

Article

A Neuro-Symbolic Framework for Ensuring Deterministic Reliability in AI-Assisted Structural Engineering: The SYNAPSE Architecture

Adriano Castagnone¹ and Giuseppe Nitti^{1,2,*} ¹ S.T.A. DATA Srl., Via Saluzzo, 71-10126 Torino, Italy; castagnone@stadata.com² Department of Structural, Geotechnical and Building Engineering, Politecnico di Torino, Corso Duca degli Abruzzi, 24-10129 Torino, Italy

* Correspondence: giuseppe.nitti@polito.it

Abstract

This paper addresses the opportunities and risks of integrating Large Language Models (LLMs) into structural engineering. Exclusive reliance on LLMs is inadequate in this field, because their probabilistic nature can lead to hallucinations and inaccuracies that are unacceptable in safety-critical domains which require rigorous calculations. To resolve this dilemma, we propose adopting Neuro-Symbolic Artificial Intelligence (NSAI), a hybrid approach that balances neural intuition with symbolic rigor. The NSAI architecture employs an intelligent query system to enrich user requests and delegate critical operations to deterministic external algorithms. This system is designed to enhance reliability and support regulatory compliance, as exemplified by the 3Muri chatbot case study, an NSAI (gemini-2.5-flash)-based intelligent assistant for structural analysis software. We developed 3Muri chatbot implementing AI processes. Our experimental results, based on over 200 questions submitted to the chatbot, show that this hybrid approach achieves 94% accuracy while keeping response times below 2 s. These results validate the feasibility of deploying AI systems in safety-critical engineering domains.

Keywords: neuro-symbolic AI; large language models; LLM; structural engineering; hybrid architecture; safety-critical systems; explainable AI

1. Introduction

The rapid advancement of Artificial Intelligence (AI) technologies has transformed numerous industrial sectors, from healthcare to finance, from manufacturing to transportation. Among these technological breakthroughs, Large Language Models (LLMs) have emerged as particularly transformative tools, demonstrating unprecedented capabilities in natural language understanding, generation, and reasoning [1,2]. These models, trained on vast corpora of text data, have shown remarkable abilities to engage in human-like conversations, answer complex questions, and even assist in creative and analytical tasks.

The structural engineering field, which inherently demands precision and unequivocal results, has naturally recognized the potential of these powerful AI systems. Engineers and researchers have envisioned applications ranging from automated code compliance checking to intelligent design assistants capable of suggesting optimal structural configurations. The promise of LLMs lies in their ability to democratize access to complex engineering knowledge, potentially lowering the barrier to expert-level reasoning and practical applica-



Academic Editors: Phuong Hoang
Dat Nguyen, Minsoo Baek and Md
Nazmus Sakib

Received: 9 December 2025

Revised: 22 January 2026

Accepted: 25 January 2026

Published: 28 January 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

tion. Recent comprehensive reviews have documented these emerging applications and highlighted the transformative potential of LLMs for structural analysis and design [3,4].

However, the adoption of LLMs in structural engineering faces an insurmountable obstacle that distinguishes this domain from many others: the hallucination problem. This phenomenon refers to the tendency of LLMs to generate outputs that appear plausible and confident but are factually incorrect or entirely fabricated [5,6]. While such errors might be tolerable or easily correctable in casual conversation or creative writing, they pose an existential threat when applied to structural calculations where human lives depend on the accuracy of every computation.

The structural engineering sector operates within a uniquely constrained context that fundamentally differentiates it from other application domains of artificial intelligence. Understanding these constraints is essential to appreciating why conventional AI approaches are insufficient and why a specialized hybrid methodology becomes necessary. The following characteristics define the operational environment:

Legal responsibility and liability: Every structural calculation carries legal weight. Engineers who sign off on designs assume personal and professional liability for the safety of structures that may stand for decades and shelter thousands of people. This legal framework demands that every calculation must be verifiable, traceable, and defensible in potential litigation. The probabilistic nature of LLM outputs, where the same query might produce different results, fundamentally conflicts with this requirement for reproducibility and accountability [7].

Stringent regulatory framework: Structural design is governed by comprehensive regulatory codes that prescribe specific methodologies, safety factors, and verification procedures. In Europe, the Eurocodes [8,9] provide a harmonized approach to structural design, while national annexes and local regulations (such as Italy's NTC-Norme Tecniche per le Costruzioni [10]) add additional requirements. Compliance with these regulations is not optional; it is a legal prerequisite for construction approval. Any AI system operating in this domain must demonstrate not merely approximate correctness but exact adherence to prescribed calculation procedures.

Safety-critical nature: Structural failures can result in catastrophic consequences, including loss of life, severe injuries, and extensive property damage. Historical disasters such as the Ronan Point collapse (1968), the Hyatt Regency walkway collapse (1981), and more recent bridge failures serve as stark reminders of the consequences of engineering errors. This safety-critical nature demands minimal tolerance for computational errors, a standard that probabilistic AI systems inherently struggle to meet [7,11]. Research on AI safety has systematically documented how hallucinations in LLMs represent an intrinsic structural consequence of their probabilistic architecture, raising particular concerns for deployment in high-stakes domains where accuracy is paramount [5,12].

Increasing project complexity: Modern structural engineering projects exhibit unprecedented complexity. Buildings incorporate mixed materials (concrete, steel, timber, masonry), non-standard geometries, and must satisfy increasingly stringent performance requirements including seismic resilience, energy efficiency, and sustainability targets. This complexity creates a genuine need for intelligent support systems that can assist engineers in navigating vast design spaces while supporting regulatory compliance [3,4].

Furthermore, recent studies emphasize how AI-based methods can address these challenges through improved modeling, analysis, and design optimization, while also acknowledging persistent concerns regarding reliability and interpretability in safety-critical contexts [13,14]. In the domain of structural optimization, iterative algorithms have been traditionally employed to determine optimal geometrical characteristics of complex structural systems, such as diagrid configurations coupled with shear walls

in tall buildings [15,16]. Such optimization procedures, which require multiple iterative steps to converge to an optimal solution, represent typical engineering tasks where AI-based approaches could potentially replace or significantly accelerate conventional computational methods.

These constraints generate a fundamental paradox at the heart of AI adoption in structural engineering. On one hand, there exists a genuine and pressing need for AI-powered innovation to help engineers manage increasingly complex projects, navigate extensive regulatory requirements, and optimize designs across multiple objectives. On the other hand, the probabilistic nature of current AI systems, particularly LLMs, makes them fundamentally unsuitable for direct application to calculations where certainty is required [4,17] as shown in Figure 1.

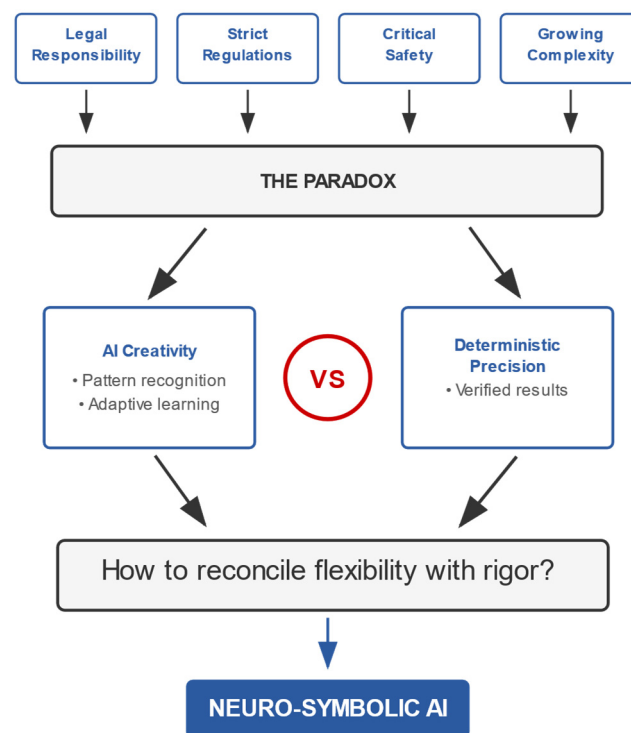


Figure 1. Stages and procedures that require absolute rigor.

The existing literature has attempted to address this tension through various approaches. Traditional expert systems provided rule-based reasoning but lacked adaptability and struggled with knowledge acquisition [18]. Machine learning methods, including neural networks, have been successfully applied to structural health monitoring, damage detection, and material property prediction [3,4], yet these approaches typically serve as supplements to, rather than replacements for, traditional engineering calculations due to their black-box nature. More recent deep learning applications have demonstrated impressive capabilities in capturing complex nonlinear relationships [4,19], but the fundamental challenge of ensuring deterministic reliability in safety-critical contexts remains unresolved. Recent surveys on AI trustworthiness have identified this gap explicitly, noting that purely data-driven approaches lack the transparency and verifiability required for critical applications [20,21]. What is missing from the current literature is a systematic framework that can combine the adaptive learning capabilities of neural AI with the rigorous, verifiable reasoning of symbolic systems specifically tailored to the constraints of structural engineering.

The equation “LLMs + Engineering = Risk” encapsulates this dilemma. The intrinsic logic of LLMs is probabilistic: they generate outputs based on statistical patterns learned

from training data, not through deterministic logical reasoning. This contrasts sharply with the deterministic nature required by structural engineering, where $2 + 2$ must always equal 4, and a beam's capacity must be calculated identically regardless of how many times the computation is performed [1,19] (Figure 2).

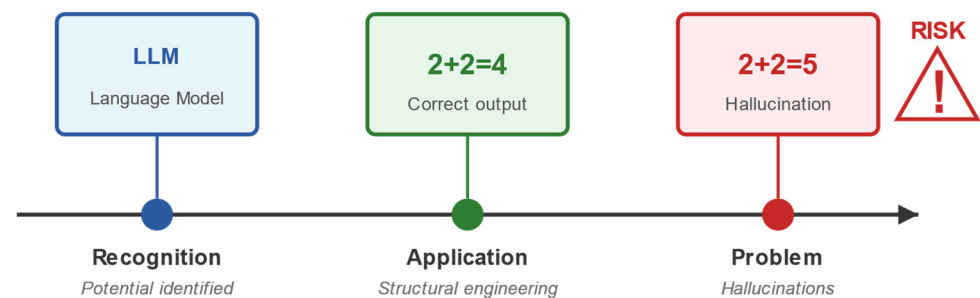


Figure 2. LLM application in structural engineering: from potential recognition to the problem of hallucinations.

To resolve this dilemma, a new paradigm is required—one that can harness the creative and interpretive capabilities of neural AI while preserving the deterministic rigor essential to engineering. It should be noted that achieving trustworthiness in AI systems is not limited to a single approach. Formal methods and verified AI techniques offer alternative pathways to ensure correctness through mathematical proofs and specification languages [22]. Additionally, hybrid explainability approaches combining logical reasoning with graph-based representations have demonstrated the feasibility of achieving both transparency and verification in language processing tasks [23]. Nevertheless, for the specific domain of structural engineering, where natural language interaction must be coupled with deterministic calculations, we propose that Neuro-Symbolic Artificial Intelligence (NSAI) [24,25] offers a particularly suitable framework. NSAI represents a hybrid approach that integrates the pattern recognition and adaptive learning strengths of neural networks with logical reasoning, rule enforcement, and knowledge representation capabilities of symbolic AI systems. By combining these complementary paradigms, NSAI enables AI systems to learn from data while adhering to explicit logical constraints and domain rules, thereby achieving both flexibility and verifiability [20,26]. In the context of structural engineering, this integration is particularly valuable: neural components can handle natural language understanding and user interaction, while symbolic components ensure that all calculations strictly follow prescribed regulatory procedures and engineering codes. This architectural separation allows NSAI to address the hallucination problem directly by delegating safety-critical computations to deterministic symbolic engines that are designed to produce reproducible, traceable results.

The primary objective of this research is to demonstrate how the Neuro-Symbolic AI paradigm can be effectively applied to structural engineering to achieve deterministic reliability while preserving the benefits of modern AI capabilities.

This paper makes the following contributions to the field of AI-assisted structural engineering:

1. A comprehensive analysis of why pure LLM approaches are unsuitable for safety-critical structural engineering applications, grounded in the specific constraints of the domain.
2. A detailed presentation of the Neuro-Symbolic AI paradigm as the appropriate solution, including its theoretical foundations and practical architecture.
3. The introduction of the SYNAPSE architecture (Symbolic Neural Architecture for Predictive Structural Engineering), demonstrating how the hybrid approach can be implemented in practice.

4. A real-world case study of the 3Muri Chatbot, providing empirical validation of the proposed approach with quantitative performance metrics.

The remainder of this paper is organized as follows: Section 2 reviews the state-of-the-art in AI for structural engineering and neuro-symbolic systems. Section 3 presents the theoretical foundations of the neuro-symbolic bridge. Section 4 details the SYNAPSE architecture. Section 5 discusses implementation challenges, introduces the 3Muri chatbot case study, and presents the experimental results. Section 6 provides a critical discussion of the results, limitations, and lessons learned. Section 7 concludes with future perspectives.

2. Literature Background

The application of artificial intelligence techniques to structural engineering problems has a rich history spanning several decades. Early efforts focused on expert systems—rule-based programs that encoded the knowledge of experienced engineers into logical rules that could be applied to specific design problems. These systems demonstrated the feasibility of computational approaches to engineering decision-making but were limited by their brittleness and the difficulty of knowledge acquisition [18].

The advent of machine learning, particularly neural networks, opened new possibilities for pattern recognition in structural data. Researchers successfully applied neural networks to problems including structural damage detection, material property prediction, and load forecasting. Thai [3] provides a comprehensive state-of-the-art review of machine learning applications in structural engineering, documenting successful applications while also noting the persistent challenge of ensuring reliability in safety-critical contexts.

More recently, deep learning approaches have been applied to increasingly complex structural engineering tasks. Solhmirzaei et al. [19] developed machine learning frameworks for predicting failure modes and shear capacity in ultra-high-performance concrete beams, demonstrating that neural approaches can capture complex nonlinear relationships that elude traditional analytical methods. However, such applications typically require extensive validation and are used to supplement rather than replace traditional engineering calculations.

Beyond survey literature, numerous experimental case studies have demonstrated AI's practical capabilities in structural engineering. Zhang and Sun [27] proposed a physics-guided machine learning approach for structural damage identification, validating their framework through both numerical simulations of a steel pedestrian bridge and experimental studies on a three-story building model. Cha et al. [28] developed autonomous structural visual inspection using region-based deep learning, successfully detecting multiple damage types in real infrastructure. In seismic damage assessment, Gonzalez and Zapico [29] demonstrated that neural networks achieved 92% accuracy in identifying seismic damage across 50 buildings, significantly outperforming the 75% accuracy of conventional inspection techniques. More recently, experimental work by Hakim et al. [30] showed that ensemble neural networks combining predictions from multiple networks can produce superior detection accuracy compared to individual networks, validated through case studies on steel girder bridges. These experimental validations confirm the practical viability of AI-based approaches while simultaneously highlighting their limitations in providing the deterministic guarantees required for design calculations.

The emergence of Large Language Models has sparked renewed interest in AI-assisted engineering. Tapeh and Naser [4] conducted a scientometric review of AI trends in structural engineering, identifying both opportunities and challenges. Sun et al. [17] reviewed machine learning applications for building structural design, noting the tension between the power of black-box models and the need for interpretability in engineering contexts.

2.1. The Hallucination Problem in Large Language Models

The phenomenon of hallucination in LLMs has emerged as a critical research topic as these models have been deployed in increasingly high-stakes applications. Huang et al. [5] provides a comprehensive survey on hallucination in Large Language Models, establishing a taxonomy of hallucination types and analyzing their underlying causes. They identify several mechanisms that contribute to hallucinations, including training data noise, the autoregressive generation process, and the models' tendency to prioritize fluency over factual accuracy.

Ji et al. [31] survey hallucination specifically in natural language generation systems, noting that the problem is not unique to LLMs but is particularly acute in models trained on diverse, unverified internet text. They observe that larger models, while generally more capable, do not necessarily exhibit reduced hallucination rates and may in fact hallucinate more confidently.

Tonmoy et al. [6] provides a comprehensive survey of hallucination mitigation techniques, categorizing approaches into training-time interventions, inference-time strategies, and post hoc verification methods. While these techniques can reduce hallucination rates, none provide the level of correctness required in safety-critical applications. This fundamental limitation motivates the search for hybrid architectures that can constrain neural outputs within verified bounds.

2.2. Neuro-Symbolic Artificial Intelligence

Neuro-symbolic AI represents what Garcez and Lamb [24] term the “third wave” of artificial intelligence, following earlier waves dominated by symbolic AI (expert systems, logic programming) and by connectionist AI (neural networks, deep learning) [32]. This third wave seeks to combine the complementary strengths of both paradigms while mitigating their respective weaknesses.

Yu et al. [25] provides a comprehensive survey of neural-symbolic learning systems, documenting the various architectural approaches that have been proposed to integrate neural and symbolic components. These range from loosely coupled systems where neural and symbolic modules operate independently to tightly integrated approaches where symbolic knowledge is embedded directly into neural network architectures.

Hitzler and Sarker [33] edited a comprehensive volume on the state of the art in neuro-symbolic AI, bringing together contributions from leading researchers in the field. The volume documents both theoretical advances and practical applications, demonstrating the growing maturity of the field. Marra et al. [34] traces the evolution from statistical relational learning to modern neuro-symbolic approaches, providing historical context for current developments.

Gibaut et al. [35] propose a taxonomy for neuro-symbolic AI systems, distinguishing between systems based on nature and degree of integration between neural and symbolic components. This taxonomy provides a useful framework for positioning different architectural approaches and understanding their relative strengths and limitations. Shakarian et al. [36] presents a comprehensive treatment of neuro-symbolic reasoning and learning, with particular attention to applications requiring verifiable reasoning. In the context of structural engineering, NSAI can be functionally defined as a hybrid architecture where: (1) neural components handle natural language processing, intent recognition, and contextual understanding of engineering queries; while (2) symbolic components enforce deterministic calculations according to building codes, verify limit state conditions through formal logic, and support regulatory compliance through rule-based reasoning. This separation is critical because structural engineering calculations must satisfy specific formal requirements: code compliance verification (e.g., Eurocode checks for member capacity),

limit state calculations (ultimate and serviceability), and safety factor enforcement. These requirements demand deterministic outputs where identical inputs must always produce identical results—a property that purely neural approaches cannot reliably provide due to their probabilistic nature. Recent work on neuro-symbolic verification [37] has demonstrated that combining neural pattern recognition with symbolic constraint satisfaction can achieve formal correctness guarantees, suggesting a viable path for safety-critical engineering applications. The symbolic component effectively acts as a “guard rail” that constrains neural outputs within verified bounds, designed to produce deterministic, traceable, and legally defensible results for any engineering calculation delegated to external symbolic engines (such as finite element analysis software or code-checking algorithms).

To further clarify the positioning of SYNAPSE relative to alternative approaches for achieving AI trustworthiness, Table 1 compares neuro-symbolic AI with verified AI methods and hallucination mitigation techniques. The comparison spans key dimensions relevant to safety-critical engineering applications.

Table 1. Comparison of Approaches for Trustworthy AI in Safety-Critical Applications.

Approach	Determinism	Explainability	Safety-Critical Reliability	Key References
Pure LLMs	No	Low	Unsuitable	[1,2,5,6,12]
Hallucination Mitigation	Partial	Low	Insufficient	[5,6,12,31]
Traditional ML/DL	No	Low	Limited	[3,4,13,14,17]
XAI Methods	No	High	Moderate	[38]
Verified AI/ Formal Methods	Yes	High	Medium	[7,11,22,23]
Neuro-Symbolic AI (general)	Partial	Medium-High	High	[20,21,24–26,33–37]
SYNAPSE	Yes	High	High	This paper

2.3. Explainable AI for Safety-Critical Systems

The deployment of AI in safety-critical systems has driven substantial research into explainable AI (XAI). Saeed and Omlin [38] provide a systematic meta-survey of XAI challenges and opportunities, noting that explainability requirements vary significantly across application domains. In structural engineering, explainability is not merely desirable but legally required: engineers must be able to justify their design decisions.

Dwivedi et al. [39] present core ideas, techniques, and solutions for explainable AI, categorizing approaches by the type of explanation provided (local vs. global, model-agnostic vs. model-specific). They note that different stakeholders require different types of explanations: regulators may need formal verification of compliance, while practicing engineers may need intuitive explanations of design recommendations.

Within civil engineering specifically, XAI research has addressed domain-specific uncertainty and interpretability challenges. Naser [40] provides an engineer’s guide to explainable AI and interpretable machine learning in structural engineering, exploring how techniques such as SHAP (SHapley Additive exPlanations), LIME (Local Interpretable Model-agnostic Explanations), and feature importance analysis can be applied to bridge damage detection and structural capacity prediction. In geotechnical applications, Love et al. [41] employed causal discovery and XAI reasoning to interpret geotechnical risks in tunnel construction, demonstrating how explainability methods can make data-driven predictions more transparent for engineering decision-making. For seismic assessment, recent

work has developed Seismo-XAI, a web-based tool that delivers transparent, interpretable insights into building performance under seismic loads [42]. The tool maps XAI techniques specifically to structural engineering tasks: attention mechanisms in Vision Transformers identify critical damage patterns in concrete microstructures, while SHAP values explain the relative importance of input features (e.g., material properties, geometric configurations) in predicting structural responses. Furthermore, physics-informed neural networks with XAI components have been applied to quantify uncertainty in seismic performance assessment, providing both point predictions and prediction intervals that account for aleatory and epistemic uncertainties inherent in earthquake engineering [42,43].

Pérez-Cerrolaza et al. [7] survey AI for safety-critical systems specifically in industrial and transportation domains, analyzing both the opportunities and the unique challenges these domains present. They emphasize that safety-critical deployment requires not only high accuracy but also predictable behavior, graceful degradation under uncertainty, and comprehensive audit trails. Kaur et al. [11] review trustworthy AI, identifying the multiple dimensions (including reliability, safety, fairness, and explainability) that must be addressed for AI systems to be trusted in high-stakes applications.

Table 2 provides a systematic comparison of key AI applications in structural engineering, summarizing the methodological approaches, application domains, and identified limitations from the literature reviewed above.

In summary, the literature reveals a significant gap at the intersection of AI capabilities and structural engineering requirements. While machine learning and deep learning have demonstrated remarkable success in pattern recognition tasks such as damage detection, material property prediction, and structural health monitoring, these approaches remain fundamentally unsuitable for safety-critical design calculations due to their probabilistic nature and lack of deterministic guarantees. Existing methods address either the learning component (neural networks, deep learning) or the reasoning component (expert systems, rule-based approaches) but fail to integrate both in a manner that satisfies the stringent requirements of structural engineering: verifiable calculations, regulatory compliance, legal accountability, and minimal tolerance for computational errors. A key limitation of the current literature stems from the inherent opacity of neural network architectures, which prevents formal verification of their outputs [22]. Additionally, while hybrid explainability approaches combining logical reasoning with neural components have shown promise in natural language processing domains [23], their systematic application to structural engineering verification remains unexplored. The features leading to these limitations include: (1) the lack of provable correctness guarantees in purely data-driven approaches, (2) the absence of formal specification languages suitable for encoding building codes into verifiable constraints, and (3) insufficient integration between symbolic reasoning engines and modern deep learning architectures. Viable paths to overcome such limitations include adopting verified AI frameworks that provide mathematical guarantees of correctness [22], leveraging hybrid explainability techniques that combine graph-based and logic-based representations for transparent decision-making [23], and developing domain-specific formal languages for encoding regulatory requirements. The neuro-symbolic paradigm offers a promising solution to bridge this gap, but its application to structural engineering has been largely unexplored. Specifically, there is no established framework demonstrating how symbolic logic can enforce deterministic safety requirements such as code compliance verification and limit state calculations while leveraging neural capabilities for natural language understanding and user interaction.

Table 2. Comparison of AI Applications in Structural Engineering.

Study	AI Method	Application Domain	Determinism	Explainability	Limitation
Thai [3]	ML/DL Survey	Structural engineering	No	Low	Black-box nature
Tapeh & Naser [4]	AI/ML/DL Scientometrics	SE trends analysis	No	Low	Reliability concerns
Etim et al. [13]	ML Survey	Structural engineering	No	Low	Limited validation
Málaga-Chuquitaype [14]	ML Review	Structural design	No	Low	Interpretability gap
Sun et al. [17]	ML Review	Building design	No	Low	Performance assessment only
Zhang & Sun [27]	Physics-guided ML	Damage identification	Partial	Medium	Limited to SHM
Cha et al. [28]	Region-based DL	Visual inspection	No	Low	Detection only
Gonzalez & Zapico [29]	Neural Networks	Seismic Damage	No	Low	Modal data dependency
Hakim et al. [30]	Ensemble NN	Damage identification	No	Low	Requires extensive data
Solhmirzaei et al. [19]	ML Framework	UHPC beam capacity	No	Low	Requires validation
Naser [40]	XAI (SHAP, LIME)	Interpretable ML	No	High	Post hoc interpretation
Love et al. [41]	XAI Reasoning	Construction risks	No	Medium	Domain-specific
Ruggieri et al. [43]	XAI Methods	Civil engineering	No	High	Predictive only
SYNAPSE (This work)	Neuro-Symbolic AI	Civil engineering	Yes	High	None identified

The SYNAPSE architecture presented in this research addresses this gap by bridging the divide between conversational AI interfaces and verified structural engineering computations, ensuring that user queries receive both contextually relevant and technically accurate responses.

3. The Neuro-Symbolic Bridge: Theoretical Foundations

Neuro-symbolic AI (NSAI) represents a transformative paradigm that aims to serve as the “intelligent bridge between neural intuition and symbolic rigor” (Figure 3). Rather than viewing neural and symbolic approaches as competing alternatives, NSAI recognizes them as complementary capabilities that, when properly integrated, can achieve results impossible for either approach alone [24,25].

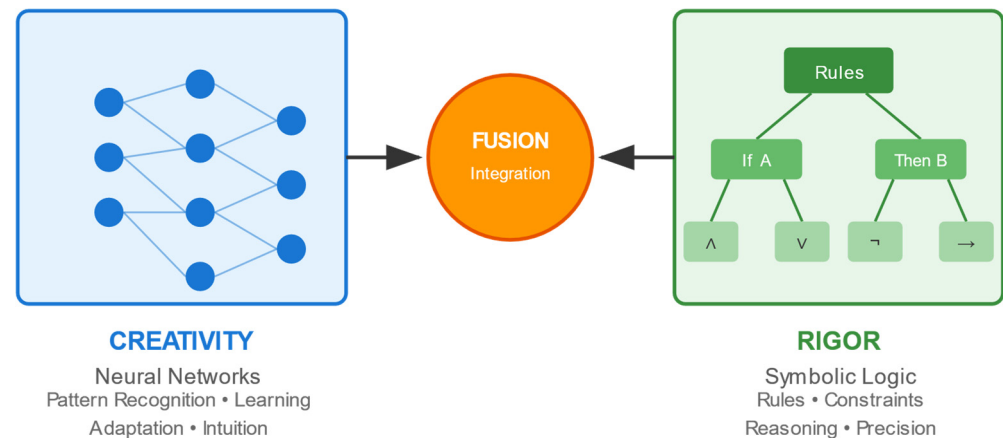


Figure 3. Neuro-Symbolic Artificial Intelligence (NSAI).

As illustrated in Figure 3, the NSAI paradigm is conceptualized as a bridge spanning two distinct computational domains. On the left side of the bridge lies the Neural AI domain, characterized by pattern recognition, learning from data, handling uncertainty, and flexible adaptation to new situations. This domain excels at tasks such as natural language understanding, image recognition, and identifying complex nonlinear relationships in data—capabilities that have made LLMs so powerful. On the right side lies the Symbolic AI domain, characterized by logical reasoning, rule-based inference, formal verification, and deterministic computation. This domain excels at tasks requiring mathematical precision, regulatory compliance checking, and producing traceable, reproducible results—capabilities essential for structural engineering calculations. The bridge itself represents the integration mechanism that allows information to flow between these domains: neural outputs are translated into symbolic representations that can be verified and constrained, while symbolic knowledge guides and constrains neural processing. In practical terms for structural engineering, this means that an engineer’s natural language query (processed by the neural component) can be translated into formal calculation requests (executed by the symbolic component), with the results verified against building codes and then communicated back in understandable language (again via the neural component). This bidirectional flow ensures both user-friendliness and computational rigor.

The fundamental insight underlying NSAI is that human intelligence itself operates through the integration of intuitive and analytical reasoning. Cognitive scientists have long recognized that human problem-solving involves both fast, intuitive pattern recognition (system 1 thinking) and slow, deliberate logical analysis (system 2 thinking). NSAI architectures seek to replicate this dual-process cognition computationally, with neural components providing intuition and symbolic components providing analysis.

In the context of structural engineering, this integration takes on particular significance. The neural component can handle the inherently fuzzy aspects of engineering practice: interpreting natural language queries, understanding hand-drawn sketches, recognizing patterns in complex datasets, and generating creative design suggestions. The symbolic component is designed to produce outputs that conform to the rigorous requirements of engineering calculation, verifiable logic, regulatory compliance, and deterministic reproducibility.

3.1. The Neural Component: Capabilities and Limitations

The neural component of an NSAI system, typically implemented using Large Language Models or other deep learning architectures, provides capabilities essential for creating usable engineering tools [44,45]. These capabilities include:

Natural language understanding: Neural models excel at interpreting human language in all its variability and ambiguity. Engineers can describe problems in their own words, using domain-specific terminology, informal descriptions, or even incomplete specifications. The neural component can parse these inputs and extract the underlying engineering intent, a task that would be extremely difficult to accomplish with rule-based systems alone.

Pattern recognition: Deep learning models can identify complex patterns in high-dimensional data, enabling capabilities such as recognizing structural elements in hand-drawn sketches, identifying relevant precedents from large document collections, or detecting anomalies in sensor data. These pattern recognition capabilities extend the reach of engineering tools to unstructured data sources that traditional software cannot process.

Flexible reasoning: Neural models can engage in approximate, analogical reasoning that mimics human intuition. They can suggest design approaches based on similarity to past projects, identify potential issues by analogy to known failure modes, or generate creative solutions by combining elements from disparate sources. This flexibility enables innovation and exploration that rigid rule-based systems cannot support.

Contextual adaptation: Neural models can adapt their behavior based on context, adjusting their outputs to match the level of detail requested, the expertise of the user, or the specific requirements of the project. This adaptability makes them effective interfaces between complex engineering systems and diverse user populations.

However, these capabilities come with fundamental limitations that make neural components unsuitable for direct application to safety-critical calculations [38]:

Probabilistic outputs: Neural models generate outputs probabilistically, meaning the same input can produce different outputs across invocations. This non-determinism is fundamentally incompatible with engineering requirements for reproducibility.

Hallucination risk: As discussed previously, neural models can generate plausible sounding but incorrect outputs with high confidence. No amount of training or fine-tuning can eliminate this risk entirely.

Opacity: Neural models are often described as “black boxes” because their decision-making processes are not directly interpretable. This opacity conflicts with engineering requirements for traceable, justifiable calculations.

Limited numerical precision: LLMs perform arithmetic through pattern matching rather than actual computation, leading to frequent errors in numerical calculations. This limitation alone disqualifies them from direct engineering calculation roles.

3.2. The Symbolic Component: Rigor and Interpretability

The symbolic component of an NSAI system provides the rigor and interpretability that neural components lack [19,20]. Symbolic AI operates on explicit representations of knowledge using formal logic, rules, and structured relationships; in the context of structural engineering, the symbolic component encompasses the parts shown in Figure 4.

Domain ontology: A formal representation of structural engineering concepts and their relationships. This ontology defines entities such as beams, columns, loads, materials, and connections, along with the properties and relationships that characterize them. The ontology provides a precise vocabulary for representing engineering problems and supports grounding all reasoning in well-defined concepts.

Verification protocols: Formal procedures for checking that designs meet specified requirements. These protocols encode the verification methods prescribed by regulatory codes, designed to perform compliance checking consistently and completely. The protocols also provide audit trails documenting exactly how compliance was verified.

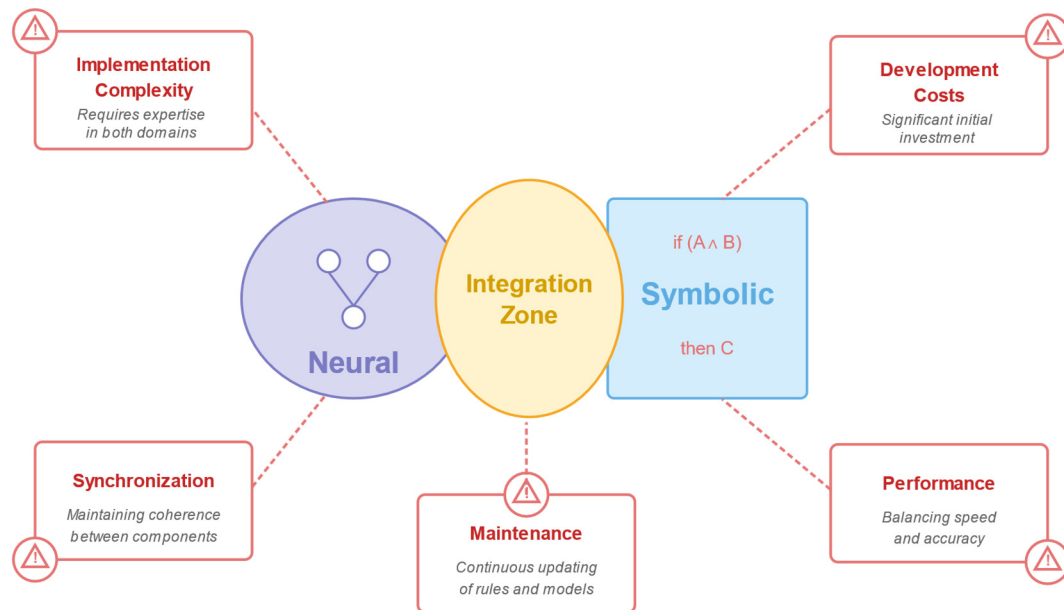


Figure 4. The complexities of integration.

Compliance rules: Explicit encodings of regulatory requirements from sources such as Eurocodes and national building codes. These rules specify the calculations that must be performed, the safety factors that must be applied, and the limits that must not be exceeded. By encoding these rules symbolically, the system is designed to produce outputs that comply with applicable regulations.

Deterministic algorithms: Mathematical algorithms for structural analysis that produce identical results for identical inputs. These algorithms implement the established methods of structural mechanics (finite element analysis, frame analysis, section capacity calculations) with numerical precision appropriate for engineering applications.

The symbolic component provides critical capabilities that complement the neural component [27]:

Deterministic precision: Symbolic calculations produce exactly reproducible results, satisfying the engineering requirement for verifiability.

Complete transparency: Every step of symbolic reasoning can be traced and justified, providing the explainability required for legal and regulatory compliance.

Strong compliance support: By encoding regulatory requirements explicitly, the symbolic component can guarantee that outputs conform to applicable codes.

However, symbolic systems also have limitations that motivate the hybrid approach:

Brittleness: Symbolic systems require inputs in precisely specified formats and cannot handle the variability and ambiguity of natural language or informal sketches.

Knowledge acquisition: Encoding expert knowledge into symbolic rules is labor-intensive and requires continuous maintenance as regulations and best practices evolve.

Limited generalization: Symbolic systems can only handle situations anticipated by their designers and encoded in their rules; they cannot generalize to novel situations.

3.3. The Integration: Achieving Synergy

The key insight of neuro-symbolic integration is that the strengths of each component precisely address the weaknesses of the other [38,39]. Neural components handle the messy, ambiguous front-end of engineering interaction—understanding what the user wants—while symbolic components handle the precise, verifiable back-end designed to deliver correct results. This integration creates systems that achieve results impossible for either approach alone:

Enhanced reliability: Every output undergoes formal verification by the symbolic component, ensuring correctness regardless of how the request was initially interpreted [35].

Full explainability: Every decision can be made through the symbolic reasoning chain, providing the justification required for engineering accountability [39].

Regulatory compliance: Direct integration with encoded regulatory requirements supports conformance of outputs to applicable codes and standards [8–10].

Error reduction: The dual verification provided by neuro-symbolic integration catches errors that might escape either component alone.

Natural interaction: Users can interact in natural language while receiving rigorously verified results.

Operational efficiency: Intelligent automation of routine tasks frees engineers to focus on higher-value activities.

4. SYNAPSE Architecture

The theoretical principles of neuro-symbolic integration must be realized in a concrete architecture that can be implemented and deployed. This section presents the SYNAPSE architecture (Symbolic Neural Architecture for Predictive Structural Engineering), which defines a precise workflow designed to delegate critical calculations to rigorous algorithms while AI serves as an intelligent interface.

The SYNAPSE architecture comprises four main components arranged in a sequential pipeline: the user interface layer, the intelligent query system (symbolic core), the LLM interface (neural interface), and the deterministic algorithm (computational core). Information flows through these components in a carefully orchestrated sequence designed to achieve both usability and rigor. The architecture is designed around a fundamental principle: the neural component is never allowed to perform safety-critical calculations directly. Instead, the neural component's role is limited to interpreting user intent and presenting results, while all actual engineering calculations are performed by verified deterministic algorithms. This separation of concerns is the key to achieving both the flexibility of AI and the reliability of traditional engineering software.

Figure 5 provides a comprehensive flowchart of the SYNAPSE architecture, illustrating the complete information flow from user input to final output. The diagram shows the four main components and their interconnections: (1) the User Interface Layer, which receives natural language queries, document uploads, and multimodal inputs from the structural engineer; (2) the Intelligent Query System (IQS), which parses, validates, and contextualizes the input while managing the engineering knowledge base; (3) the LLM Interface, which handles intent recognition, parameter extraction, and natural language generation; and (4) the Deterministic Calculation Engine, which executes verified numerical algorithms and produces traceable results. The arrows in Figure 5 indicate the sequential flow of information: user input enters through the interface layer, is processed by the IQS for validation and context enrichment, passes to the LLM for interpretation and structuring, and finally reaches the deterministic engine for calculation. Results flow back through the same pathway, with the LLM transforming numerical outputs into comprehensible explanations and the IQS verifying compliance before presentation to the user. This architecture is designed so that at no point does the probabilistic LLM component directly perform safety-critical calculations—all engineering computations are delegated to the deterministic engine, supporting reproducibility and regulatory compliance. The subsequent subsections detail each component of this architecture.

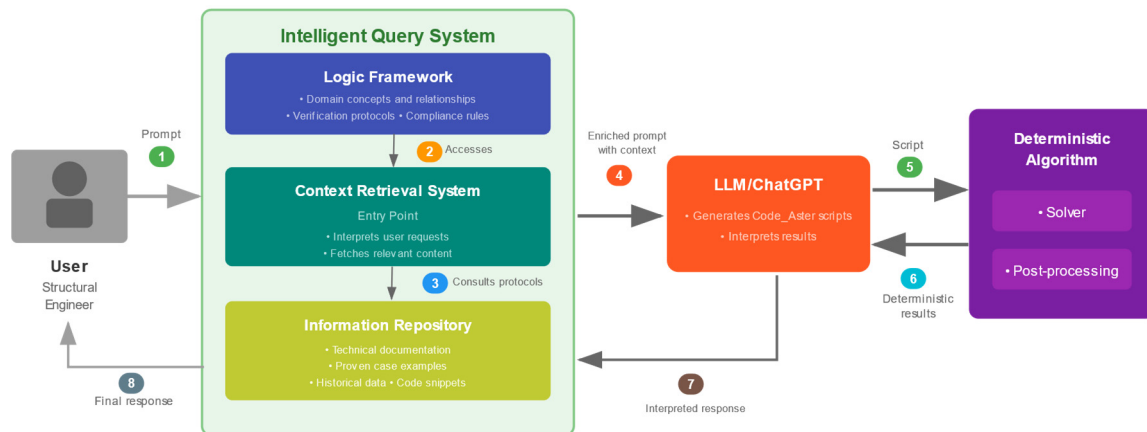


Figure 5. Connections between user, intelligent query system, LLM, and deterministic algorithms.

4.1. The User Input Layer

The interaction process begins with the user (the structural engineer) providing input in natural language. The system is designed to accept a wide variety of input formats, reflecting the diverse ways engineers naturally express their needs:

Textual queries: Engineers can type questions or requests in natural language, such as “verify this steel beam for the given loads” or “what is the maximum span for an HEA 200 section under these conditions?”

Hand-drawn sketches: The system can interpret hand-drawn sketches showing structural configurations, dimensions, and loading conditions (Figure 6). This capability is particularly valuable during preliminary design phases when formal CAD (Computer-Aided Design) drawings may not yet exist.

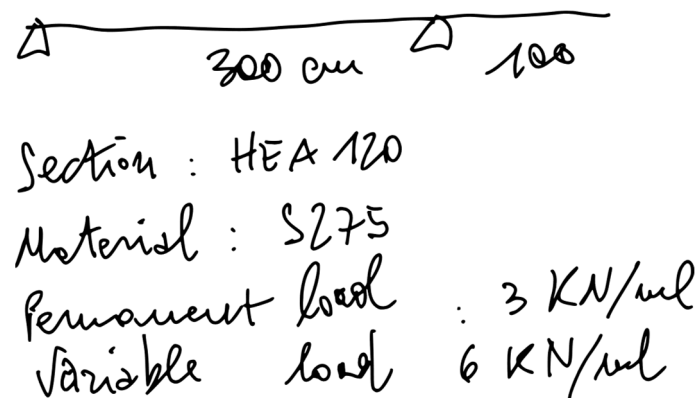


Figure 6. Hand-drawn sketches showing structural configurations, dimensions, and loading conditions of the steel-made beam.

Mixed media: Users can combine text and images, providing sketches with textual annotations or verbal descriptions accompanied by supporting diagrams.

The system’s ability to interpret informal inputs significantly lowers the barrier to use. Engineers do not need to learn specialized input formats or navigate complex menu structures; they can simply describe what they need in their own words and let the system handle the translation to formal specifications.

For example, a user might submit a hand-drawn sketch showing a continuous beam with two spans. Despite the informality of the input, the system’s recognition capabilities can extract the essential information: structural typology (continuous beam), geometry (span lengths $L_1 = 300$ cm, $L_2 = 100$ cm), section details (HEA 120 in S275 steel grade),

and loading (permanent load $g_k = 3 \text{ kN/m}$, variable load $q_k = 6 \text{ kN/m}$). This extracted information forms the basis for subsequent rigorous analysis.

4.2. The Intelligent Query System

The intelligent query system (IQS) constitutes the symbolic core of the SYNAPSE architecture (Figure 5). It serves as the critical bridge between informal user input and rigorous engineering calculation, designed to properly contextualize all requests and verify all outputs.

The IQS implements a unified architecture for intelligent query processing comprising three integrated layers that work together to transform user requests into verified engineering outputs:

Layer 1—Information repository: This foundational layer contains all verified resources necessary to support engineering calculations. The repository includes comprehensive technical documentation covering structural analysis methods, material properties, and design procedures. It also contains proven case examples demonstrating correct application of engineering methods to real problems, historical data from previous analyses that can inform current work, and validated code snippets implementing standard calculations. The repository is curated so that all information is accurate, current, and consistent with applicable regulations.

Layer 2—Logic framework: This layer houses the symbolic rigor of the system. It contains the domain ontology defining structural engineering concepts and relationships, verification protocols specifying how compliance must be checked, and compliance rules encoding the requirements of applicable codes (Eurocodes, NTC, etc.). The logic framework is designed to ground all reasoning in verified engineering knowledge and to produce outputs that conform to regulatory requirements.

Layer 3—Context retrieval system: This layer serves as the entry point to the IQS, receiving user requests and orchestrating the response. It interprets requests to identify the engineering task required, retrieves relevant information from the repository, consults the logic framework to determine applicable verification protocols, and assembles the enriched context that will guide subsequent processing. The Context retrieval system is designed to handle every request with full awareness of relevant requirements and constraints.

The three-layer architecture ensures comprehensive coverage of engineering knowledge while maintaining clear separation between information (what we know), logic (how we reason), and context (what is relevant now).

4.3. The Neural Interface

The LLM component serves as the neural interface of the system, providing natural language capabilities while operating within the constraints established by the symbolic core. Critically, the LLM does not operate autonomously but receives an enriched prompt that has been augmented by the intelligent query system.

The enriched prompt contains not only the original user request, but also additional context provided by the IQS:

Regulatory requirements: Specific code provisions applicable to the requested analysis, ensuring that the LLM's response accounts for all relevant regulations.

Calculation methods: Prescribed procedures for performing the required analysis, guiding the LLM to prepare data in the format expected by the deterministic algorithms.

Predefined templates: Structured formats for input data and output presentation, supporting consistency and completeness.

The LLM's role in architecture is carefully circumscribed. It is responsible for interpreting user intent from natural language input, preparing structured data for the deterministic

calculation routines, generating scripts (e.g., Code Aster scripts) that invoke the appropriate solvers, and interpreting and presenting results in user-friendly format. Crucially, the LLM is not responsible for performing any safety-critical calculations itself. This separation ensures that the LLM's probabilistic nature and potential for hallucination cannot compromise the reliability of engineering outputs.

4.4. The Deterministic Calculation Engine

The deterministic algorithm component is the computational heart of the SYNAPSE architecture, where actual structural analysis occurs. Unlike the LLM's probabilistic logic, calculation in this component is deterministic and therefore rigorously reproducible.

The SYNAPSE architecture integrates Code Aster as its primary finite element solver. Code Aster (Analyse des Structures et Thermomécanique pour des Études et des Recherches) was selected for several compelling reasons: (1) it is an open-source, extensively validated solver developed by Électricité de France (EDF) with over 30 years of industrial deployment in nuclear and civil engineering applications; (2) it provides comprehensive structural analysis capabilities including linear/nonlinear static analysis, modal analysis, seismic response spectrum analysis, and pushover analysis essential for masonry structures; (3) its Python (Version 3.12)-based command interface (`code_aster.py`) enables seamless integration with the LLM-generated scripts; and (4) its extensive verification and validation documentation satisfies the traceability requirements of safety-critical engineering applications. The solver operates through a command file syntax that the LLM is trained to generate, with each command file containing mesh definitions, material properties, boundary conditions, loading scenarios, and solution parameters structured according to Code Aster's validated templates.

The component performs two primary functions:

Solver function: The solver implements verified numerical methods for structural analysis. Depending on the problem type, this may include linear and nonlinear finite element analysis, frame analysis using stiffness methods, section capacity calculations according to code provisions, and stability analyses. The solver operates on the structured input prepared by the LLM, performing calculations with full numerical precision and complete reproducibility.

Post-processing function: After the solver completes its calculations, the post-processor extracts and organizes the results. This includes identifying critical values (maximum stresses, deflections, utilization ratios), comparing results against code limits, generating compliance reports, and preparing data for visualization. The post-processor is designed to present results in formats useful for engineering decision-making and regulatory documentation.

The verification protocols implement a multi-stage validation process for LLM-generated scripts before execution. Stage 1 (Syntactic Validation) performs automated parsing of the generated Code Aster command file to verify correct syntax, proper command sequencing, and valid keyword usage. Stage 2 (Semantic Validation) checks engineering logic consistency: material property values are verified against physically plausible ranges (e.g., Young's modulus for steel must be between 190–210 GPa, masonry compressive strength typically 1–20 MPa); boundary conditions are checked for static determinacy; load combinations are validated against applicable building codes (e.g., Eurocode load factors $\gamma_G = 1.35$ for permanent loads, $\gamma_Q = 1.5$ for variable loads). Stage 3 (Dimensional Consistency) verifies unit compatibility across all inputs and ensures mesh density is appropriate for the analysis type. Stage 4 (Regulatory Compliance) cross-references the analysis parameters against the specified building code requirements stored in the IQS

knowledge base. Only scripts passing all four validation stages are permitted to execute on the deterministic engine.

The error handling logic addresses three categories of failures to maintain deterministic reliability. Category 1 (Pre-execution Errors): If validation protocols detect script errors, the system returns a structured error report to the LLM interface, which regenerates the script with corrections. A maximum of three regeneration attempts is permitted before escalating to user intervention with specific guidance on the detected issues. Category 2 (Convergence Failures): For nonlinear analyses where the solver fails to converge, the system implements an adaptive strategy: first, automatic refinement of load stepping (reducing increment size by 50%); second, if still non-convergent, switching to alternative solution algorithms (e.g., from Newton–Raphson to arc-length method); third, if convergence remains unachieved, the system reports the failure with diagnostic information including the last converged load factor, displacement state, and suspected cause (geometric instability, material softening, or numerical ill-conditioning). Category 3 (Runtime Errors): Memory allocation failures, numerical overflow, or unexpected solver termination trigger immediate job termination with full state logging. The system preserves all intermediate results and generates a diagnostic report enabling post-mortem analysis. In all error scenarios, the fundamental principle is maintained: no unverified or partially completed results are ever presented to the user as valid engineering outputs.

The separation between the LLM’s probabilistic processing and the algorithm’s deterministic calculation is fundamental to the architecture’s reliability. The LLM may interpret user intent with some uncertainty, but the calculation itself is performed with complete precision. Any errors in interpretation would be caught by the verification protocols, while the calculations themselves are performed with high reliability by the deterministic algorithms.

In the last stage, the LLM interprets the rigorous response from the algorithm and transforms it into a final answer presented to the user. This interpretation includes explaining the results in natural language, highlighting key findings, and providing context for engineering decision-making. Throughout this process, the original numerical results remain available for verification, supporting complete traceability.

5. Implementation and Case Study: The 3Muri Chatbot

This section presents the practical realization of the SYNAPSE architecture through its implementation in the 3Muri chatbot, demonstrating how the theoretical principles established in previous sections translate into a working system. The 3Muri chatbot was designed by S.T.A. DATA, an Italian software company; the authors of this paper are, respectively, the CEO and the scientific technical consultant of S.T.A. DATA, providing direct insight into both the software’s technical requirements and the chatbot implementation following NSAI principles as described in Section 4.

The development of NSAI systems requires expertise spanning multiple technical domains. On the neural side, developers must understand deep learning architectures, LLM fine-tuning, and prompt engineering. On the symbolic side, expertise is needed in formal logic, knowledge representation, and ontology engineering.

5.1. The 3Muri Software Context

3Muri is recognized as the leading Italian software for seismic and static analysis of masonry and mixed structures. The software employs the innovative FME (Frame by Macro Element) method, which models masonry structures using macro-elements that capture the essential mechanical behavior. 3Muri supports analysis of structures incorporating multiple materials (masonry, reinforced concrete, steel, and timber). Despite the software’s technical sophistication, user support presented persistent challenges: documentation complexity

(manuals exceeding 2000 pages), support burden from repetitive queries, response time expectations, and the need for contextually precise responses.

These challenges motivated exploration of AI-assisted solutions that could provide immediate, accurate, and contextually relevant support.

5.2. Solution Design

The 3Muri chatbot (Figure 7) was designed as an NSAI-based intelligent assistant implementing the SYNAPSE architecture principles described in Section 4. The design combines the LLM's natural language understanding capabilities with a rule-based expert system specific to 3Muri. The symbolic core is built on a purpose-designed ontology covering structural engineering concepts across multiple material types and analysis types. The logic framework encodes compliance rules from Eurocodes and Italian NTC. The neural component utilizes state-of-the-art language models with careful prompt engineering, while extensive fine-tuning improves domain-specific performance. Through this integration, the chatbot achieves verified reliability, full traceability, regulatory compliance, and intuitive interaction.

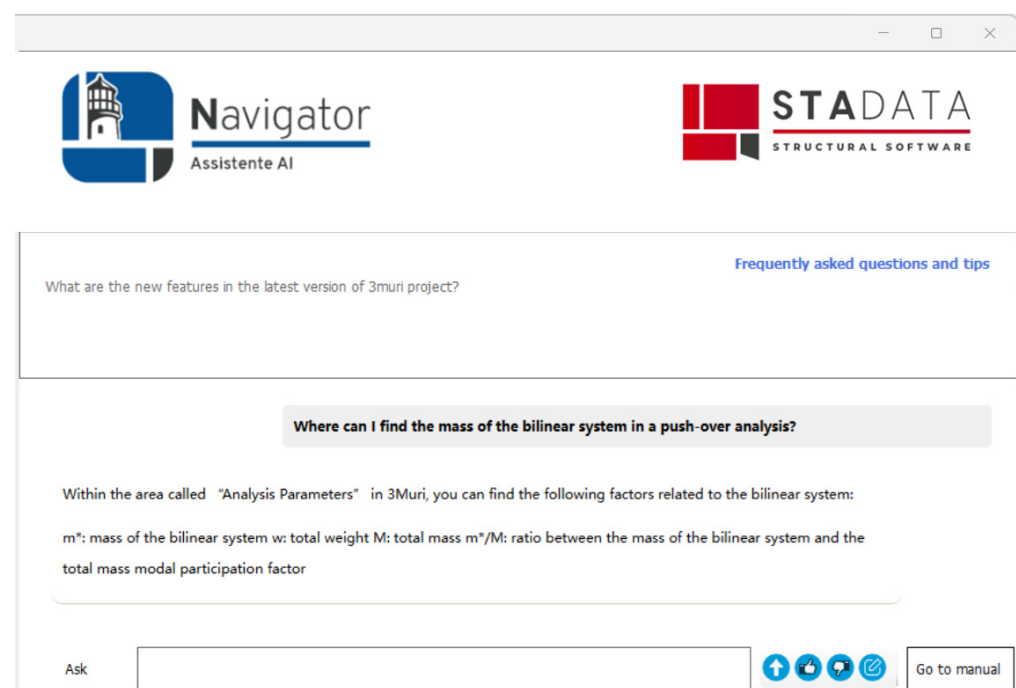


Figure 7. The graphical interface of the 3Muri chatbot.

Knowledge engineering, the process of encoding domain expertise into symbolic rules and ontologies, is particularly labor-intensive. Every regulatory requirement must be carefully analyzed, translated into formal rules, and validated against authoritative sources.

To ensure reproducibility and provide transparency regarding our implementation, we detail the technical specifications of the 3Muri chatbot system. The neural component employs OpenAI's GPT-4 model accessed via the OpenAI API, configured with a temperature of 0.2 to minimize output variability while maintaining response quality. The Retrieval-Augmented Generation (RAG) pipeline is implemented using LangChain (version 0.3.26) with a FAISS vector store for efficient similarity search. Document embeddings are generated using Google's text-embedding-004 model with a chunk size of 1024 tokens and 80-token overlap to preserve context across document segments. The symbolic computation engine integrates Code Aster, the open-source finite element solver developed by EDF, for

all structural calculations. The knowledge base comprises approximately 2000 pages of 3Muri documentation, over 500 solved examples, and the complete Italian NTC 2018 and Eurocode provisions for masonry structures, totaling approximately 15,000 text chunks. Domain adaptation was achieved through supervised fine-tuning on a curated dataset of over 200 question–answer pairs validated by senior structural engineers. Hyperparameter optimization was performed using a held-out validation set of 200 queries, with the final configuration achieving optimal balance between response accuracy (94%) and latency (1.8 s average). The system operates on cloud infrastructure (AWS) with automated scaling to handle concurrent user requests.

5.3. Experimental Results

The 3Muri chatbot has been deployed and evaluated in production use, providing quantitative validation of the NSAI approach. The validation process involved asking the chatbot over 200 questions. Structural engineers from the development team evaluated the answers, with each response scored based on their completeness and accuracy. Key performance metrics include:

- Response Accuracy of 94% (queries receiving responses rated as accurate by expert evaluators);
- Average Response Time of 1.8 s (median time from query submission to response delivery);
- User Satisfaction of 4.7/5 based on user surveys. Users particularly value immediate availability (24/7 access), contextual relevance of answers, and natural interaction style.

Maintaining coherence between neural and symbolic components presents ongoing challenges, as discussed in Section 6.

6. Discussion

The 94% accuracy achieved by the 3Muri chatbot warrants critical interpretation in the context of structural engineering safety requirements. While this result demonstrates the practical viability of the NSAI approach, the 6% failure rate requires careful analysis.

6.1. Analysis of Failed Cases

Detailed examination of the failed queries reveals that the majority (approximately 70% of failures) fall into categories that do not directly impact structural safety: queries about software licensing, installation issues, or features not yet implemented in the current version. These failures represent user experience limitations rather than safety-critical deficiencies.

The remaining failures (approximately 30% of the 6%, representing less than 2% of total queries) involve technically correct but incomplete responses to complex structural engineering questions. These cases typically arise when users pose questions that span multiple interconnected topics or require synthesizing information from disparate sections of the knowledge base. Importantly, the NSAI architecture’s symbolic verification layer prevented any incorrect structural calculations from being presented to users. When the system could not provide a complete answer with high confidence, it acknowledged uncertainty rather than generating potentially dangerous misinformation—a critical safety feature that distinguishes the NSAI approach from pure LLM implementations.

Balancing response time with thoroughness is addressed through caching and tiered verification.

6.2. Technical Limitations

The most labor-intensive aspect of the 3Muri implementation was the formalization of regulatory codes into symbolic rules. The Italian NTC (Norme Tecniche per le Costruzioni) and Eurocode provisions for masonry structures contain numerous interconnected requirements, conditional clauses, and cross-references that proved challenging to encode systematically. Each regulatory requirement had to be: (1) extracted from the normative text, (2) decomposed into atomic logical predicates, (3) expressed in formal rule syntax, and (4) validated against authoritative interpretations. This process required approximately 6 person-months for the initial NTC masonry provisions alone, with ongoing maintenance effort as regulatory updates are published.

Maintaining coherence between neural and symbolic components presents ongoing challenges. During development, we encountered cases where the LLM correctly understood user intent but mapped this understanding to incorrect symbolic categories. For example, queries about “wall resistance” were sometimes classified as referring to in-plane shear capacity when the user intended out-of-plane flexural capacity. Resolving such ambiguities required iterative refinement of both LLM prompts and IQS ontology definitions. Similarly, kinematic analysis of masonry walls subjected to out-of-plane failure mechanisms currently relies on iterative rigid macro-block models that evaluate interlocking effects between structural elements [46]. Such iterative procedures for kinematic analysis could benefit from AI-assisted approaches that leverage the pattern recognition capabilities of neural networks while maintaining the deterministic verification provided by symbolic engines.

Version synchronization between components added complexity. When the base LLM was updated from GPT-3.5 to GPT-4 during the project, behavioral changes affected integration with the symbolic component, requiring recalibration of prompt templates and confidence thresholds. The four-stage verification process also introduced latency that required optimization through caching frequently requested analyses and implementing tiered verification protocols. The current implementation is limited to Italian and European regulatory frameworks; extending to other jurisdictions (such as American ACI/ASCE codes or Japanese building standards) would require substantial knowledge engineering effort for each code family.

A further methodological limitation concerns the absence of ablation studies isolating the individual contributions of each architectural component. The current validation relies on end-to-end accuracy metrics (94% on 200+ queries) but does not separately quantify the performance contributions of the symbolic rules, the IQS context enrichment mechanism, and the multi-stage verification layer. Such component-level ablation experiments would strengthen the empirical foundation by demonstrating which architectural elements are most critical for achieving reliable performance. We identify ablation studies as a priority for future work to provide more granular insights into the NSAI system’s behavior.

6.3. Lessons Learned

The 3Muri chatbot deployment has provided practical insights that inform future NSAI development in structural engineering applications. Knowledge base quality is critical: the accuracy of the system depends fundamentally on the quality of its knowledge base. Investment in curating, organizing, and verifying knowledge base content pays direct dividends in system performance. Conversely, gaps or errors in the knowledge base translate directly into user-visible failures. This finding emphasizes the importance of involving domain experts throughout the development process, not just during initial knowledge acquisition. Graceful uncertainty builds trust: users accept that an AI system cannot answer every question. What damages trust is false confidence—answers that are

wrong but presented with certainty. The system's design to recognize and acknowledge uncertainty has proven essential to user acceptance. Engineers prefer receiving "I don't know" over receiving incorrect guidance that could affect structural safety.

Continuous improvement is essential: the chatbot improves over time through systematic analysis of queries it handles poorly. Each failure represents an opportunity to expand the knowledge base or refine processing logic. This continuous improvement cycle requires dedicated resources for monitoring, analysis, and iterative enhancement.

Integration with existing workflows matters: the chatbot's success depends partly on its accessibility within users' normal workflows. Integration with the software interface, email support systems, and documentation portals allows users to access assistance without disrupting their work.

6.4. Future Perspectives

Several directions for future work emerge from this research, addressing both the expansion of NSAI applications and the enhancement of the underlying architecture.

Extension to additional domains: the NSAI approach demonstrated here for structural engineering applies equally to other safety-critical engineering domains including geotechnical engineering, hydraulic engineering, and mechanical system design. Each domain presents specific challenges but shares the fundamental need for reliable AI assistance that combines natural interaction with verifiable accuracy.

BIM and digital twin integration represents a natural evolution of the SYNAPSE approach. As building information models become increasingly rich and accessible, NSAI systems can leverage this structured data for enhanced context understanding and more precise responses. Integration with digital twin platforms would enable real-time monitoring and AI-assisted decision support throughout the building lifecycle.

Extended multimodal capabilities including processing of hand-drawn sketches, photographs of existing structures, and CAD drawings would further expand the system's utility. Engineers often communicate through visual media, and enabling the NSAI system to interpret and respond to such inputs would significantly enhance its practical value in real-world engineering workflows.

7. Conclusions

This paper addresses the fundamental challenge of integrating artificial intelligence into structural engineering, a safety-critical domain where the probabilistic nature and hallucination tendencies of Large Language Models make their direct application unacceptable. The research makes four key contributions to the field.

First, we have provided a comprehensive problem analysis explaining why pure LLM approaches are unsuitable for structural engineering, grounding this analysis in the specific constraints of the domain: legal responsibility, regulatory compliance, safety criticality, and the need for deterministic reproducibility. Secondly, we have presented the theoretical framework of neuro-symbolic AI as the appropriate paradigm for this domain, demonstrating how the complementary strengths of neural and symbolic components can be integrated to achieve both flexibility and rigor. Thirdly, we have developed the SYNAPSE architecture which provides a concrete blueprint for implementing neuro-symbolic systems in structural engineering applications. Fourthly, through the 3Muri chatbot case study, we have provided empirical validation demonstrating 94% accuracy with response times under 2 s in production deployment.

Three principles emerge from this research that should guide AI adoption in structural engineering. First, neuro-symbolic AI represents the future of computational engineering: neither pure neural approaches (too unreliable) nor pure symbolic approaches (too rigid)

can meet the needs of modern engineering practice. The hybrid approach, on the other hand, provides the path forward, combining learning capability with formal verification, natural interaction with deterministic calculation. Secondly, the balance between innovation and reliability is not only possible but necessary: the engineering profession must not choose between embracing AI and maintaining safety standards. The neuro-symbolic approach demonstrates that both goals can be achieved simultaneously through careful architectural design. Finally, our test results using 3Muri chatbot demonstrate that neuro-symbolic systems can be deployed successfully in real-world applications, achieving performance levels that satisfy both user expectations and safety requirements.

The structural engineering profession stands at a pivotal moment. SYNAPSE architecture and its successful implementation in the 3Muri chatbot demonstrate that the engineering profession can embrace AI while maintaining the safety standards that society depends upon. As AI capabilities continue to advance, the neuro-symbolic approach provides a principled framework for harnessing these capabilities responsibly in safety-critical applications.

Author Contributions: Conceptualization, A.C.; methodology, A.C.; software, A.C. and G.N.; validation, G.N.; formal analysis, A.C. and G.N.; investigation, A.C. and G.N.; resources, A.C.; data curation, G.N.; writing—original draft preparation, A.C. and G.N.; writing—review and editing, G.N.; visualization, G.N.; supervision, A.C.; project administration, A.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The 3Muri chatbot is a proprietary component of commercial software developed and owned by S.T.A. DATA Srl. Accordingly, the underlying data, algorithms, and knowledge base are not publicly available due to intellectual property restrictions.

Conflicts of Interest: Author Adriano Castagnone and Giuseppe Nitti were employed by the company S.T.A. DATA Srl, Via Saluzzo, 71–10126, Torino, Italy. All authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Zhao, W.X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. A Survey of Large Language Models. *arXiv* **2023**, arXiv:2303.18223.
2. Naveed, H.; Khan, A.U.; Qiu, S.; Saqib, M.; Anwar, S.; Usman, M.; Akhtar, N.; Barnes, N.; Mian, A. A Comprehensive Overview of Large Language Models. *arXiv* **2023**, arXiv:2307.06435. [\[CrossRef\]](#)
3. Thai, H.-T. Machine Learning for Structural Engineering: A State-of-the-Art Review. *Structures* **2022**, *38*, 448–491. [\[CrossRef\]](#)
4. Tapeh, A.T.G.; Naser, M.Z. Artificial Intelligence, Machine Learning, and Deep Learning in Structural Engineering: A Scientometrics Review of Trends and Best Practices. *Arch. Comput. Methods Eng.* **2023**, *30*, 115–159. [\[CrossRef\]](#)
5. Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; et al. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Comput. Surv.* **2025**, *43*, 42. [\[CrossRef\]](#)
6. Tonmoy, S.M.; Zaman, S.M.M.; Jain, V.; Ber, A.; Das, A.; Smita, A.S.; Coppolillo, E.; Patwary, M.; Tiwary, K.; Majumder, A.K.; et al. A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models. *arXiv* **2024**, arXiv:2401.01313. [\[CrossRef\]](#)
7. Pérez-Cerrolaza, J.; Abella, J.; Cazorla, F.J.; Taber, B.; Keefe, K.; Kott, G.; Kosmidis, L. Artificial Intelligence for Safety-Critical Systems in Industrial and Transportation Domains: A Survey. *ACM Comput. Surv.* **2024**, *56*, 176. [\[CrossRef\]](#)
8. EN 1990:2002; Eurocode—Basis of Structural Design. European Committee for Standardization: Brussels, Belgium, 2002.
9. Gulvanessian, H.; Calgaro, J.A.; Holický, M. *Designers' Guide to EN 1990: Eurocode: Basis of Structural Design*; Thomas Telford: London, UK, 2002.
10. Ministero delle Infrastrutture e dei Trasporti. Decreto Ministeriale 17 gennaio 2018: Aggiornamento delle Norme Tecniche per le Costruzioni. *Gazzetta Ufficiale della Repubblica Italiana, Serie Generale n. 42*. 20 February 2018. Available online: <https://www.gazzettaufficiale.it/eli/id/2018/2/20/18A00716/sg> (accessed on 5 January 2026). (In Italian)

11. Kaur, D.; Uslu, S.; Rittichier, K.J.; Durresi, A. Trustworthy Artificial Intelligence: A Review. *ACM Comput. Surv.* **2022**, *55*, 39. [CrossRef]
12. Xu, Z.; Jain, S.; Kankanhalli, M. Hallucination is Inevitable: An Innate Limitation of Large Language Models. *arXiv* **2024**, arXiv:2401.11817. [CrossRef]
13. Etim, B.; Al-Ghosoun, A.; Renno, J.; Seaid, M.; Mohamed, M.S. Machine Learning-Based Modeling for Structural Engineering: A Comprehensive Survey and Applications Overview. *Buildings* **2024**, *14*, 3515. [CrossRef]
14. Málaga-Chuquitaype, C. Machine Learning in Structural Design: An Opinionated Review. *Front. Built Environ.* **2022**, *8*, 815717. [CrossRef]
15. Lacidogna, G.; Nitti, G.; Scaramozzino, D.; Carpinteri, A. Diagrid systems coupled with closed- and open-section shear walls: Optimization of geometrical characteristics in tall buildings. *Procedia Manuf.* **2020**, *44*, 402–409. [CrossRef]
16. Lacidogna, G.; Nitti, G.; Scaramozzino, D.; Carpinteri, A. Diagrid system coupled with shear walls: Analytical investigation on the dynamical response in tall buildings. In Proceedings of the 8th International Conference on Computational Methods in Structural Dynamics and Earthquake Engineering, Athens, Greece, 28–30 June 2021; pp. 1793–1802. [CrossRef]
17. Sun, H.; Burton, H.V.; Huang, H. Machine Learning Applications for Building Structural Design and Performance Assessment: State-of-the-Art Review. *J. Build. Eng.* **2021**, *33*, 101816. [CrossRef]
18. Saka, M.P.; Dogan, E.; Aydogdu, I. Analysis of Swarm Intelligence-Based Algorithms for Constrained Optimization. In *Swarm Intelligence and Bio-Inspired Computation*; Yang, X.S., Cui, Z., Xiao, R., Gandomi, A.H., Karamanoglu, M., Eds.; Elsevier: Amsterdam, The Netherlands, 2013; pp. 25–48. [CrossRef]
19. Solhmirzaei, R.; Salehi, H.; Kodur, V.; Naser, M.Z. Machine Learning Framework for Predicting Failure Mode and Shear Capacity of Ultra High Performance Concrete Beams. *Eng. Struct.* **2020**, *224*, 111221. [CrossRef]
20. Gaur, M.; Sheth, A. Building Trustworthy NeuroSymbolic AI Systems: Consistency, Reliability, Explainability, and Safety. *AI Mag.* **2024**, *45*, 139–155. [CrossRef]
21. Michel-Delétie, C.; Sarker, M.K. Neuro-Symbolic Methods for Trustworthy AI: A Systematic Review. *Neurosymbolic Artif. Intell.* **2024**. Available online: <https://neurosymbolic-ai-journal.com/system/files/nai-paper-726.pdf> (accessed on 5 January 2026).
22. Seshia, S.A.; Sadigh, D.; Sastry, S.S. Toward Verified Artificial Intelligence. *Commun. ACM* **2022**, *65*, 46–55. [CrossRef]
23. Fox, O.R.; Bergami, G.; Morgan, G. Verified Language Processing with Hybrid Explainability. *Electronics* **2025**, *14*, 3490. [CrossRef]
24. Garcez, A.; Lamb, L.C. Neurosymbolic AI: The 3rd Wave. *Artif. Intell. Rev.* **2023**, *56*, 12387–12406. [CrossRef]
25. Yu, D.; Yang, B.; Liu, D.; Wang, H.; Pan, S. A Survey on Neural-Symbolic Learning Systems. *Neural Netw.* **2023**, *166*, 105–126. [CrossRef]
26. Garcez, A.d.; Gori, M.; Lamb, L.C.; Serafini, L.; Spranger, M.; Tran, S.N. Neural-Symbolic Computing: An Effective Methodology for Principled Integration of Machine Learning and Reasoning. *arXiv* **2019**, arXiv:1905.06088. [CrossRef]
27. Zhang, Z.; Sun, C. Structural Damage Identification via Physics-Guided Machine Learning: A Methodology Integrating Pattern Recognition with Finite Element Model Updating. *Struct. Health Monit.* **2021**, *20*, 1013–1039. [CrossRef]
28. Cha, Y.J.; Choi, W.; Suh, G.; Mahmoudkhani, S.; Büyüköztürk, O. Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types. *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *33*, 731–747. [CrossRef]
29. Gonzalez, M.; Zapico, J. Seismic Damage Identification in Buildings Using Neural Network and Modal Data. *Comput. Struct.* **2008**, *86*, 416–426. [CrossRef]
30. Hakim, S.J.S.; Abdul Razak, H.; Ravanfar, S.A. Ensemble Neural Networks for Structural Damage Identification Using Modal Data. *Int. J. Damage Mech.* **2021**, *30*, 1457–1483. [CrossRef]
31. Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y.J.; Madotto, A.; Fung, P. Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.* **2023**, *55*, 248. [CrossRef]
32. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
33. Hitzler, P.; Sarker, M.K. (Eds.) *Neuro-Symbolic Artificial Intelligence: The State of the Art*; IOS Press: Amsterdam, The Netherlands, 2022.
34. Marra, G.; Dumančić, S.; Manhaeve, R.; De Raedt, L. From Statistical Relational to Neurosymbolic Artificial Intelligence: A Survey. *Artif. Intell.* **2024**, *328*, 104062. [CrossRef]
35. Gibaut, W.; Pereira, L.; Grassiotto, F.; Osorio, A.; Gadioli, E.; Munoz, A.; Gomes, S.; Dos Santos, C. Neurosymbolic AI and Its Taxonomy: A Survey. *arXiv* **2023**, arXiv:2305.08876. [CrossRef]
36. Shakarian, P.; Baral, C.; Simari, G.I.; Xi, B.; Pokala, L. *Neuro Symbolic Reasoning and Learning*; Springer: Cham, Switzerland, 2023. [CrossRef]
37. Xie, Y.; Xu, Z.; Meel, K.S.; Kankanhalli, M.; Soh, H. Neuro-Symbolic Verification of Deep Neural Networks. In Proceedings of the IJCAI 2022, Vienna, Austria, 23–29 July 2022; pp. 3622–3628. [CrossRef]
38. Saeed, W.; Omlin, C. Explainable AI (XAI): A Systematic Meta-Survey of Current Challenges and Future Opportunities. *Knowl.-Based Syst.* **2023**, *263*, 110273. [CrossRef]

39. Dwivedi, R.; Dave, D.; Naik, H.; Singhal, S.; Omer, R.; Patel, P.; Qian, B.; Wen, Z.; Shah, T.; Morgan, G.; et al. Explainable Artificial Intelligence (XAI): Core Ideas, Techniques, and Solutions. *ACM Comput. Surv.* **2023**, *55*, 194. [[CrossRef](#)]
40. Naser, M.Z. An Engineer's Guide to eXplainable Artificial Intelligence and Interpretable Machine Learning: Navigating Causality, Forced Goodness, and the False Perception of Inference. *Autom. Constr.* **2021**, *129*, 103821. [[CrossRef](#)]
41. Love, P.E.D.; Fang, W.; Matthews, J.; Porter, S.; Luo, H.; Ding, L. Explainable Artificial Intelligence (XAI): Precepts, Models, and Opportunities for Research in Construction. *Adv. Eng. Inform.* **2023**, *57*, 102024. [[CrossRef](#)]
42. Noureldin, M. Applications of Artificial Intelligence for Explainability and Uncertainty Quantification for Performance-Based Design and Damage Classification in Civil Engineering. Ph.D. Thesis, Aalto University, Espoo, Finland, 2025.
43. Ruggieri, S.; Cardellicchio, A.; Leggieri, V.; Uva, G. Bridging AI and Explainability in Civil Engineering: The Yin-Yang of Predictive Power and Interpretability. *AI Civ. Eng.* **2025**, *4*, 21. [[CrossRef](#)]
44. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions. *J. Big Data* **2021**, *8*, 53. [[CrossRef](#)]
45. Wan, Z.; He, L.; Tao, X.; Xie, Z.; Wang, J.; He, G. Towards Cognitive AI Systems: A Survey and Prospective on Neuro-Symbolic AI. *arXiv* **2024**, arXiv:2401.01040. [[CrossRef](#)]
46. Casapulla, C.; Castagnone, A.; Nitti, G.; Borgesa, L.; Seni, D. Interlocking effects of masonry walls on their out-of-plane failure using rigid macro-block model. In Proceedings of the 18th World Conference on Earthquake Engineering, Milan, Italy, 30 June–5 July 2024.

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