) and Commentary (ACI 318R-14)*. Farmington Hills, MI, USA: American Concrete Institute, 2014.
- [19]. J. M. Fisher and L. A. Kloiber, *Base Plate and Anchor Rod Design*, AISC Design Guide 1, 2nd ed. Chicago, IL, USA: American Institute of Steel Construction, 2006.
- [20]. C. Boje, A. Guerriero, S. Kubicki, and Y. Rezgui, "Towards a semantic construction digital twin: Directions for future research," *Automation in Construction*, vol. 114, Art. no. 103179, 2020.